

Universidad Miguel Hernández de Elche

**MASTER UNIVERSITARIO EN
ROBÓTICA**



**“Programación de robots ABB para una estación
de trabajo de pintura de carrocerías de
automóvil”**

Trabajo de Fin de Máster

2020 - 2021

**Autor: Arturo López López
Tutor/es: Carlos Pérez Vidal**

AGRADECIMIENTOS

Agradecer a mi familia y amigos, y en especial a mi pareja Aida, todo el apoyo, cariño y paciencia recibidos durante el tiempo invertido en la realización de este proyecto.

Agradecer también a todos los profesores del Master de Robótica, por los conocimientos en robótica que he adquirido gracias a ellos, y en especial a mi tutor, Carlos Pérez Vidal, y al subdirector del máster, Luis Paya Castello, por la ayuda y las facilidades prestadas, solucionando cualquier duda que surgiera a lo largo del desarrollo del proyecto.

Por último, agradecer a la Universidad Miguel Hernández y al Departamento de Ingeniería de Sistemas y Automática poner a mi disposición la licencia, y un servidor para la utilización de la misma, del software RobotStudio.





RESUMEN

En el presente trabajo se realizará el diseño y programación de una estación robótica de pintado en seguimiento para la aplicación de la capa base del recubrimiento de la carrocería de un modelo de automóvil seleccionado. En el diseño de la estación se buscará dotar a la misma de una gran versatilidad de modo que en otro escenario de aplicación, es decir, con otra programación de los robots de la estación, se pudiera variar el modelo de vehículo al que se aplica la capa base.

La capa base, como se expondrá en detalle dentro de trabajo, es la capa del recubrimiento de la carrocería que le proporciona, entre otras cosas, el color a la misma, por lo tanto se puede decir que se realizará el diseño y programación de una estación robótica que confiere el color a la carrocería del modelo de vehículo seleccionado.

Para realizar esta tarea se empleará el software de ingeniería RobotStudio, desarrollado por la empresa ABB para la programación y control de sus sistemas robóticos. La programación de la estación estará limitada por su complejidad real y por el hecho de no llevarse a cabo en el marco de una empresa de programación real, lo cual implica que no se dispone de todos los elementos, datos, señales, etc. que intervendrían en una estación de pintado real.

Esta estación robótica estará formada por seis robots de pintura del fabricante ABB, cuatro de aplicación y dos de asistencia, todos ellos montados sobre tracks para aumentar su espacio de trabajo, que trabajando al unísono realizarán la aplicación completa de la capa base sobre el exterior y el interior de la carrocería del modelo de vehículo seleccionado sin detener el avance del mismo a lo largo de estación de recubrimiento.

Para dotar a la estación de una mayor flexibilidad y mantener la limpieza del ambiente de la misma, aspecto muy importante en la aplicación robotizada de pintura, además de programar la aplicación de la capa base, se programará la estación para que sea posible realizar tareas de mantenimiento y limpieza de los aplicadores sin necesidad de acceder al interior de la estación, evitando así la contaminación de su ambiente. También se programarán ciertas medidas de seguridad para evitar daños en el equipo y el personal.

Para llevar a cabo de forma correcta el presente proyecto, en primer lugar se realizó un estudio del estado del arte de las estaciones robóticas de recubrimiento de carrocerías de vehículos, de forma que se adquirieran los conocimientos necesarios para el correcto diseño de la estación del presente proyecto.

A continuación se realizó un estudio del software RobotStudio, con el fin de comprenderlo y así obtener los conocimientos y habilidades necesarias para crear, diseñar y programar lo más fielmente posible a la realidad la estación de pintado. Para ello se han empleado una gran variedad de manuales, recursos web, video tutoriales, etc.

Una vez adquiridos estos conocimientos, se realizó el diseño y programación paso a paso de la estación.

Finalmente se comprobó el correcto funcionamiento de todas las características de la estación, mediante la ejecución y simulación de los programas de robot, analizando y evaluando los resultados obtenidos.

ABSTRACT

In this project, the design and programming of a robotic painting in tracking station for the application of the base coat of the bodywork coating of a selected car model will be carried out. The design of the station will seek to provide it with great versatility so that in another application scenario, that is, with another programming of the station robots, the vehicle model to which the base coat is applied could be varied.

The base coat, as will be explained in detail in this project, is the coating layer of the bodywork that provides, among other things, the color to it, therefore it can be said that the design and programming of a robotic station that confers the color to the bodywork of the selected vehicle model will be carried out.

The engineering software RobotStudio, developed by ABB for the programming and control of their robotic systems, will be used for this project. The programming of the station will be limited by its real complexity and by the fact that it will not be carried out in the framework of a real programming company, which means that not all the elements, data, signals, etc. that would be involved in a real painting station are available.

This robotic station will consist of six ABB painting robots, four application robots and two assistance robots, all of them mounted on tracks to increase their work area, which, working in unison, will perform the complete application of the base coat on the exterior and interior of the body of the selected vehicle model without stopping its progress along the coating station.

In order to provide the station with greater flexibility and maintain the cleanliness of its environment, a very important aspect in the robotic application of paint, in addition to programming the application of the base coat, the station will be programmed to allow maintenance and cleaning of the applicators without the need to access the interior of the station, thus avoiding contamination of its environment. Certain safety measures will also be programmed to avoid damage to equipment and personnel.

In order to correctly carry out the present project, first of all, a study of the state of the art of robotic stations for the coating of vehicle bodyworks was carried out, in order to acquire the necessary knowledge for the correct design of the station of this project.

Next, a study of the RobotStudio software was carried out in order to understand it and thus obtain the necessary knowledge and skills to create, design and program the painting station as closely as possible to reality. For this purpose, a wide variety of manuals, web resources, video tutorials, etc. were used.

Once this knowledge was acquired, the station was designed and programmed step by step.

Finally, the correct operation of all the features of the station was checked by executing and simulating the robot programs, analyzing and evaluating the results obtained.



ÍNDICE

AGRADECIMIENTOS	I
RESUMEN	III
ABSTRACT.....	IV
1. INTRODUCCIÓN.....	1
1.1 Motivación	3
1.2 Objetivos	4
1.3 Estructura de la memoria	4
2. ESTADO DEL ARTE	7
2.1 Evolución histórica del pintado robótico en ABB.....	7
2.2 Pintado de vehículos	11
2.2.1 Estación de recubrimiento.....	11
2.2.2 Etapas de la estación de recubrimiento.....	12
2.2.3 Distribución en planta de la estación de recubrimiento	20
2.2.4 Estación de pintado	21
2.2.4.1 Capa de imprimación	23
2.2.4.2 Capas superiores.....	34
2.2.5 Automatización del proceso de pintado	42
2.2.6 Distribución en planta y equipamiento necesario	43
2.2.7 Consideraciones prácticas del proceso de aplicación de la capa base	56
3. METODOLOGÍA.....	61
3.1 Introducción al software RobotStudio	61
3.1.1 Descripción general	61
3.1.2 Términos y conceptos generales	61
3.1.3 Interfaz gráfica de usuario	63
3.1.4 Creación de una estación.....	73
3.1.5 Importación de robots y otros componentes de la estación	74
3.1.6 Creación/importación de tracks y mecanismos	78
3.1.7 Importación del efector final del robot	84
3.1.8 Modelado de componentes en RobotStudio	87
3.1.9 Modificación de componentes en RobotStudio.....	96
3.1.10 Descripción de los sistemas de coordenadas.....	101
3.1.11 Posicionamiento de robots y otros componentes de la estación	106
3.1.12 Creación de un controlador virtual	112

3.1.13	Creación de transportadores y conexión con el controlador	116
3.1.14	Movimiento manual de mecanismos.....	120
3.1.15	Creación de objetivos y trayectorias.....	123
3.1.16	Orientaciones y configuraciones de los objetivos.....	136
3.1.17	Plantillas de instrucciones	142
3.1.18	Sincronización del controlador virtual con RAPID.....	145
3.1.19	Lógica de estación	147
3.1.20	Componentes inteligentes.....	155
3.1.21	Señales de E/S del sistema de robot.....	167
3.1.22	Simulación de la estación	169
3.1.23	Detección de colisiones	174
3.1.24	Complementos y PowerPacs	177
3.2	Programación básica en RAPID.....	181
3.2.1	Elementos básicos del lenguaje.....	181
3.2.2	Datos de programa	184
3.2.3	Módulos.....	192
3.2.4	Rutinas.....	194
3.2.5	Instrucciones	202
3.2.5.1	Control del flujo del programa.....	203
3.2.5.2	Movimiento del robot.....	210
3.2.5.3	Señales E/S	225
3.2.5.4	Comunicaciones.....	227
3.2.5.5	Interrupciones	229
3.2.5.6	Otras instrucciones	231
3.3	Otros softwares utilizados	232
4.	DISEÑO Y PROGRAMACIÓN DE UNA ESTACIÓN ROBÓTICA DE PINTADO EN SEGUIMIENTO DE LA CARROCERÍA DE UN AUTOMÓVIL	235
4.1	Determinación de las necesidades y limitaciones	235
4.2	Componentes de la estación	237
4.2.1	Selección y posicionamiento de los equipos de ABB	237
4.2.1.1	Selección y posicionamiento de los robots y los tracks	237
4.2.1.2	Selección de las herramientas	241
4.2.2	Selección y/o diseño de componentes mediante RobotStudio	243
4.2.3	Selección y/o diseño de componentes mediante softwares externos.....	249
4.3	Creación y programación de la estación en RobotStudio	263
4.3.1	Creación de la estación vacía.....	266

4.3.2	Importación/creación y posicionamiento de componentes fundamentales	266
4.3.3	Creación del transportador.....	271
4.3.4	Creación de los controladores virtuales	272
4.3.5	Creación de las conexiones del transportador con los controladores	275
4.3.6	Creación de la plantilla de instrucciones y objetos de trabajo.....	278
4.3.7	Creación de puntos objetivo y programación de trayectorias, procedimientos y rutinas	279
4.3.7.1	Objetivos.....	280
4.3.7.2	Trayectorias, procedimientos y rutinas	288
4.3.7.3	Rutinas TRAP.....	333
4.3.8	Desarrollo de la lógica de estación	343
4.3.9	Creación de señales de E/S.....	372
4.3.10	Configuración de la simulación.....	373
5.	RESULTADOS OBTENIDOS	377
5.1	Simulación del proceso de aplicación de la capa base de la carrocería	377
5.2	Simulación de la rutina TRAP de limpieza.....	392
5.3	Simulación de la rutina TRAP de mantenimiento	395
5.4	Simulación de la rutina TRAP de seguridad de puertas	397
5.5	Simulación de la rutina TRAP paro de emergencia	399
6.	CONCLUSIONES Y LÍNEAS FUTURAS DE TRABAJO.....	401
6.1	Conclusiones.....	401
6.2	Líneas futuras de trabajo.....	402
7.	Bibliografía	403
8.	Anexos.....	406
8.1	Código RAPID de los programas de robot.....	406
8.1.1	Controlador virtual Cont_Inf_Izq	406
8.1.2	Controlador virtual Cont_Inf_Der	450
8.1.3	Controlador virtual Cont_Sup_Izq	491
8.1.4	Controlador virtual Cont_Sup_Der	510
8.1.5	Controlador virtual Cont_Puertas_Izq	528
8.1.6	Controlador virtual Cont_Puertas_Der	532
8.2	Fichas técnicas de los robots de la estación	538

ÍNDICE DE FIGURAS

Figura 1.1 - Utilización de los robots en el sector logístico, público y militar en el mundo	1
Figura 1.2 - Densidad de robots en el año 2019 para algunos de los principales países manufactureros.....	2
Figura 2.1 - Primer robot de pintura para uso industrial.....	7
Figura 2.2 - Robot de pintura IRB 5500-27	8
Figura 2.3 - ABB Ability™ Connected Atomizer o RB1000i.....	9
Figura 2.4 - ABB FeatherDuster	10
Figura 2.5 - Cronología de los avances de ABB en la aplicación robotizada de pintura	10
Figura 2.6 – Cronología de las mejoras que han supuesto los avances de ABB en la aplicación robotizada de pintura.....	11
Figura 2.7 - Proceso de electrorecubrimiento catódico	14
Figura 2.8 - Sistema industrial de electrorecubrimiento catódico.....	15
Figura 2.9 - Aplicación de los materiales de sellado	16
Figura 2.10 - Preservación de las cavidades mediante aplicación robotizada	17
Figura 2.11 - Control de calidad del recubrimiento mediante robots	18
Figura 2.12 - Medición laser del espesor del recubrimiento	18
Figura 2.13 - Fotómetro espectral multiangular.....	19
Figura 2.14 - Cadena de procesos en una estación de recubrimiento de vehículos	19
Figura 2.15 - Limpieza automática de la superficie del vehículo mediante rodillos de plumas ..	20
Figura 2.16 - Proceso estándar de pintado de una carrocería	22
Figura 2.17 - Proceso de pintado sin capa de imprimación	22
Figura 2.18 - Proceso de pintado Wet-on-wet	22
Figura 2.19 - Estructura de las capas del recubrimiento de un vehículo	22
Figura 2.20 - Distribución de la demanda global de imprimaciones (2003)	24
Figura 2.21 - Distribución de la demanda global de imprimaciones (2018)	24
Figura 2.22 - Caleo del electrorecubrimiento.....	26
Figura 2.23 - Ejemplo de delaminación debida al caleo del electrorecubrimiento.....	26
Figura 2.24 - Atomización electrostática centrífuga en sistemas a base de disolvente.....	31
Figura 2.25 - Atomización electrostática centrífuga mediante electrodos externos en sistemas a base de agua	31
Figura 2.26 - Aplicación de imprimaciones en polvo mediante carga triboeléctrica	33
Figura 2.27 - Aplicación de imprimaciones en polvo mediante el efecto corona	33
Figura 2.28 - Tendencia de aplicación de los colores principales a nivel global	34
Figura 2.29 - Reflexión de la luz blanca según el color en el que incide	35
Figura 2.30 - Variación de la viscosidad en función de la velocidad de corte	38
Figura 2.31 - Emisiones producidas en la aplicación de la capa base y de revestimiento según la tecnología de aplicación.....	40
Figura 2.32 - Esquema de una cabina de pintado automatizada.....	43
Figura 2.33 - Ejes de movimiento para un robot de pintura industrial	45
Figura 2.34 - Muñeca tipo Three-roll.....	46
Figura 2.35 - Muñeca tipo Lemma.....	46
Figura 2.36 - Muñeca tipo Hollow	47
Figura 2.37 - Disposición de los equipos de suministro y mezcla en el brazo robótico	47
Figura 2.38 - Robots de pintura con cubiertas desechables	48
Figura 2.39 - Simulación mediante ordenador de una estación de pintura robótica	49

Figura 2.40 - Robots de pintura instalados en pared.....	50
Figura 2.41 - Robots de pintura instalados sobre rail.....	50
Figura 2.42 - Atomizador rotativo de alta velocidad de plato de disco.....	51
Figura 2.43 - Atomizador rotativo de alta velocidad de plato de campana.....	51
Figura 2.44 - Detalle del plato de campana de un atomizador rotativo.....	52
Figura 2.45 - Esquema de un atomizador neumático.....	53
Figura 2.46 - Cambiador de color automático.....	54
Figura 2.47 - Ejemplo de un sistema de suministro.....	55
Figura 2.48 - Ejemplo de transportador aéreo.....	56
Figura 2.49 - Ejemplo de transportador de suelo.....	56
Figura 2.50 - Técnica de pintado de trazo e indexado.....	57
Figura 2.51 - Perfil de la película de pintura depositada por una sola pasada.....	58
Figura 2.52 - Perfil del espesor de pintura tras 4 trazos con un 50% de solapamiento.....	58
Figura 2.53 - Perfil del espesor de pintura tras 4 trazos con un 66% de solapamiento.....	59
Figura 2.54 - Perfil del espesor de pintura tras 4 trazos con un 75% de solapamiento.....	59
Figura 2.55 - Modificación del perfil de pintura de la Figura 2.51 mediante el caudal de aire...60	60
Figura 2.56 - Nuevo perfil del espesor de pintura tras 4 trazos con un 50% de solapamiento...60	60
Figura 3.1 - Interfaz gráfica del software RobotStudio.....	64
Figura 3.2 - Pestaña Archivo.....	65
Figura 3.3 - Pestaña Modelado.....	65
Figura 3.4 - Pestaña Simulación.....	66
Figura 3.5 - Pestaña Controlador.....	66
Figura 3.6 - Pestaña RAPID.....	67
Figura 3.7 - Pestaña Complementos.....	67
Figura 3.8 - Navegador Diseño.....	68
Figura 3.9 - Navegador Trayectorias y puntos.....	68
Figura 3.10 - Navegador Física.....	69
Figura 3.11 - Navegador Controlador.....	69
Figura 3.12 - Accesos rápidos en la ventana gráfica.....	70
Figura 3.13 - Ventana gráfica de la Lógica de estación.....	71
Figura 3.14 - Ventana gráfica de la configuración del controlador de robot.....	71
Figura 3.15 - Ventana gráfica con el código RAPID de programa de robot para un controlador.....	72
Figura 3.16 - Barra de estado.....	72
Figura 3.17 - Opciones para la creación de una nueva estación robótica.....	73
Figura 3.18 - Opción de creación de estación Solución con estación y controlador virtual.....	74
Figura 3.19 - Robots disponibles para su importación.....	75
Figura 3.20 - Opciones de configuración del modelo de robot IRB 5500 a importar.....	75
Figura 3.21 - Ejemplo de importación de un manipulador de robot.....	75
Figura 3.22 - Opciones para importar una geometría CAD.....	76
Figura 3.23 - Tipos de archivos CAD que permite importar RobotStudio.....	76
Figura 3.24 - Ejemplo de importación de geometría CAD.....	77
Figura 3.25 - Opciones para importar un archivo de biblioteca de RobotStudio.....	77
Figura 3.26 - Modelos de robots, posicionadores y tracks disponibles para su importación.....	78
Figura 3.27 - Opciones de configuración para el modelo de track IRB 5500 Elevated Rail.....	79
Figura 3.28 - Ejemplo de importación de un modelo de track.....	79
Figura 3.29 - Panel Crear mecanismo.....	80
Figura 3.30 - Cuadro de diálogo Crear Eslabón.....	81
Figura 3.31 - Mecanismo sencillo de ejemplo.....	81

Figura 3.32 - Configuración de ejes para el mecanismo de ejemplo	82
Figura 3.33 - Cuadro de dialogo Crear sistema de coordenadas	82
Figura 3.34 - Cuadro de dialogo Crear datos de herramienta	83
Figura 3.35 - Cuadro de dialogo Crear calibración.....	83
Figura 3.36 - Ejemplo del mecanismo sencillo creado.....	84
Figura 3.37 - Herramientas disponibles para su importación	85
Figura 3.38 - Cuadro de dialogo para el modelo de herramienta RB 1000 EXT	85
Figura 3.39 - Ejemplo de importación de una herramienta.....	86
Figura 3.40 - Primera opción de conexión de la herramienta al robot	86
Figura 3.41 - Ejemplo de conexión de herramienta a robot	87
Figura 3.42 - Grupo Crear de la pestaña Modelado.....	87
Figura 3.43 - Ejemplo de un grupo de componentes.....	88
Figura 3.44 - Ejemplo de agrupación de cuerpos en una pieza inicialmente vacía	88
Figura 3.45 - Opciones de creación de solidos primitivos	89
Figura 3.46 - Ejemplos del panel Crear para diferentes sólidos primitivos.....	89
Figura 3.47 - Opciones de creación de superficies	90
Figura 3.48 - Ejemplos del panel Crear para diferentes superficies.....	90
Figura 3.49 - Opciones de creación de curvas	91
Figura 3.50 - Ejemplos del panel Crear para diferentes curvas	91
Figura 3.51 - Ejemplo de los diferentes tipos de borde que pueden crearse	92
Figura 3.52 - Opciones para la creación de ejes o articulaciones	92
Figura 3.53 - Ejemplos del panel de configuración para la creación de distintas articulaciones	93
Figura 3.54 - Ejemplo de articulación de rotación	93
Figura 3.55 - Ejemplo del panel de la herramienta Unión	94
Figura 3.56 - Resultado de la intersección de los cuerpos Cubo y Esfera	94
Figura 3.57 - Resultado de la resta de los cuerpos Cubo y Esfera	94
Figura 3.58 - Resultado de la unión de los cuerpos Cubo y Esfera	95
Figura 3.59 - Ejemplo de diferentes tipos de curvas/superficies.....	95
Figura 3.60 - Panel extrudir superficie o curva	96
Figura 3.61 - Ejemplos de solidos o superficies creados mediante la herramienta Extrudir superficie.....	96
Figura 3.62 - Opciones de la herramienta modificar	97
Figura 3.63 - Pestaña Modificar para un componente de la estación	97
Figura 3.64 - Herramienta Seleccionar color	98
Figura 3.65 - Ejemplos de la herramienta Apariencia de gráficos	98
Figura 3.66 - Herramientas para la modificación de la física de una componente	99
Figura 3.67 – Ejemplo de geometría de colisión de calidad baja	100
Figura 3.68 - Ejemplo de geometría de colisión de calidad media	101
Figura 3.69 - Ejemplo de geometría de colisión de calidad alta	101
Figura 3.70 - Ejemplo de sistema de coordenadas del punto central de la herramienta	102
Figura 3.71 - Ejemplo de sistema de coordenadas mundo de RobotStudio (RS-WCS)	102
Figura 3.72 - Ejemplo de sistema de coordenadas de la base (BF)	103
Figura 3.73 - Ejemplo de sistema de coordenadas de la tarea (TF)	103
Figura 3.74 - Ejemplo de conexión de un sistema de coordenadas del objeto de trabajo a un componente de la estación	104
Figura 3.75 - Creación de un nuevo sistema de coordenadas del objeto de trabajo.....	105
Figura 3.76 - Panel Crear objeto de trabajo	105
Figura 3.77 - Herramientas para crear un sistema de coordenadas de usuario	106

Figura 3.78 - Herramientas para el posicionamiento de un componente	107
Figura 3.79 - Panel Definir posición	108
Figura 3.80 - Ejemplo de aplicación de la herramienta Posición de Offset.....	108
Figura 3.81 - Panel Girar	109
Figura 3.82 - Herramientas de posicionamiento a mano alzada	109
Figura 3.83 - Reposicionamiento de un robot una vez creado su controlador	110
Figura 3.84 - Cuadro de dialogo Configuración de movimiento	111
Figura 3.85 - Panel Modificar sistemas de coordenadas de tareas	111
Figura 3.86 - Opciones para crear un controlador virtual de robot	112
Figura 3.87 - Cuadro de dialogo Creación de controlador desde diseño	113
Figura 3.88 - Cuadro de dialogo Creación de controlador desde diseño - Mecanismos.....	113
Figura 3.89 - Cuadro de dialogo Creación de controlador desde diseño - Tareas del controlador	114
Figura 3.90 - Cuadro de dialogo Creación de controlador desde diseño - Opciones del controlador	114
Figura 3.91 - Cuadro de dialogo Creación de controlador desde diseño - Cambiar opciones ..	114
Figura 3.92 - Cuadro de dialogo Añadir un nuevo controlador	115
Figura 3.93 - Cuadro de dialogo Añadir controlador existente.....	116
Figura 3.94 - Panel Crear transportador.....	117
Figura 3.95 - Opciones del controlador - Conveyor Tracking.....	117
Figura 3.96 - Panel Crear conexión.....	118
Figura 3.97 - Visualización de la conexión en un transportador.....	119
Figura 3.98 - Panel Objeto del transportador.....	120
Figura 3.99 - Panel Movimiento del transportador	120
Figura 3.100 - Herramientas de mano alzada para mover mecanismos.....	121
Figura 3.101 - Herramienta Movimiento de ejes de mecanismo	121
Figura 3.102 - Panel Movimiento de ejes	122
Figura 3.103 - Panel Movimiento lineal.....	123
Figura 3.104 - Herramientas para la creación de objetivos y trayectorias	123
Figura 3.105 - Herramientas de la opción Posición para crear objetivos.....	124
Figura 3.106 - Ejemplo de visualización de objetivos	124
Figura 3.107 - Ejemplo de visualización de objetivos de ejes	125
Figura 3.108 - Panel Puntos de arista	126
Figura 3.109 - Herramientas de modificación de objetivos	127
Figura 3.110 - Herramientas de la opción Modificar posición	128
Figura 3.111 - Opción Ruta para la creación de trayectorias	129
Figura 3.112 - Ejemplos del panel Trayectoria automática	129
Figura 3.113 - Herramientas para insertar instrucciones en una trayectoria	131
Figura 3.114 - Panel Crear instrucción Move.....	131
Figura 3.115 - Herramientas de modificación de trayectorias.....	132
Figura 3.116 - Herramientas de la opción Trayectoria para modificar una trayectoria	134
Figura 3.117 - Herramienta Editar instrucción	135
Figura 3.118 - Herramientas de la opción Modificar una instrucción.....	136
Figura 3.119 - Números de los cuadrantes de configuración para los ejes 1, 4 y 6	139
Figura 3.120 - Números de los cuadrantes de configuración para los ejes lineales.....	139
Figura 3.121 - Panel Configuraciones (desde objetivo)	140
Figura 3.122 - Panel configuraciones (desde instrucción de movimiento)	141

Figura 3.123 - Modificación de los argumentos de la instrucción de movimiento en la Barra de estado	143
Figura 3.124 - Editar plantilla de instrucciones	143
Figura 3.125 - Plantilla de instrucciones predeterminada	143
Figura 3.126 - Plantilla de instrucciones con la opción Conveyor Tracking activada en el controlador y el Painting PowerPac	144
Figura 3.127 - Cuadro de dialogo Añadir instrucción	145
Figura 3.128 - Cuadro de dialogo Sincronizar con RAPID	146
Figura 3.129 - Ejemplo de visualización de objetivos y trayectorias sincronizadas con RAPID	147
Figura 3.130 - Pestaña Componer de la Lógica de estación	148
Figura 3.131 - Opciones de la herramienta Añadir componente.....	149
Figura 3.132 - Cuadro de dialogo Guardar estado actual	150
Figura 3.133 - Pestaña Diseño de la Lógica de estación	151
Figura 3.134 - Pestaña Propiedades y enlazamientos	153
Figura 3.135 - Pestaña Señales y conexiones	154
Figura 3.136 - Componente inteligente LogicGate	157
Figura 3.137 - Componente inteligente LogicSplit.....	158
Figura 3.138 - Componente inteligente LogicSRLatch	158
Figura 3.139 - Componente inteligente Comparer	159
Figura 3.140 - Componente inteligente LineSensor	160
Figura 3.141 - Componente inteligente Rotator2.....	160
Figura 3.142 - Componente inteligente PoseMover.....	161
Figura 3.143 - Componente inteligente RapidVariable	162
Figura 3.144 - Componente inteligente Highlighter	163
Figura 3.145 - Componente inteligente LightControl	164
Figura 3.146 - Componente inteligente ColorTable.....	165
Figura 3.147 - Componente inteligente PaintApplicator	166
Figura 3.148 - Ventana llamada Configuración – I/O System	168
Figura 3.149 - Cuadro de dialogo Instancia editor con los parámetros comunes a todas las señales	168
Figura 3.150 - Ventana Configuración de simulación	170
Figura 3.151 - Ejemplo de señales de E/S de un controlador, de una estación y de un componente inteligente.....	172
Figura 3.152 - Panel Cronómetro	173
Figura 3.153 - Panel Modificar conjunto de colisión	175
Figura 3.154 - Visualización de un caso de casi colisión y un caso de colisión.....	176
Figura 3.155 – Configuración de las opciones de simulacion de colisiones de RobotStudio	176
Figura 3.156 - Painting PowerPac.....	178
Figura 3.157 - Complemento Equipment Builder	180
Figura 3.158 - Ejemplo de los paneles del complemento Equipment Builder	180
Figura 3.159 - Ejemplo sobre las reglas de los diferentes ámbitos de los datos.....	186
Figura 3.160 - Tipos de módulos de RAPID.....	192
Figura 3.161 - Cuadro de dialogo Crear módulo.....	193
Figura 3.162 - Rutinas de RAPID	194
Figura 3.163 - Cambio del valor del límite de velocidad máxima del TCP del robot	200
Figura 3.164 - Grupos de instrucciones	202
Figura 3.165 - Logotipo de SolidWorks.....	232
Figura 3.166 - Ejemplos de diseños mediante SolidWorks	232

Figura 3.167 - Logotipo del software Blender	233
Figura 3.168 - Modelo 3D CAD final de la carrocería Mercedes Benz Clase A A45 AMG	233
Figura 3.169 - Logotipo del software SketchUp	234
Figura 3.170 - Logotipo del software MeshLab	234
Figura 4.1 - Disposición de los robots y los tracks del lado derecho de la estación	238
Figura 4.2 - Disposición global de la estación	238
Figura 4.3 - Modelo de robot IRB 5500-23	239
Figura 4.4 - Modelo de robot IRB 5500-25	239
Figura 4.5 – Modelo de robot IRB 5350 – versión de 4 ejes	241
Figura 4.6 - Modelo de atomizador IRB1000i SAD-L.....	242
Figura 4.7 - Modelo de agarradera IRB 5350 Gripper.....	242
Figura 4.8 - Componente Transportador	243
Figura 4.9 – Componente cabina de control IRC5P	244
Figura 4.10 - Componente FlexPendant	245
Figura 4.11 - Componente Pedestal	245
Figura 4.12 - Componente Puerta	246
Figura 4.13 - Componente Suelo	246
Figura 4.14 - Componente Mecanismo trampilla de mantenimiento	247
Figura 4.15 - Componente Paredes y techo de la estación.....	247
Figura 4.16 – Dimensiones en el alzado, planta y perfil del componente Paredes y techo de la estación.....	248
Figura 4.17 - Componente Valla de protección	249
Figura 4.18 - Vista en perspectiva del modelo inicial de carrocería	250
Figura 4.19 - Vistas del modelo inicial de carrocería	251
Figura 4.20 - Modelo de la puerta sin interior	252
Figura 4.21 - Modelo tridimensional de puerta genérico con armazón interior.....	252
Figura 4.22 - Modelos de puertas terminadas.....	253
Figura 4.23 - Vista en perspectiva del modelo final de carrocería con las puertas cerradas	253
Figura 4.24 - Vista en perspectiva del modelo final de carrocería con las puertas abiertas	254
Figura 4.25 - Vistas del modelo final de la carrocería con las puertas abiertas	255
Figura 4.26 - Vistas del modelo final de la carrocería con las puertas cerradas	255
Figura 4.27 - Medidas del modelo final de carrocería (mm)	256
Figura 4.28 - Pilar soporte conjunto robótico superior	257
Figura 4.29 - Análisis estático simple del pilar de soporte.....	258
Figura 4.30 - Estructura de soporte de la carrocería	258
Figura 4.31 - Herramientas de agarre para el robot superior izquierdo (izquierda) y el robot superior derecho (derecha).....	259
Figura 4.32 - Elementos de agarre para el capó (izquierda) y el maletero (derecha)	259
Figura 4.33 - Limpiador automático del atomizador	260
Figura 4.34 - Posicionamiento de los limpiadores de los atomizadores	260
Figura 4.35 - Suelo de rejilla de la estación de pintado	261
Figura 4.36 - Componente Torre de señalización luminosa industrial.....	262
Figura 4.37 - Componente Operario.....	262
Figura 4.38 - Componente Puesto de control.....	263
Figura 4.39 - Conjunto de puntos objetivo y trayectorias de la estación finalizada	265
Figura 4.40 - Importación del robot IRB 5500-23	266
Figura 4.41 - Importación del robot IRB 5500-25	266
Figura 4.42 -Importación del robot IRB 5350 lado izquierdo.....	267

Figura 4.43 -Importación del robot IRB 5350 lado derecho	267
Figura 4.44 -Importación del track IRB 5500 Paint rail extended	267
Figura 4.45 -Importación del track IRB 5500 Elevated Rail	268
Figura 4.46 -Importación del track IRB 5350 Rail	268
Figura 4.47 - Importación de los atomizadores	269
Figura 4.48 - Robots y herramientas del lado derecho conectados.....	270
Figura 4.49 - Conjunto carrocería más bancada	270
Figura 4.50 - Posición del origen local del conjunto carrocería más bancada	271
Figura 4.51 - Creación del transportador	272
Figura 4.52 - Creación de los controladores - Primera ventana	273
Figura 4.53 - Creación de los controladores - Segunda ventana	273
Figura 4.54 - Creación de los controladores - Tercera ventana	274
Figura 4.55 - Creación de la conexión transportador - controlador	275
Figura 4.56 - Conexión creada para cada robot.....	276
Figura 4.57 - Parámetros de movimiento del transportador	277
Figura 4.58 - Parámetros del objeto del transportador.....	277
Figura 4.59 - Plantilla de instrucciones final de la estación	278
Figura 4.60 - Parámetros del objeto de trabajo WO_In_Int	279
Figura 4.61 - Vista superior de la nube de puntos objetivo	280
Figura 4.62 - Vista trasera en perspectiva de la nube de puntos objetivo.....	281
Figura 4.63 - Carrocería situada en el inicio de la Ventana de inicio del transportador	282
Figura 4.64 - Herramienta Insert Paint Stroke.....	284
Figura 4.65 - Panel Manual Path	284
Figura 4.66 - TCPs del atomizador RB1000i_EXT_SAD_L	285
Figura 4.67 - Creación de una cuadrícula sobre el elemento a pintar	286
Figura 4.68 - Cuadrícula creada para el componente capó	286
Figura 4.69 - Dimensiones del patrón de pintado	287
Figura 4.70 – Desplegable del Module1 del controlador Cont_Sup_Izq.....	290
Figura 4.71 - Procedimiento Pint_Bajo_Capo - Cont_Inf_Izq	296
Figura 4.72 - Procedimiento Pint_PI_Del - Cont_Inf_Izq	297
Figura 4.73 - Procedimiento Pint_PI_Tras - Cont_Inf_Izq.....	298
Figura 4.74 - Procedimiento Pint_Malet_Inf - Cont_Inf_Izq.....	298
Figura 4.75 - Procedimiento Pint_Bajos_1 - Cont_Inf_Izq	299
Figura 4.76 - Procedimiento Pint_Lateral - Cont_Inf_Izq.....	299
Figura 4.77 - Procedimiento Pint_Bajos_2 - Cont_Inf_Izq	300
Figura 4.78 - Procedimiento Pint_Bajo_Capo - Cont_Inf_Der	303
Figura 4.79 - Procedimiento Pint_PD_Del - Cont_Inf_Der.....	304
Figura 4.80 - Procedimiento Pint_PD_Tras - Cont_Inf_Der	304
Figura 4.81 - Procedimiento Pint_Maletero_Inf - Cont_Inf_Der	305
Figura 4.82 - Procedimiento Pint_Bajos_1_1 - Cont_Inf_Der	305
Figura 4.83 - Procedimiento Pint_Lat - Cont_Inf_Der.....	306
Figura 4.84 - Procedimiento Pint_Bajos_2_2 - Cont_Inf_Der	306
Figura 4.85 - Procedimiento Abertura_Capo - Cont_Sup_Izq.....	310
Figura 4.86 - Curva auxiliar para la programación de la trayectoria de apertura del capó	310
Figura 4.87 - Procedimiento Cierre_Capo - Cont_Sup_Izq	311
Figura 4.88 - Procedimiento Pint_Capo_Inf - Cont_Sup_Izq	312
Figura 4.89 - Procedimiento Pint_Capo_Sup - Cont_Sup_Izq.....	312
Figura 4.90 - Procedimiento Pint_Barra - Cont_Sup_Izq	313

Figura 4.91 - Procedimiento Pint_Techo - Cont_Sup_Izq	313
Figura 4.92 - Procedimiento Pint_Malet_Sup - Cont_Sup_Izq.....	314
Figura 4.93 - Procedimiento Abertura_Maletero - Cont_Sup_Der	317
Figura 4.94 - Curva auxiliar para la programación de la trayectoria de apertura del maletero	317
Figura 4.95 - Procedimiento Cierre_Maletero - Cont_Sup_Der.....	318
Figura 4.96 - Procedimiento Pint_Capo_Inf - Cont_Sup_Der	319
Figura 4.97 - Procedimiento Pint_Capo_Sup - Cont_Sup_Der.....	319
Figura 4.98 - Procedimiento Pint_Barra - Cont_Sup_Der	320
Figura 4.99 - Procedimiento Pint_Techo - Cont_Sup_Der	320
Figura 4.100 - Procedimiento Pint_Malet_Sup - Cont_Sup_Der	321
Figura 4.101 - Procedimiento Abertura_Cierre_PI_Del - Cont_Puertas_Izq.....	325
Figura 4.102 - Curva auxiliar para la programación de la trayectoria de apertura y cierre de la puerta delantera izquierda.....	326
Figura 4.103 - Procedimiento Abertura_Cierre_PI_Tras - Cont_Puertas_Izq	327
Figura 4.104 - Curva auxiliar para la programación de la trayectoria de apertura y cierre de la puerta trasera izquierda.....	327
Figura 4.105 - Procedimiento Abertura_Cierre_PD_Del - Cont_Puertas_Der	330
Figura 4.106 - Curva auxiliar para la programación de la trayectoria de apertura y cierre de la puerta delantera derecha	331
Figura 4.107 - Procedimiento Abertura_Cierre_PD_Tras - Cont_Puertas_Der.....	332
Figura 4.108 - Curva auxiliar para la programación de la trayectoria de apertura y cierre de la puerta trasera derecha	332
Figura 4.109 - Lógica de estación completa	343
Figura 4.110 - Conjunto de componentes inteligentes para detectar los componentes de la carrocería	344
Figura 4.111 – Posición de los sensores lineales LineSensor_P1 y LineSensor_P2	345
Figura 4.112 - Lógica del componente inteligente Detector_Componentes	345
Figura 4.113 - Posicionamiento de los sensores que detectan los componentes de la carrocería	346
Figura 4.114 - Componente inteligente Abertura_Capo	347
Figura 4.115 - Lógica del componente inteligente Abertura_Capo	347
Figura 4.116 - Componente inteligente Abertura_Maletero.....	348
Figura 4.117 - Lógica del componente inteligente Abertura_Maletero	349
Figura 4.118 - Componente inteligente Abertura_PIs	350
Figura 4.119 - Lógica del componente inteligente Abertura_PIs.....	351
Figura 4.120 - Componente inteligente Abertura_PDs	352
Figura 4.121 - Lógica del componente inteligente Abertura_PDs	353
Figura 4.122 - Conjunto de componentes inteligentes Paro_Puerta.....	354
Figura 4.123 - Posición del sensor lineal LineSensor_Puerta_Der	355
Figura 4.124 - Conjunto de componentes inteligentes Obtener_Contador	356
Figura 4.125 - Componente inteligente LogicGate_Pos_Parada	357
Figura 4.126 - Lógica del componente inteligente LogicGate_Pos_Parada	358
Figura 4.127 - Conjunto inteligente Paro_Emergencia.....	358
Figura 4.128 - Componente inteligente Transportador.....	359
Figura 4.129 - Conjunto de componentes inteligentes Abertura_Mantenimiento_Izq.....	360
Figura 4.130 - Conjunto de componentes inteligentes Abertura_Mantenimiento_Der.....	361
Figura 4.131 - Componente inteligente Aplicadores_Pintura	362
Figura 4.132 - Lógica del componente inteligente Aplicadores_Pintura	365

Figura 4.133 - Lógica del componente inteligente Obtener_Color.....	366
Figura 4.134 - Lógica del componente inteligente Aplicador_Sup_Izq.....	367
Figura 4.135 - Componente inteligente ColorTable.....	368
Figura 4.136 - Conjunto de componentes inteligentes Señales_Luminosas_Global	368
Figura 4.137 - Lógica del componente inteligente Comparador_Señales	369
Figura 4.138 - Lógica del componente inteligente Señales_Luminosas.....	370
Figura 4.139 - Configuración final de la simulación	373
Figura 4.140 - Estado_Inicial de la estación.....	374
Figura 4.141 - Lista de usuarios de señales de E/S para el controlador Cont_Sup_Izq	375
Figura 4.142 - Cronómetros de la estación.....	375
Figura 4.143 - Conjuntos de colisión de la estación	376
Figura 5.1 - Inicio de la simulación	377
Figura 5.2 - Simulación - Abertura del capó	378
Figura 5.3 - Simulación - Abertura del capó 2.....	379
Figura 5.4 - Simulación - Pintado de la zona del chasis debajo de la carrocería	380
Figura 5.5 - Simulación - Abertura puerta izquierda delantera	380
Figura 5.6 - Simulación - Abertura puerta derecha delantera y pintado de la superficie interior del capó.....	381
Figura 5.7 - Simulación - Pintado interior de las puertas delanteras	381
Figura 5.8 - Simulación - Cierre del capó	382
Figura 5.9 - Simulación - Pintado de la superficie exterior del capó.....	382
Figura 5.10 - Simulación - Cierre puertas delanteras, pintado barra y pintado techo	383
Figura 5.11 - Simulación - Abertura puertas traseras	383
Figura 5.12 - Simulación - Pintado interior puertas traseras	384
Figura 5.13 - Simulación - Pintado exterior del maletero	384
Figura 5.14 - Simulación - Cierre puertas traseras y apertura maletero	385
Figura 5.15 - Simulación - Pintado de la superficie interior del maletero.....	385
Figura 5.16 - Simulación - Limpieza del atomizador superior izquierdo	386
Figura 5.17 - Simulación - Cambio de color del atomizador superior izquierdo	386
Figura 5.18 - Simulación - Desplazamiento de los robots IRB5350 a su nueva posición intermedia.....	387
Figura 5.19 - Simulación - Pintado de la zona de la rueda delantera izquierda	387
Figura 5.20 - Simulación - Cierre del maletero	388
Figura 5.21 - Simulación - Pintado de la zona de la rueda delantera derecha	388
Figura 5.22 - Simulación - Pintado de los laterales de la carrocería	388
Figura 5.23 - Pintado de la zona de las ruedas traseras	389
Figura 5.24 - Simulación - Limpieza de los atomizadores inferiores	389
Figura 5.25 - Simulación - Cambio de color de los atomizadores inferiores	390
Figura 5.26 - Simulación - Torres de señalización luminosa	390
Figura 5.27 - Simulación - Torres de señalización luminosa 2	391
Figura 5.28 - Simulación - Tiempo de un ciclo completo de pintado	391
Figura 5.29 - Simulación - Tiempo de un ciclo de pintado sencillo	391
Figura 5.30 - Simulación - Tiempo 10 ciclos de pintado completos	391
Figura 5.31 - Señales de E/S del controlador Cont_Inf_Der	392
Figura 5.32 - Paro del robot.....	393
Figura 5.33 - Ejecución de la trayectoria de salida	393
Figura 5.34 - Ejecución de la trayectoria de salida 2	393
Figura 5.35 - Limpieza del atomizador inferior derecho	393

Figura 5.36 - Posición intermedia de reposo	394
Figura 5.37 - Iluminación de las torres de señalización en verde	394
Figura 5.38 - Reposicionamiento en el punto de paro	394
Figura 5.39 - Mantenimiento atomizador inferior derecho.....	396
Figura 5.40 - Mantenimiento atomizador inferior derecho 2.....	396
Figura 5.41 - Reanudación del movimiento de todos los robots donde se detuvieron	397
Figura 5.42 - Paro debido a la abertura de una de las puertas de acceso a la estación.....	398
Figura 5.43 - Robots esperando en la posición de reposo	398

ÍNDICE DE TABLAS

Tabla 2.1 – Características principales de tres imprimaciones diferentes, una de cada tipo	28
Tabla 2.2 - Comparativa de diferentes propiedades para los tres procesos de aplicación principales.....	39
Tabla 3.1 - Configuraciones posibles para el cuadrante cfx para los robots de 6 ejes con eslabones en serie.....	137
Tabla 3.2 - Configuraciones posibles para el cuadrante cfx para los robots de 7 ejes.....	138
Tabla 3.3 - Palabras reservadas que hacen referencia a elementos propios del lenguaje de RAPID	182
Tabla 3.4 - Operadores y expresiones aritméticas en RAPID	183
Tabla 3.5 - Operadores y expresiones lógicas en RAPID	184
Tabla 4.1 - Posiciones y orientaciones de los tracks respecto del sistema de coordenadas mundo.....	269
Tabla 4.2 - Procedimientos en cada uno de los controladores de robot	289
Tabla 4.3 - Rutinas TRAP en cada uno de los controladores de robot	289
Tabla 4.4 - Señales de E/S de todos los controladores virtuales de la estación.....	372

1. INTRODUCCIÓN

Alrededor del mundo los diferentes sectores industriales se enfrentan cada año a múltiples adversidades. La primera de ellas es la continua demanda de productos personalizados, es decir, un mercado cambiante que demanda nuevos productos constantemente, aspecto que hace que el ciclo de producción sea cada vez más complejo y cambiante. La segunda sería la escasez de mano de obra cualificada en algunos países, lo cual causa, en particular en las pequeñas y medianas empresas, que no se puedan hacer frente a picos de producción. Además de estas dos se tienen muchas otras como la incertidumbre geopolítica, el exceso de regulación, etc.

Para intentar hacer frente a esto, los fabricantes están recurriendo a aumentar la automatización para seguir siendo competitivos en este entorno difícil. Así pues, a finales de 2019 se tenían 2,7 millones de robots trabajando en todo el mundo, de los cuales, más de tres cuartas partes en la industria manufacturera, sobre todo en los sectores de la automoción y la electrónica.

Estos dos sectores en conjunto representan el 59% de los robots en funcionamiento, y esta cifra continúa en aumento. Entre 2014 y 2019 el número de robots en el sector automovilístico creció un 10% de media anual y en el sector de la electrónica un 19% de media anual.

En otros sectores industriales, como el de la alimentación y las bebidas, el textil, los productos de madera y los plásticos, menos familiarizados con la automatización robótica años atrás, los actuales avances en las tecnologías robóticas, como la movilidad, los sistemas de visión, los sistemas de agarre, la conectividad y la facilidad de programación, están contribuyendo a aumentar su presencia de forma considerable, aunque por ahora continúa haciéndose paulatinamente. En la siguiente Figura 1.1 se observa el crecimiento de la utilización de robots en diferentes sectores.

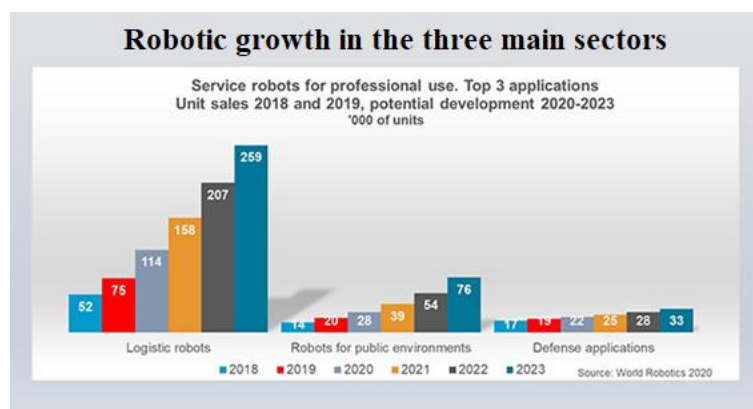


Figura 1.1 - Utilización de los robots en el sector logístico, público y militar en el mundo

Desde el punto de vista humano, el número de robots por trabajador sigue aumentando año tras año. En el año 2019, el número de robots por cada 10.000 trabajadores, también conocido como densidad de robots, en la industria manufacturera era de 113, representando esto un aumento del 12% respecto al año anterior. Esta densidad varía notablemente en función del

país, desde 855 robots por cada 10.000 trabajadores en Corea del Sur hasta 1 en Egipto, Perú o Ucrania. En la siguiente Figura 1.2 se puede observar la densidad de robots en el año 2019 para algunos de los principales países manufactureros.

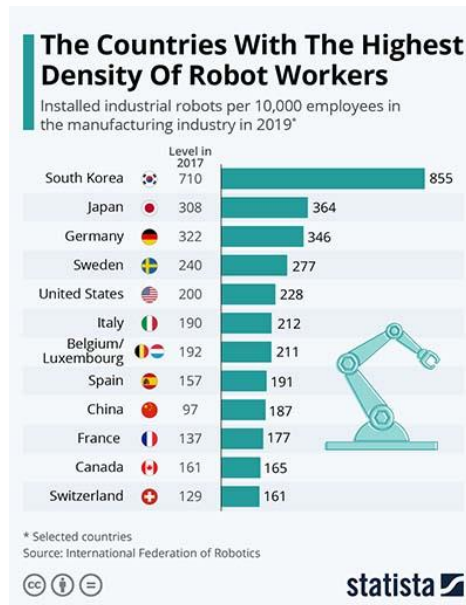


Figura 1.2 - Densidad de robots en el año 2019 para algunos de los principales países manufactureros

Los continuos avances en las tecnologías de automatización han ampliado el rango de tareas que pueden llevar a cabo los robots, siendo estos cada vez más fáciles de instalar, programar y reutilizar. La digitalización y los avances en los protocolos de comunicación facilitan la conexión de los robots con otras máquinas del proceso de producción.

Uno de estos importantes avances es el continuo desarrollo, por parte de los fabricantes de sistemas robóticos, de softwares destinados al control de sus sistemas robóticos. Estos softwares permiten realizar el modelado, la configuración, la programación y la simulación de células robotizadas fuera de línea. Lógicamente, también permiten realizar la configuración y programación de los robots físicos directamente en la planta de producción.

En muchas de las aplicaciones para las que se emplean robots se tienen cientos o miles de puntos objetivo que el robot deberá alcanzar. Programar tal cantidad de objetivos trasladando el robot manualmente hasta cada uno de ellos en la estación real conllevaría parar la producción y un enorme gasto de tiempo y personal, lo que en definitiva conlleva una pérdida monetaria y de competitividad. En su lugar esta tarea se lleva a cabo mediante la programación fuera de línea u offline, unida a la simulación, representado por lo tanto un gran avance para la automatización.

La simulación puede ser utilizada por la compañía que se encarga de la programación de los robots como un herramienta de demostración conceptual del funcionamiento del sistema robótico, y por lo tanto, como una herramienta de ventas. Mediante la simulación se pueden detectar posibles colisiones entre el robot, las herramientas, las fijaciones y cualquier elemento de seguridad de la estación. Además, se pueden analizar los límites de los ejes del robot, las singularidades y los problemas de alcance.

Además de estas, una de las principales ventajas de la programación fuera de línea es que la producción de la estación real puede continuar en marcha mientras se programan por otro lado

los próximos trabajos de esa estación. Una vez se tienen programados en el ordenador y la estación real esta lista, simplemente se cambia su programa y se retoma la producción en un espacio de tiempo relativamente bajo.

Dicho esto, cabe destacar que la programación fuera de línea es idónea y rentable para aplicaciones de programación de trayectorias complejas que requieren muchos puntos objetivo, como la soldadura, el acabado, el corte por láser, el desbarbado, la pulverización térmica, la pintura, el revestimiento por láser, la fabricación aditiva, etc. Por el contrario, para aplicaciones que requieren programar simplemente unos pocos puntos objetivo, como aplicaciones sencillas de pick-and-place, ensamblaje, embalaje o paletización, la programación fuera de línea no es rentable debido a que se invierte una mayor cantidad de tiempo y de mano de obra especializada, siendo más interesante programar el robot manualmente en la estación real.

1.1 Motivación

Como se ha expuesto en los párrafos anteriores, el sector de la automatización robótica es un sector en auge, que está tomando una enorme importancia tanto a nivel industrial como a nivel social, ya que, junto con las inteligencias artificiales, provocarán que muchos de los puestos de trabajo hasta ahora ocupados por trabajadores humanos pasen a ser ocupados por sistemas robóticos inteligentes.

Así pues, la motivación principal del presente proyecto es la de adquirir la mayor cantidad posible de conocimientos en lo referente a la programación fuera de línea de sistemas robóticos. Particularmente, puesto que la universidad dispone de un acuerdo con la empresa ABB para proporcionar a los alumnos que lo soliciten la licencia de su software de programación de robots RobotStudio, se buscará aprender y ganar experiencia en la programación de robots de ABB fuera de línea mediante este software. Esto puede resultar interesante, y en cierto modo ventajoso, de cara a solicitar un puesto de trabajo en una empresa dedicada a realizar tareas similares, como por ejemplo una empresa integradora de sistemas robóticos.

Para ello se ha seleccionado el sector de la pintura industrial robotizada, particularmente en el sector de la automoción, puesto que, aunque es un sector ampliamente desarrollado y perfeccionado, se concibe como un tema muy interesante, se entiende que es un sector con una gran demanda y que a través de la investigación del mismo se obtendrán un gran número de conocimientos referentes a los sistemas robóticos y su programación fuera de línea.

Otra de las motivaciones es tener un primer contacto, aunque de una forma sencilla y más o menos teórica, con el sector robótico industrial, de forma que un futuro se puedan ampliar estos conocimientos y se pueda optar a formar parte del mismo.

Por último, también es una motivación el poder llevar a la práctica algunos de los conocimientos adquiridos a lo largo del Master en Robótica.

1.2 Objetivos

El objetivo principal del presente trabajo será el diseño y programación, mediante el software RobotStudio del fabricante ABB, de una estación robótica de pintado en seguimiento para la aplicación de la capa base del recubrimiento de la carrocería del modelo de automóvil Mercedes Benz Clase A A45 AMG, adquiriendo a lo largo de su desarrollo la experiencia y los conocimientos necesarios para llevarlo a cabo correctamente y de la forma más fiel a la realidad posible.

Otro objetivo de este trabajo es dotar a la estación de la mayor flexibilidad posible en cuanto a la aplicación de la capa base, el mantenimiento de los equipos y la limpieza de la estación y los equipos.

El último de los objetivos será reducir al máximo posible el tiempo de ciclo, es decir, el tiempo empleado para llevar a cabo la aplicación de la capa base sobre la carrocería del vehículo.

1.3 Estructura de la memoria

La memoria de este trabajo se estructura en 6 capítulos principales, de forma que cada capítulo proporcione los conocimientos necesarios para la correcta comprensión de los siguientes. Además de los capítulos principales se tendrán los capítulos de Bibliografía y Anexos.

Capítulo 1 - Introducción. En este apartado se expone el contexto del presente trabajo, la motivación para realizarlo, los objetivos que se persiguen y como se encuentra estructurada la memoria del trabajo para una comprensión preliminar del mismo.

Capítulo 2 - Estado del arte. En este apartado se expone la investigación documental realizada sobre el fabricante ABB en el ámbito de la pintura industrial robotizada y sobre las estaciones de recubrimiento de carrocerías de vehículos. Este apartado es de vital importancia para obtener los conocimientos necesarios, en cuanto al pintado de carrocerías de vehículos, que permitirán el correcto diseño de la estación del presente proyecto.

Capítulo 3 – Metodología. En este apartado se expone una extensa explicación de los conceptos, herramientas y mecanismos clave del software RobotStudio y su lenguaje de programación RAPID, cuyo conocimiento es imprescindible para llevar a cabo el diseño y programación de la estación. Para una correcta comprensión de todos estos conceptos se exponen multitud de ejemplos e imágenes a lo largo de las explicaciones, ya que al tratarse de un software con una parte gráfica muy importante y potente, se entiende que esta es una mejor forma para comprenderlos.

Capítulo 4 - Diseño y programación de una estación robótica de pintado en seguimiento de la carrocería de un automóvil. En este apartado se expone y explica todo el proceso que se ha

llevado a cabo para realizar el diseño y programación de la estación robótica, de manera que se obtengan los resultados expuestos en el siguiente capítulo.

Capítulo 5 - Resultados obtenidos. En este apartado se describe el funcionamiento de la estación diseñada, basándose en la simulación del proceso completo de aplicación de la capa base sobre la carrocería y en la simulación de los procesos de mantenimiento, limpieza y seguridad programados.

Capítulo 6 - Conclusiones y líneas futuras de trabajo. En este apartado se exponen las conclusiones que se han obtenido tras finalizar el desarrollo del proyecto, basándose en los objetivos y requisitos impuestos en su inicio. Por otro lado, también se exponen posibles nuevas líneas de investigación para mejorar o desarrollar más la estación diseñada en el presente proyecto.

Capítulo 7 – Bibliografía. En este capítulo se indican todas las referencias bibliográficas empleadas para realizar el presente trabajo.

Capítulo 8 – Anexos. Este apartado se emplea para añadir información complementaria, incluyendo los programas completos de los robots que forman la estación y las hojas de características de los mismos.





2. ESTADO DEL ARTE

2.1 Evolución histórica del pintado robótico en ABB

La historia del fabricante ABB en la pintura industrial robotizada comienza en 1969 en Bryne, Noruega, cuando un grupo de ingenieros mecánicos de la empresa Trallfa, nombre original de la factoría de robots ABB en Bryne, encabezados por Ole Molaug, desarrollaron el primer robot de pintura para uso industrial (Figura 2.1), siendo poco después exportado a una compañía sueca para realizar trabajos de esmaltado de bañeras, donde impresionó por tener una precisión nunca vista anteriormente. Este robot fue el pionero que inspiró el desarrollo por parte de ABB de un gran número de soluciones para la pintura industrial robotizada a lo largo de los siguientes 50 años.



Figura 2.1 - Primer robot de pintura para uso industrial

Hoy en día los robots de pintura de ABB tienen poco que ver con el prototipo hidráulico simple que se encuentra expuesto en el centro ABB R&D en Bryne, Noruega. Actualmente estos robots son máquinas de alta tecnología industrial con múltiples funcionalidades integradas, siendo así capaces de obtener los acabados más precisos requeridos en las diferentes industrias.

A lo largo de su historia ABB ha desarrollado muchas innovaciones industriales que han transformado el estado del arte de la pintura industrial. ABB reparó rápidamente en los beneficios que proporcionaba habilitar digitalmente sus robots de uso general e introdujo el control mediante ordenador o control computarizado de los robots en 1975, hecho que cambió el almacenamiento de programas, desde el almacenamiento en cinta al almacenamiento digital en un disquete. Este cambio permitió realizar copias de programas evitando así invertir una gran cantidad de tiempo programando cada robot.

Menos de una década después, en 1984, ABB marcó el comienzo de la era del atomizador de pintura con el Robobell II, un atomizador rotativo de alta eficiencia que fue diseñado para recubrir superficies complejas con pintura a base de disolvente.

En el ámbito de la automatización de la pintura, ABB comenzó centrando sus esfuerzos en aumentar la eficiencia y la facilidad de utilización de sus sistemas. Una vez alcanzados estos propósitos, ABB se ha centrado en hacer de sus robots máquinas de pintura potentes con una serie de funcionalidades integradas en un solo dispositivo. Su último lanzamiento es un claro ejemplo de esto, el robot IRB 5500 – 27 (Figura 2.2).



Figura 2.2 - Robot de pintura IRB 5500-27

Este robot es una variante de 7 ejes del robot IRB 5500 FlexPainter altamente especializada para cubrir las exigencias de la industria automotriz en la aplicación de pintura. Sus 7 ejes le permiten tener un rango de trabajo más amplio y un mejor acceso a todas las superficies de pintado de los vehículos, tanto exteriores como interiores. Debido a este mayor rango de trabajo y su versatilidad de movimientos el robot puede emplearse para trabajar con diversos tamaños de carrocerías de automóviles, pudiendo reducir el número de robots necesarios en la línea de pintado hasta en un 25%. Además, con la integración del modelo dinámico de ABB para el control de todos los ejes se consigue una reducción del consumo total de energía, se aumenta la precisión y se asegura que los movimientos del robot sean completamente sincronizados y equilibrados.

El compromiso de la firma ABB con sus clientes unido a su habilidad para adaptarse rápidamente a las demandas cambiantes del mercado le han llevado a ser líderes del mercado a nivel mundial. Un claro ejemplo de la adaptación de la marca a la actual y futura industria 4.0 fue el lanzamiento en 2018 del primer atomizador robótico de pintura digital conectado del mundo, el ABB Ability™ Connected Atomizer o RB1000i (Figura 2.3).



Figura 2.3 - ABB Ability™ Connected Atomizer o RB1000i

El RB1000i es un gran avance para industrias como la de la automoción, que requieren acabados de pintura impecables en cuestión de minutos. Este atomizador aumenta las capacidades digitales de la compañía implementando sensores y la plataforma digital ABB Ability en el mismo, para obtener datos inteligentes que permiten mejorar la eficiencia del pintado y reducir en un 75% el desperdicio de pintura debido a los frecuentes cambios de color. Los sensores detectan problemas de calidad a medida que se lleva a cabo el proceso de pintado, ayudando así a las empresas a detectar y corregir los acabados inadecuados que anteriormente solo podían ser detectados durante las inspecciones tras el proceso completo de pintado. Los sensores de temperatura y vibración y un giroscopio integrados en el atomizador transmiten un conjunto de datos acerca de la estado general del dispositivo y la eficiencia de transferencia de la pintura. Al controlar el estado de los componentes clave del atomizador, como las campanas rotativas, los motores neumáticos y el cabezal de aire, así como las variables de aceleración, presión, vibración y temperatura, se puede aumentar la eficiencia de la transferencia de pintura hasta en un 10% y se puede reducir el consumo de aire comprimido hasta en un 20%, lo que puede suponer un ahorro significativo a largo plazo. Por último, los datos extraídos de varios atomizadores conectados se pueden analizar conjuntamente para crear una perspectiva completa del proceso de pintado en tiempo real.

Mientras que el proceso de pintado es muy similar desde hace muchos años, ABB tiene el compromiso de desarrollar y mejorar su equipamiento para la aplicación de pintura, de modo que se mantenga líder en este sector proporcionando una excelente precisión de pintado.

Aprovechando su amplia red de experiencia y colaboración con sus clientes, ABB ha diseñado y producido equipos de proceso que hacen posible cambios de color rápidos que requieren un mantenimiento mínimo, así como atomizadores que reducen los residuos de pintura y aumentan la eficiencia de transferencia. También ha desarrollado equipo especializado para el proceso de limpieza, como el ABB FeatherDuster (Figura 2.4), el cual elimina las partículas de polvo de un vehículo antes de comenzar el proceso de sellado y pintado, asegurando resultados de pintado perfectos y protección de principio a fin.



Figura 2.4 - ABB FeatherDuster

Por otro lado, los controladores específicos para robots de pintura como el IRC5P aportan la flexibilidad, la seguridad y el control necesarios para el proceso de pintado, mientras que el software de automatización de pintura minimiza el tiempo necesario para la programación del robot. RobotStudio, el software de programación offline de ABB, reduce los plazos de entrega al permitir a los usuarios programar y verificar los tiempos de ciclo en sus ordenadores antes incluso de que el robot físico se coloque en la fábrica.

Los 50 años de historia y experiencia de la compañía ABB en el ámbito de la pintura industrial robotizada muestran cómo el equipo de ABB combina la ciencia y la tecnología con la innovación y la imaginación humana.

En las siguientes Figuras 2.5 y 2.6 se muestra la cronología simplificada de los principales hitos de la compañía ABB en el ámbito de la pintura industrial robotizada.

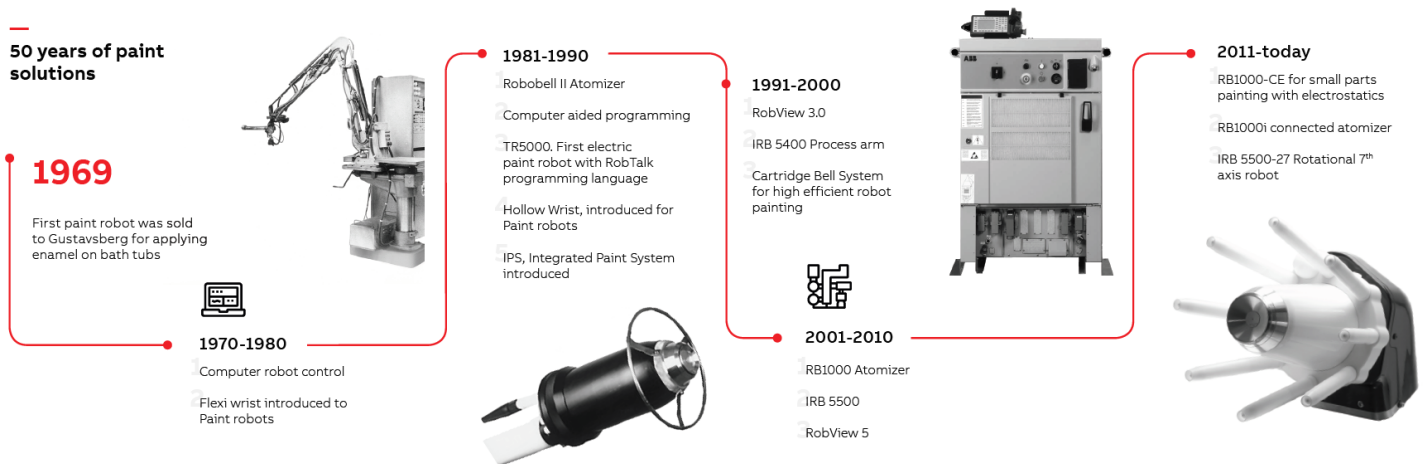


Figura 2.5 - Cronología de los avances de ABB en la aplicación robotizada de pintura

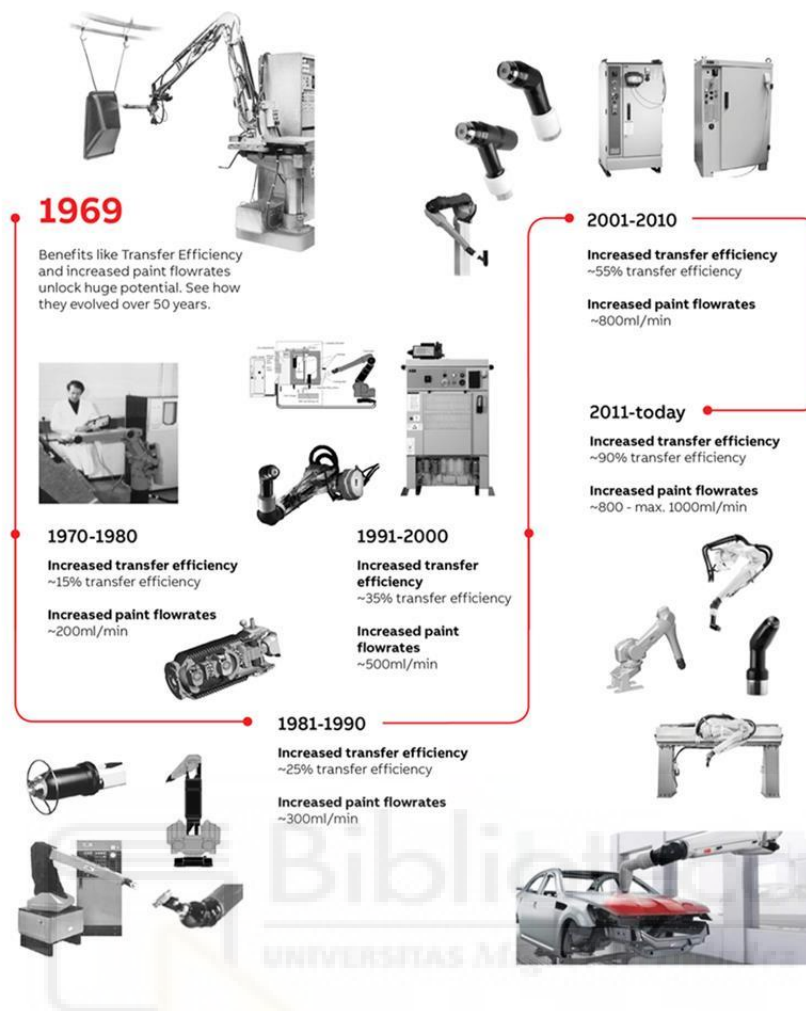


Figura 2.6 – Cronología de las mejoras que han supuesto los avances de ABB en la aplicación robotizada de pintura

A lo largo de estos 50 años de historia, ABB ha establecido seis centros líderes en China, India, Japón, Alemania, Italia y Estados Unidos que apoyan los proyectos y capacidades de ingeniería de la compañía. En todo el mundo alrededor de 550 empleados se centran en llevar la pintura industrial a la perfección.

2.2 Pintado de vehículos

2.2.1 Estación de recubrimiento

El proceso de fabricación de un vehículo consta de un gran número de áreas de producción, siendo la estación de recubrimiento una de las más complejas. En esta, la calidad visual y funcional de la pintura, la productividad y el impacto medioambiental son los requisitos prioritarios. Para poder alcanzarlos las estaciones de recubrimiento han evolucionado hasta alcanzar un gran nivel de automatización.

En la mayoría de estaciones de recubrimiento cada uno de los procesos de recubrimiento de los diferentes componentes está organizado de manera que se obtenga un flujo de material lo más simple y coherente posible, en relación a las conexiones entre la estación de pintado y las áreas de producción colindantes, la estación de carrocería, la línea de ensamblado, etc.

Una línea de recubrimiento de automóviles estándar capaz de realizar 60 unidades por hora tiene una longitud de alrededor de 2 kilómetros, siendo el tiempo medio de permanencia del vehículo dentro de la misma de entre 6 y 11 horas. En una estación de recubrimiento totalmente automatizada, se necesitan entre 30 y 50 empleados por turno para realizar tareas de mantenimiento, control de procesos y solución de problemas.

La cadena de procesos de la estación incluye procesos de valor añadido (VA) y procesos de no valor añadido (NVA). Los procesos de no valor añadido normalmente son trabajos manuales como reparación de fallos en la carrocería, lijado y pulido, limpieza, repintado, etc. Uno de los objetivos futuros es la eliminación completa o la reducción al máximo de estos procesos de no valor añadido, ya que repercuten en las pérdidas económicas. Por otro lado, actualmente, los procesos de valor añadido han alcanzado casi por completo su total automatización. El constante requerimiento de reducción de costes de producción por unidad lleva al desarrollo y mejora de los procesos de recubrimiento y a una mejor asociación entre los clientes y los proveedores.

2.2.2 Etapas de la estación de recubrimiento

La estación de recubrimiento engloba una amplia variedad de procedimientos dentro de los procesos de preparación, pintado y acabado del automóvil. Estos se agrupan y clasifican según el lugar en el que se llevan a cabo a lo largo de la cadena de procesos. A continuación se exponen y explican brevemente cada uno de ellos.

Pretratamiento

El pretratamiento es la primera etapa de la estación de recubrimiento. En él se acondiciona el sustrato metálico de la pieza o carrocería para que pueda ser posible un correcto recubrimiento posterior de la misma.

En las operaciones del taller de carrocería los lubricantes, que contienen muchos compuestos como el grafito o el disulfuro de molibdeno, deben evitarse siempre que sea posible. Además, normalmente se aplican selladores en el taller de carrocería. El material que se ha aplicado descuidadamente o que está en exceso debe ser retirado, realizándose manualmente si es necesario. En cualquier caso, para obtener resultados libres de defectos, se debe instalar una estación de limpieza de carrocerías en el taller de carrocería para eliminar todos los contaminantes pesados en la medida de lo posible antes de entrar en la zona de pretratamiento.

El pretratamiento consta de un determinado número de procesos de prelavado o limpieza previa, desengrasado, purgado y fosfatado. El prelavado elimina los contaminantes/suciedad de mayor tamaño. El desengrasado disuelve las grasas, como por ejemplo, la grasa procedente de los procesos de embutición profunda, aceites, ceras y otros contaminantes procedentes de

procesos de trabajo anteriores. El fosfatado, tras el proceso de purga, se emplea como protección temporal frente a la corrosión y para mejorar la capacidad de adhesión de la película de pintura en su posterior aplicación.

Electrorecubrimiento o electrodeposición

El proceso de electrorecubrimiento es el segundo proceso de modificación de la superficie de trabajo después del pretratamiento y antes de la aplicación de la capa de imprimación o de la capa base en la misma. El electrorecubrimiento en la industria de la automoción está diseñado para proporcionar a las carrocerías de los vehículos la mayor protección posible contra la corrosión.

El proceso de electrorecubrimiento presenta una serie de beneficios, siendo los más relevantes los siguientes:

- Consigue un recubrimiento completo y uniforme, incluso dentro de los huecos y cavidades.
- Logra una gran protección frente a la corrosión del material base.
- Tras la aplicación no se forman gotas o corrimientos de la pintura.
- Empleando un correcto método de purgado posterior, se tienen unas pérdidas de pintura muy bajas.
- Buena capacidad de procesado y de control de calidad
- Las instalaciones de electrorecubrimiento son completamente automáticas y por lo tanto pueden procesar piezas de forma continua.

Por otro lado este proceso también presenta algunos inconvenientes:

- Requiere una inversión alta para la construcción de la instalación.
- Coste de material relativamente alto.
- Gastos considerables para el control del baño de pintura.

Desde alrededor de 1975 se emplea el electrorecubrimiento catódico por inmersión en el proceso de pintado de vehículos, ya que presenta una serie de ventajas sobre el recubrimiento anódico como son una mejor protección frente a la corrosión, la distribución uniforme del espesor de la capa de pintura, un buen recubrimiento de bordes y un menor consumo de energía. Este proceso presentaba una serie de desventajas en sus inicios, como un menor espesor de la capa de pintura y la necesidad de altas temperaturas de cocción, que han sido considerablemente reducidas gracias a los avances en los materiales de pintura.

Para llevar a cabo este proceso la carrocería se sumerge verticalmente en el baño catódico, realizando un movimiento rotativo dentro del baño (Figura 2.7). Las pinturas empleadas en el baño son suspensiones de agentes aglutinantes y pigmentos en agua desionizada con una pequeña proporción de disolventes orgánicos, aproximadamente un 3%. El agua desionizada actúa como medio portador para los aglutinantes y pigmentos que están bajo agitación constante. Los agentes aglutinantes son la columna vertebral de la capa de pintura final y son los responsables de la protección contra la corrosión, su durabilidad y su dureza. Los pigmentos se emplean para proporcionar color y brillo.



Figura 2.7 - Proceso de electrorecubrimiento catódico

El electrorecubrimiento contempla todos los procesos de pintura por inmersión, donde la pintura precipita en la pieza de trabajo debido a la conversión química y la coagulación asociada al aglutinante. Las conversiones químicas se producen debido a un flujo de corriente eléctrica que circula desde un electrodo externo a través de la pintura conductora hasta la pieza de trabajo.

Debido a su gran protección contra la corrosión y su buena capacidad de adhesión con el material base y con las capas superiores de pintura, se emplean resinas epoxi como aglutinante para las pinturas utilizadas en el recubrimiento catódico por inmersión.

Normalmente estas resinas no son solubles al agua, por lo que mediante una reacción, llamada reacción de neutralización, con ácidos orgánicos, como el ácido acético, se modifican para alcanzar un estado en el que sí lo son. El agua en los electrodos y en la pieza de trabajo es electrolizada por medio de la corriente eléctrica, liberándose los gases H_2 y O_2 y los iones. Los iones hidroxilo OH^- creados en la interfaz de la pieza de trabajo catódicamente polarizada causan una inversión de la reacción de neutralización y conducen a una coagulación del aglutinante.

La capa de partículas de pintura coaguladas que está suelta inicialmente es empujada hacia el electrodo por las partículas de pintura que están vagando debido a que todavía están cargadas eléctricamente. En el electrodo, el material coagulado con alto contenido de agua es fijado y comprimido por el material que se continúa acumulando. Al mismo tiempo, el campo eléctrico provoca la eliminación de iones situados a lo largo de las capas de hidratación. Esto conduce además a una deshidratación y solidificación de la capa de pintura por medio de electro-ósmosis. En poco tiempo, la parte no volátil de la capa de pintura aumenta a más del 90%, aunque el baño de pintura en sí sólo tiene alrededor del 20%. Así, la capa de pintura alcanza una firmeza que le permite resistir los posteriores procesos de aclarado.

Así pues, la precipitación de las partículas de pintura no se lleva a cabo por la atracción de las mismas a la superficie de la pieza de trabajo a causa del campo eléctrico, más bien los factores determinantes para el recubrimiento son los procesos electroquímicos en la capa de difusión controlada en la superficie catódica y las reacciones de los aglutinantes. El transporte de masa se determina únicamente por convección. Por este motivo se requiere una recirculación constante del baño de inmersión.

Un sistema industrial de electrorecubrimiento catódico empleado en la industria de la automoción se compone de los siguientes componentes principales, que pueden observarse en la Figura 2.8:

- Un tanque de electrorecubrimiento y un tanque de aclarado.



Figura 2.9 - Aplicación de los materiales de sellado

Pintado

Una vez se ha protegido el metal base frente a la corrosión y se ha conseguido una superficie suave y uniforme se procede al pintado de la misma. El proceso de pintado consta de un número variable de etapas según el procedimiento que se elija, pudiendo existir o no una primera capa de imprimación, una o varias capas base, que pueden aplicarse mediante diferentes métodos, y una capa de revestimiento transparente.

Puesto que el presente trabajo se centra en esta etapa del proceso de recubrimiento de la carrocería, esta se expondrá de forma mucho más detallada en el siguiente apartado [2.2.4 Estación de pintado](#), exponiendo las diferentes capas junto con sus características y métodos de aplicación, así como las diferentes tecnologías existentes para su aplicación.

Preservación de cavidades

Este es el último proceso de recubrimiento y su función es la de intensificar la protección contra la corrosión dentro de las cavidades de la carrocería mediante el sellado de las mismas con materiales de cera.

Para llevar a cabo este proceso, dos son los procedimientos empleados normalmente, pulverización e inundación. En el procedimiento de pulverización se insertan unas boquillas especiales en las cavidades y se pulveriza una cantidad exacta de material dentro de cada cavidad. Para realizar el procedimiento de inundación, las cavidades se rellenan bajo presión con cera de relleno.

Para la aplicación de la cera se emplean dos procesos automatizados. El primero es la utilización de robots equipados con boquillas plegables neumáticas para la inserción de las mismas en las

cavidades y así aplicar el material (Figura 2.10). Este proceso posee la ventaja de ser muy flexible en cuanto a la posición de los huecos donde se debe introducir la boquilla.



Figura 2.10 - Preservación de las cavidades mediante aplicación robotizada

En el otro procedimiento automático, un número determinado de boquillas están dispuestas en una plataforma de acuerdo a la posición de los huecos en el vehículo. Este se posiciona correctamente encima de plataforma, esta se eleva hasta que todas las boquillas quedan insertadas simultáneamente dentro de las cavidades del vehículo y se aplica el material. La ventaja de este procedimiento es que pueden ser selladas muchas cavidades en un corto periodo de tiempo siempre y cuando sean alcanzables.

Control de calidad

Una vez se han finalizado todos los procesos de recubrimiento del vehículo, se procede a realizar el control de calidad del mismo. A lo largo de este proceso el recubrimiento completo se examina para detectar defectos como inclusiones de suciedad, alteraciones por humectación, corrimientos y otros defectos similares. Además, se miden regularmente el espesor de capa y los parámetros visuales como el tono de color, el brillo y la uniformidad de la capa.

Anteriormente, la inspección de la calidad y la monitorización se realizaban visualmente mediante mediciones realizadas por personal especializado tras el proceso de pintura y posterior secado. Dado que el personal, aun siendo especializado, puede tener un desempeño diferente y errático a cada momento, el proceso de control de calidad ha sido automatizado mediante el empleo de robots equipados con los diferentes instrumentos de medición empleados, como por ejemplo los mostrados en la Figura 2.11.



Figura 2.11 - Control de calidad del recubrimiento mediante robots

El espesor del recubrimiento, que es el valor más importante para la calidad del recubrimiento de pintura, puede medirse directamente y sin contacto después de la aplicación, en la etapa en la que el recubrimiento superior continúa húmedo. Para ello se dispone de sistemas de medición como el que se muestra en la siguiente Figura 2.12. El sensor funciona según el principio fototérmico, en el que un pulso de láser muy corto se dirige a la superficie de la pintura que se va a medir, y esta se calienta brevemente algunos grados Kelvin. La progresión de la radiación de calor reflejada se registra por un sistema óptico de infrarrojos y un detector. A partir de las características de la curva de medición, el sistema de medición determina el correspondiente espesor de la capa individual.

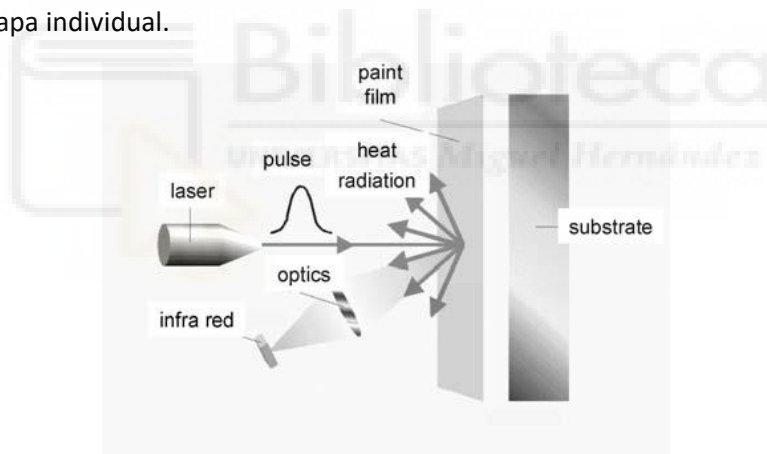


Figura 2.12 - Medición laser del espesor del recubrimiento

La medición del color sin contacto se realiza con fotómetros espectrales multiangulares que han sido diseñados para llevar a cabo la operación junto con robots industriales (Figura 2.13). Los sensores proporcionan los valores de reflexión espectral y otros valores colorimétricos necesarios para la caracterización de la tonalidad de color. La superficie a medir se ilumina con una luz de xenón a 45° y los valores de medición se registran en ángulos de 15°, 25°, 45° y 75°. La temperatura de la superficie se mide con un sensor de infrarrojos y se compensa con respecto a la tonalidad de color debido a la dependencia termocrómica de varias tonalidades de color, especialmente el rojo. Dado que la medición de los tonos de color depende del ángulo de medición, el posicionamiento de la cabeza del sensor debe realizarse con mucha precisión.



Figura 2.13 - Fotómetro espectral multiangular

Además de la medición del espesor del recubrimiento y de las características de calidad óptica, actualmente también se puede realizar una inspección automatizada de fallos con sistemas de procesamiento de imágenes. Normalmente se reconocen todos los fallos del recubrimiento que provocan un cambio de la topografía en la superficie. Para ello se utilizan cámaras e iluminación guiadas por robots (Figura 2.11). Las imágenes captadas son evaluadas por ordenadores y clasificadas según su relevancia para el cliente. Posteriormente, los fallos relevantes son marcados para ser reparados.

En la siguiente Figura 2.14 se muestra un resumen de la cadena de procesos que se llevan a cabo en una estación de recubrimiento de vehículos.

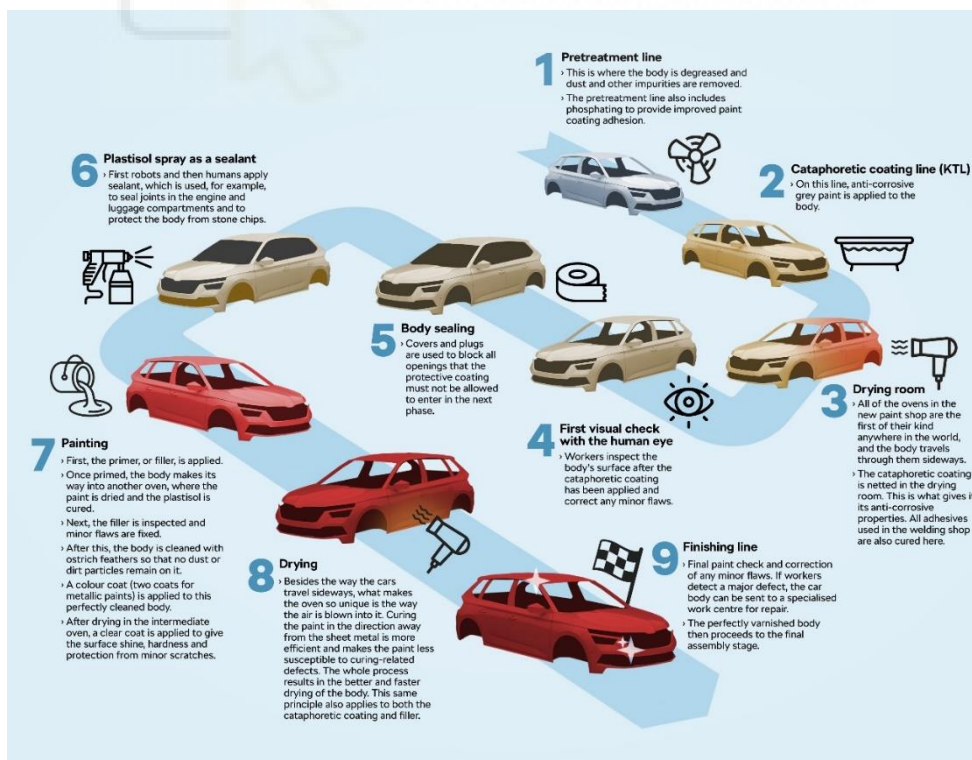


Figura 2.14 - Cadena de procesos en una estación de recubrimiento de vehículos

2.2.3 Distribución en planta de la estación de recubrimiento

Los diferentes procesos que conforman la estación de recubrimiento se distribuye en diferentes niveles o plantas de la factoría. En los niveles inferiores se encuentran los tanques de inmersión para el pretratamiento, el electrorecubrimiento catódico y todo el sistema de tratamiento de aguas residuales y aire residual o aire de escape. En el nivel superior a este se encuentra la línea de pintado y las plataformas de trabajo para tareas manuales. En el siguiente nivel se encuentran los secaderos y la zona de almacenamiento de carrocerías. Las unidades de suministro de aire se encuentran ubicadas en el nivel más alto, también llamado ático.

En las líneas de pintado es imprescindible una gran limpieza para evitar contaminantes que provoquen resultados defectuosos, por lo que estas áreas están diseñadas de acuerdo al concepto de sala blanca, estando la entrada a esta área restringida y requiriendo el empleo de ropa especial.

Las carrocerías se trasladan a esta área después de una limpieza automática a fondo. Esta limpieza se puede realizar en seco o en mojado. En ambos casos el proceso de limpieza tiene dos etapas. En la primera etapa se eliminan las posibles partículas de suciedad de las zonas internas y los cordones de soldadura mediante soplado de aire. La carrocería se traslada a través de un área con una instalación de boquillas de aire orientadas verticalmente, horizontalmente y en otras direcciones por las cuales se bombean corrientes de aire con velocidades de más de 40 m/s. Las partículas de suciedad son arrastradas por estas corrientes hasta el final de la instalación donde son atrapadas por una serie de filtros. Una vez filtrado el aire, este se vuelve a emplear como suministro para las boquillas.

En la segunda etapa se lleva a cabo la limpieza de la superficie de la carrocería mediante rodillos giratorios equipados con plumas, normalmente de avestruz por su gran carga electrostática y su robustez. Estas atrapan las posibles partículas de suciedad, que son inmediatamente aspiradas y eliminadas de las plumas. Estos rodillos son programables y giran a la distancia justa de la superficie de la carrocería (Figura 2.15). Un ejemplo de automatización robótica de este proceso se puede observar en la Figura 2.4.



Figura 2.15 - Limpieza automática de la superficie del vehículo mediante rodillos de plumas

Otro método para realizar la limpieza de posibles contaminantes presentes en la superficie de la carrocería es mediante un lavado. La carrocería se sumerge y limpia con cepillos giratorios

similares a los de una estación de lavado de vehículos convencional y para las zonas internas se emplean chorros de agua a alta presión. Para obtener resultados óptimos pueden utilizarse sistemas robóticos que guían las boquillas en la dirección adecuada.

Tras la limpieza de la carrocería, esta debe secarse por completo de forma adecuada en un secadero o túnel de secado. Esto previene la aparición de marcas de agua y la acumulación de gotas de agua que pueden perjudicar la adhesión de la pintura posterior.

Actualmente gracias al soporte de los sistemas de diseño 3D asistidos por computadora (3D CAD) es posible alcanzar un alto nivel de optimización en el diseño y distribución de la estación de recubrimiento. La inversión en este aspecto es importante, ya que además de aumentar la seguridad de la planta, reducirá las pérdidas derivadas de fallos de diseño. Por ejemplo, mediante una buena distribución y diseño se pueden evitar colisiones de las piezas con el equipamiento o restricciones de espacio a la hora de instalar equipos de gran tamaño. Para ello mediante las herramientas 3D CAD se pueden simular las estaciones robotizadas al completo, el flujo de material a lo largo de las mismas y planificar su trabajo coordinado, e incluso pueden utilizarse herramientas de realidad virtual para inspeccionar la estación de recubrimiento antes de su inauguración.

En un diseño óptimo deben tenerse en cuenta los siguientes requisitos:

- Plazos de entrega cortos.
- Flujo de material rectilíneo.
- Ahorro de espacio con un diseño compacto.
- Una correcta implementación de medidas de protección contra contaminantes externos.
- Concentración de las diferentes tecnologías de la instalación y las plataformas de trabajo para tareas manuales.
- Acceso rápido y sencillo para operarios y personal de servicio.
- Mantenimiento y condiciones de servicio sencillas.

2.2.4 Estación de pintado

Una de las etapas de la estación de recubrimiento es en la que se centrará el presente trabajo, la estación de pintado. En ella además de dotar al vehículo del color y el brillo que le harán atractivo para el consumidor final, también se le proporciona la protección necesaria frente a la radiación ultravioleta, químicos del medioambiente, posibles impactos de gravilla presente en la calzada y pequeños arañazos. El proceso estándar de pintado empleado por los fabricantes de automóviles durante años consta de una capa de imprimación, una primera capa base, una segunda capa base y una capa de revestimiento transparente, como se muestra en la siguiente Figura 2.16. Las franjas grises que se observan en esta figura entre las diferentes etapas de aplicación de las capas son procesos de secado u horneado de las mismas.

Standard process

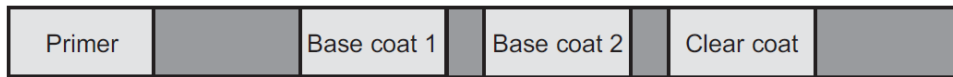


Figura 2.16 - Proceso estándar de pintado de una carrocería

Actualmente se han desarrollado e implantado procesos que conllevan tiempos de proceso más cortos, consiguiéndose esto mediante la eliminación de la capa de imprimación (Figura 2.17) o mediante la utilización de la técnica mojado sobre mojado (wet-on-wet) donde todas las capas son aplicadas sin etapas de secado intermedio (Figura 2.18).

Primerless processes

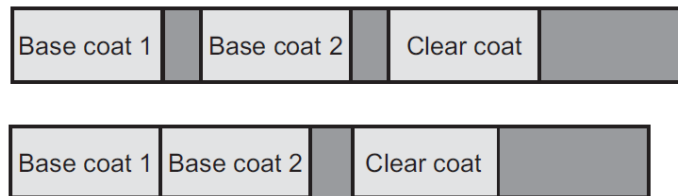


Figura 2.17 - Proceso de pintado sin capa de imprimación

3 wet-on process (solvent-based paints)

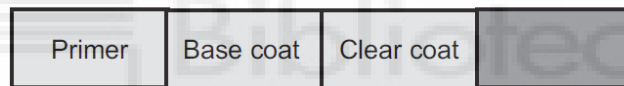


Figura 2.18 - Proceso de pintado Wet-on-wet

En la siguiente Figura 2.19 se muestra la estructura de las capas del recubrimiento de un vehículo con la aplicación de una capa base de efecto metálico, que constaría de dos capas base. En ella, además de la distribución de capas, se observa el grosor medio de las mismas, así como su contribución a las diferentes protecciones necesarias.



Figura 2.19 - Estructura de las capas del recubrimiento de un vehículo

A continuación se expondrán de forma más detallada cada una de las capas aplicadas junto con su metodología de aplicación y las tecnologías disponibles para ello.

2.2.4.1 Capa de imprimación

La capa de imprimación ejerce como vínculo entre el electrorecubrimiento catódico y las capas base superiores. Las funciones principales de esta capa son aumentar la protección frente a la corrosión, proporcionar protección frente a la radiación ultravioleta y frente a los posibles impactos de gravilla presente en la calzada, contribuir al color final deseado, proporcionar una superficie suave y libre de imperfecciones y mejorar la adhesión entre el electrorecubrimiento y la primera capa base.

La capa de imprimación rellena y cubre los pequeños fallos que pueden haberse producido en etapas anteriores o en procesos de lijado y pulido intermedios realizados para eliminar fallos, suavizando así la superficie y protegiéndola por completo contra la corrosión. Además, normalmente, las carrocerías de los vehículos presentan imperfecciones en su superficie debidas a los procesos de prensado, estampado y soldadura realizados en la estación de carrocería, los cuales se acentúan durante el proceso de electrorecubrimiento, lo que denota la importancia de la capa de imprimación.

La capa de imprimación desempeña una función importante en la resistencia del recubrimiento frente a los posibles impactos de gravilla presente en la calzada. Si se produjera un impacto y este lograra perforar el recubrimiento dejando expuesto el metal base sería inevitable que se produjera corrosión en el mismo.

También desempeña un papel de vital importancia en la protección del electrorecubrimiento catódico frente a la radiación ultravioleta procedente del sol, ya que las capas superiores de color y acabado no tienen la capacidad de protección necesaria frente a esta radiación. Dado que el electrorecubrimiento no es resistente a esta radiación, este termina por degradarse produciéndose delaminación de las capas superiores, si no se aplica una capa de imprimación adecuada.

Por otro lado, si se elimina en alguna zona la capa de color del recubrimiento debido a un impacto o cualquier otra causa es muy importante que la apariencia de esa zona sufra el menor cambio posible, por lo que es habitual que se incorporen pigmentos a la mezcla de la capa de imprimación de modo que los posibles fallos queden enmascarados.

Actualmente existen tres tipos principales de capas de imprimación, siendo estas la imprimación a base de disolvente, la imprimación a base de agua y la imprimación en polvo. Hace 40 años las imprimaciones a base de disolvente eran las únicas disponibles. Progresivamente se han ido sustituyendo de acuerdo con los requerimientos legales medioambientales para reducir las emisiones de disolventes altamente contaminantes.

La zona europea y norte americana han tomado caminos diferentes en esta sustitución. En Europa se optó por sustituir las imprimaciones a base de disolvente por imprimaciones a base de agua, mientras que en Norte América las imprimaciones en polvo tienen una notable importancia. A pesar de los requerimientos legales, la sustitución de las imprimaciones a base de disolvente no se ha llevado a cabo a nivel global, continuando siendo estas las imprimaciones con mayor demanda. En Asia el 97% de las imprimaciones empleadas son a base de disolvente, debido a sus menores costes de producción y sus mayores velocidades de línea. Para cumplir con los requerimientos legales se equipan las líneas de producción con sistemas de limpieza de las aguas residuales. En la siguiente Figura 2.20 puede observarse la distribución de la demanda global de imprimaciones en el año 2003. En la Figura 2.21 se observa como actualmente, año

2018, las imprimaciones a base de agua están ganando mercado llegando a tener la misma importancia que las imprimaciones a base de disolvente.

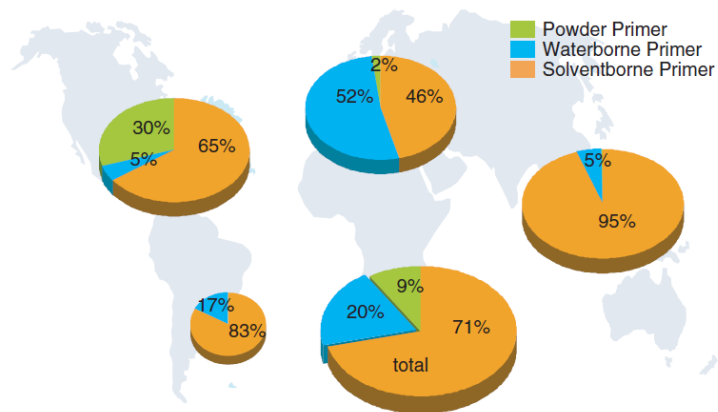


Figura 2.20 - Distribución de la demanda global de imprimaciones (2003)

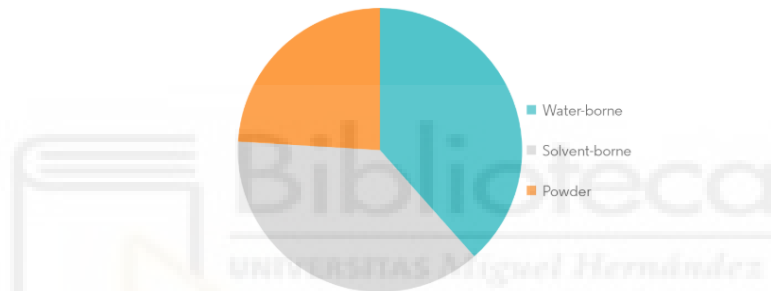


Figura 2.21 - Distribución de la demanda global de imprimaciones (2018)

Requisitos para la capa de imprimación

Requisitos legales

La industria automotriz y la industria química de fabricación de recubrimientos deben lidiar con una serie de estrictos requisitos legales para garantizar el pleno cumplimiento de la normativa vigente a escala industrial.

Como se ha explicado anteriormente, desde hace 40 años se ha reducido considerablemente la emisión de disolventes y otros componentes químicos derivados de su fabricación, gracias al impulso de las respectivas autoridades gubernamentales a la sustitución de las imprimaciones y pinturas a base de disolvente por las imprimaciones y pinturas a base de agua. Pese a que la situación se ha mejorado notablemente a escala global, la pugna por dicha sustitución sigue siendo una realidad en muchas partes del mundo.

Requisitos mecánicos

La capa de imprimación tiene una gran importancia en el conjunto global de recubrimiento en la resistencia contra impactos de pequeños sólidos presentes en la calzada, como gravilla. Debe evitarse totalmente que un impacto atraviese el recubrimiento por completo llegando al metal base, ya que esto conllevará inevitablemente la corrosión del mismo, causando así el deterioro tanto del metal base como del recubrimiento. La elasticidad de la capa proporciona un indicador de la capacidad potencial de protección frente a impacto de gravilla de esta capa por sí sola. Dicho esto, en ciertas zonas de la carrocería más expuestas es bastante difícil resistir los continuos impactos de gravilla, por lo que normalmente se añade color a la mezcla empleada para la capa de imprimación, de modo que en caso de que las capas superiores a esta sean perforadas por el impacto de algún sólido, el defecto en el recubrimiento se enmascare lo mejor posible.

Para garantizar un rendimiento óptimo del conjunto de recubrimiento, se precisa una buena adhesión de la capa de imprimación con la capa de electrorecubrimiento catódico y con las capas base superiores, evitándose así la delaminación de las capas en caso de perforación de alguna de ellas.

La dureza superficial de la capa de imprimación es un indicador de su compatibilidad con las medidas de corrección de defectos, como el lijado y el pulido. Esta es la última ocasión que se tiene para eliminar ciertos defectos, si no se eliminan tras la aplicación de la capa de imprimación deberán ser eliminados tras finalizar el proceso completo de recubrimiento, lo cual conlleva notables pérdidas económicas.

Requisitos climatológicos

A lo largo de la vida útil del vehículo el recubrimiento completo del mismo se ve expuesto continuamente a condiciones variables de humedad y temperatura, lo cual puede provocar efectos osmóticos adversos. Si no existe un buen rendimiento de la capa de imprimación y una buena interacción de la misma con el conjunto global de recubrimiento pueden aparecer defectos en el conjunto como formación de ampollas y pérdida de adhesión entre capas.

Puesto que las capas base superiores son transparentes a la radiación ultravioleta, la capa de imprimación se ve expuesta directamente a ella, siendo la única barrera. Si esta radiación causa la desintegración de capa de imprimación, se producirá el fenómeno conocido como caleo o "chalking" del electrorecubrimiento, que conllevará una pérdida drástica de la adhesión y la delaminación de las capas superiores (Figura 2.22).

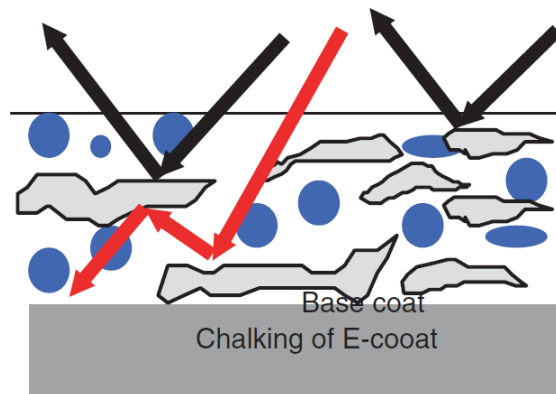


Figura 2.22 - Caleo del electrorecubrimiento

Como ejemplo de la importancia de la resistencia de la capa de imprimación a la climatología, en los años ochenta se generalizó la implementación de los electrorecubrimientos catódicos llegando a obtener muy buenos rendimientos de los mismos y buenas calidades de superficie, lo cual derivó en Estados Unidos en la tendencia de realizar el recubrimiento de los vehículos de pasajeros eliminando la capa de imprimación del proceso. Inicialmente se observó que al eliminar esta capa se reducía el tiempo total, lo cual conllevaba un notable ahorro, pero con el paso del tiempo, puesto que las capas superiores no tienen apenas poder de protección contra la radiación ultravioleta, se produjo la desintegración del electrorecubrimiento (chalking) y la delaminación de las capas superiores, como puede observarse en la Figura 2.23.



Figura 2.23 - Ejemplo de delaminación debida al caleo del electrorecubrimiento

Requisitos superficiales

El buen rendimiento del recubrimiento del vehículo en cuanto a apariencia es un requisito altamente exigido por parte de los productores. La capa de imprimación se diseña para desempeñar un papel importante en conjunto global del recubrimiento. La evaluación del rendimiento de la capa de imprimación tiene que realizarse en combinación con el resto de capas del recubrimiento.

Para describir la calidad del recubrimiento en términos físicos, el método comúnmente empleado es el escaneo de onda. Los equipos de escaneo de onda utilizan un rayo láser para detectar los diferentes tipos de textura de la superficie. Las mediciones de ondulación se

correlacionan automáticamente en términos matemáticos de acuerdo con otras escalas de medición visual de toda la industria, incluyendo el valor de tensión y la clasificación estándar de piel de naranja. Las texturas de la superficie se diferencian en seis categorías, identificadas por longitud de onda.

Las texturas de la superficie se dividen en rangos de onda corta y onda larga. Siendo las de onda corta para la percepción a corta distancia y las de onda larga para la percepción a larga distancia. Los valores de onda corta bajos significan que hay poca perturbación de la apariencia visual a una distancia de visualización larga. Los valores bajos de onda larga significan que la apariencia apenas se perturba cuando se observa también desde una distancia larga.

Requisitos de aplicación

Los requisitos para la aplicación de la capa de imprimación son altamente complejos y a menudo no están estandarizados. Desde mediados de los años noventa muchos fabricantes de automóviles han delegado la responsabilidad de definir correctamente estos requisitos a los proveedores de recubrimientos. Así estos deben ajustar las características reológicas del material de la capa de imprimación para un comportamiento óptimo durante su almacenamiento y aplicación, así como sus características de evaporación y secado.

Los materiales empleados deben optimizarse para garantizar la estabilidad de circulación cuando se encuentran en el sistema de suministro o circulación, es decir, cuando se hallan en los conductos de suministro al aplicador.

Para la aplicación líquida de capas de imprimación se emplea tanto la atomización neumática como la atomización centrífuga de alta velocidad. Para esta última, los materiales de aplicación deben estar optimizados para que su reología garantice su pulverización y para evitar cualquier problema durante la evaporación del disolvente y la tendencia a causar corrimientos. Para las aplicaciones en polvo se debe ajustar la reología del material de forma que se asegure su correcta fusión y unión a temperaturas superiores a 80 °C.

Tecnologías de aplicación y características

Existen dos tipos principales de imprimaciones líquidas, las imprimaciones a base de agua (waterborn) y las imprimaciones a base de disolvente (solventborne). Para la fabricación de cualquiera de los dos tipos se deben mezclar los siguientes cuatro tipos de componentes hasta alcanzar una mezcla heterogénea:

- Aglutinantes y resinas.
- Pigmentos y extensores.
- Disolventes, siendo Compuestos Orgánicos Volátiles (VOC) para las imprimaciones a base de disolvente y agua para las imprimaciones a base de agua.
- Aditivos.

Las imprimaciones a base de agua pueden ser diluidas y los aplicadores pueden ser limpiados con agua. Como ya se ha mencionado, estas producen un volumen significativamente más bajo

de emisiones de compuestos orgánicos volátiles (VOC) que las imprimaciones a base de disolvente. Las imprimaciones a base de agua también tienen mejores propiedades termoplásticas y son más duraderas que las a base de disolvente. Por otro lado, las imprimaciones a base de disolvente producen un acabado más duro que es más tolerante a las condiciones adversas durante el período de secado.

Por otro parte se tienen las imprimaciones en polvo (powder), con una composición similar a las líquidas pero con la gran diferencia de estar compuestas casi exclusivamente por materiales no volátiles, siendo este su mayor atractivo ya que se eliminan por completo las emisiones de los mismos, además de ser completamente sólidas. Otra de sus principales ventajas es que cualquier material resultante de la sobrepulverización puede reciclarse por completo, suponiendo esto un ahorro significativo.

En la siguiente Tabla 2.1 se muestran a modo de ejemplo algunas de las características más relevantes de tres imprimaciones diferentes, una de cada tipo, de un fabricante.

Properties	Units	Solvent-borne primer	Waterborne primer	Powder primer
Solid content	%	50–65	35–45	100
VOC	g l ⁻¹	390–420	170–230	0
Density	g cm ⁻³	1.1–1.3	1.1–1.3	0.5 – 0.7 ^a
Bake conditions	min °C ⁻¹	20/130–165	20/130–165	20/160–190
Film build	µm	35–50	25–35	55–100
Viscosity (20 °C)	mPa at 1000 s ⁻¹	60–100	60–100	Solid
Storage temperature	°C	5–35	5–35	5–30
Shelf life	Mo	6	6	12

^a Bulk Density.

Tabla 2.1 – Características principales de tres imprimaciones diferentes, una de cada tipo

A continuación se explicarán brevemente los aspectos más relevantes de cada uno de los componentes que forman los diferentes tipos de capas de imprimación.

Las propiedades básicas de la mezcla a aplicar, como la viscosidad, la reactividad, la dureza y la elasticidad de la capa de imprimación una vez aplicada, se establecen al seleccionar el aglutinante o la resina de la mezcla.

El poliéster como aglutinante es el componente básico de casi todos los tipos de imprimaciones. El poliéster normalmente se suministra en una solución con disolventes. Para los sistemas de imprimación a base de polvo se requieren resinas sólidas y libres de disolventes. Para los sistemas de imprimación a base de agua el poliéster debe ser suministrado en la fase acuosa.

Con resinas epoxi es posible crear recubrimientos con excelentes propiedades mecánicas y tecnológicas y también con una protección eficaz contra la corrosión debido a su buena adhesión a las superficies metálicas.

Para las imprimaciones en polvo existen dos formulaciones clásicas, las imprimaciones carboxi-epoxi y las imprimaciones de poliuretano (PUR). En las imprimaciones carboxi-epoxi los poliésteres de ácidos se combinan con resinas epoxi. Estos productos se denominan polvos híbridos.

Existen alternativas como las resinas cicloalifáticas y los acrilatos de metacrilato de glicilo (GMA). Además, debe tenerse en cuenta la existencia de endurecedores epoxi-funcionales resistentes a la radiación ultravioleta basados en el anillo isocianurato. El ejemplo más conocido, el diisocianurato de triglicidilo (TGIC), era anteriormente el endurecedor estándar para recubrimientos en polvo, pero está asociado con problemas fisiológicos en el lugar de trabajo y por lo tanto se está eliminando gradualmente.

Los pigmentos en las capas de imprimación no solo se utilizan para dar color, sino que también se emplean para apoyar el rendimiento mecánico de la misma. Este aspecto se refuerza mediante el uso de extensores que afectan a las propiedades y el perfil de rendimiento de la capa mejorando así el brillo, la resistencia a los impactos de gravilla y su reología, pero sin afectar a la protección contra la radiación ni al color.

Los aditivos se emplean para controlar y corregir las interacciones dentro de la capa de recubrimiento y de las capas límite. El uso de aditivos en los sistemas de imprimación a base de agua es altamente complejo dado que además de la resina y el pigmento, el agua debido a su alta actividad superficial y características físicas especiales, es responsable de una dimensión adicional de los efectos de la capa límite. Por lo tanto, las capas de imprimación a base de agua tienen una mezcla de aditivos más compleja que las a base de disolventes.

Actualmente las imprimaciones a base de agua, con cada vez menor cantidad de disolventes, están jugando un papel mucho más significativo que antaño a causa de los exhaustivos controles medioambientales. A pesar de esto, la mayoría de las composiciones de las imprimaciones están basadas en mezclas de disolventes y diluyentes orgánicos.

En las imprimaciones a base de disolventes convencionales el término disolvente es demasiado general. Normalmente existe un equilibrio entre disolvente y diluyentes (hidrocarburos aromáticos) que se utilizan como diluyentes y para reducir el coste de la formulación. Se necesita este equilibrio para obtener la apariencia de capa deseada bajo condiciones particulares de aplicación dentro de un rango específico de contenido de sólidos.

Los factores principales que afectan a la elección de un disolvente u otro son la capacidad de disolución, la viscosidad, punto o rango de ebullición, tasa de evaporación, punto de inflamación, naturaleza química, toxicidad, y coste. Particularmente, la toxicidad, el olor y el coste se han vuelto cada vez más importantes.

Métodos de aplicación

Para la aplicación de la capa de imprimación se requiere la máxima fiabilidad del equipamiento empleado. Para ello, se utilizan sistemas robóticos capaces de mejorar la adaptación del patrón de pulverizado al contorno de la carrocería, de forma que se aumente la eficiencia en términos de cantidad de producción y de eficacia de transferencia (TE).

El éxito y la calidad del pulverizado de esta capa dependen principalmente del tipo de aplicación. Esta elección también supone un mayor o menor desaprovechamiento de material de imprimación, siendo la capa de imprimación la que mayor porcentaje de malgasto de material supone a la estación de pintado.

La sobrepulverización es el término empleado para referirse a las partículas de recubrimiento que no se depositan en la pieza de trabajo pero pasan volando sobre la misma durante la pulverización. Si la sobrepulverización se consigue reducir, se reduce automáticamente la cantidad de desechos emitidos y la cantidad de material que debe adquirirse, reduciéndose así también los gastos. Dicho esto, como es lógico, será preferible un equipo de aplicación que limite la sobrepulverización.

Existen métodos de aplicación sin sobrepulverizado basados en otros principios, como la aplicación a rodillo, por fundición, por inmersión o por inundación, aunque estos no tienen ninguna relevancia para la aplicación de capas de imprimación sobre vehículos bien porque la geometría de la carrocería del vehículo no los permite o porque su eficacia en términos de colores variables es insuficiente.

Para las imprimaciones líquidas, a base de agua y a base de disolvente, el principio común de todos los procesos de aplicación es que estas son atomizadas en el aire y transferidas al metal base o al electrorecubrimiento si se da. Los principales métodos de pulverizado son los siguientes:

- Aplicación sin aire (airless) mediante boquillas de una sola sustancia.
- Aplicación neumática mediante boquillas de doble sustancia.
- Atomización neumática con carga electrostática.
- Atomización electrostática centrífuga mediante rotación a alta velocidad.

El proceso más empleado actualmente es la atomización electrostática centrífuga mediante rotación a alta velocidad, ya que mediante este proceso se obtienen resultados óptimos en cuanto a atomización y eficacia de transferencia. Este proceso también es el que mejores reducciones de sobrepulverización consigue de todos los equipos de pulverizado. Los procesos de atomización sin aire, por aire comprimido o neumáticos y el método de alto volumen y baja presión (HVLP) tienen una relevancia secundaria.

En la atomización electrostática centrífuga mediante rotación a alta velocidad, se crea un campo eléctrico entre la carrocería del vehículo (potencial de tierra) y el dispositivo de pulverización. Las gotas de recubrimiento se cargan electrostáticamente mediante un electrodo, desplazándose con su carga a lo largo de las líneas de campo hasta llegar a la carrocería, aplicándose así sobre la misma. La carga eléctrica fluye a través de la carrocería del vehículo hasta la toma de tierra.

En los sistemas de imprimación a base de disolvente, el imprimador tiene una resistencia eléctrica tan baja que el alto voltaje necesario se puede aplicar directamente en el suministro de recubrimiento, dentro del dispositivo de pulverización (Figura 2.24).

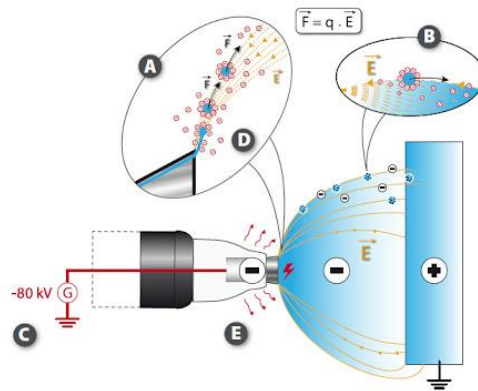


Figura 2.24 - Atomización electrostática centrífuga en sistemas a base de disolvente

Por el contrario, los sistemas de imprimación a base de agua son mucho más conductores. Así que para evitar que las altas tensiones, de hasta 80 kV, se filtren al suministro de recubrimiento y a las líneas de suministro se debe establecer estructuralmente una separación de potencial antes de aplicar la alta tensión o la superficie de imprimación debe ser primero atomizada en gotas de recubrimiento antes de que se carguen en la fase de transferencia. En este último caso, se generan iones de aire con la ayuda de electrodos aplicados externamente, que posteriormente cargan las gotas de recubrimiento (Figura 2.25). Esto es menos efectivo que la carga directa, como en las imprimaciones a base de disolvente, pero se utiliza más a menudo ya que el coste de las instalaciones para evitar los cortocircuitos en el sistema de suministro de recubrimiento, debido a la alta conductividad de las imprimaciones, es muy elevado.

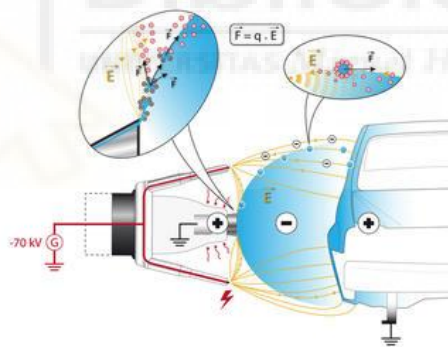


Figura 2.25 - Atomización electrostática centrífuga mediante electrodos externos en sistemas a base de agua

La atomización centrífuga de alta velocidad se lleva a cabo mediante el uso de atomizadores rotativos de alta velocidad con plato de campana de un diámetro de 60 a 100 mm. El recubrimiento se acelera en el plato de campana y se atomiza en el borde exterior debido a las fuerzas centrífugas generadas. La velocidad estándar de rotación del plato de campana es de alrededor de 20000 rpm. Los atomizadores de altas prestaciones pueden alcanzar velocidades de rotación de hasta 60000 rpm.

Mediante la utilización de este procedimiento se pueden obtener tasas de eficacia de transferencia de hasta el 90%, aunque de forma realista se tienen tasas para la imprimación a base de disolvente del 85% y para la imprimación a base de agua del 75%.

La suavidad de la capa de imprimación y la calidad de flujo dependen de la capacidad del plato de campana para atomizar finamente las gotas de recubrimiento. Para esto se requerirá que la imprimación tenga una viscosidad baja. Por el contrario se requiere una alta viscosidad para que

cuando las gotas lleguen a la superficie de la carrocería no se formen corrimientos de pintura. Puesto que los esfuerzos cortantes son mucho más altos en el proceso de atomización que cuando la imprimación se halla en la superficie de la carrocería, estos dos requerimientos opuestos se pueden alcanzar por medio de la viscosidad estructural intrínseca.

En las imprimaciones en polvo, las partículas de polvo presentan un tamaño relativamente pequeño, aproximadamente 25 – 35 μm , lo cual hace que tengan una superficie específica alta. Las interacciones que se dan entre ellas son las fuerzas de repulsión de Coulomb y las fuerzas de atracción de Van der Waals, aumentando estas en proporción directa a la superficie específica, lo cual también aumenta la aglomeración. Una reducción del 20% en el contenido de finos de una imprimación en polvo puede conllevar a una reducción del 80% en su área de superficie.

Para aplicar las imprimaciones en polvo se requiere la fluidización del polvo de manera que sea transportable neumáticamente. Cuanto mayor es el tamaño medio de las partículas menor es la tendencia a aglomerarse y cuanto mayor es la densidad mejor fluidización se puede conseguir. Por el contrario, cuanto menor sea el tamaño de las partículas mejores propiedades de flujo se obtienen y cuanto menor sea la densidad, mejor eficacia de transferencia se obtiene en superficies horizontales. Por lo tanto debe optimizarse la relación entre densidad y tamaño de partículas.

Para la aplicación de las imprimaciones en polvo se utiliza el principio de dispersión en el aire, en lugar de la atomización de las imprimaciones líquidas. El polvo suministrado debe tener el tamaño de partículas óptimo. Para que sea posible el transporte del polvo hasta el aplicador este se fluidifica. Para ello se insufla aire a través del suelo poroso del tanque que contiene el polvo, generando un vórtice giratorio de polvo denominado lecho fluidizado. De esta forma el polvo y aire en conjunto se comportan como un fluido pudiendo ser transportados.

Las pistolas neumáticas convencionales para aplicar las imprimaciones en polvo emplean boquillas de una sola sustancia. Normalmente la pistola es una simple tubería con una placa deflectora o una boquilla de hendidura en la salida. También pueden emplearse aplicadores neumáticos con plato de campana. En estos las partículas de polvo se dispersan mediante las fuerzas centrífugas de rotación. La velocidad de rotación en este caso es alrededor de 10 veces menor a la velocidad de los atomizadores para imprimaciones líquidas, en primer lugar para evitar la aglomeración y en segundo lugar porque la atomización en gotas ya no es necesaria.

Existen dos métodos principales para realizar la carga electrostática de las partículas, la carga triboeléctrica y la carga mediante el efecto corona. La carga triboeléctrica aprovecha el hecho de que el polvo no es conductor. El rozamiento de las partículas de polvo con una superficie no polar hace que estas reciban una carga electrostática positiva y por lo tanto se sentirán atraídas por la carrocería conectada a tierra (Figura 2.26). Así pues no será necesaria la aplicación de una fuente de alta tensión, como sí lo era en las imprimaciones líquidas. Este método es adecuado para superar cualquier problema de las jaulas Faraday formadas por el metal a recubrir. Actualmente no es común el empleo de la cara triboeléctrica en aplicaciones del sector automotriz.

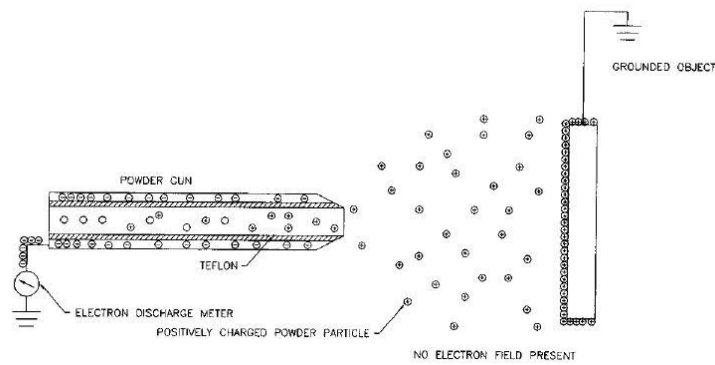


Figura 2.26 - Aplicación de imprimaciones en polvo mediante carga triboeléctrica

Por otro lado, en el método de carga mediante el efecto corona, un electrodo, situado en la salida del dispositivo de pulverización, genera iones de aire que cargan las partículas de polvo (Figura 2.27). Existe el problema de que se genera un exceso de iones de aire que se depositan junto con las partículas de polvo en la carrocería del vehículo. Este efecto puede descontrolarse si existen altas tensiones, baja distancia a la pieza y un alto espesor de capa aplicada. El exceso de carga en la capa que se está aplicando hace que el polvo ya aplicado sea despegado de nuevo. Este fenómeno se conoce como ionización posterior. Una vez se ha aplicado el polvo es esencial que se mantenga la carga. La adhesión al sustrato se mantiene sólo debido a la carga electrostática. Esto explica por qué la dificultad de aplicación si la humedad relativa alcanza aproximadamente el 75% o más, ya que se produce la descarga. Por lo tanto, no parece haber razón para un límite mínimo de humedad del aire. Sin embargo, la experiencia muestra que la ionización posterior también aumenta si existe un porcentaje de humedad relativa menor al 40%.

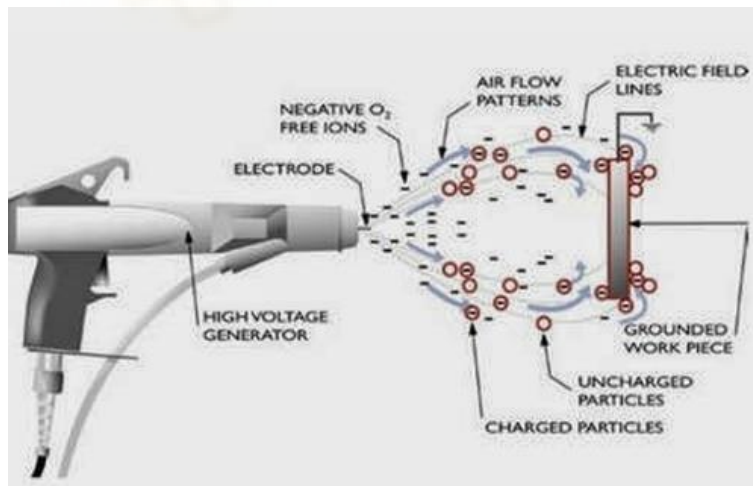


Figura 2.27 - Aplicación de imprimaciones en polvo mediante el efecto corona

Debido a la mejor estabilidad y reproducibilidad del proceso de aplicación, la carga mediante efecto corona es el método de carga que se utiliza normalmente en la fabricación de vehículos.

Para aplicar la imprimación al exterior de los automóviles se emplean aplicadores neumáticos con plato de campana y carga mediante efecto corona. Actualmente mediante este método se pueden alcanzar porcentajes de eficacia de transferencia de entre el 70% y el 80%. En la mayoría

de los casos, la sobrepulverización se recircula y, por lo tanto, no se desperdicia. Con el fin de optimizar este proceso, la humedad relativa no debe exceder el 60% porque de lo contrario la manipulación del polvo reciclado será más difícil. Cabe destacar que el contenido de partículas finas es mayor en el sobrepulverizado reciclado que en el producto suministrado, lo cual conlleva una menor fluidez y fluidización. La distribución del tamaño de partículas en el producto suministrado debe tener en cuenta el aumento del contenido de partículas finas a lo largo del proceso.

2.2.4.2 Capas superiores

Una vez se ha realizado el pretratamiento y el electrorecubrimiento, para proteger el metal base frente a la corrosión, y se ha aplicado la capa de imprimación, para dar uniformidad a la estructura y proteger las capas inferiores de los impactos de gravilla, el paso final es dotar al vehículo del color deseado y proporcionar durabilidad al sistema de recubrimiento mediante las capas superiores. Estas capas son las últimas a aplicar en el proceso de recubrimiento del vehículo y consisten en dos tipos diferentes de capa, la capa base y la capa de revestimiento transparente.

La capa base contiene el pigmento principal que proporciona el color al vehículo. Actualmente existen más de 40000 colores disponibles para la capa base y cada año se añaden alrededor de 1000 colores nuevos a la lista de los ya disponibles. Con el uso generalizado de pigmentos de efecto, como escamas de aluminio, micas y otros tipos de pigmentos de interferencia, las capacidades de diseño de color son ilimitadas. En la siguiente Figura 2.28 se muestra la tendencia de aplicación de los colores principales a nivel global.

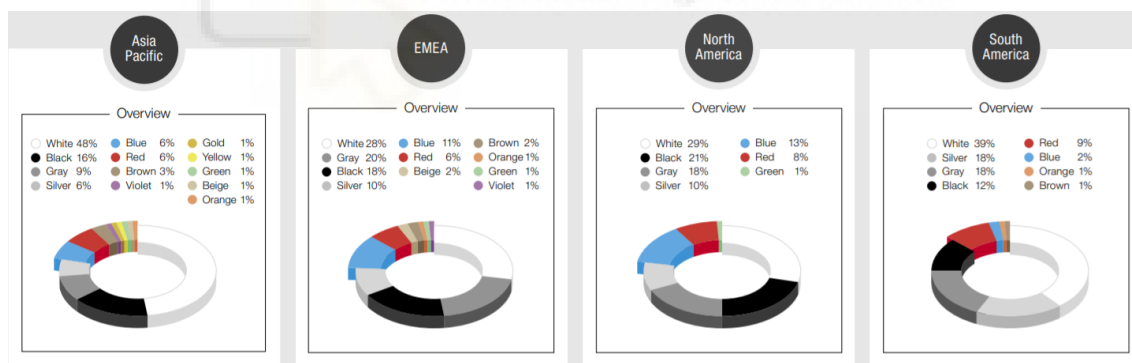


Figura 2.28 - Tendencia de aplicación de los colores principales a nivel global

Para proteger la capa base, y por lo tanto el color del vehículo, se aplica una capa de revestimiento transparente sobre esta. Esta capa de revestimiento proporciona protección frente a la luz solar, los efectos ambientales, la corrosión, los posibles arañazos, los diferentes ataques químicos a los que se ve sometida la carrocería a lo largo de su vida útil y además proporciona una gran retención del color y un acabado suave y sin manchas.

Las capas superiores han sido y continúan siendo la mayor fuente de emisiones de contaminantes de la estación de recubrimiento completa. Actualmente se ha conseguido reducir las emisiones de compuestos orgánicos volátiles (VOC), resultantes de la aplicación de las capas superiores de un vehículo, de 150 g/m², en las estaciones de los años setenta, a menos de 35 g/m², en las estaciones actuales.

El aspecto y el color de la capa superior dependen altamente de las condiciones de aplicación. La aplicación de recubrimientos en la industria automotriz no está estandarizada y se lleva a cabo en una multitud de estaciones de pintura diferentes en todo el mundo utilizando diferentes equipos para su aplicación, procesos de aplicación y condiciones. Los materiales de revestimiento y el proceso de aplicación deben estar perfectamente seleccionados y adaptados para obtener resultados de alta calidad y una buena productividad.

Como ya se ha mencionado, para proporcionar el color deseado al vehículo se utilizan pigmentos. Un pigmento puede definirse como cualquier sólido de partículas de color, negro, blanco o fluorescente, que sea insoluble en, y esencialmente no se vea afectado por, el medio en el que se incorpora. El pigmento mantendrá su estructura cristalina o de partículas durante todo el proceso de coloración. Alterará la apariencia de un objeto por la absorción y/o dispersión selectiva de la luz que recibe (Figura 2.29).

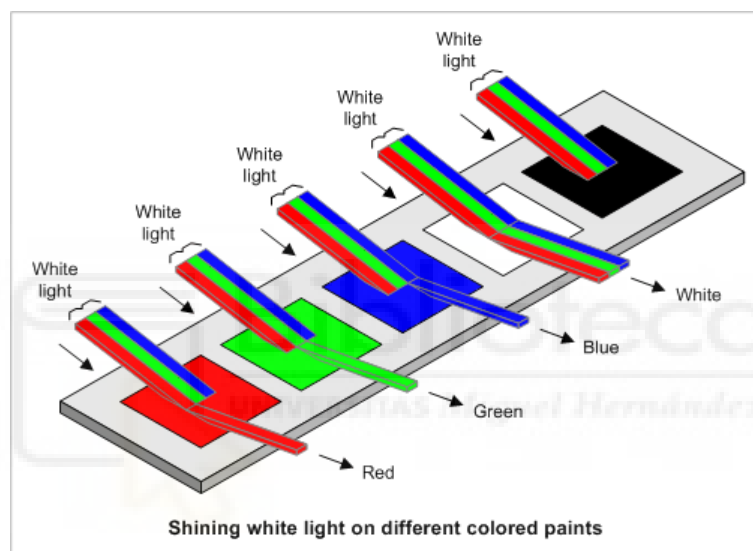


Figura 2.29 - Reflexión de la luz blanca según el color en el que incide

Los diferentes tipos de pigmentos pueden clasificarse en:

- Pigmentos orgánicos.
- Pigmentos inorgánicos.
- Pigmentos de aluminio.
- Pigmentos de efecto, sin aluminio.
- TiO_2 , negro de carbón.
- Pigmentos funcionales (corrosión, extensores, nanopartículas, agente mateante, etc).

Los colores se describen según los siguientes parámetros:

- **Tono.** Puede definirse como el grado en el cual un estímulo puede ser descrito como similar o diferente de los estímulos, como rojo, amarillo y azul. Puede representarse cuantitativamente mediante un solo número, correspondiente a una posición angular alrededor de un punto o eje central o neutro en un diagrama de coordenadas del espacio

de color o una rueda de color, o por su longitud de onda dominante o la de su color complementario.

- Luminosidad o brillo. Puede definirse como la percepción no lineal que se tiene de la cantidad de luz recibida. Cuanto más oscuro es un color menor es su luminosidad.
- Saturación. Se puede definir la saturación de un color como una combinación de su intensidad luminosa y la distribución de sus diferentes longitudes de onda en el espectro de colores. El color más puro se consigue usando una sola longitud de onda a una intensidad muy alta, como con un láser. Si la intensidad luminosa disminuye, la saturación también.

Mediante la utilización de pigmentos de efecto de aluminio para tonos metálicos, el brillo dependerá del ángulo de observación. El efecto metálico se obtiene por la reflexión de la luz en la superficie de las partículas de aluminio. Con partículas de mayor tamaño se consigue una mejor reflexión y un mayor brillo. Con pigmentos de aluminio más gruesos, las partículas se vuelven más visibles, lo que produce granulosidad o textura. El aspecto metálico también depende de la orientación de las partículas de metal en la capa de recubrimiento, la forma de las partículas, la transparencia de la matriz aglutinante y la presencia de otros colorantes. Las partículas desorientadas causarán apariencia incorrecta de la superficie y al aumento de la estructura de ondas cortas o el efecto de neblina de la superficie recubierta.

Con los pigmentos de efecto mica para tonos perlados o con pigmentos de interferencia, el tono dependerá del ángulo de observación.

Además de por las propiedades del color anteriormente descritas, los pigmentos presentan las siguientes propiedades técnicas:

- En las pinturas líquidas:
 - Dispersabilidad.
 - Reología.
 - Floculación y sedimentación.
 - Comportamiento en almacenamiento.
- En la capa aplicada de recubrimiento:
 - Resistencia a la luz.
 - Resistencia climatológica.
 - Resistencia a disolventes.
 - Resistencia química.
 - Resistencia al calor
 - Resistencia a la decoloración.

Estas propiedades de los pigmentos se describen mediante su composición química en forma de un índice de color, y por la forma y el tamaño de las partículas de pigmento, siendo este establecido por el proceso de fabricación del pigmento y por el proceso de fabricación de la pintura en etapas de dispersión o molienda.

Capa base

Como se ha descrito anteriormente, la capa base proporciona el color deseado al vehículo. Esta capa se aplica sobre la capa de imprimación y se cubre mediante la capa de revestimiento transparente para protegerla del exterior.

Para colores sólidos, es decir colores sin efectos, solo se aplica una capa base, mientras que para colores con efecto, como el efecto metálico o el efecto perla, se requiere una segunda capa base.

El espesor de la capa base dependerá del poder de ocultación de la pintura elegida, es decir, como de capaz es la pintura de ocultar el sustrato sobre el que es depositada, lo cual dependerá de la pigmentación. Para tonos plateados, que muestran una buena ocultación negro-blanco, el espesor típico de la capa es de alrededor de 10 μm , para tonos blancos alrededor de 20 μm , y para tonos amarillos/rojos hasta 30 μm . Para los colores con efecto, la orientación de las partículas metálicas o las partículas de mica paralelas al sustrato controlan la diferencia de brillo.

La orientación de las partículas se ve influenciada en gran medida por la reología de la pintura. Las capas base a base de disolvente cuya formulación se base en polímeros solubles y pigmentos muestran un comportamiento de flujo newtoniano. Sin la adición de agentes de control de la reología las partículas de las capas base tendrían una orientación incorrecta, el espesor de la capa en las zonas verticales sería insuficiente y además se producirían corrimientos de la pintura en estas zonas.

A lo largo del proceso de aplicación el material de la capa base experimenta una serie de diferentes velocidades de corte, que conllevan diferentes esfuerzos cortantes. Cuando se encuentra almacenado en el tanque de almacenamiento y no se agita, por lo que la velocidad de corte es cercana a cero, la alta viscosidad previene que los pigmentos sedimenten. Posteriormente el material se agita ligeramente y se bombea a través de la línea de circulación, aumentando así la velocidad de corte, de forma que la viscosidad disminuye para facilitar el bombeo. En la boquilla del pulverizador o en el plato de campana del atomizador, la velocidad de corte es máxima, por lo que la viscosidad es sumamente baja, ayudando así a la pulverización en gotas. Cuando las gotas impactan contra la superficie de la carrocería y se deforma y aplanan, la viscosidad tiene que ser baja de nuevo para apoyar el proceso de orientación de las partículas. Una vez que se completa esta reorientación, la velocidad de corte vuelve a bajar a cero, y la viscosidad vuelve a subir a niveles altos, evitando así corrimientos de pintura. En la siguiente Figura 2.30 se observa esta variación de la viscosidad en función de la velocidad de corte de forma clara.

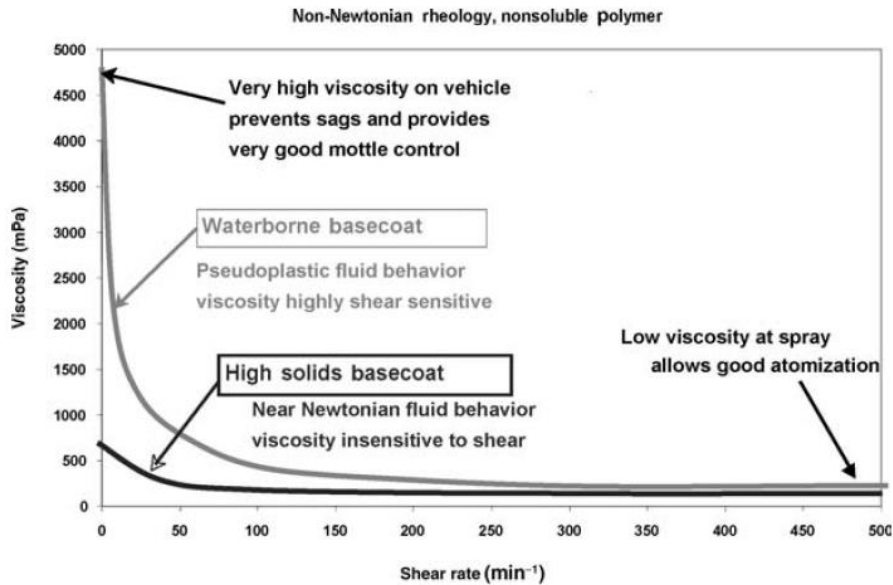


Figura 2.30 - Variación de la viscosidad en función de la velocidad de corte

Tecnologías de aplicación y características

En la industria automotriz existen tres procesos de aplicación principales para la capa base de recubrimiento:

- De sólidos medios a base de disolventes (MS).
- De sólidos altos a base de disolventes (HS).
- A base de agua.

Mientras que en Europa se emplean principalmente las capas base a base de agua, en Norte América se utilizan mayoritariamente las capas base de sólidos altos a base de disolventes.

Inicialmente las capas base de efecto se formularon con niveles de sólidos bajos, 10% – 15%, y después con niveles de sólidos medios, 15% – 20%, teniendo así una viscosidad de 100 mPa a 1000 rpm. El poliéster, en combinación con butirato de acetato de celulosa (CAB), SCA y ceras, se emplea como aglutinante principal. En esta tecnología, los colores sólidos suelen tener entre un 25 y un 45 por ciento de sólidos.

Las formulaciones de las capas base de sólidos altos a base de disolvente se desarrollaron a partir de las de sólidos medios. Para cumplir los requisitos medioambientales impuestos por la EPA (Agencia de Protección del Medio Ambiente en Estados Unidos) se tuvo que aumentar el nivel de sólidos de los recubrimientos de nivel medio y se tuvo que reducir la cantidad de disolventes orgánicos. Para mantener las buenas propiedades de pulverizado se redujo el peso molecular de las resinas empleadas. Esto conlleva una reducción de la viscosidad y por lo tanto se requiere menos disolventes para obtener la viscosidad de aplicación correcta. Por el contrario, estos recubrimientos presentan un menor secado físico y son más sensibles al corrimiento del mismo. Para prevenir esto último, el recubrimiento debe ser pseudoplástico,

mostrando un comportamiento pseudoplástico a altas velocidades de corte, o tener un límite elástico que muestre un comportamiento de alta viscosidad por debajo de una cierta velocidad de corte. El método más empleado para realizar esto es la incorporación de microgeles compuestos por polimerización de emulsiones no acuosas. Los microgeles poliméricos y las capas orgánicas son los agentes de control reológico comunes en los recubrimientos HS ya que no aumentan significativamente la viscosidad del recubrimiento a altas velocidades de corte, lo cual es fundamental para conseguir una atomización correcta durante la pulverización.

Actualmente la tecnología principal de aplicación de capas base son las capas base a base de agua. Desde su primera implantación industrial en 1987-1988 esta tecnología ha conquistado la mayor parte del mercado en Europa, Estados Unidos y Japón. Su éxito se debe a los beneficios medioambientales que presenta, a su rendimiento superior y a sus robustas propiedades de aplicación.

En Europa se emplean normalmente las aminas DMEA (N, N-dimetiletanol amina) y en Estados Unidos se emplean las AMP (2-amino-2-metil-1-propanol). El aglutinante principal normalmente es un polímero de emulsión de núcleo parcialmente reticulado basado en acrílicos y/o poliéster. Para mejorar la resistencia a los posibles impactos, el flujo y la apariencia se introducen resinas auxiliares como poliésteres acuosos o dispersiones de poliuretano.

La reología de las capas base a base de agua proviene en parte de los aglutinantes utilizados, pero en su mayor parte se crea mediante espesantes alcalinos dilatantes (del tipo ácido poliacrílico) o una mezcla de silicatos en capas.

En la siguiente Tabla 2.2 puede observarse una comparativa de diferentes propiedades, tales como, viscosidad, cantidad de compuestos orgánicos volátiles, contenido de sólidos, espesor de capa y tiempo y temperatura de secado antes de la aplicación de la capa de revestimiento, para los tres procesos de aplicación principales.

		Medium solids	High solids	Waterborne
Solid content (%)	solids	25–40	45–60	20–45
	Effect	15–25	40–50	15–25
VOC (g l ⁻¹)		450–600	250–400	100–150
Applied viscosity (s) DIN4/23 °C		20–30	15–20	35–60
(mPas) (1000 rpm)		40–50	30–40	60–120
Film build μm	solids	15–25	20–30	15–25
	Effect	10–15	15–20	10–15
Flash off before clear coat		2–3 min 23 °C	3–5 min 23 °C	3–8 min 50–80 °C

Tabla 2.2 - Comparativa de diferentes propiedades para los tres procesos de aplicación principales

En la siguiente Figura 2.31 puede observarse una comparativa de la cantidad de emisiones que son producidas en la aplicación de la capa base y la capa de revestimiento según la tecnología de aplicación seleccionada.

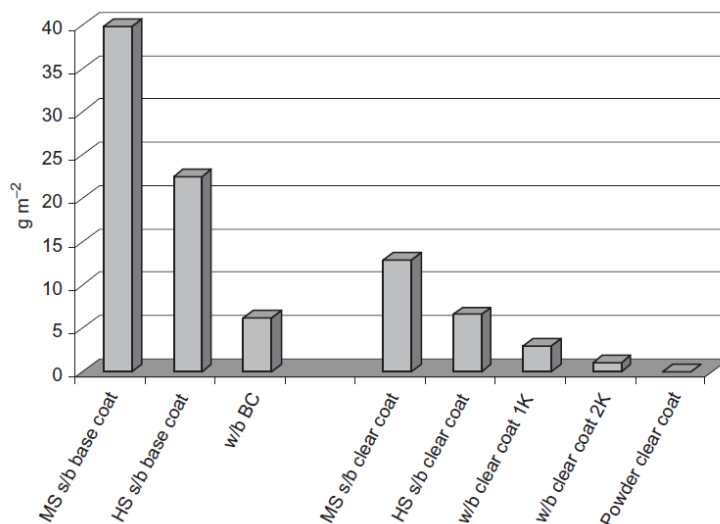


Figura 2.31 - Emisiones producidas en la aplicación de la capa base y de revestimiento según la tecnología de aplicación

En el caso de la capa base, como ya se ha mencionado anteriormente, las capas a base de disolvente con niveles de sólidos medios son las que más emisiones producen, seguidas de las capas a base de disolvente con niveles de sólidos altos y siendo las capas a base de agua las que menores emisiones producen. Si se reemplaza un sistema de recubrimiento de capa base a base de disolvente con niveles de sólidos medio por uno a base de agua, puede reducirse el total de emisiones a 35 g/m².

Métodos de aplicación

Los métodos de aplicación para la capa base, puesto que el material es líquido en todos los casos, son los mismos que se han expuesto en el apartado [2.2.4.1 Capa de imprimación](#) para las capas de imprimación líquidas.

Dicho esto, el método mayoritariamente empleado en el sector automotriz para la aplicación de la capa base, ya sea a base de disolvente o a base de agua, es la atomización electrostática centrífuga mediante rotación a alta velocidad, en el cual se emplea un atomizador con plato de campana. Este método se ha explicado ampliamente en dicho apartado, por lo que no es necesaria su explicación de nuevo.

Capa de revestimiento transparente

La capa superior y final del sistema de recubrimiento de un automóvil es la capa de revestimiento transparente. Como se ha explicado anteriormente esta capa sirve como protección a la capa base frente a los efectos ambientales, posibles arañazos, posibles ataques químicos, posibles decoloramientos debidos a la radiación ultravioleta y proporciona al vehículo un acabado suave y sin manchas. Además esta capa hace que la reparación y el mantenimiento del recubrimiento sean más sencillos.

Mientras que, como se ha visto, las tecnologías de aplicación para la capa de imprimación y la capa base son muy similares, el número de productos químicos utilizados y el número de formas de suministro es mayor con las capas de revestimiento transparentes. Para estas se tienen las siguientes tecnologías principales, denominadas por sus siglas en inglés:

- SBCC1. 1K capa de revestimiento transparente a base de disolvente (todos los tipos).
- SBCC2. 2K capa de revestimiento transparente a base de disolvente PUR (poliuretano).
- WBCC. 1K capa de revestimiento transparente a base de agua.
- PCC. Capa de revestimiento transparente en polvo.

Tecnologías de aplicación y características

Capas de revestimiento transparente líquidas

Las capas de revestimiento transparente líquidas dominan ampliamente el mercado a nivel global. Este hecho se debe a que las técnicas de aplicación y la química de las mezclas se han desarrollado y optimizado ampliamente, lo cual garantiza un rendimiento excelente del recubrimiento completo del automóvil.

La química de las mezclas varía ligeramente en función de las necesidades del cliente o del mercado al que se destina. Aun así, la mayoría de estos sistemas químicos se basan en redes centrales de resina acrílica, que son funcionalizados con grupos OH para la reticulación con resinas melamínicas o poliuretanos reactivos, con grupos epoxídicos para la reticulación con grupos carboxílicos o con grupos de carbamato para la reticulación con resinas melamínicas. En su mayoría están formulados como sistemas de un solo componente que deben ser suficientemente estables para su almacenamiento y manipulación a temperatura ambiente.

Las principales capas de revestimiento líquidas son las siguientes:

- Capa de revestimiento transparente de melamina acrílica de un componente (1K).
- Capa de revestimiento transparente basada en carbamato-melamina de un componente (1K).
- Capa de revestimiento transparente de poliuretano de dos componentes (2K).
- Capa de revestimiento transparente de poliuretano de un componente (1K).
- Capa de revestimiento transparente de ácido epoxi de uno (1K) y dos (2K) componentes.
- Capa de revestimiento transparente a base de agua.

Capas de revestimiento transparente en polvo

Esta tecnología es la más respetuosa con el medio ambiente ya que no emite ninguna cantidad de compuestos orgánicos volátiles durante su aplicación. Además de esta, otra de sus ventajas principales es que todo el material sobrepulverizado se recicla completamente y directamente, es decir, el material sobrepulverizado recogido puede volverse a utilizar sin necesidad de procesos intermedios de separación o limpieza. También se tienen otras ventajas como que no produce residuos, aguas residuales ni lodos de pintura, no se requieren disolventes para la limpieza de los equipos de aplicación, solo aspiración, se reduce la energía total consumida y

además se obtienen resultados de espesor y apariencia similares en las superficies horizontales y verticales.

Puesto que el material de aplicación está totalmente compuesto por sólidos, se requieren velocidades de flujo menores en los aplicadores, lo cual ayuda a mejorar la eficacia de transferencia a tasas superiores al 90%.

Uno de los futuros objetivos es reducir el espesor de la capa de revestimiento transparente del nivel actual de 65 μm a niveles inferiores a 40-50 μm , los cuales son comunes en las capas de revestimiento transparente líquidas.

Métodos de aplicación

Los métodos de aplicación para la capa de revestimiento transparente son los mismos que se han expuesto para las capas de imprimación, en el apartado [2.2.4.1 Capa de imprimación](#).

Dicho esto, el método mayoritariamente empleado en el sector automotriz para la aplicación de la capa de revestimiento transparente es la atomización electrostática centrífuga mediante rotación a alta velocidad para las capas líquidas y aplicación neumática mediante pulverizadores con plato de campana para capas en polvo. Estos métodos se han explicado ampliamente anteriormente, por lo que no es necesaria su explicación de nuevo.

2.2.5 Automatización del proceso de pintado

La evolución de los procesos de aplicación de pintura ha pasado por numerosas etapas hasta alcanzar su total automatización. En las primeras fases, el recubrimiento de las áreas exteriores de las carrocerías de los vehículos se realizaba mediante máquinas de pintado. Las capas individuales de pintura ya se aplicaban en ese momento mediante atomizadores rotativos de alta velocidad, excepto una, la segunda capa base de efecto metálico. Para ello, eran necesarios atomizadores neumáticos que se caracterizan principalmente por su baja eficiencia de transferencia. El concepto de máquina de pintado se desarrolló hasta alcanzar el concepto de robot de pintura. La búsqueda de una mayor flexibilidad y eficiencia impulsaron este movimiento de avance tecnológico.

Las instalaciones robotizadas proporcionan más flexibilidad en cuanto al diseño del vehículo y permiten una mayor eficiencia debido a la reducción del número de atomizadores, el aumento de la eficacia de transferencia y por lo tanto una reducción de los gastos generales. El atomizador neumático para la aplicación de la segunda capa de efecto metálico ha sido sustituido por el atomizador rotativo de alta velocidad. Actualmente, tanto para el recubrimiento exterior como para el interior se emplean atomizadores rotativos de alta velocidad, alejándose así el desarrollo de los procesos de pintura de la aplicación neumática.

Existen dos técnicas diferentes, aunque con algunas características en común, de movimiento o transporte de la carrocería para su pintado mediante robots. La primera técnica es la llamada pintura en seguimiento, que se caracteriza por el movimiento continuo, es decir, sin realizar

ninguna parada, de la carrocería a través de la estación robotizada, y la segunda es la pintura mediante el método “stop and go”. En esta última técnica es necesario que se lleve a cabo la entrada de la carrocería a la estación robotizada de forma rápida, para posteriormente fijarla y posicionarla para que los robots puedan aplicar el recubrimiento mientras está parada.

La selección de una u otra técnica tiene un impacto significativo en el diseño de la estación de pintado. La pintura en seguimiento utiliza principalmente robots con seis ejes. Este concepto se caracteriza por una tecnología menos compleja y menores costes de inversión. Además permite una mayor capacidad de pintado ya que no se necesita tiempo adicional para las técnicas especiales de transporte y posicionamiento. Sin embargo, la técnica de “stop and go” también tiene sus ventajas. El posicionamiento de la carrocería es más preciso, no viéndose influido así el proceso de aplicación por las posibles variaciones en la posición de la misma causadas por el sistema de transporte. El número de robots necesarios puede reducirse, a costa de un tiempo de permanencia en la estación más largo. En caso de fallo técnico por parte de alguno de los robots, el resto de robots pueden hacerse cargo del pintado, siempre y cuando se instalen dos robots a cada lado de la carrocería.

2.2.6 Distribución en planta y equipamiento necesario

Cabina de pintura

Una estación de pintado se divide en varias zonas. La pintura se aplica bajo las condiciones necesarias de temperatura y humedad en la cabina de pintura (Figura 2.32). El sobrepulverizado que se genera durante el pintado es absorbido por el aire que fluye verticalmente a través de la cabina y dirigiéndose hacia el lavado.



Figura 2.32 - Esquema de una cabina de pintado automatizada

La cabina se alimenta con aire mediante una instalación de suministro de aire. El aire fresco aspirado por los ventiladores se filtra en varias etapas, se calienta, se humidifica y se bombea a la cabina de manera uniforme a través de un filtro desde el techo por medio de una sala de presión, con una velocidad de corriente descendente de 0.2 – 0.5 m/s, dependiendo del proceso de aplicación.

El aire saliente, que está cargado con el sobrepulverizado, fluye a través de un sistema de lavado, donde las partículas del sobrepulverizado son absorbidas por el agua. El aire resultante se guía a través de un intercambiador de calor giratorio aire-aire o también denominado intercambiador de calor de absorción. En este, el calor y la humedad se intercambian con el suministro de aire entrante que fluye en dirección opuesta. El material de pintura retenido en el agua de lavado, se aglutina mediante la adición de un agente de coagulación y se elimina del agua mediante el uso de diversas tecnologías. El agua de la cabina circula a través de un circuito cerrado. El agua evaporada y el agua recogida del lodo de pintura se agrega de nuevo al agua de la cabina.

Robots de pintura

Buscando mejorar los procesos de pintado de piezas de todos los tamaños y formas, los fabricantes del sector automotriz están recurriendo a la tecnología robótica para aumentar la calidad, la consistencia y el rendimiento del acabado, a la vez que reducen drásticamente los costes operativos y disminuyen el material desperdiciado y los entornos peligrosos para los operadores humanos. Fáciles de instalar y programar, los robots son rápidos, fiables y pueden aplicar exactamente la misma calidad de acabado una y otra vez sin realizar paradas.

La pintura robótica proporciona un ahorro de pintura de entre un 25% y un 30% en comparación con un proceso manual. Otras ventajas de la pintura robotizada son un menor mantenimiento y limpieza, menor gasto en productos químicos para el lavado/filtrado del agua de limpieza de la cabina y reducción de las emisiones de compuestos orgánicos volátiles. Los robots de pintura mejoran la seguridad de los operadores humanos al reducir la exposición de estos a los vapores de la pintura y otros riesgos ambientales, así como al reducir las lesiones producidas por llevar a cabo movimientos repetitivos.

Los robots pueden ser utilizados para la aplicación de la capa de imprimación, de la capa o capas base y la capa transparente, así como para el proceso de sellado, utilizando materiales a base de agua, a base de disolvente, polvo, esmaltes y pegamentos o adhesivos. Un robot puede ser programado para llevar a cabo una trayectoria de pulverización compleja, aplicando los materiales de recubrimiento en diferentes zonas de la pieza con varios espesores de capa sin imperfecciones.

Un robot de pintura puede definirse como una máquina de pintura automática de libre programación con una arquitectura de cadena cinemática de tipo serie, que tiene seis ejes rotativos, tres ejes principales de posicionamiento y tres ejes de muñeca de orientación. Además los robots de pintura pueden estar instalados sobre posicionadores que permiten el desplazamiento del robot a lo largo de un eje, aumentando así su área de trabajo. Con un robot de pintura actual se pueden llevar a cabo todos los procesos de pintado del interior y el exterior de las carrocerías de los automóviles.

Los robots de pintura se componen de seis ejes de movimiento, mostrados en la Figura 2.33.

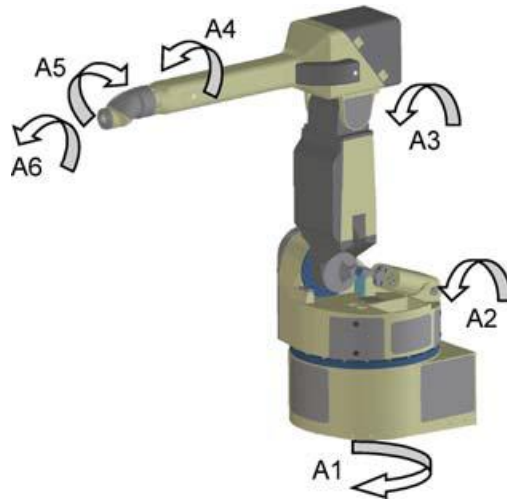


Figura 2.33 - Ejes de movimiento para un robot de pintura industrial

- Eje 1. Realiza el movimiento de giro horizontal del eslabón 1 respecto de la base.
- Eje 2. Realiza el movimiento pivotante del eslabón 2 respecto al eslabón 1.
- Eje 3. Realiza el movimiento pivotante del eslabón 3 respecto al eslabón 2.
- Eje 4. Realiza el movimiento rotatorio del eje de la muñeca respecto al eslabón 3.
- Eje 5. Realiza el movimiento de desplazamiento angular rotatorio del componente del eje medio de la muñeca.
- Eje 6. Realiza el movimiento rotatorio del eje de la muñeca en la brida final donde se instala el aplicador correspondiente.

Existen tres tipos principales de modelos de muñeca para los robots de pintura, cada una diseñada para facilitar el pintado de tipos específicos de piezas.

Muñeca Three-roll

Este diseño se caracteriza por tres rotaciones alrededor de ejes de balanceo consecutivos, en un ensamblaje de muñeca compacto (Figura 2.34). Esta arquitectura no realiza un movimiento esférico ya que los tres ejes no se cruzan en un punto común, pero la posibilidad de obtener grandes rotaciones de las articulaciones aumenta la destreza de la muñeca. Este diseño es muy adecuado para el pintado de contornos complejos, como el interior de las carrocerías de los vehículos, el interior de los objetos en forma de caja y otras piezas con contornos cerrados.

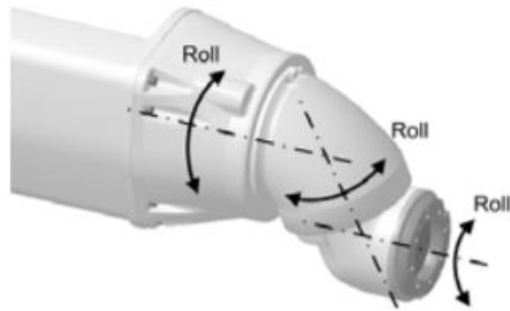


Figura 2.34 - Muñeca tipo Three-roll

Muñeca Lemma

Este diseño también proporciona tres ejes de movimiento, pero tiene un poco menos de flexibilidad para maniobrar en lugares estrechos (Figura 2.35). La muñeca Lemma es muy adecuada para pintar o recubrir a alta velocidad formas de piezas menos complejas en planos horizontales y verticales, como un marco o el exterior de un armario.

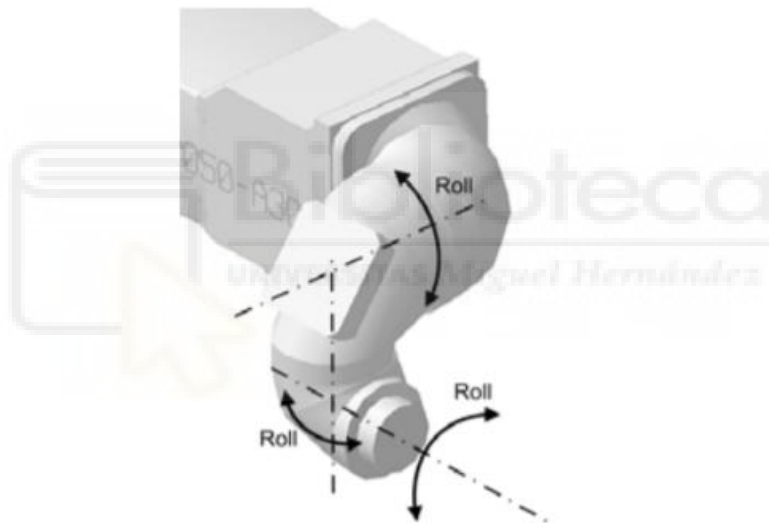


Figura 2.35 - Muñeca tipo Lemma

Muñeca Hollow

Este diseño tiene el mismo tipo de movimiento que una muñeca "Three-roll", pero incluye una gran abertura a través de la misma para las mangueras, los cables y la conexión directa de varios dispositivos de aplicación (Figura 2.36). Con este diseño se evita la interferencia entre los diferentes equipos de suministro y las piezas de trabajo o el resto de componentes de la instalación, lo cual asegura un tiempo de ciclo y un alcance y acceso del robot. La programación del robot también se simplifica sin preocupaciones por posibles interferencias con los equipos de suministro. Sin embargo, este tipo de muñecas cuestan aproximadamente entre un 10% y un 15% más que los modelos no huecos.



Figura 2.36 - Muñeca tipo Hollow

Algunas de las características importantes que debe poseer un robot de pintura son una capacidad de carga suficiente, una dinámica alta y una alta precisión de la trayectoria. Para reducir el tiempo empleado en la pintura de interiores, pueden emplearse los llamados robots de asistencia, que se emplean para las tareas de abrir las puertas y capós. Por otro lado, si el tiempo de pintura se evalúa adecuadamente, la abertura de puertas y capós mediante el propio robot de pintura puede reducir el coste de la inversión, ya que se requieren menos robots.

Los robots de pintura pueden equiparse con diferentes sistemas de atomización como el atomizador neumático, el atomizador rotativo de alta velocidad y el atomizador de polvo, que se expondrán en más detalle en apartados posteriores. Además, los robots de pintura son adecuados para su empleo en procesos de pintado que impliquen carga electrostática.

En los robots de pintura actuales todas las líneas de suministro al atomizador, cableado y mangueras, están situadas dentro del propio brazo robótico, de forma que estas no dificulten o restrinjan el movimiento altamente dinámico del mismo, haciendo así que sea más fácil de programar y pueda acceder dentro de pequeños espacios sin problemas. Por la misma razón los equipos de mezcla, la unidad de control de aire, el cambiador de color, la bomba dosificadora y otros equipos de aplicación estarán situados en el propio brazo robótico lo más cerca posible del atomizador (Figura 2.37).



Figura 2.37 - Disposición de los equipos de suministro y mezcla en el brazo robótico

En algunas aplicaciones se utilizan cubiertas desechables colocadas sobre el robot de pintura para protegerlo de la sobrepulverización de material y permitir una limpieza más sencilla (Figura 2.38). Actualmente algunos robots de pintura incluyen un revestimiento de teflón en el interior y exterior del último eslabón previo a la muñeca, el cual está hueco o tiene una configuración para posibilitar la instalación de ciertos equipos de suministro, y de la muñeca para reducir el desgaste de la manguera y el mantenimiento.



Figura 2.38 - Robots de pintura con cubiertas desechables

Los robots son ideales para operaciones de pintura líquida donde los vapores generados pueden ser peligrosos para los trabajadores humanos. Normalmente los robots utilizados para operaciones de pintura líquida deben tener una clasificación de seguridad intrínseca del tipo Factory Mutual (FM) de Clase 1 División 1, o equivalente de otro organismo de certificación, es decir, a prueba de explosiones. Para reducir el riesgo de explosión, se utiliza la purga de aire para presurizar positivamente el robot y mantener los vapores inflamables alejados de los motores eléctricos. Los recubrimientos en polvo no suelen ser inflamables como los recubrimientos líquidos, sin embargo, en algunos casos, los robots de pintura y los controladores intrínsecamente seguros siguen siendo necesarios debido al riesgo de explosión de los materiales en polvo aplicados electrostáticamente.

El control de los robots debe ser altamente eficiente, es decir, un robot debe tener un sistema operativo capaz de realizar varias tareas a la vez. Los componentes de hardware y software del sistema de control deben tener una arquitectura modular. Funciones específicas de pintado como por ejemplo una función para el reposicionamiento exacto y la continuación del proceso de pintura después de una parada del transportador, o una función que permita al robot retroceder automáticamente en la trayectoria realizada después de una parada de emergencia, lo cual es especialmente importante para la pintura de interiores, hacen que el funcionamiento del robot sea más seguro para el procedimiento y el resto de equipos y piezas de trabajo.

Algunos controladores avanzados para robots de pintura tienen mandos de programación o "Pendant" con menús incorporados específicos para la pintura, con funciones como encendido/apagado de la pistola, cambio de color y archivos de condiciones del proceso de pintado que controlan el fluido, el aire del ventilador, el aire de atomización, el alto voltaje electrostático, la velocidad del plato de campana y el aire de modelado.

Actualmente para planificar una estación de pintura robótica se utiliza la simulación mediante ordenador para seleccionar el modelo de robot y el tipo de muñeca y optimizar la distribución de los diferentes equipos en la cabina, determinando la mejor colocación de los robots en

relación con los transportadores y las paredes para eliminar posibles interferencias. Estas simulaciones permiten simular los movimientos de los robots, de las piezas y las cintas transportadoras, así como el proceso real de pulverización para determinar el mejor patrón de recubrimiento que proporcione el acabado requerido con el menor desperdicio de materiales y el mejor tiempo de producción. Los programas de simulación permiten a los fabricantes desarrollar programas para el robot fuera de línea, es decir, en un ordenador y descargarlos directamente al robot. Esto reduce o elimina el tiempo de inactividad necesario para la programación de los robots punto a punto. En la siguiente Figura 2.39 se muestra un ejemplo de simulación mediante ordenador de una estación de pintura robótica, concretamente la programada en el presente proyecto.

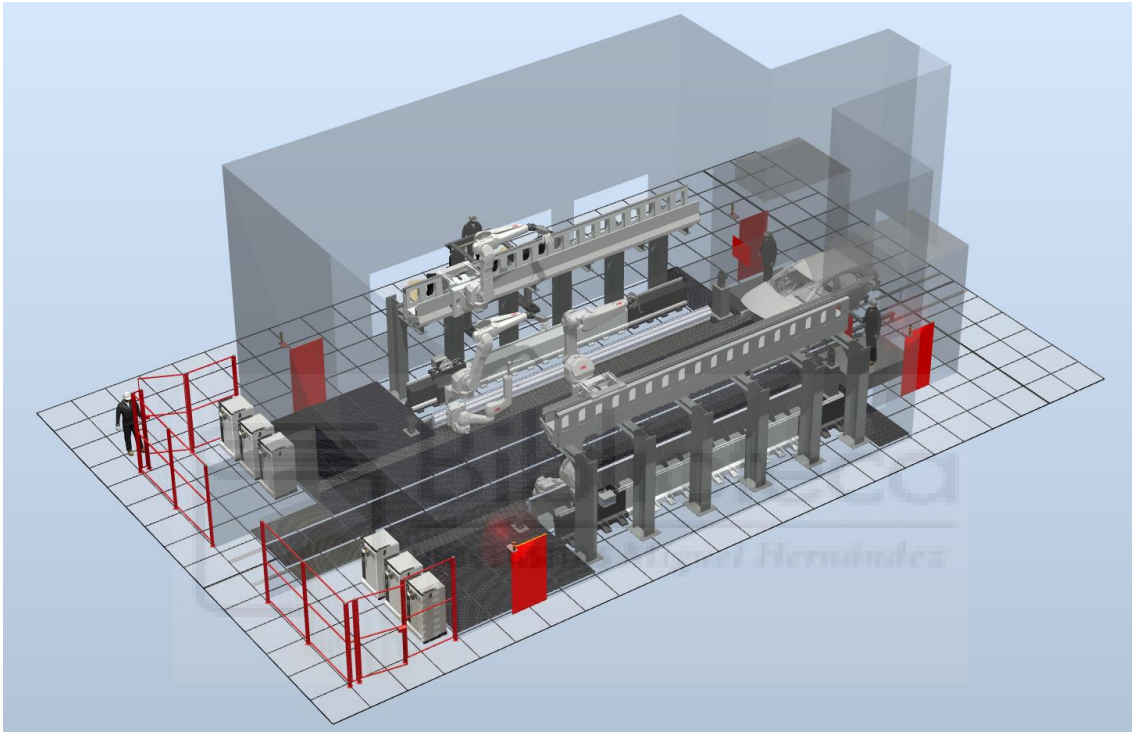


Figura 2.39 - Simulación mediante ordenador de una estación de pintura robótica

Anteriormente, los robots de pintura casi siempre estaban instalados en el suelo, lo que creaba un área de trabajo limitada debido a la interferencia entre el robot, el aplicador y la carrocería del vehículo que se desplazaba por la línea. Actualmente, los robots de pintura pueden estar instalados en la pared (Figura 2.40) o colgados del techo o de railes (Figura 2.41), lo que ofrece ventajas significativas al ampliar el área de trabajo efectiva y disminuir las potenciales interferencias. Este tipo de montajes reducen los requisitos de espacio en el suelo, lo que resulta en un ahorro adicional de costes. También provocan menos contaminación por sobrepulverizado sobre los propios robots y en las bases de transporte, lo cual reduce el mantenimiento y limpieza de estos.



Figura 2.40 - Robots de pintura instalados en pared



Figura 2.41 - Robots de pintura instalados sobre rail

Atomizador

El término atomizador hace referencia a los diferentes tipos de equipos empleados para la pulverización de los materiales de pintado. En el caso de los materiales de pintado líquidos, mediante el atomizador se lleva a cabo la atomización de los mismos, es decir, la descomposición de estos en gotas de tamaño minúsculo mediante diferentes mecanismos. En el caso de los materiales de pintura en polvo, mediante el atomizador se lleva a cabo el pulverizado de los mismos mediante aire.

Los principales requisitos de un atomizador automático son la reproducibilidad de todos los parámetros del mismo y del patrón de pulverización. Este último aspecto no sólo depende de las características de los sistemas de control, sino también de la calidad de fabricación de los componentes utilizados para la atomización, como la boquilla, el casquillo de aire, el plato de campana, etc.

Las funciones del atomizador están controladas, en su mayoría, neumáticamente por electroválvulas. El sistema de control también procesa las señales de retroalimentación del atomizador, como la posición de la aguja y la velocidad del plato de campana. Para la aplicación automática, es importante que el atomizador posea un comportamiento de respuesta muy rápido. Tanto el accionamiento como el apagado del atomizador y los cambios requeridos del valor de los volúmenes de pintura y de aire deben realizarse exactamente en los puntos programados. Este requisito no es de fácil cumplimiento, especialmente porque los sistemas neumáticos, que utilizan aire comprimido, tienen tiempos de reacción más largos, mientras que las velocidades de movimiento de los robots de pintura actuales son superiores a 1000 mm/s. Si el tiempo de conmutación del atomizador es, por ejemplo de 50 milisegundos, lo cual es un valor bastante bueno, un robot de pintura se mueve 50 mm en su trayectoria durante este tiempo. Por lo tanto, si el atomizador tiene que ser accionado en un punto determinado, las señales de control para la aguja del atomizador, la bomba dosificadora, la unidad de control de aire y demás equipos necesarios deben ser ajustadas antes.

La disponibilidad de las estaciones de pintura automática no puede verse afectada por períodos de inactividad largos para llevar a cabo tareas de mantenimiento y/o reparación. Por lo tanto, debe ser posible cambiar cada componente en el menor tiempo posible.

Actualmente los atomizadores más empleados en las estaciones de pintado del sector automotriz son los atomizadores rotativos de alta velocidad, seguidos de los atomizadores neumáticos.

Tanto los atomizadores neumáticos como los atomizadores rotativos de alta velocidad distribuyen la pintura atomizada en un patrón mediante el uso de chorros de aire. Los atomizadores neumáticos usan una combinación de "atomización" y "aire de abanico", y los rotativos usan "aire modelado" para ajustar el diámetro del patrón.

Atomizador rotativo de alta velocidad

Existen dos variantes principales de atomizadores rotativos de alta velocidad, el atomizador de plato de disco (Figura 2.42) y el atomizador de plato de campana (Figura 2.43).

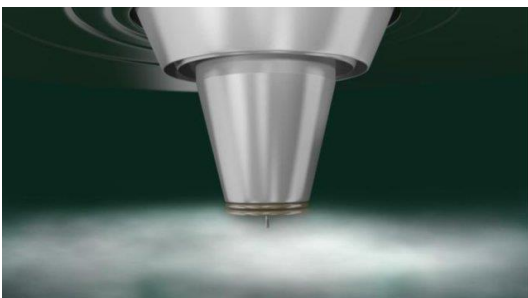


Figura 2.42 - Atomizador rotativo de alta velocidad de plato de disco

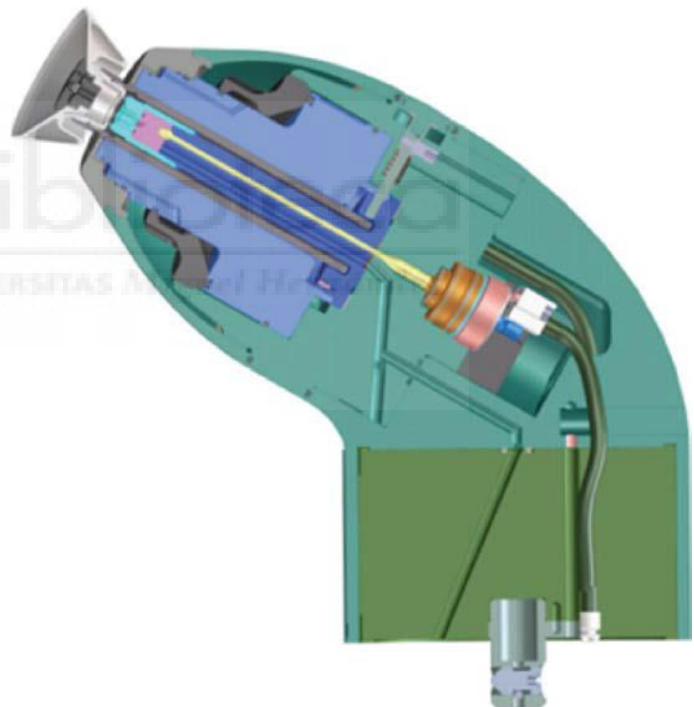


Figura 2.43 - Atomizador rotativo de alta velocidad de plato de campana

En ambos tipos de atomizador, el fluido a aplicar se alimenta centralmente hacia el disco rotativo o la campana rotativa. Cuando el fluido entra en contacto con la superficie interior del plato, disco o campana, se extiende formando una película y adquiere una carga mientras es acelerado hacia el borde por la fuerza centrífuga generada por la rotación a alta velocidad. Al contactar con el aire exterior la película se descompone primero en ligamentos y finalmente en gotas. El impulso rotacional impartido a los ligamentos de fluido hace que estos sean expulsados desde el borde del exterior del plato con una trayectoria envolvente en espiral. Dentro del ligamento, las fuerzas capilares se equilibran con las fuerzas elásticas, estando estas últimas definidas por

la reología del fluido. Cuando las fuerzas capilares son dominantes, la tensión superficial rompe el ligamento en múltiples gotas. Estas gotas conservan el impulso generado al ser expulsadas del plato en rotación, lo que hace que se extiendan radialmente hacia el exterior.

Puesto que el patrón de pulverizado generado de esta forma tendría un diámetro excesivamente grande y las gotas viajarían con poca velocidad, el atomizador está provisto de un anillo o anillos de orificios de aire de conformación. Estos están situados sobre el plato y proyectan chorros de aire que fluyen sobre el plato, dirigiendo así el fluido atomizado hacia la superficie en la que debe depositarse. Al aumentar el volumen de aire de conformación se reduce el diámetro del patrón de pulverizado. El límite inferior del diámetro del patrón lo determina el tamaño del plato y el ángulo con el que el aire de conformación se desvía alrededor del mismo. El caudal típico de aire de conformación oscila entre los 100 y 500 litros por minuto, y los diámetros típicos del patrón de pulverizados son de 150 a 500 mm.

Además del flujo de aire de conformación ya mencionado, las fuerzas del campo eléctrico, generadas al cargar eléctricamente el fluido, también intervienen en el transporte de las gotas de fluido hasta la superficie en la que se tienen que depositar, que se encuentra conectada a tierra, por lo que con este tipo de atomizadores se puede conseguir una eficiencia de transferencia de hasta un 85-90%.

La superficie interior del plato puede ser lisa o tener líneas mecanizadas en el borde llamadas estrías, con una longitud de aproximadamente de 1 a 2 mm. La finalidad de las estrías es formar intencionadamente ligamentos cuando el fluido sale del borde del plato, es decir, las estrías están diseñadas para impedir que el fluido salga del plato en forma de película. En la siguiente Figura 2.44 se muestra la disposición de los estos elementos descritos en un atomizador rotativo de plato de campana.

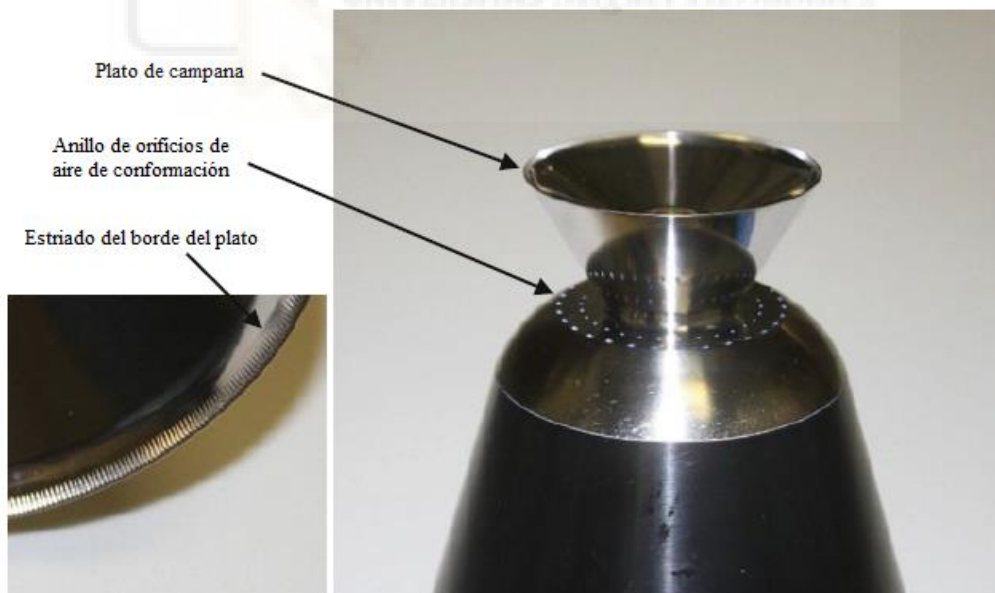


Figura 2.44 - Detalle del plato de campana de un atomizador rotativo

Los atomizadores rotativos de alta velocidad constan principalmente de un plato de campana atomizador y su accionamiento, siendo estos últimos en la mayoría de los casos turbinas de aire. Dependiendo del fluido a atomizar y de la cantidad de producción, los atomizadores funcionan a velocidades de rotación de entre 15.000 y 60.000 rpm. Esta velocidad se suele contralar

mediéndola ópticamente. Habitualmente los platos de campana tienen un diámetro de alrededor de 50 mm, con diferentes geometrías y superficies de borde.

Atomizador neumático

En los atomizadores neumáticos, el fluido se expulsa a través de una boquilla. Cuando el chorro de fluido sale de la boquilla a baja velocidad, la superficie del fluido experimenta un alto cizallamiento inducido por un flujo anular de aire de alta velocidad, generando así crestas de onda en la superficie del chorro conocidas como inestabilidad Kelvin-Helmholtz. A continuación, a medida que las crestas de las ondas crecen, el aire es arrastrado hacia la columna de líquido provocando la inestabilidad conocida como inestabilidad de Raleigh-Taylor. Por último, se forman ligamentos que se estiran hasta que se desprenden del núcleo líquido, para transformarse finalmente en gotas como resultado de la inestabilidad de Raleigh Plateau. En la siguiente Figura 2.45 se muestra un ejemplo de un atomizador neumático.

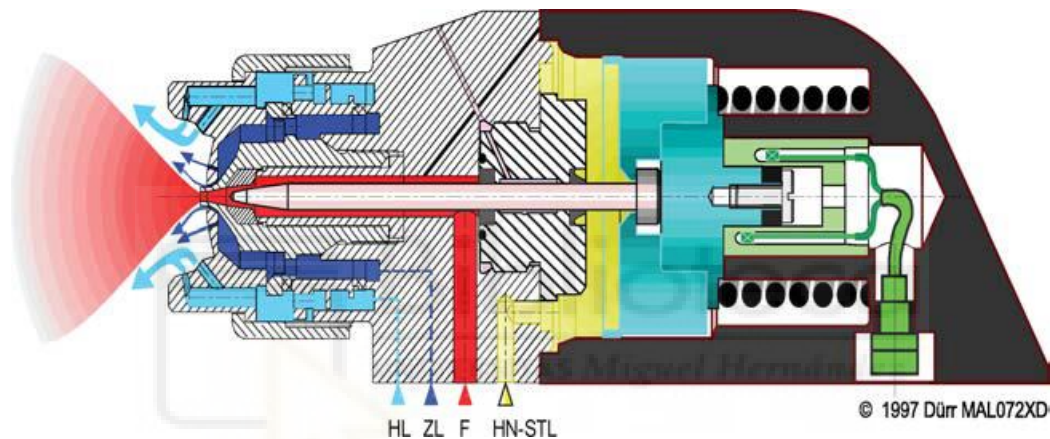


Figura 2.45 - Esquema de un atomizador neumático

El cabezal de aire y la boquilla de fluido son los componentes fundamentales para llevar a cabo la atomización del fluido, y son factores decisivos para conseguir una buena calidad de la película depositada. El aire de atomización que causa la atomización del chorro de fluido que sale desde la boquilla, fluye de forma anular a través del orificio del cabezal de aire. Otros chorros de aire procedentes de orificios circundantes sustentan el proceso de atomización y evitan la formación de una zona de baja presión con presión negativa delante del cabezal de aire, lo cual causaría el ensuciamiento del cabezal del atomizador.

Para regular la forma del patrón de pulverizado se tienen otros orificios de chorro de aire de conformación. El aire que fluye de estos orificios puede formar desde un chorro de aire redondo, que formará un patrón de pulverizado casi circular, hasta un chorro de aire en abanico que formará un patrón de pulverizado elíptico. Tanto el aire de atomización como el aire de conformación se alimentan y controlan por separado.

La boquilla de fluido se cierra y se abre mediante una aguja de boquilla que se activa neumáticamente con el aire de control. El caudal de fluido se ajusta con bombas dosificadoras. Un conducto de retorno sirve para purgar el atomizador durante el proceso de cambio de color de la pintura.

Otro equipamiento de la estación de pintado

Cambiador de color automático

Las estaciones de pintura robotizadas, especialmente las del sector automovilístico, deben ser capaces de permitir la utilización de pinturas con diferentes colores. Para ello el sistema robótico debe estar equipado con un cambiador de color automático y un sistema de dosificación (Figura 2.46).



Figura 2.46 - Cambiador de color automático

El cambio de color se debe realizar en un periodo corto de tiempo y las pérdidas derivadas del purgado del sistema de suministro deben ser mínimas.

Un sistema de cambio de color debe estar diseñado de tal manera que el cambio de color pueda realizarse en el espacio de tiempo existente entre dos piezas de trabajo consecutivas. Dependiendo de la distancia entre las piezas y de la velocidad del transportador que las mueve, a menudo se dispone de unos segundos para realizar el cambio de color. Para lograr un tiempo de cambio de color bajo y reducir al mínimo las pérdidas de pintura y disolvente generadas durante el mismo, el cambiador de color suele estar instalado directamente delante del atomizador, sobre el brazo robótico, como se muestra en la Figura 2.37.

El cambiador de color está formado por varios módulos funcionales que permiten encadenar cualquier número de conexiones de pintura. El componente central es la regleta de conexiones en la que se enroscan las válvulas de control de la pintura. Las válvulas de control de la pintura tienen conexiones de aire y se activan neumáticamente mediante electroválvulas. En la regleta de conexión se encuentra el bloque de conexión que sirve para fijar las mangueras de pintura mediante placas de conexión de mangueras.

El cambiador de color termina en el regulador de presión de la pintura. Además de las conexiones para los distintos colores, se dispone de una conexión para el disolvente y otra para el aire de impulsión.

Sistema de suministro

La tarea de un sistema de suministro de una estación de pintado es suministrar a la tecnología seleccionada para la aplicación las cantidades necesarias del material a aplicar, a la presión requerida (Figura 2.47). Para ello, con el fin de conseguir una alta calidad en la aplicación de la pintura, todos los parámetros del fluido deben mantenerse imperativamente dentro de un

rango establecido previamente. Por ejemplo, la variación de la temperatura del material de pintura provoca cambios de viscosidad en el mismo, lo cual modifica su estructura interna o puede causar corrimientos en el recubrimiento depositado.



Figura 2.47 - Ejemplo de un sistema de suministro

En la mayoría de los sistemas de pintura existe una tendencia a la sedimentación del aglutinante y del pigmento contenidos en el material de pintura. La prevención de estos movimientos de sedimentación es una tarea de suma importancia en un sistema de suministro de pintura. Esto se consigue mediante los equipos de almacenamiento de la pintura y mediante las propiedades del material de pintura. Un material de pintura no homogéneo causará inevitablemente un resultado de pintado incorrecto. Los componentes de pintura sedimentados pueden provocar atascos y el bloqueo gradual de las líneas de suministro.

En el sector industrial, con instalaciones de pintura para un gran número de piezas de tamaños variables y un consumo de material de pintura elevado, normalmente se emplean sistemas de suministro de pintura con circulación de pintura para alimentar los atomizadores. Estos sistemas de suministro de pintura se pueden clasificar en dos áreas, la sala de mezcla de pintura, con los grupos de contenedores y todos los componentes necesarios para el reparto y el suministro de los materiales de pintura, y en el sistema de líneas de circulación, con los puntos de destino.

Equipo de transporte

Las distintas zonas de la estación de recubrimiento, con sus respectivos procesos de aplicación de recubrimientos, están conectadas mediante sistemas de transporte. La ejecución de estos sistemas se ajusta según el proceso.

Salvo para los procesos de inmersión, como el electrorecubrimiento inicial, y los recubrimientos de los bajos, en los que se emplean sobre todo transportadores aéreos (Figura 2.48), en el resto de los procesos de recubrimiento se emplean transportadores de suelo (Figura 2.49). Los distintos niveles de la estación de recubrimiento se superan mediante estaciones de elevación. Las distintas áreas de la instalación están desacopladas entre sí por medio de pilas de almacenamiento intermedio y de clasificación por colores.



Figura 2.48 - Ejemplo de transportador aéreo



Figura 2.49 - Ejemplo de transportador de suelo

Mediante sistemas de identificación conectados a la carrocería y numerosos puntos de lectura presentes en la estación, se tiene conocimiento en todo momento de la posición de cada pieza de trabajo. Esto es necesario para suministrar al control de las diferentes máquinas automáticas de la estación información sobre el número identificador de la pintura y el modelo del vehículo.

2.2.7 Consideraciones prácticas del proceso de aplicación de la capa base

Puesto que el presente proyecto se centra en la aplicación en seguimiento de la capa base de una carrocería de un vehículo, en este apartado se expondrán de forma sencilla algunos aspectos prácticos a tener en cuenta en este proceso de aplicación, ya que en este proceso intervienen una multitud de aspectos complejos, siendo difícil analizarlos todos.

El objetivo principal del proceso de aplicación de la capa base sobre una carrocería de un vehículo es obtener el espesor de película de pintura requerido y una correcta uniformidad, produciendo la mínima cantidad de residuos e imperfecciones.

Las variaciones en el espesor de la película de pintura pueden conllevar a una serie de defectos relacionados con la apariencia y el flujo. Los defectos más conocidos son la llamada piel de naranja, que puede producirse si el espesor de la película es demasiado bajo, y los corrimientos, si el espesor es demasiado grueso.

Además de la resistencia a los impactos, los arañazos, la resistencia a los ácidos, la solidez del color, la adherencia y la flexibilidad, también se requieren excelentes cualidades ópticas. La calidad de la apariencia y la durabilidad requiere un control preciso tanto de la uniformidad del revestimiento depositado como de la forma en que se deposita.

Para generar una película uniforme de pintura sobre una gran superficie, es importante optimizar el patrón de pulverización producido por el atomizador. Tanto los atomizadores neumáticos como los atomizadores de plato de campana rotativos de alta velocidad distribuyen la pintura atomizada en un patrón mediante el uso de chorros de aire. Los atomizadores

neumáticos utilizan una combinación de "aire de atomización" y "aire de conformado", y los atomizadores rotativos emplean "aire de conformación" para ajustar el diámetro del patrón. La geometría del chorro de pulverización se convierte en el "cepillo" que deposita la pintura sobre a pieza de trabajo.

Es posible lograr una película de pintura más uniforme ajustando el aire de atomización y/o el aire del conformado y el caudal de pintura. Estos parámetros pueden modelarse estadísticamente mediante la metodología estándar de diseño de experimentos (DOE).

Por otro lado, tomando como técnica de aplicación la técnica de trazo e indexado, muy utilizada en pintura robotizada, mostrada en la Figura 2.50, en la cual el patrón de pintado se desplaza por un panel a pintar realizando trazos verticales representados por las líneas negras y al final de cada trazo el aplicador se desplaza o indexa un distancia fija para preparar el siguiente trazo. Los trazos continúan hasta que el patrón ha recorrido toda la superficie del panel.

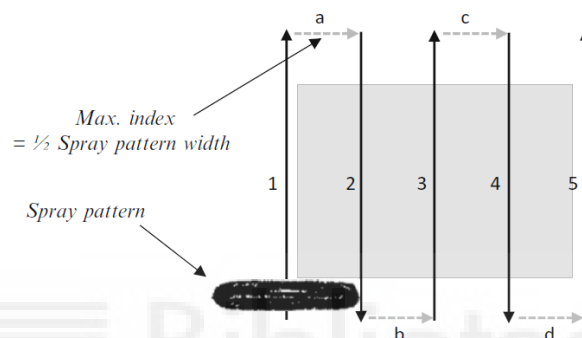


Figura 2.50 - Técnica de pintado de trazo e indexado

Los trazos en esta técnica pueden ser horizontales, verticales o incluso diagonales a través de la pieza que se está pintando, pero los trazos deben permanecer paralelos entre sí con una distancia de indexación o desplazamiento constante entre cada uno. El objetivo es depositar una película uniforme de pintura de borde a borde mediante la superposición del patrón a través de una serie de trazos paralelos, que avanzan de un extremo a otro de la pieza. Para proporcionar un solapamiento suficiente del patrón, la distancia de indexación máxima no debe ser mayor que la mitad de la anchura del patrón de pulverización, como se muestra en la Figura 2.50.

El siguiente proceso puede emplearse para optimizar la uniformidad del espesor de la película de pintura. La uniformidad del espesor final de la película de pintura sobre un panel plano pintado es una función del espesor de la película depositada a partir de una sola pasada del aplicador de pulverización y la cantidad de solapamiento entre las sucesivas pasadas. La figura 2.51 muestra un gráfico del perfil de espesor de película depositada por una sola pasada utilizando un caudal de aire y pintura específicos.

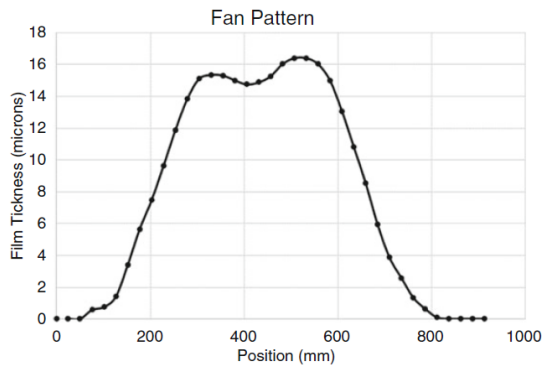


Figura 2.51 - Perfil de la película de pintura depositada por una sola pasada

Los datos de espesor del perfil se obtuvieron estableciendo los valores del caudal de aire y pintura del aplicador. A continuación, se activó el aplicador mientras se desplazaba por un panel de prueba a una velocidad y distancia definidas por el cliente. Después de hornear el panel, se tomaron mediciones del espesor de la película en intervalos regulares a través del revestimiento seco, para trazar el espesor de la película de borde a borde con el fin de crear el perfil de espesor de la película frente a la posición. La Figura 2.52 muestra el perfil de espesor de la película de la Figura 2.51 indexado tres veces sobre cuatro trazos (perfiles grises). La línea negra oscura sobre los perfiles representa el espesor final de la película y se obtiene sumando los valores de espesor de cada trazo individual. Por lo tanto, puede utilizarse para proporcionar una estimación de la uniformidad que puede esperarse del proceso. Dado que la anchura total del patrón en este caso es de 800 mm, la distancia de indexado de 400 mm mostrada representa un solapamiento del 50% de las capas pulverizadas entre dos trazos de pulverización adyacentes. Con estos valores del caudal de aire y pintura y un solapamiento del 50%, cabe esperar una variación de 5 μm de pico a valle.

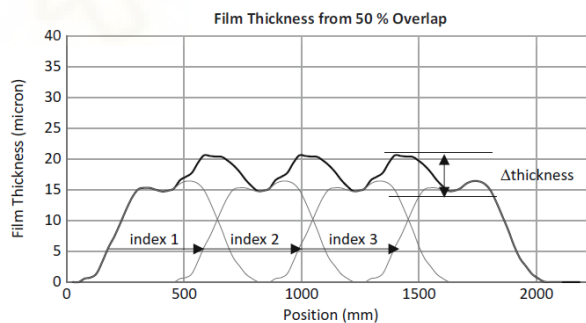


Figura 2.52 - Perfil del espesor de pintura tras 4 trazos con un 50% de solapamiento

El mismo perfil con un solapamiento del 66% da una variación de 12 μm , como se muestra en la Figura 2.53. En este caso, la distancia de indexado es de 266 mm o 1/3 de la anchura del patrón.

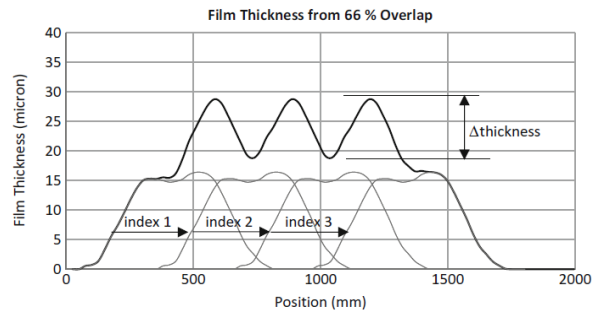


Figura 2.53 - Perfil del espesor de pintura tras 4 trazos con un 66% de solapamiento

El mismo perfil con un solapamiento del 75% da una variación de $2.5 \mu\text{m}$, como se muestra en la Figura 2.54. En este caso, la distancia de indexado es de 200 mm o $1/4$ de la anchura del patrón.

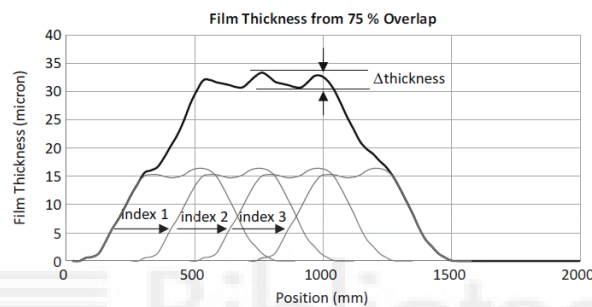


Figura 2.54 - Perfil del espesor de pintura tras 4 trazos con un 75% de solapamiento

Este ejemplo muestra la importancia del solapamiento en la uniformidad final de la película y describe un método fuera de línea para optimizar la uniformidad de un proceso de pulverización. Estos gráficos también demuestran cómo el espesor medio de la película (línea negra) es directamente proporcional al solapamiento.

El ejemplo anterior muestra que es posible optimizar la uniformidad del espesor de la película optimizando la distancia de indexado o distancia entre trazos. Sin embargo, lo más frecuente es que esta distancia no sea la primera variable que se considere cambiar. La creación de grandes trayectorias de pintado robotizadas lleva mucho tiempo y un cambio en el índice requiere una cantidad de trabajo significativa. Por este motivo, los ingenieros suelen optar por optimizar el caudal de aire. Modificando el caudal de aire utilizado en el perfil de la Figura 2.51, es posible generar el perfil mostrado en la Figura 2.55. Puede observarse cómo este perfil se responde a un solapamiento del 50% en la Figura 2.56.

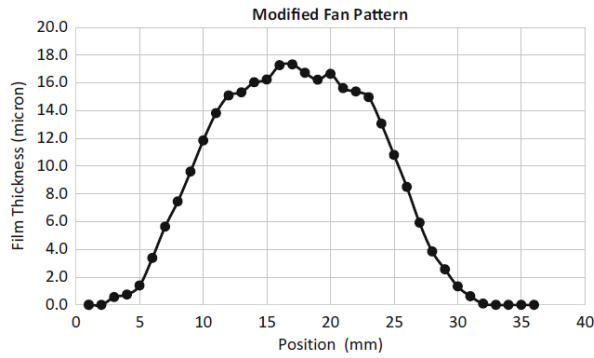


Figura 2.55 - Modificación del perfil de pintura de la Figura 2.51 mediante el caudal de aire

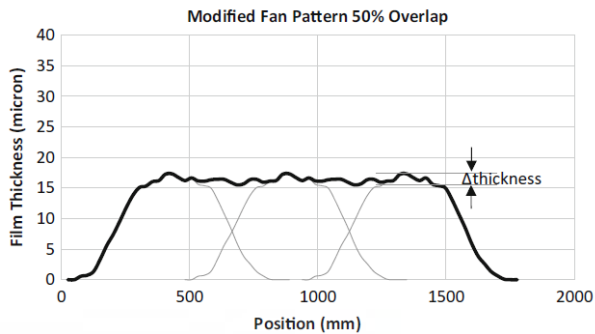


Figura 2.56 - Nuevo perfil del espesor de pintura tras 4 trazos con un 50% de solapamiento

Después de determinar el caudal de aire óptimo y el perfil de solapamiento resultante, un ajuste final del proceso de pintado implica aumentar o disminuir la velocidad de la pistola o el caudal de pintura para lograr el espesor de película especificado. Todos los colores empleados en las capas base para automoción tienen un espesor de película recomendado basado en la capacidad de lograr la "ocultación" u opacidad. El poder de ocultación es una medida de la capacidad de un recubrimiento para ocultar un fondo de color. Algunos colores, como el plateado metálico o el negro, se vuelven opacos con un espesor de película bajo, mientras que otros colores, como el rojo o el amarillo, requieren una película más gruesa para esto.

El grosor de la película de ocultación para un color específico se cuantifica pulverizando una película sobre una tira de prueba en blanco y negro, variando gradualmente el grosor de la misma de fino a grueso. Tras secarse, el espesor en el que un observador ya no puede ver una diferencia de color entre el lado blanco y el negro de la tira de prueba se denomina espesor de ocultación de la película. Para la mayoría de los colores, este espesor es también el espesor de la película de proceso recomendado. Sin embargo, en el caso de los colores que alcanzan la ocultación con un espesor de película muy bajo, suele haber una especificación mínima para garantizar las propiedades óptimas de la pintura. El proveedor de la pintura puede proporcionar un requisito de espesor de película para el sistema de recubrimiento. Una vez conocidos los valores de ocultación, el ingeniero debe considerar cómo ajustar el caudal del proceso para conseguir el espesor de película final especificado para cada color.

3. METODOLOGÍA

3.1 Introducción al software RobotStudio

3.1.1 Descripción general

RobotStudio es un software de ingeniería desarrollado por la empresa ABB destinado al control de sus sistemas robóticos. Este software permite realizar el modelado, la configuración, programación y simulación de células robotizadas fuera de línea. En el modo fuera de línea u Offline, el sistema está conectado a un controlador virtual (VC), siendo este una copia virtual de un controlador IRC5 real que se ejecuta localmente en un ordenador o un controlador IRC5 real sin conexión.

Este modo de trabajo es muy ventajoso ya que permite realizar un diseño preliminar óptimo de la célula de trabajo robótica, configurar y programar los diferentes robots sin tener que detener la producción, optimizar programas que ya están siendo utilizados sin tener que detener la producción y además permite formar a los usuarios sin ningún tipo de riesgo, tanto material como personal. De esta forma el software RobotStudio en su modo de trabajo offline permite aumentar la rentabilidad y la productividad de los sistemas robóticos de ABB.

Por supuesto, también permite realizar la configuración y programación de los robots físicos directamente en la planta de producción conectándose al controlador real del robot. Este modo de trabajo se denomina en línea u Online.

Para realizar la programación de los robots, RobotStudio emplea el lenguaje de programación propio RAPID, descrito detalladamente en el apartado [3.2 Programación básica en RAPID](#).

En el presente proyecto se ha empleado la versión de RobotStudio 2019.5.4 con una licencia de red premium de tipo escolar proporcionada por la Universidad Miguel Hernández de Elche.

3.1.2 Términos y conceptos generales

Los siguientes términos hacen referencia a los componentes de hardware estándar de una célula robotizada.

- **Manipulador de robot.** Este término hace referencia a un robot industrial ABB.
- **Módulo de control o Control Module.** Este término hace referencia a la unidad de ordenador principal que controla el movimiento del manipulador de robot. Un mismo

módulo de control puede estar conectado a entre 1 y 4 módulos de accionamiento. En el caso de los robots de pintura, un mismo módulo de control solo puede estar conectado a un módulo de accionamiento.

- **Módulo de accionamiento o Drive Module.** Este término hace referencia a una unidad que contiene los componentes electrónicos que alimentan los motores de un manipulador de robot. Un módulo de accionamiento puede contener hasta nueve unidades de accionamiento, cada una de ellas encargada de controlar un eje del manipulador de robot. Puesto que los manipuladores de robot estándar de ABB poseen seis ejes, lo habitual es emplear un módulo de accionamiento por cada manipulador de robot.
- **FlexController.** Este término hace referencia al armario de controlador de robot IRC5. Un FlexController se compone de un módulo de control y un módulo de accionamiento para cada manipulador de robot que forma el sistema.
- **FlexPendant.** Este término hace referencia a la unidad de programación in situ, es decir, la programación online o en línea. El FlexPendant está conectado al módulo de control.
- **Herramienta.** Este término hace referencia al dispositivo montado en la brida de la muñeca de un manipulador de robot de uso general que le permite realizar una aplicación particular.

Como componentes de hardware opcionales se tienen los siguientes.

- **Track.** Este término hace referencia a un soporte móvil sobre el que se monta el manipulador de robot para dotarle de un área de trabajo más grande. Si el módulo de control controla el movimiento de un manipulador de robot sobre track, este se conoce como un eje externo de track.
- **Posicionador.** Este término hace referencia a un soporte móvil sobre el que puede montarse una pieza de trabajo o un útil. Si el módulo de control controla el movimiento de un manipulador de posicionador, este se conoce como un eje externo.
- **FlexPositioner.** Este término hace referencia a un segundo manipulador de robot que desempeña la función de manipulador de posicionador. Este se controla desde el mismo módulo de control que el manipulador de robot.
- **Herramienta estacionaria.** Este término hace referencia al dispositivo montado en un posición fija que permite al manipulador de robot realizar una aplicación particular. El manipulador de robot sostiene la pieza de trabajo y la acerca al dispositivo fijo para llevar a cabo tareas concretas, como por ejemplo aplicar adhesivo o rectificar.
- **Pieza de trabajo.** Este término hace referencia al producto sobre el que se está realizando la tarea determinada.

- **Útil.** Este término hace referencia a un utensilio o estructura que sostiene la pieza de trabajo en una posición determinada, de forma que sea posible mantener la repetitividad de la tarea.

Los siguientes términos se emplean en la programación de los manipuladores de robot.

- **Programación en línea u online.** Este término hace referencia a la programación mediante conexión a un controlador real. Esto también implica el uso y movimiento del robot real para crear posiciones y configuraciones.
- **Programación fuera de línea u offline.** Este término hace referencia a la programación sin una conexión al robot ni al controlador real. En esta programación se emplea una copia virtual de un controlador IRC5 real que se ejecuta localmente en un ordenador.
- **Programación real fuera de línea.** Este término hace referencia a la programación mediante la conexión de un entorno de simulación a un controlador virtual real. Esto permite crear programas y realizar las pruebas y la optimización de los programas fuera de línea.
- **Controlador virtual.** Este término hace referencia a la copia virtual de un controlador IRC5 real (FlexController) que permite la ejecución local en un ordenador del mismo software empleado para controlar los robots, consiguiéndose el mismo comportamiento de los robots fuera de línea que el que se consigue en línea.
- **MultiMove.** Este término hace referencia a la ejecución de varios manipuladores de robot con el mismo módulo de control.
- **Sistemas de coordenadas.** Este término hace referencia a los sistemas de referencia que se utilizan para definir posiciones y orientaciones.
- **Frame.** Este término es un sinónimo de sistema de coordenadas.
- **Calibración de objetos de trabajo.** Si todos los objetivos se refieren a objetos de trabajo, sólo es necesario calibrar los objetos de trabajo al implementar los programas creados fuera de línea.

3.1.3 Interfaz gráfica de usuario

La interfaz gráfica de usuario de RobotStudio consta de siete elementos principales, mostrados en la siguiente Figura 3.1. A continuación se expondrán brevemente las principales características de los mismos.

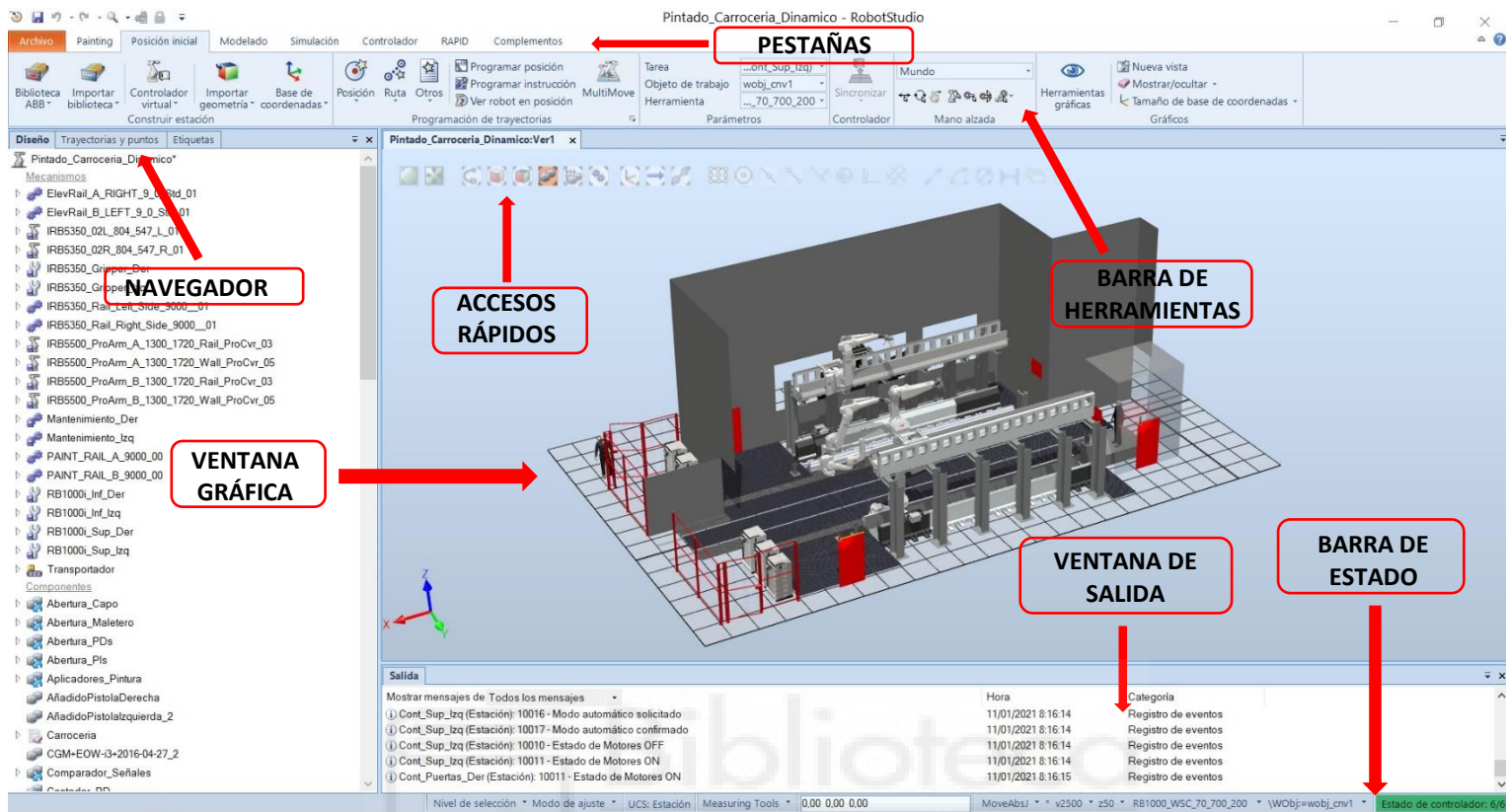


Figura 3.1 - Interfaz gráfica del software RobotStudio

Pestañas

Las pestañas contienen los diferentes menús con todas las herramientas de diseño y opciones que proporciona RobotStudio. Seleccionando la pestaña deseada se accede a ese menú que a su vez está dividido en grupos de herramientas de diseño. Por ejemplo, en la Figura 3.1, se observa que la pestaña **Posición inicial** está dividida en los grupos **Construir estación**, **Programación de trayectorias**, **Parámetros**, **Controlador**, **Mano alzada** y **Gráficos**.

Las diferentes pestañas de las que se compone RobotStudio son las siguientes:

- **Archivo.** Esta pestaña contiene las opciones para crear una nueva estación, crear un sistema de robot, conectarse a un controlador, guardar la estación, muestra información y metadatos acerca de la estación activa actualmente, enumera las estaciones abiertas recientemente, etc. (Figura 3.2)

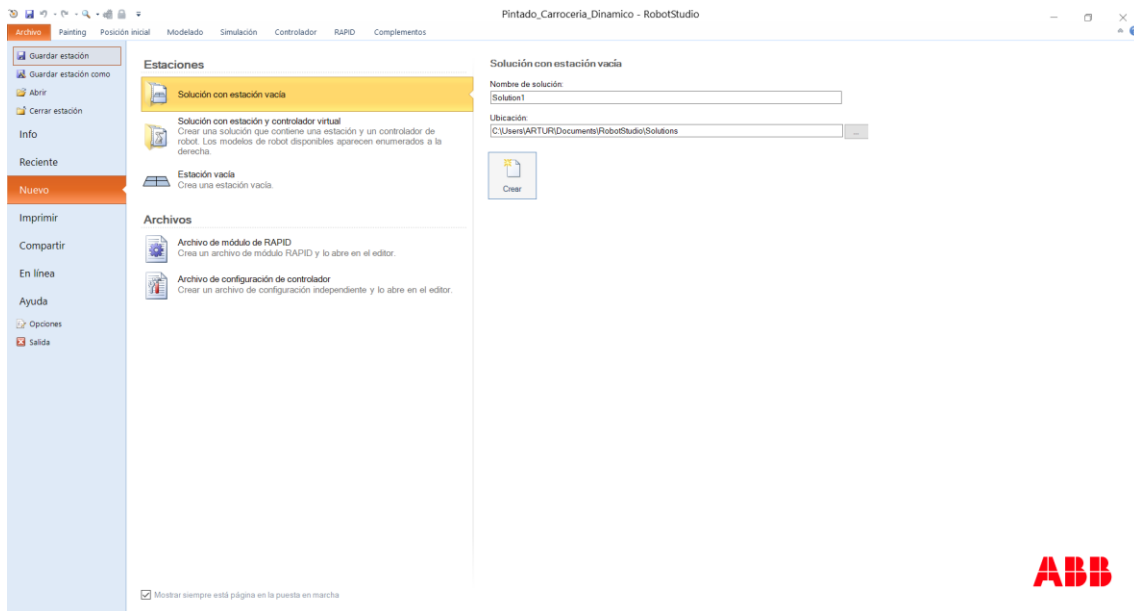


Figura 3.2 - Pestaña Archivo

- **Posición inicial.** Esta pestaña contiene las herramientas para construir estaciones, crear controladores de robot, programar objetivos y trayectorias, colocar elementos y personalizar los gráficos. Esta pestaña ofrece además los navegadores **Diseño, Trayectorias y puntos y Etiquetas.** (Figura 3.1)
- **Modelado.** Esta pestaña contiene las herramientas para crear y agrupar componentes gráficos, crear cuerpos, ejes, cables, etc., realizar mediciones, operaciones de CAD y crear mecanismos, como transportadores, herramientas, etc. Esta pestaña ofrece además los navegadores **Física, Diseño y Etiquetas.** (Figura 3.3)

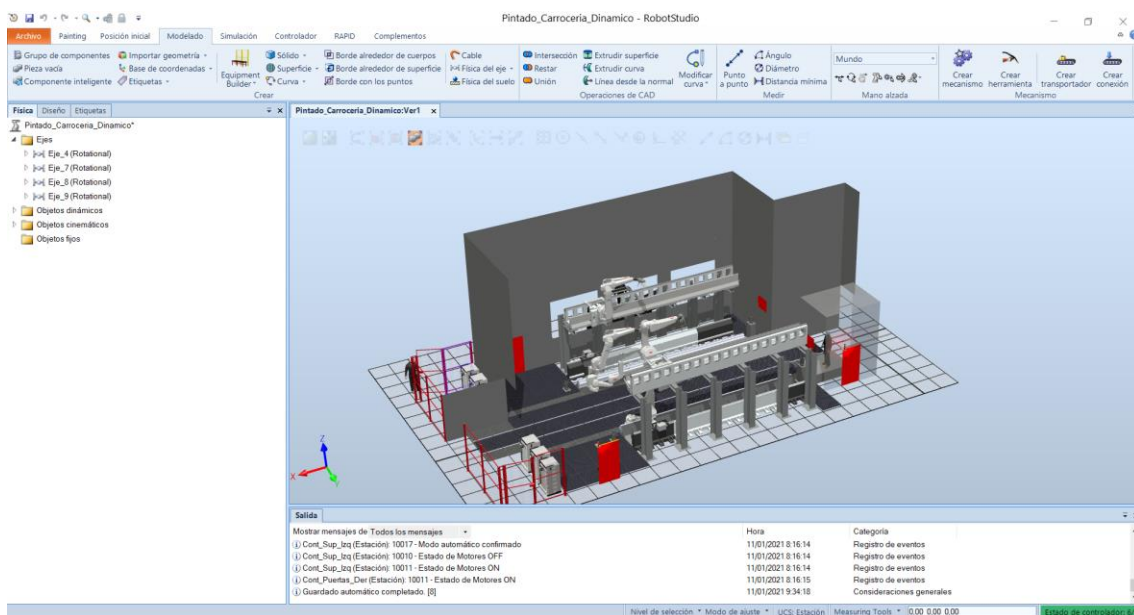


Figura 3.3 - Pestaña Modelado

- **Simulación.** Esta pestaña contiene las herramientas para iniciar, configurar, controlar, monitorizar y grabar simulaciones, crear conjuntos de colisión y analizar las señales de la estación. Esta pestaña ofrece además los navegadores **Diseño**, **Traectorias y puntos** y **Etiquetas.** (Figura 3.4)

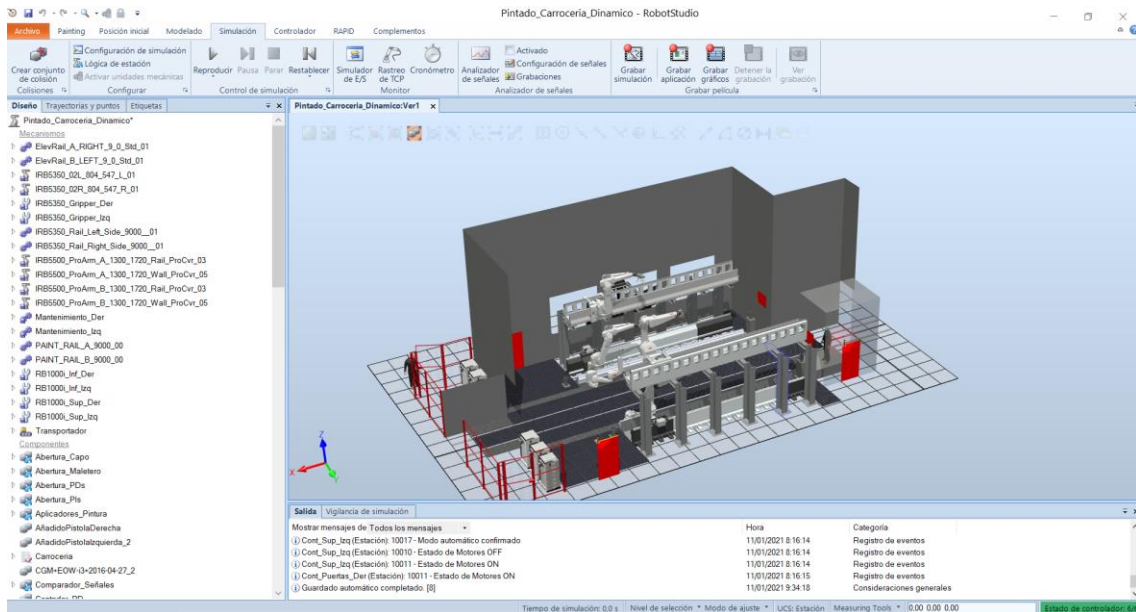


Figura 3.4 - Pestaña Simulación

- **Controlador.** Esta pestaña contiene los controles para la sincronización, configuración y tareas asignadas al controlador virtual. También contiene controladores para gestionar los controladores reales. Esta pestaña ofrece además el navegador **Controlador.** (Figura 3.5)

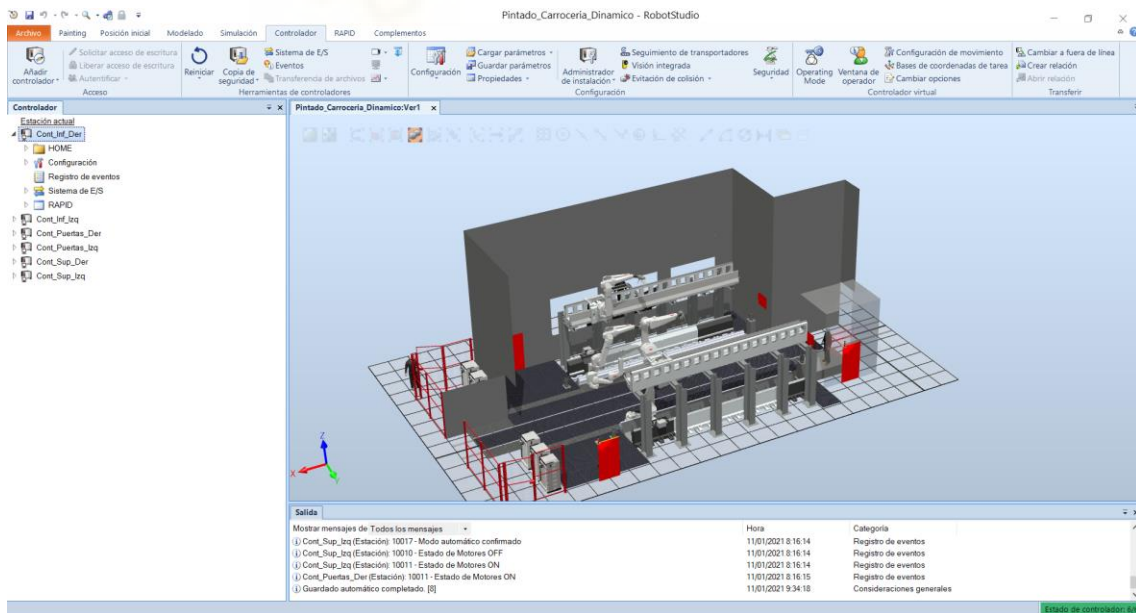


Figura 3.5 - Pestaña Controlador

- **RAPID.** Esta pestaña contiene el Editor de lenguaje RAPID integrado, que se emplea para programar, editar y administrar el conjunto de instrucciones que forman la tarea del robot. Esta pestaña ofrece además los navegadores **Controlador** y **Archivos**. (Figura 3.6)

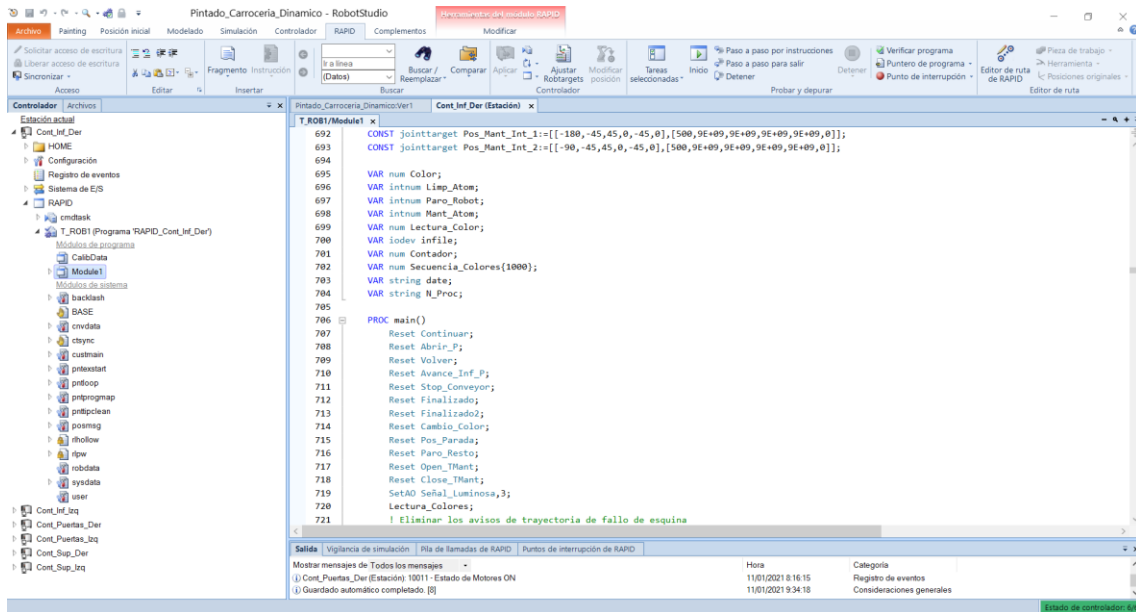


Figura 3.6 - Pestaña RAPID

- **Complementos.** Esta pestaña contiene los controles de los PowerPacs y RobotWare, así como una herramienta de predicción de calor de caja reductora y las RobotApps, una serie de recursos, como aplicaciones, modelos, SmartComponents, etc., compartidos por la comunidad de RobotStudio. Esta pestaña ofrece además el navegador **Complementos**. (Figura 3.7)

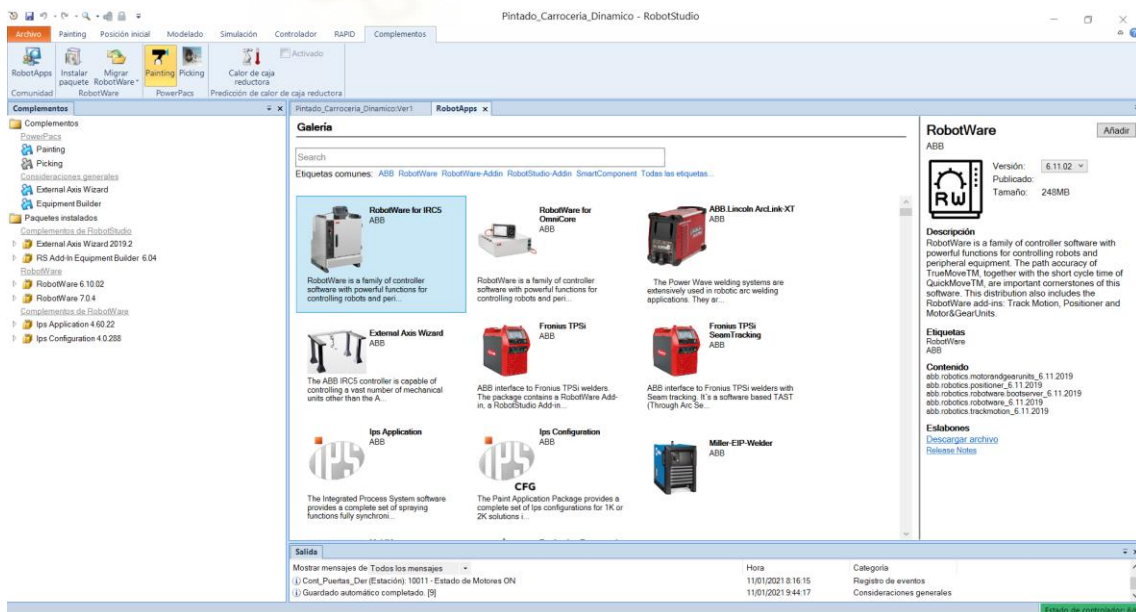


Figura 3.7 - Pestaña Complementos

Navegador

Esta ventana ofrece diferentes opciones según la pestaña en la que se encuentre el usuario. Los más relevantes son los siguientes:

- **Diseño.** En este navegador se muestran diferentes mecanismos y componentes gráficos que forman la estación. (Figura 3.8)

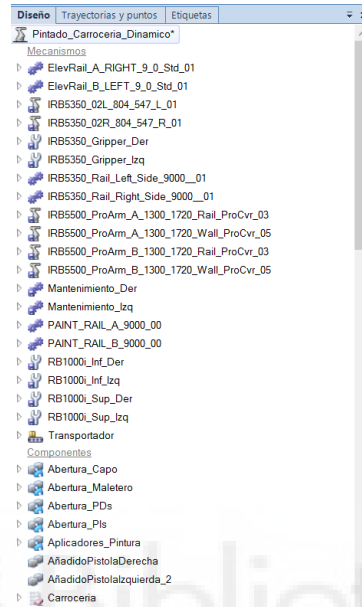


Figura 3.8 - Navegador Diseño

- **Trayectorias y puntos.** En este navegador se muestran los objetos de trabajo, objetivos, trayectorias y procedimiento programados para cada controlador. (Figura 3.9)

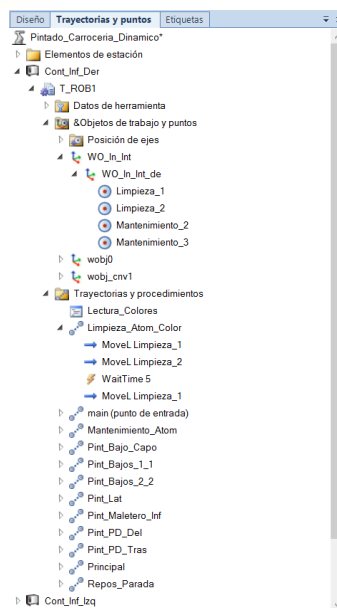


Figura 3.9 - Navegador Trayectorias y puntos

- **Física.** En este navegador se muestran las posibles articulaciones de objetos presentes en la estación y el tipo de comportamiento físico de los componentes de la estación. (Figura 3.10)

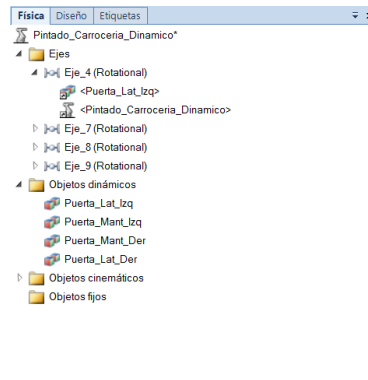


Figura 3.10 - Navegador Física

- **Controlador.** En este navegador se muestran los principales parámetros y configuraciones de cada controlador virtual presente en la estación. (Figura 3.11)

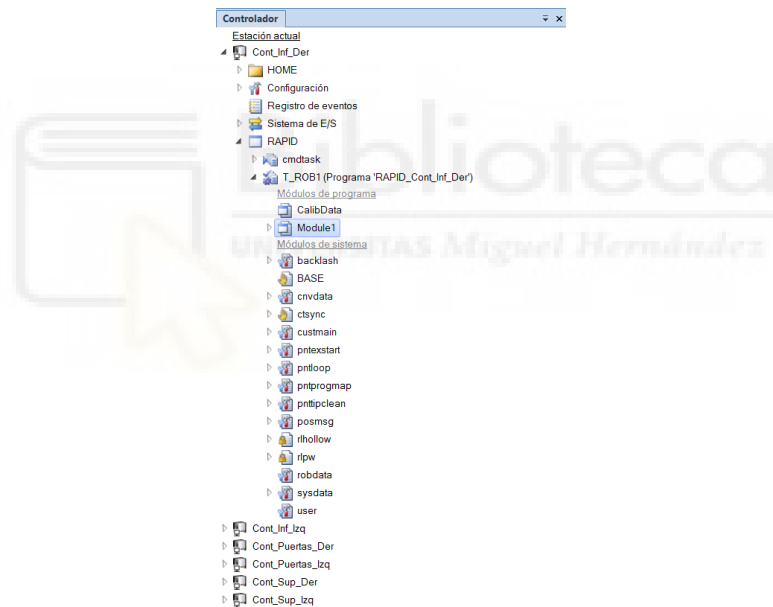


Figura 3.11 - Navegador Controlador

Accesos rápidos

Como su propio nombre indica, se trata de accesos rápidos, mostrados en la ventana gráfica, a las herramientas de programación gráfica utilizadas habitualmente. Como por ejemplo, cambiar el nivel de vista, selección de superficies, cuerpos, piezas, etc., herramientas de medición, etc. (Figura 3.12)

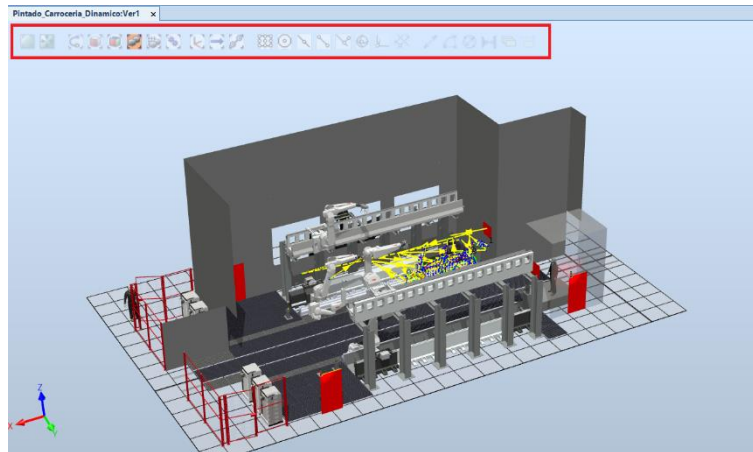


Figura 3.12 - Accesos rápidos en la ventana gráfica

Barra de herramientas

La barra de herramientas de cada pestaña agrupa todos los comandos, herramientas, opciones, etc. que hacen posible el diseño y programación de la estación robótica. Cada barra de herramientas a su vez se divide en grupos de herramientas de diseño.

Ventana gráfica

En la ventana gráfica se muestra una o varias representaciones gráficas tridimensionales del espacio de trabajo de la estación y todos sus componentes gráficos. Mediante esta representación se hace más sencilla y productiva la programación de los objetivos y las instrucciones de movimiento, que permiten al manipulador de robot llevar a cabo su tarea, y la colocación de los componentes en su posición correcta.

Si nos encontramos en la pestaña de **Simulación**, en la ventana grafica también pueden representarse la configuración de la simulación y la lógica de estación, permitiendo modificarla y optimizarla de una forma más visual y sencilla. (Figura 3.13)

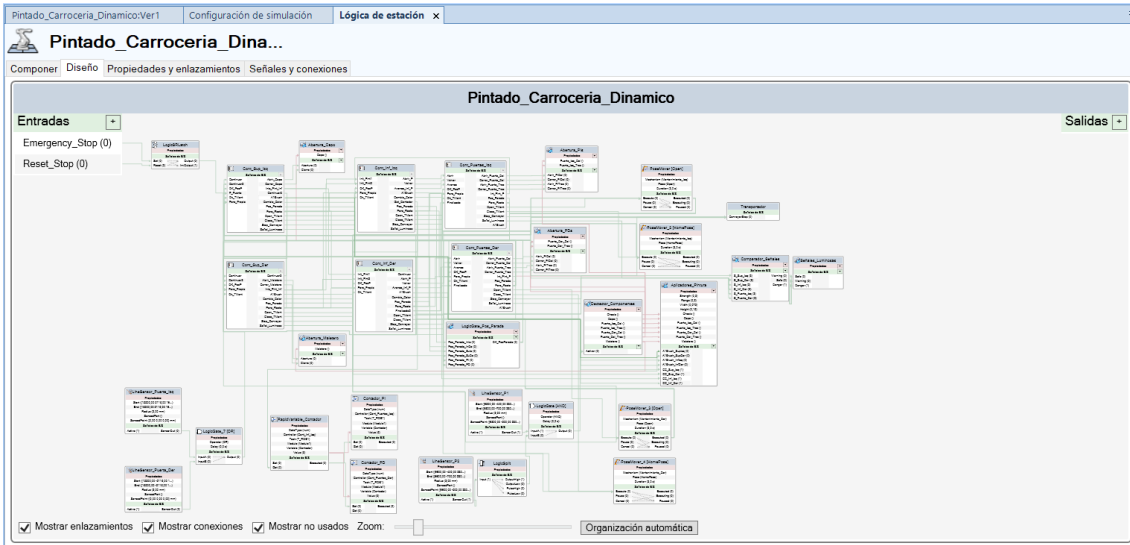


Figura 3.13 - Ventana gráfica de la Lógica de estación

Si nos encontramos en la pestaña **Controlador** o la pestaña **RAPID**, en la ventana grafica también pueden representarse las diferentes opciones de configuración del controlado del robot (Figura 3.14) y el código de RAPID seleccionado (Figura 3.15).

Tipo	Name	Type of Signal	Assigned to Device	Signal Identification Label	Device Mapping	Category	Access Level	Default Value	Filter Time Passive (ms)
Access Level	A1BellRotating	Digital Input			N/D		All	0	0
Cross Connection	A1Brush	Group Output			N/D		All	0	N/D
Device Trust Level	A1BrushTable	Analog Input			N/D		All	0	N/D
DeviceNet Command	A1BrushTabLimEn	Digital Output			N/D		All	0	N/D
DeviceNet Device	A1BrushTabNdx	Group Output			N/D		All	0	N/D
DeviceNet Internal Device	A1Erasable	Digital Output			N/D		All	0	N/D
EtherNet/IP Device	A1Error	Digital Output			N/D		All	0	N/D
EtherNet/IP Command	A1Fluid	Analog Output			N/D		All	0	N/D
EtherNet/IP Device	A1FluidAccu	Analog Output			N/D		All	0	N/D
ICI Device	A1FluidEn	Digital Output			N/D		All	0	N/D
Industrial Network	A1FluidPITO	Digital Output			N/D		All	0	N/D
PROFINET Common Data	A1FluidPIWS	Digital Output			N/D		All	0	N/D
PROFINET Device	A1HVE	Digital Output			N/D		All	0	N/D
PROFINET Internal Device	A1HVErrNo	Group Output			N/D		All	0	N/D
RobCI Device	A1Material	Group Output			N/D		All	0	N/D
RobCI Device	A1NeedleEn	Digital Output			N/D		All	0	N/D
RobCI Server	Abrr_P	Digital Output			N/D		All	0	N/D
Route	AppEnabled	Digital Output			N/D		Default	0	N/D
Signal	AppError	Digital Output			N/D		Default	0	N/D
Signal Safe Level	AS1	Digital Input	PANEL	Automatic Stop chain(X5:11 to X5:6) and (X5:9 to X5:1)	13	safety	ReadOnly	0	0
System Input	AS2	Digital Input	PANEL	Automatic Stop chain backup(X5:5 to X5:6) and (X5:3 to X5:1)	14	safety	ReadOnly	0	0
System Output	AUTO1	Digital Input	PANEL	Automatic Mode(X9:6)	5	safety	ReadOnly	0	0
System Output	AUTO2	Digital Input	PANEL	Automatic Mode backup(X9:2)	6	safety	ReadOnly	0	0
	AutoModeOn	Digital Output			N/D		All	0	N/D
	Avance_Inf_P	Digital Output			N/D		All	0	N/D
	BellRotating	Digital Output			N/D		Default	0	N/D
	c1CntFromEnc	Group Input	CTM1	Sync1Enc1.Value			Default	0	0
	c1CntToEnc	Group Output			N/D		Default	0	N/D
	c1CntToEncStr	Digital Output			N/D		All	0	N/D
	c1Connected	Digital Output			N/D		Default	0	N/D
	c1Counts	Group Input	CTM1	Enc1.position			Default	0	0
	c1CountsPerSec	Analog Input	CTM1	Enc1.velocity			Default	0	N/D
	c1DropWObj	Digital Output			N/D		All	0	N/D
	c1NewObjStrobe	Digital Output			N/D		All	0	N/D
	c1NullSpeed	Digital Output			N/D		Default	0	N/D

Figura 3.14 - Ventana gráfica de la configuración del controlador de robot

```

685 CONST robtarg Pos_Sal_Malet1:=[[133.378,-297.145,2145.168],[0.591311481,-0.570551212,0.411216842,-0.394617227],[-1,-1,-1,0],[-300,9E+09,9E+09,9E+09]]
686 CONST robtarg Pos_Sal_Malet2:=[[-32.728188909,-2386.802457234,1985.830432144],[0.286787956,-0.736279172,0.286787956,0.541662549],[0,0,-1,0],[1200,9E+09,9E+09,9E+09]]
687 CONST robtarg Pos_Sal_Baj1_1:=[[3937.765,-692.525,914.824],[0.46818888,0.331201992,-0.44717531,-0.686395407],[-2,0,0,-1],[1200,9E+09,9E+09,9E+09]]
688 CONST robtarg Pos_Sal_Baj2_1:=[[1058.881,-855.003,964.62],[0.635068602,-0.333697338,0.33237484,-0.612258869],[-1,-2,0,0],[-480,9E+09,9E+09,9E+09]]
689 CONST jointtarget Pos_Mant_Int_3:=[[0,-45,45,0,-45,0],[500,9E+09,9E+09,9E+09,9E+09,0]];
690 CONST robtarg Mantenimiento_2:=[[2025.24,-350,1360],[0.707106781,0,0.707106781,0],[-1,1,-2,-1],[0,9E+09,9E+09,9E+09,9E+09,0]];
691 CONST robtarg Mantenimiento_3:=[[2675.24,-350,1360],[0.707106781,0,0.707106781,0],[-1,1,-2,-1],[0,9E+09,9E+09,9E+09,9E+09,0]];
692 CONST jointtarget Pos_Mant_Int_1:=[[-180,-45,45,0,-45,0],[500,9E+09,9E+09,9E+09,9E+09,0]];
693 CONST jointtarget Pos_Mant_Int_2:=[[-90,-45,45,0,-45,0],[500,9E+09,9E+09,9E+09,9E+09,0]];
694
695 VAR num Color;
696 VAR intnum Limp_Atomo;
697 VAR intnum Paro_Robot;
698 VAR intnum Mant_Atomo;
699 VAR num Lectura_Color;
700 VAR iodev infile;
701 VAR num Contador;
702 VAR num Secuencia_Colores{1000};
703 VAR string date;
704 VAR string N_Proc;
705
706 PROC main()
707     Reset Continuar;
708     Reset Abrir_P;
709     Reset Volver;
710     Reset Avance_Inf_P;
711     Reset Stop_Conveyor;
712     Reset Finalizado;
713     Reset Finalizado2;
714     Reset Cambio_Color;

```

Figura 3.15 - Ventana gráfica con el código RAPID de programa de robot para un controlador

Ventana de salida

En la ventana de salida se muestra la información sobre los eventos que se producen en la estación, como inicios de simulación, errores, etc. Esta información es muy importante a la hora de detectar un error y poder comprenderlo para poder corregirlo.

Barra de estado

En esta barra se muestra el estado y el modo en el que se encuentran los controladores virtuales de la estación, la instrucción de movimiento activa de la plantilla de instrucciones, junto con sus argumentos, y una serie de accesos rápidos a diferentes opciones de diseño. (Figura 3.16)

Controlador	Estado	Modo
Controladores de estación		
Cont_Sup_Izq	Iniciado	Auto
Cont_Sup_Der	Iniciado	Auto
Cont_Inf_Izq	Iniciado	Auto
Cont_Puertas_Izq	Iniciado	Auto
Cont_Inf_Der	Iniciado	Auto

Figura 3.16 - Barra de estado

3.1.4 Creación de una estación

Un vez expuestos y comprendidos los aspectos iniciales de RobotStudio puede iniciarse la creación de una nueva estación.

El primer paso será diferenciar entre los términos **estación** y **solución**. Una **estación** es la representación tridimensional de la célula de robótica mediante un conjunto de componentes que presentan una relación espacial. En el documento de estación se guarda información sobre los distintos componentes que la forman, la lógica de estación, los gráficos tridimensionales, los datos de CAD de los componentes y los datos sobre la parte gráfica del programa de robot. El documento de estación está vinculado con el controlador virtual, pero los datos de este son externos al documento de estación. El documento de estación tiene la extensión .rsstn.

Una **solución** es el nombre que se le da a la carpeta que contiene la estructura para estaciones, bibliotecas y todos los elementos relacionados. La solución ayuda a estructurar los datos de estación, agrupando los archivos de tipo similar en subcarpetas. En la estructura de soluciones existen subcarpetas dedicadas para la estación, los controladores virtuales, la biblioteca, las diferentes geometrías, los programas de RAPID, posibles paquetes de distribución y copias de seguridad. La carpeta de solución se crea automáticamente cuando el usuario crea una nueva solución. Cuando se crea una nueva solución se crean automáticamente las subcarpetas Backups, Libraries, RAPID Programs, SignalAnalyzer, Stations y Virtual Controllers.

Una vez expuesto esto, el software RobotStudio ofrece diferentes opciones para crear una nueva estación, como se muestra en la Figura 3.17.

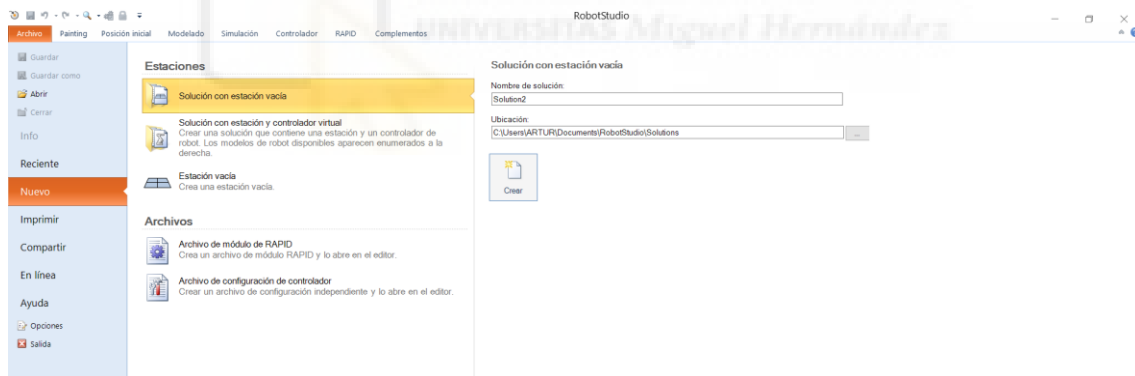


Figura 3.17 - Opciones para la creación de una nueva estación robótica

- **Estación vacía.** Crea un archivo de estación completamente vacía, sin componentes ni controladores, de forma que el usuario pueda diseñarla totalmente desde 0. En este caso no debe especificarse ningún dato de inicio.
- **Solución con estación vacía.** Esta opción crea una carpeta de solución que incluye las subcarpetas Libraries y Stations y un archivo de estación completamente vacía. Cuando se añada el controlador del robot se crearán automáticamente el resto de subcarpetas que debe tener una solución. En este caso deberá especificarse inicialmente el nombre de la solución y la ubicación donde se quiere guardar.

- **Solución con estación y controlador virtual** (Figura 3.18). Esta opción crea una carpeta de solución que incluye todas las subcarpetas y un archivo de estación con un robot seleccionado. Inicialmente se debe especificar el nombre de la solución, la ubicación donde se quiere guardar, el nombre del controlador del robot, la ubicación donde se guarda dicho controlador, el robot que se quiere importar y la versión de dicho robot en el caso de que disponga de más de una. También puede seleccionarse la casilla de personalizar opciones, para seleccionar más opciones del controlador además de las que vienen por defecto. Por otro lado, puede crearse el controlador a partir de otro controlador que se tenga guardado, siempre que estén configurados para el mismo robot.

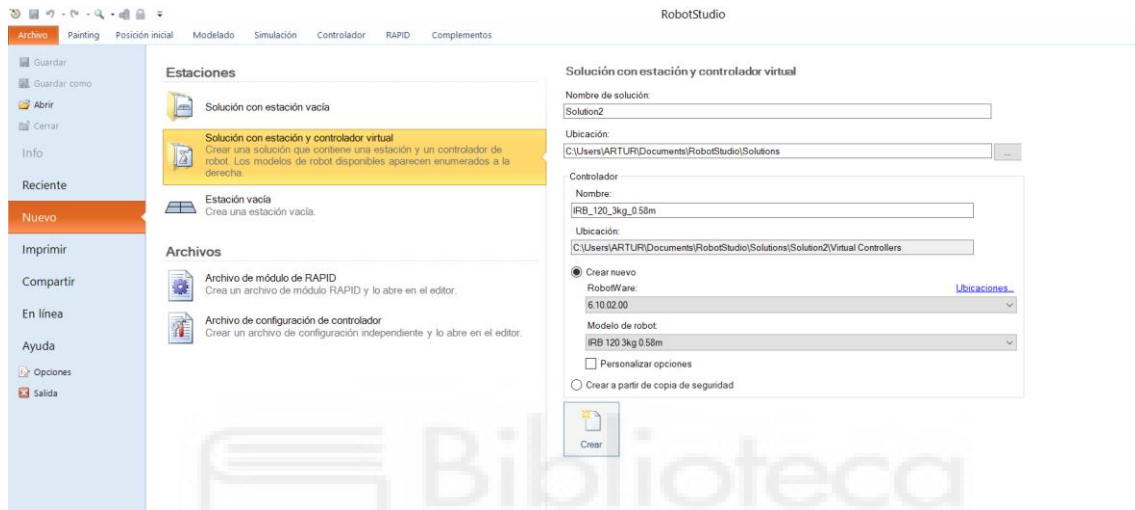


Figura 3.18 - Opción de creación de estación Solución con estación y controlador virtual

3.1.5 Importación de robots y otros componentes de la estación

El siguiente paso tras crear una estación vacía o una solución vacía es importar los manipuladores de robot requeridos y el resto de componentes de la estación.

Esta tarea puede llevarse a cabo de forma progresiva y conjuntamente con el desarrollo de la estación, aunque existen ciertos componentes, controlados por el controlador virtual, que es altamente recomendable que se añadan y posicionen antes de añadir el controlador virtual del robot, como por ejemplo los robots, un track o un posicionador.

Para importar un manipulador de robot en la estación se deben seguir estos pasos:

1. Se accede a la pestaña **Posición inicial** seleccionándola con el cursor del ratón.
2. En el grupo **Construir estación**, se selecciona la opción **Biblioteca ABB**.
3. Aparece un desplegable con todos los modelos de robot disponibles agrupados por tipo de robot. Se selecciona el robot requerido para llevar a cabo la tarea. (Figura 3.19)

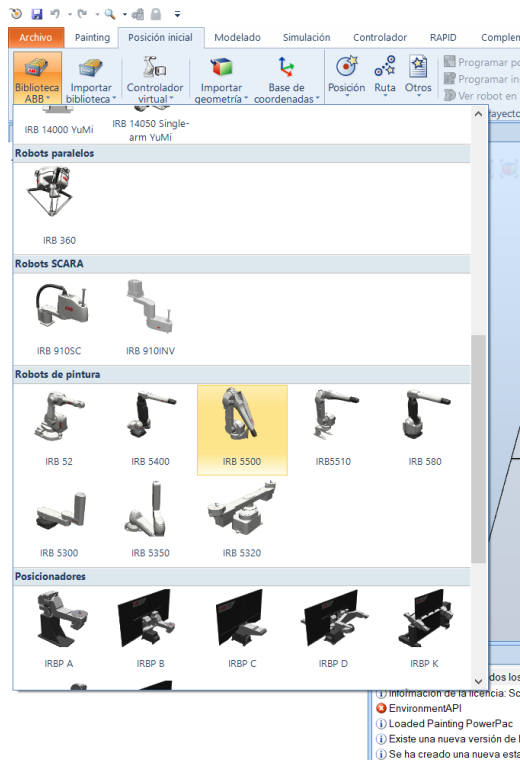


Figura 3.19 - Robots disponibles para su importación

4. Cuando se selecciona el robot aparece el cuadro de diálogo del modelo de robot elegido. En este cuadro de diálogo se especifica la versión del robot y otras características que dependen del robot seleccionado. En la siguiente Figura 3.20 se expone el cuadro de diálogo para el robot IRB 5500.

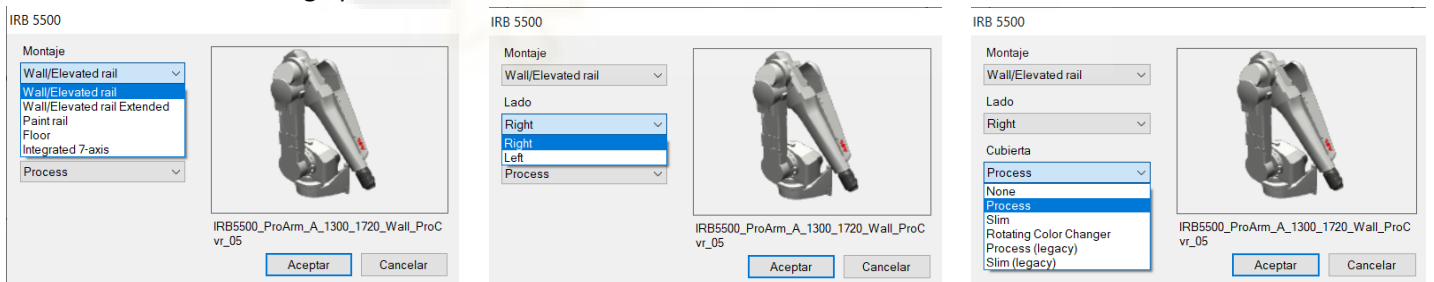


Figura 3.20 - Opciones de configuración del modelo de robot IRB 5500 a importar

5. Una vez seleccionas las opciones, se pulsa **Aceptar** y el robot se añade a la estación, mostrándose en la **Ventana gráfica** y en el navegador **Diseño**. (Figura 3.21)

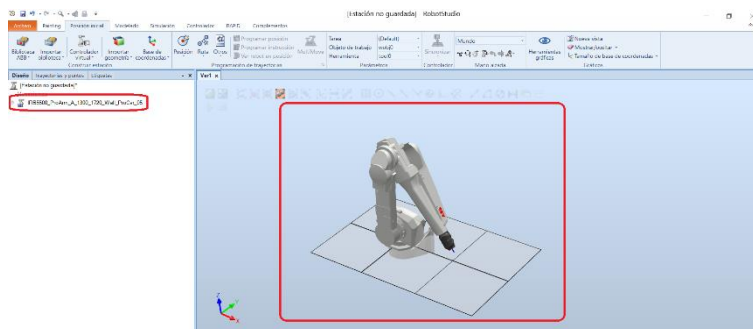


Figura 3.21 - Ejemplo de importación de un manipulador de robot

Para importar otros componentes a la estación se dispone de dos opciones diferentes.

Para importar componentes creados mediante otros softwares de diseño CAD (SolidWorks, Inventor, CATIA, Blender, SketchUp, etc.) los pasos a seguir son los siguientes:

1. Se accede a la pestaña **Posición inicial** o la de **Modelado** seleccionándola con el cursor del ratón.
2. En el caso de estar en la pestaña de **Posición inicial** en el grupo **Construir estación**, se selecciona la opción **Importar geometría**.
3. En el caso de estar en la pestaña de **Modelado** en el grupo **Crear**, se selecciona la opción **Importar geometría**.
4. Aparece el desplegable mostrado en la Figura 3.22. En él se puede elegir entre la opción de **Geometría de usuario**, en el caso de que se hayan guardado previamente, la opción de **Documentos**, mediante la cual se abre un cuadro de dialogo con diferentes opciones para buscar la geometría a importar, la opción de **Ubicaciones**, que muestra un cuadro de dialogo con las posibles ubicaciones de geometría dentro de RobotStudio y la opción de **Buscar geometría**, mediante la cual se abre el explorador de archivos de Windows para buscar la geometría a importar. Este última es la opción más empleada por su versatilidad.

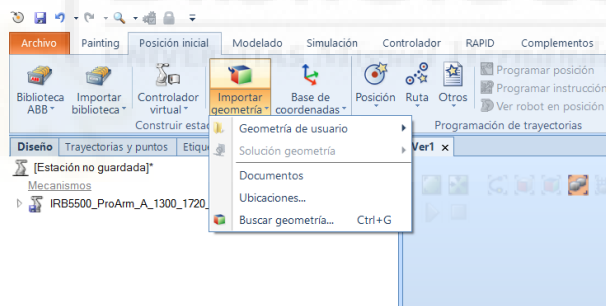


Figura 3.22 - Opciones para importar una geometría CAD

5. Los tipos de archivo que RobotStudio permite importar son se muestran el siguiente Figura 3.23. Si el archivo es de otro tipo primero se deberá convertir a uno de los permitidos mediante algún software externo.

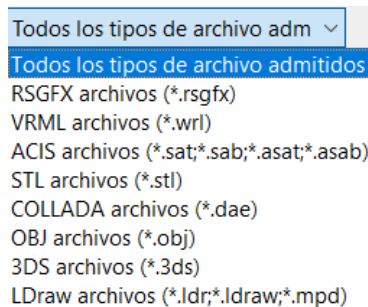


Figura 3.23 - Tipos de archivos CAD que permite importar RobotStudio

6. Cuando se abre la geometría deseada esta se muestra en la **Ventana gráfica** y en el navegador **Diseño**. La geometría se ubica en la estación haciendo coincidir su origen local con el sistema de coordenadas mundo de la estación. En apartados posteriores se explicará cómo posicionar correctamente todos los componentes que forman la estación. (Figura 3.24)

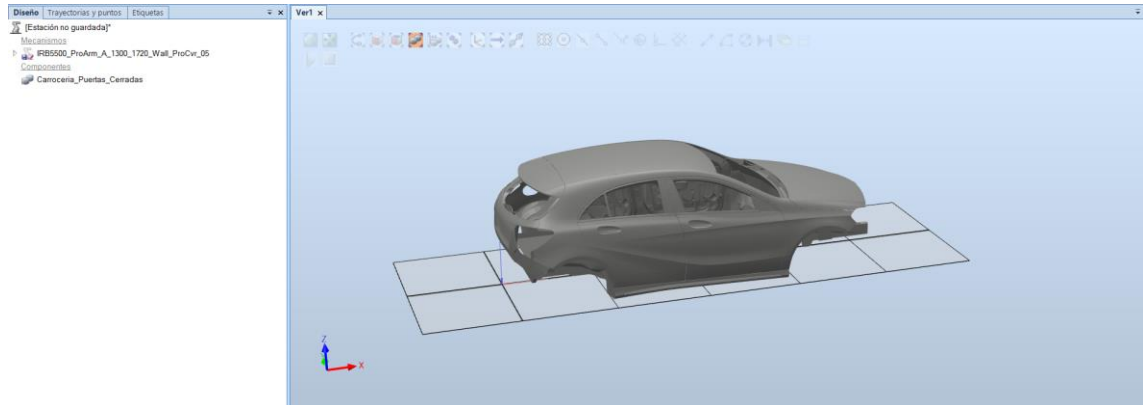


Figura 3.24 - Ejemplo de importación de geometría CAD

La otra opción de importación de componentes se emplea para importar componentes creados en RobotStudio o importados a RobotStudio y que se han guardado anteriormente como biblioteca. Los archivos de biblioteca de RobotStudio tienen la extensión .rslib. Para utilizar esta opción se tienen que seguir los siguientes pasos:

1. Se accede a la pestaña **Posición inicial** seleccionándola con el cursor del ratón.
2. En el grupo **Construir estación**, se selecciona la opción **Importar biblioteca**.
3. Aparece el desplegable mostrado en la Figura 3.25. En él se puede elegir entre la opción de **Biblioteca del usuario**, en el caso de que se hayan guardado geometrías como biblioteca de RobotStudio previamente, la opción de **Equipamiento**, mediante la cual RobotStudio ofrece diferentes geometrías de equipamiento y herramientas, la opción de **Documentos** y la opción de **Ubicaciones**, ya explicadas en el caso anterior, y la opción **Buscar biblioteca**, mediante la cual se abre el explorador de archivos de Windows para buscar la geometría de biblioteca a importar. Este última es la opción más empleada por su versatilidad.

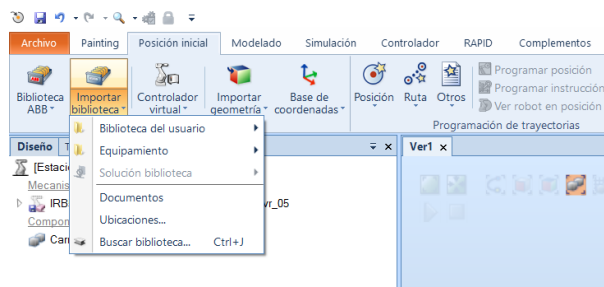


Figura 3.25 - Opciones para importar un archivo de biblioteca de RobotStudio

4. Cuando se importa la biblioteca deseada esta se muestra en la **Ventana gráfica** y en el navegador **Diseño**. Como en el caso anterior, la geometría se ubica en la estación haciendo coincidir su origen local con el sistema de coordenadas mundo de la estación.
5. La geometría importada queda conectada a la biblioteca. Si se quiere editar la geometría primero deberá desconectarse de la biblioteca.

3.1.6 Creación/importación de tracks y mecanismos

La opción más sencilla y práctica es la de importar uno de los tracks o uno de los posicionadores ABB que proporciona RobotStudio prediseñados. Las principales ventajas de esta opción es que estos componentes ya están diseñados y programados para tener un funcionamiento óptimo y encajan a la perfección con los manipuladores robóticos de ABB. Además existen diversas opciones de personalización para cada uno de los mismos. Para llevar a cabo esto se deben seguir los siguientes pasos:

1. Se accede a la pestaña **Posición inicial** seleccionándola con el cursor del ratón.
2. En el grupo **Construir estación**, se selecciona la opción **Biblioteca ABB**.
3. Aparece un desplegable con todos los modelos de robot, los posicionadores y tracks disponibles. Se selecciona el track o posicionador requerido para llevar a cabo la tarea. (Figura 3.26)



Figura 3.26 - Modelos de robots, posicionadores y tracks disponibles para su importación

4. Cuando se selecciona el track o posicionador aparece el cuadro de dialogo del modelo de elegido. En este cuadro de dialogo se especifica la versión del mismo y otras

características que dependen del modelo seleccionado. En la siguiente Figura 3.27 se expone el cuadro de dialogo para el track IRB 5500 Elevated Rail.

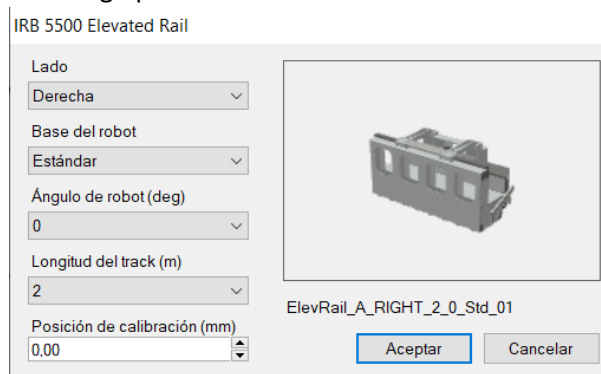


Figura 3.27 - Opciones de configuración para el modelo de track IRB 5500 Elevated Rail

- Una vez seleccionas las opciones, se pulse **Aceptar** y el track se añade a la estación, mostrándose en la **Ventana gráfica** y en el navegador **Diseño**. El track se ubica en la estación haciendo coincidir su origen local con el sistema de coordenadas mundo de la estación. (Figura 3.28)

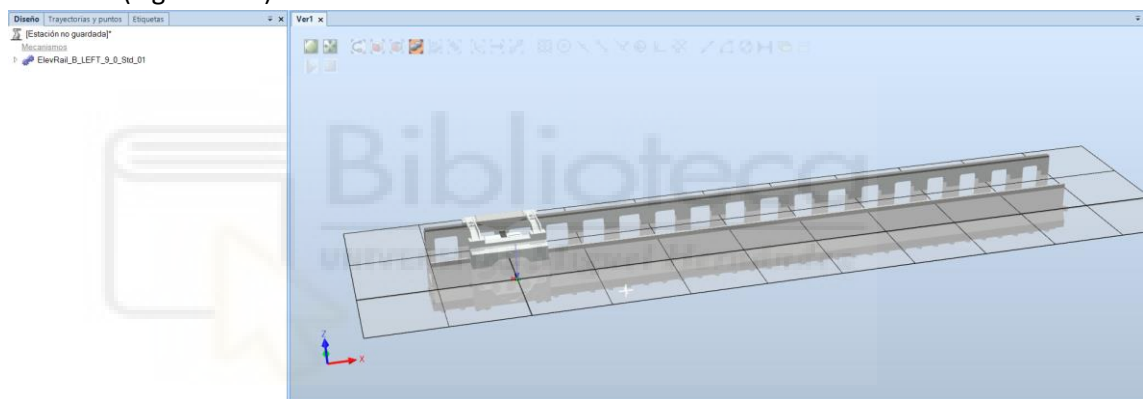


Figura 3.28 - Ejemplo de importación de un modelo de track

Si se tiene un mecanismo de cualquier tipo almacenado como biblioteca de RobotStudio, este puede importarse a la estación como se ha explicado en el apartado anterior [3.1.5 Importación de robots y otros componentes de la estación.](#)

Por otro lado, RobotStudio permite la creación de mecanismos propios para simular objetos externos como, por ejemplo, mesas giratorias, pinzas, posicionadores, etc.

Existen cuatro tipos diferentes de mecanismo que pueden ser creados:

- **Mecanismo de robot.** Este es un mecanismo que posee TCP y se controla mediante un controlador virtual.
- **Dispositivo.** Este mecanismo no posee TCP.
- **Ejes externos.** Estos mecanismos no tienen TCP pero si pueden ser controlador mediante un controlador virtual, como por ejemplo un track o un posicionador.
- **Herramientas.** Estos mecanismos actúan como efector final y por lo tanto poseen TCP y son movidos por el manipulador de robot.

Una vez se han creado o importado a la estación los componentes que formarán el mecanismo, deben seguirse los siguientes pasos para crearlo:

1. Se accede a la pestaña de **Modelado** seleccionándola con el cursor del ratón.
2. En el grupo **Mecanismo**, se selecciona la opción **Crear mecanismo**. Aparece el panel **Crear Mecanismo** mostrado en la Figura 3.29.

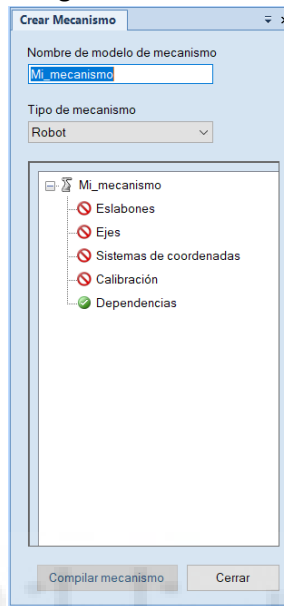


Figura 3.29 - Panel Crear mecanismo

3. Primero se debe especificar el nombre que se le quiere dar al mecanismo.
4. En el apartado **Tipo de mecanismo** se debe seleccionar uno de los cuatro tipos de mecanismo explicados anteriormente.
5. En la estructura de árbol que aparece a continuación se selecciona el apartado **Eslabones**, se hace clic con el botón derecho del ratón y se selecciona **Añadir Eslabón**. Este paso es común para todos los tipos de mecanismos.
6. En el cuadro de dialogo que aparece **Crear Eslabón** (Figura 3.30) primero debe especificarse el nombre que se le quiere dar a ese eslabón y después en el apartado **Componente seleccionado** se debe seleccionar el componente de la estación que actuará como ese eslabón. Si el eslabón seleccionado es la base del mecanismo se debe seleccionar la casilla **Establecer como eslabón base**. Una vez hecho esto se pulsa en la flecha verde que aparece en el cuadro al lado de los parámetros para añadir el componente al cuadro de la lista de componentes y se pulsa en **Aplicar**. Se repite esto para cada uno de los eslabones que compongan el mecanismo.
Para facilitar la comprensión de este proceso se crea un mecanismo sencillo de tipo dispositivo mostrado en la Figura 3.31. Este se compone de una pared, que actúa como eslabón base, y una trampilla que se moverá verticalmente para abrirse.

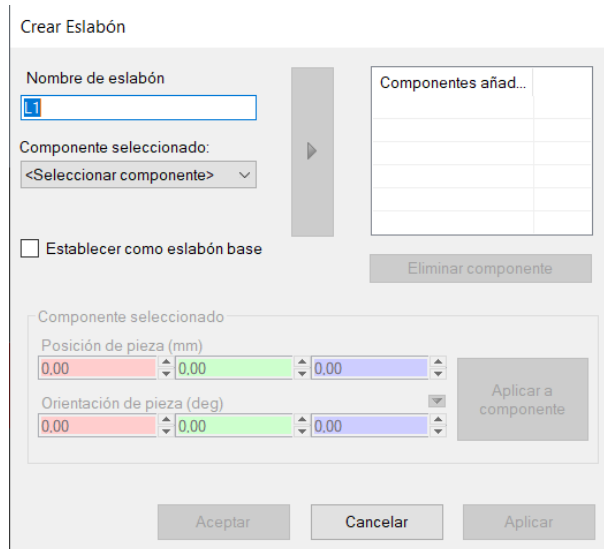


Figura 3.30 - Cuadro de diálogo Crear Eslabón

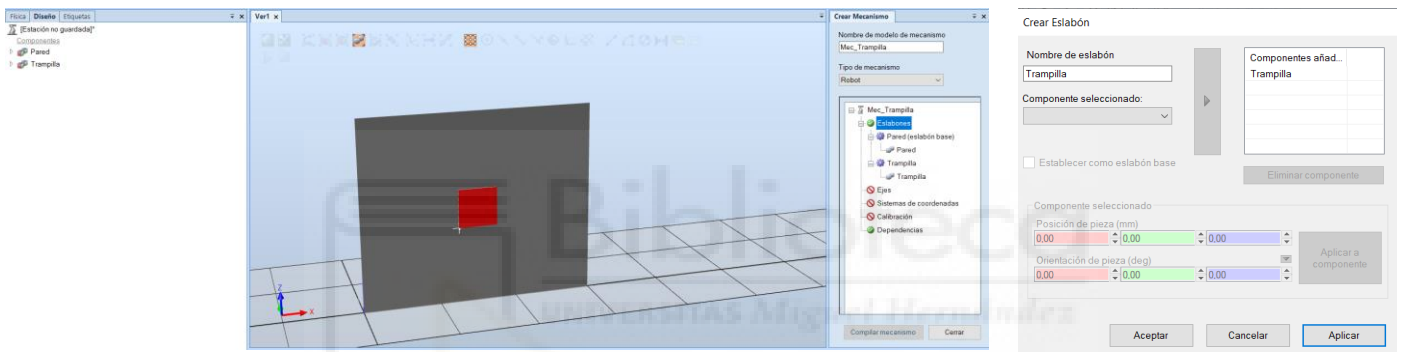


Figura 3.31 - Mecanismo sencillo de ejemplo

- Una vez se tienen todos los eslabones se procede a crear los ejes del mecanismo. Este paso es común para todos los tipos de mecanismos. Se hace clic con el botón derecho del ratón en el apartado **Ejes** y se selecciona **Añadir Eje**. En el cuadro de diálogo que aparece se especifica el nombre del eje, el tipo de eje (de rotación o prismático), el eslabón principal (por defecto el eslabón base) y el eslabón secundario. Una vez se tiene esto también debe indicarse posición del eje mediante dos posiciones espaciales y la dirección en X, Y, Z del mismo. Para finalizar se puede añadir un límite constante o variable al eje, especificando también su límite inferior y superior. Cuando se tiene todo se clicca en **Aplicar**. La Figura 3.32 muestra la configuración para el ejemplo simple anteriormente expuesto.

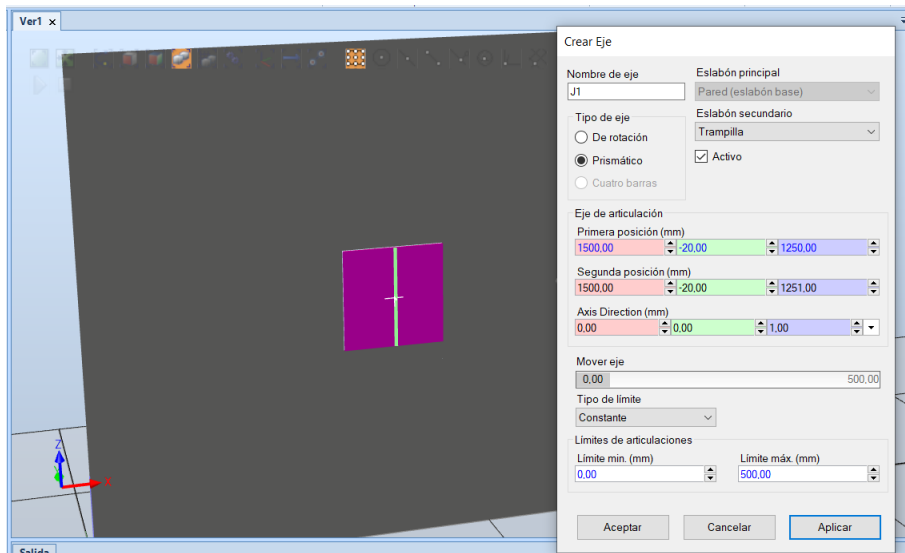


Figura 3.32 - Configuración de ejes para el mecanismo de ejemplo

En el caso de este sencillo mecanismo de tipo dispositivo ya se tendría definido por completo.

- En el caso de los mecanismos de tipo robot y eje externo el siguiente paso sería añadir los sistemas de coordenadas. Para esto se clicca con el botón derecho del ratón en el apartado **Sistemas de coordenadas** y se selecciona **Añadir sistema de coordenadas**. Se completan los diferentes campos que aparecen en la Figura 3.33 y se pulsa aceptar.

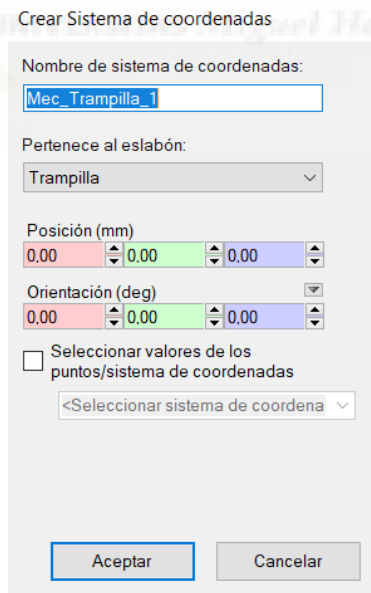


Figura 3.33 - Cuadro de diálogo Crear sistema de coordenadas

- En el caso de los mecanismos de tipo herramienta el siguiente paso sería añadir los datos de la herramienta. Para esto se clicca con el botón derecho del ratón en el apartado **Datos de herramienta** y se selecciona **Añadir datos de herramienta**. Se completan los diferentes campos que aparecen en la Figura 3.34 y se pulsa aceptar.

Crear Datos de herramienta

Nombre de datos de herramienta:
Mec. Trampilla

Pertenece al eslabón:
Trampilla

Posición (mm)
0,00 0,00 0,00

Orientación (deg)
0,00 0,00 0,00

Seleccionar valores de los puntos/sistema de coordenadas
<Seleccionar sistema de coordena >

Datos de herramienta

Masa (Kg)
1,00

Centro de gravedad (mm)
0,00 0,00 0,00

Momento de inercia bx. ly. lz (kgm²)
0,00 0,00 0,00

Aceptar Cancelar

Figura 3.34 - Cuadro de dialogo Crear datos de herramienta

Para obtener los movimientos correctos cuando el programa se ejecute, los parámetros de la herramienta deben estar especificados en los datos de la herramienta. El dato más importante de una herramienta es el TCP, es decir, la posición del punto central de la herramienta respecto de la muñeca del robot.

Con este paso se completaría la definición de los mecanismos de tipo herramienta.

- Para terminar la definición de los mecanismos de tipo robot y eje externo el siguiente paso sería añadir la calibración. Para esto se clic con el botón derecho del ratón en el apartado **Calibración** y se selecciona **Añadir calibración**. Se completan los diferentes campos que aparecen en la Figura 3.35 y se pulsa aceptar.

Crear Calibración

Calibración perteneciente al eje:
<Seleccionar eje>

Posición (mm)
0,00 0,00 0,00

Orientación (deg)
0,00 0,00 0,00

Aceptar Cancelar

Figura 3.35 - Cuadro de dialogo Crear calibración

- Una vez completados todos los apartados que definen el mecanismo se pulsa en **Compilar mecanismo** para crearlo.
- El mecanismo se añade al navegador **Diseño** y ahora es posible crear y editar poses para el mecanismo y en el caso de que se tenga más de un eje puede editarse la correlación entre los mismos. En la Figura 3.36 se muestra el resultado del sencillo mecanismo creado para ilustrar estos pasos.

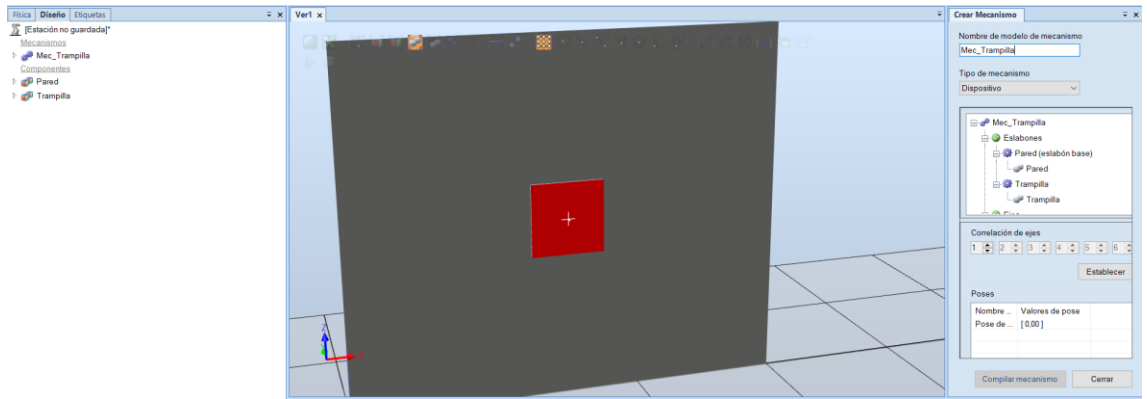


Figura 3.36 - Ejemplo del mecanismo sencillo creado

3.1.7 Importación del efector final del robot

Como se ha visto en el apartado anterior puede crearse una herramienta de efector final de robot mediante la opción de **Crear Mecanismo**. También existe la opción de crear herramientas sencillas mediante la opción **Crear herramienta**, del grupo **Mecanismo** de la pestaña **Modelado**, pero esta opción no permite crear herramientas con movimiento o complejas.

Por otro lado, RobotStudio nos ofrece una serie de herramientas de ABB prediseñadas que pueden ser importadas.

Para esto deben seguirse los siguientes pasos:

1. Se accede a la pestaña **Posición inicial** seleccionándola con el cursor del ratón.
2. En el grupo **Construir estación**, se selecciona la opción **Importar biblioteca**. Aparece el desplegable mostrado en la Figura 3.37. En él se selecciona la opción **Equipamiento**. Aparece así un desplegable con los diferentes componentes y herramientas prediseñadas que ofrece RobotStudio. Se selecciona la herramienta requerida para llevar a cabo la tarea.

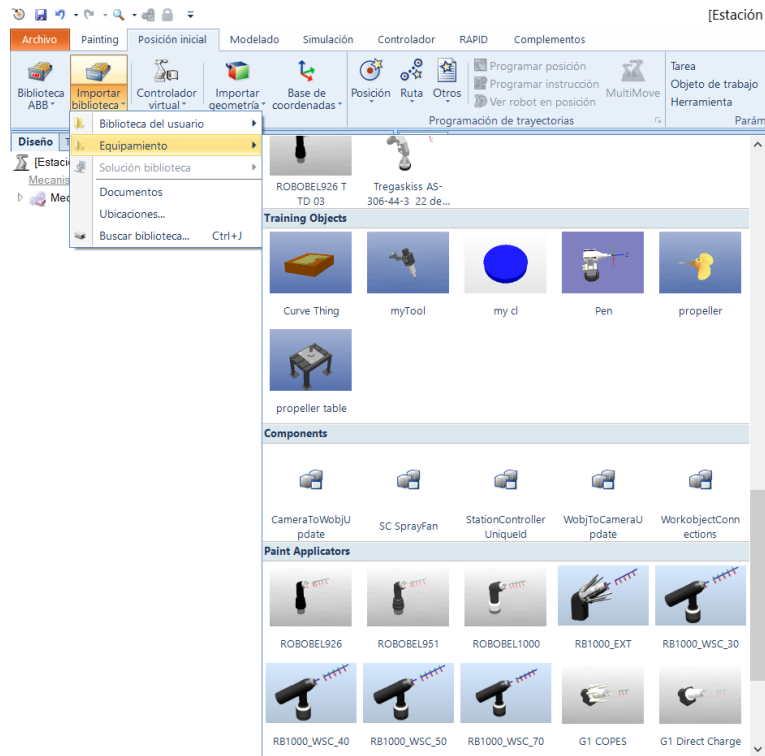


Figura 3.37 - Herramientas disponibles para su importación

3. Cuando se selecciona la herramienta aparece el cuadro de dialogo del modelo de herramienta elegido. En este cuadro de dialogo se especifican las características de la herramienta. (Figura 3.38)

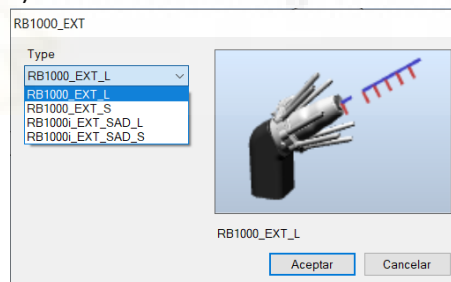


Figura 3.38 - Cuadro de dialogo para el modelo de herramienta RB 1000 EXT

4. Una vez especificadas las opciones, se pulsa **Aceptar** y la herramienta se añade a la estación, mostrándose en la **Ventana gráfica** y en el navegador **Diseño**. La herramienta se ubica en la estación haciendo coincidir su origen local con el sistema de coordenadas mundo de la estación. (Figura 3.39)

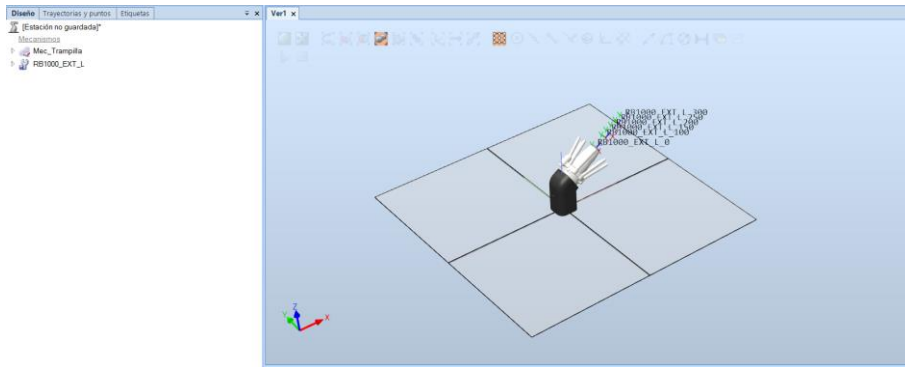


Figura 3.39 - Ejemplo de importación de una herramienta

Una vez se tiene la herramienta en la estación, esta se debe conectar al manipulador de robot para que este la mueva y se tenga en cuenta su TCP para realizar los movimientos y acciones que se programarán.

Para conectarla existen dos opciones:

En la primera, en el navegador **Diseño** se hace clic con el botón derecho en la herramienta deseada, se selecciona **Conectar a** y se selecciona el robot al que se desea conectar. (Figura 3.40)

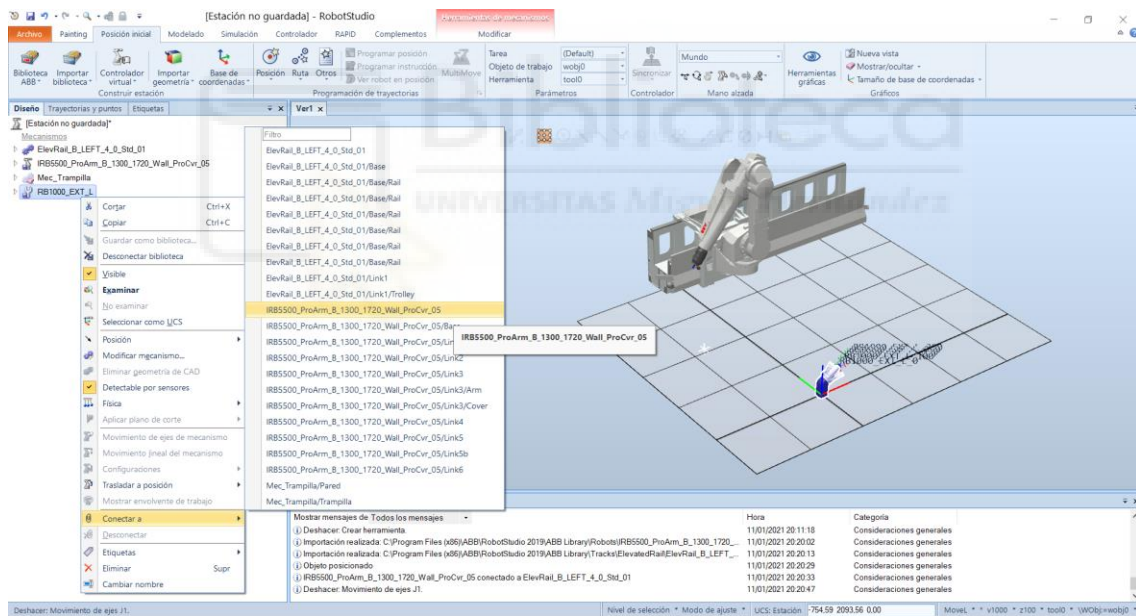


Figura 3.40 - Primera opción de conexión de la herramienta al robot

En la segunda opción, la herramienta se selecciona en el navegador **Diseño** y manteniendo pulsado el botón izquierdo del ratón en la herramienta, se arrastra hasta el nombre del robot al que se quiere conectar y una vez encima se suelta.

En ambos casos, en la ventana de diálogo que aparece **Actualizar posición** se debe hacer clic en **Sí**. De esta forma la herramienta se conecta a la brida del robot y el TCP de la misma se tiene en cuenta. (Figura 3.41)

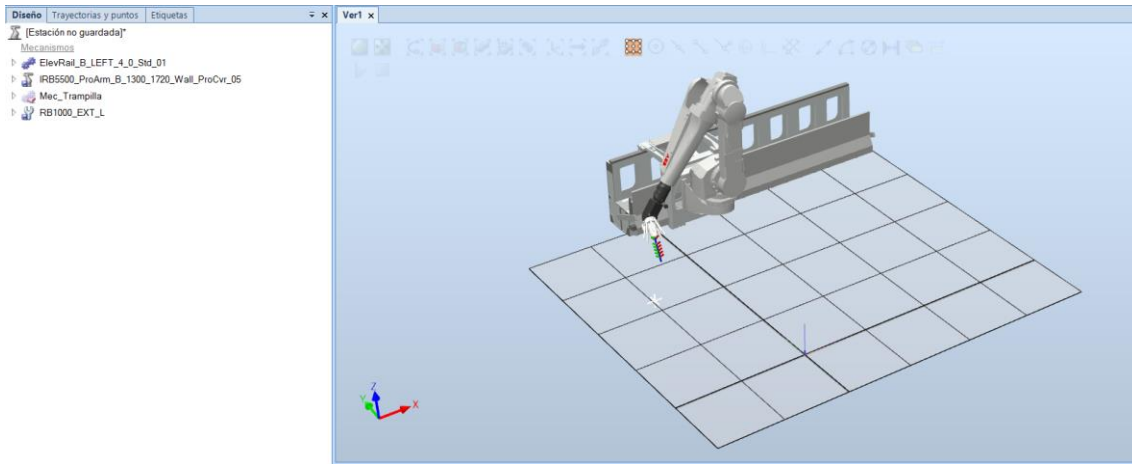


Figura 3.41 - Ejemplo de conexión de herramienta a robot

3.1.8 Modelado de componentes en RobotStudio

RobotStudio ofrece un gran número de herramientas para crear diferentes tipos de componentes gráficos relativamente sencillos. Todas estas herramientas se encuentran en la pestaña **Modelado** en el grupo **Crear** (Figura 3.42). A continuación se expondrán brevemente estas herramientas de creación de componentes.

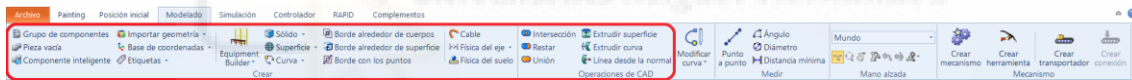


Figura 3.42 - Grupo Crear de la pestaña Modelado

Creación de un grupo de componentes

Se denomina **Grupo de componentes** a una geometría que está formada por varios componentes, también denominados piezas (Figura 3.43). Mediante la creación de un grupo de componentes vacío se pueden agrupar dentro de este distintos componentes para formar una única geometría más compleja, con un único origen local, de forma que pueden moverse y modificarse todos los componentes como un único bloque. Para añadir un componente a un grupo de componentes simplemente se debe seleccionar con el botón izquierdo del ratón y arrastrarlo encima del nombre del grupo. Si se desea extraer un componente de un grupo, se debe seleccionar con el botón izquierdo del ratón y arrastrarlo hasta el nombre de la estación en el navegador **Diseño**.

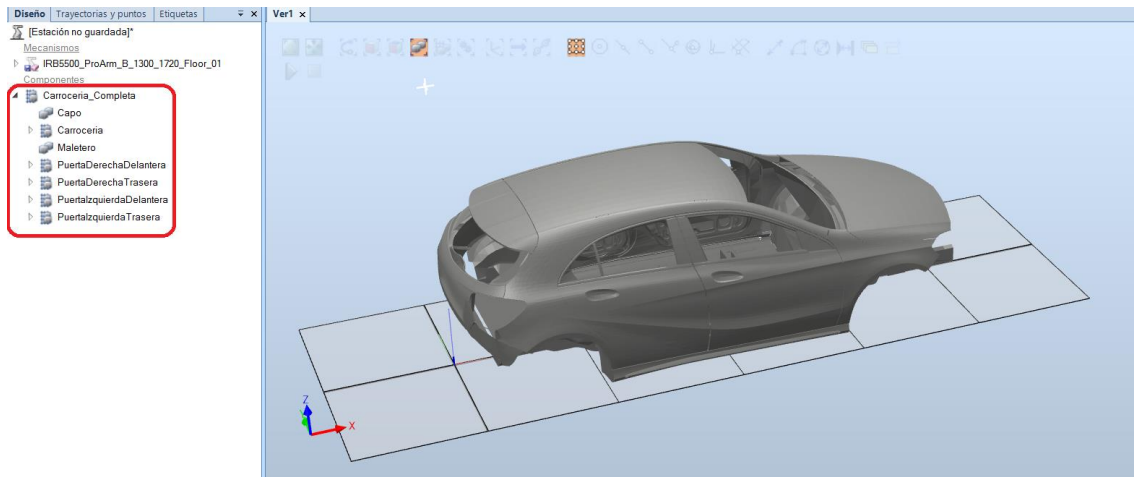


Figura 3.43 - Ejemplo de un grupo de componentes

Creación de una pieza vacía

Esta herramienta crea una pieza sin representación gráfica en el navegador **Diseño**. Una pieza es cualquier objeto físico de RobotStudio. Las piezas contienen cuerpos como sólidos, superficies o curvas.

Una pieza vacía actúa del mismo modo que el grupo de componentes pero en este caso con cuerpos. Los cuerpos creados o los cuerpos de otras piezas pueden agruparse dentro de una pieza vacía. Para ello deben seleccionarse con el botón izquierdo del ratón y arrastrarlos encima del nombre de la pieza vacía. De esta forma se compone una nueva pieza con cuerpos procedentes de distintos lugares, como se muestra en la Figura 3.44.

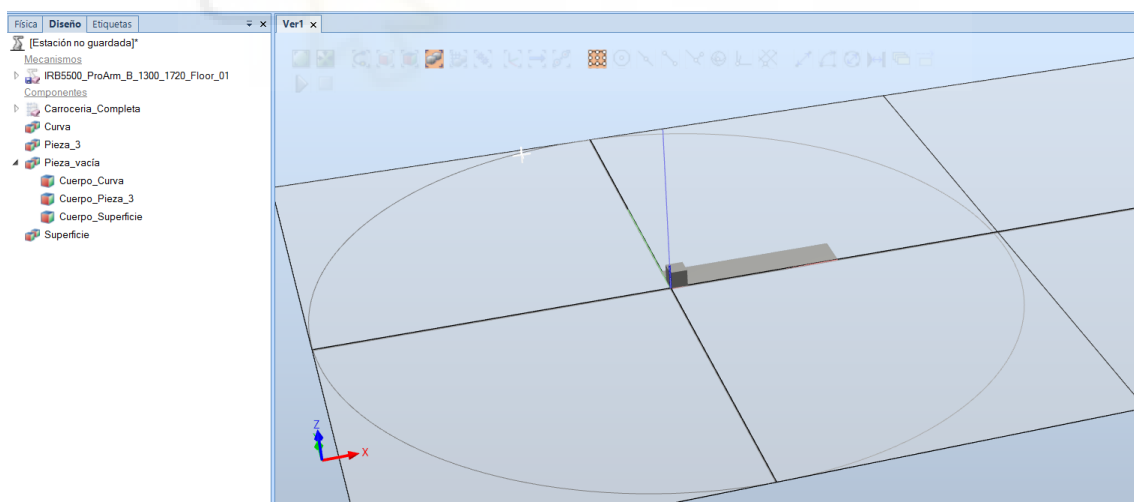


Figura 3.44 - Ejemplo de agrupación de cuerpos en una pieza inicialmente vacía

Creación de un sólido

Mediante esta herramienta se pueden crear diferentes cuerpos primitivos de tipo sólido. Un sólido primitivo es un conjunto de formas tridimensionales básicas que forman un cuerpo tridimensional sencillo. Mediante la combinación de sólidos primitivos se pueden obtener cuerpos más complejos.

Los sólidos primitivos que pueden crearse son los mostrados en la siguiente Figura 3.45.

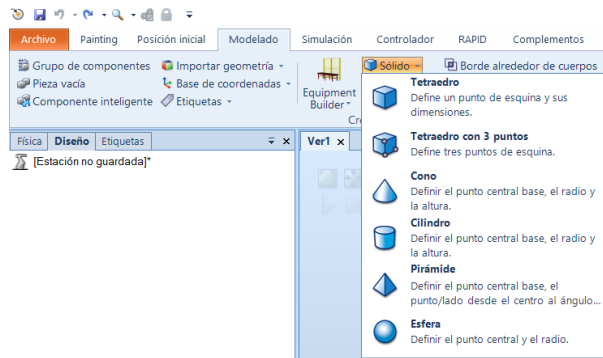


Figura 3.45 - Opciones de creación de sólidos primitivos

Al seleccionar el sólido deseado en este desplegable se abre el panel **Crear (Tipo de sólido)**, en el cual deben especificarse básicamente el punto de origen de la geometría y sus dimensiones. En la Figura 3.46 se muestran algunos ejemplos de este panel.

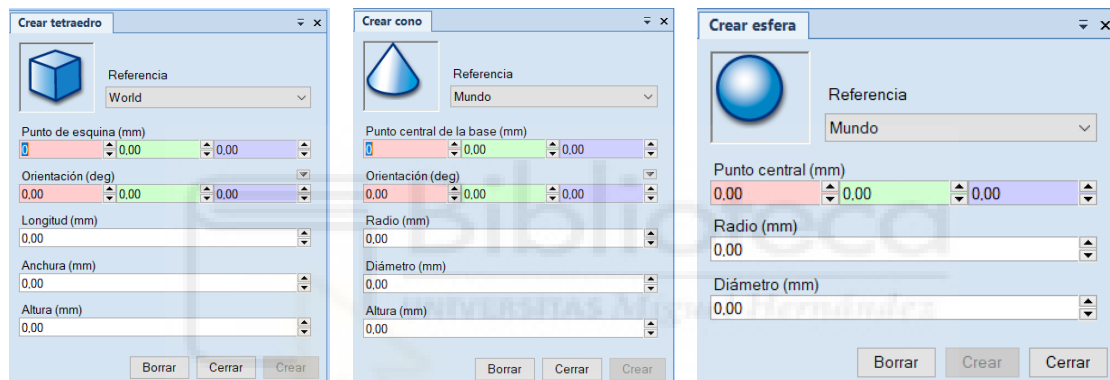


Figura 3.46 - Ejemplos del panel Crear para diferentes sólidos primitivos

Una vez especificadas las características del sólido, se pulsa en **Crear** y una pieza que contiene el sólido se añade a la estación, mostrándose en la **Ventana gráfica** y en el navegador **Diseño**. El origen local, es decir el sistema de coordenadas del objeto, siempre estará situado coincidiendo con el sistema de coordenadas mundo de la estación. Posteriormente podrá cambiarse de posición, para que por ejemplo coincida con el centro de una cara del objeto.

Creación de una superficie

Mediante esta herramienta se pueden crear diferentes cuerpos de tipo superficie. Las superficies que pueden crearse son las mostradas en la siguiente Figura 3.47.

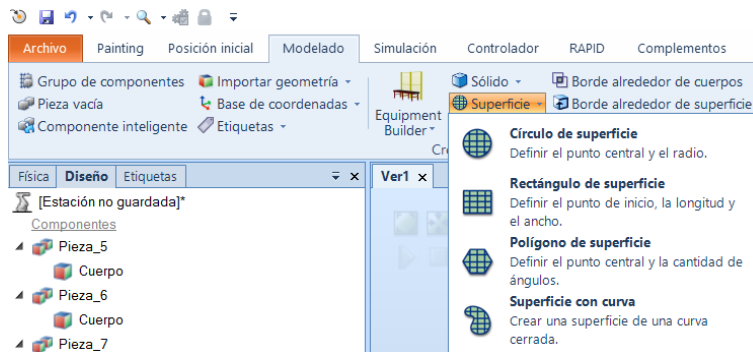


Figura 3.47 - Opciones de creación de superficies

Al seleccionar la superficie deseada en este desplegable se abre el panel **Crear (Tipo de superficie)**, en el cual deben especificarse básicamente el punto de origen de la superficie y sus dimensiones. En la Figura 3.48 se muestran algunos ejemplos de este panel.

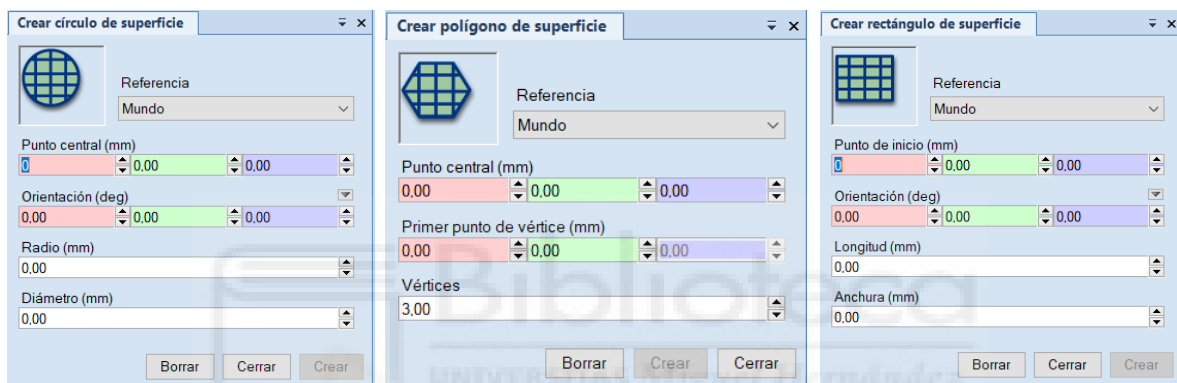


Figura 3.48 - Ejemplos del panel Crear para diferentes superficies

Una vez especificadas las características de la superficie, se pulsa en **Crear** y una pieza que contiene la superficie se añade a la estación, mostrándose en la **Ventana gráfica** y en el navegador **Diseño**. El origen local, es decir el sistema de coordenadas de la superficie, siempre estará situado coincidiendo con el sistema de coordenadas mundo de la estación. Posteriormente podrá cambiarse de posición, para que por ejemplo coincida con un vértice de la superficie.

Creación de una curva

Mediante esta herramienta se pueden crear diferentes cuerpos de tipo curva. Las curvas que pueden crearse son las mostradas en la siguiente Figura 3.49.

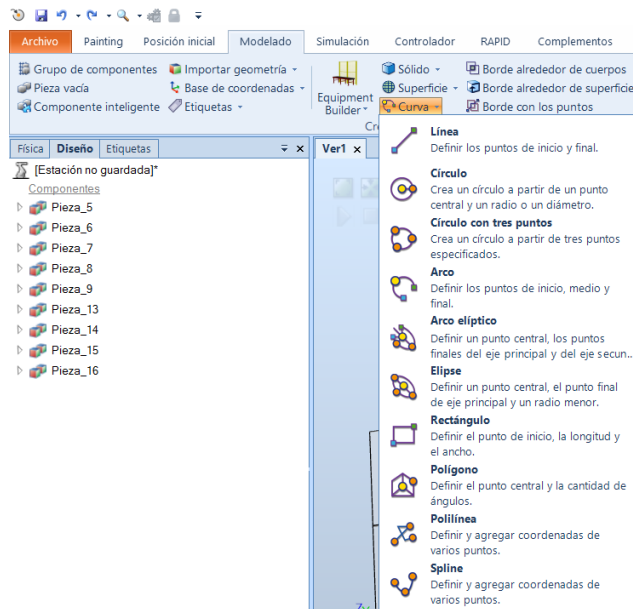


Figura 3.49 - Opciones de creación de curvas

Al seleccionar la curva deseada en este desplegable se abre el panel **Crear (Tipo de curva)**, en el cual deben especificarse básicamente el punto de origen de la curva y sus dimensiones. En la Figura 3.50 se muestran algunos ejemplos de este panel.



Figura 3.50 - Ejemplos del panel Crear para diferentes curvas

Una vez especificadas las características de la curva, se pulsa en **Crear** y una pieza que contiene la curva se añade a la estación, mostrándose en la **Ventana gráfica** y en el navegador **Diseño**. El origen local, es decir el sistema de coordenadas de la curva, siempre estará situado coincidiendo con el sistema de coordenadas mundo de la estación. Posteriormente podrá cambiarse de posición, para que por ejemplo coincida con un vértice o el centro de la curva.

Creación de un borde

Mediante las tres herramientas disponibles para la creación de un borde se pueden crear cuerpos de tipo curva alrededor de un sólido o una superficie, especificando puntos en un sólido o generando el borde que forma la intersección de dos sólidos (Figura 3.51).

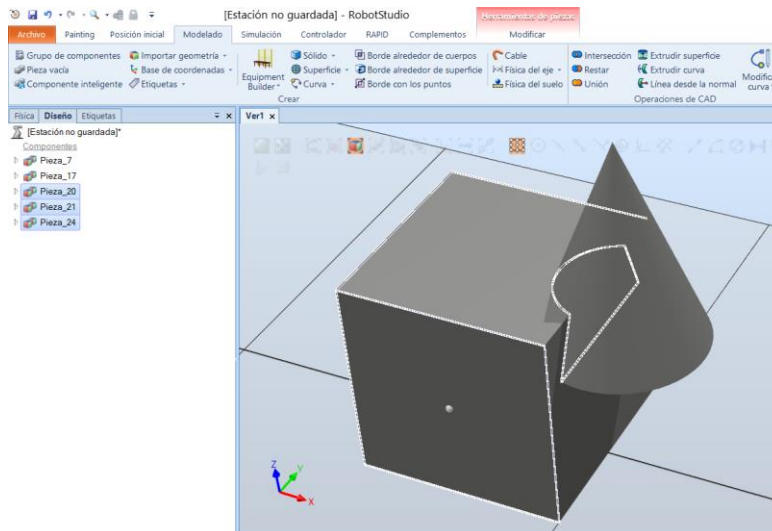


Figura 3.51 - Ejemplo de los diferentes tipos de borde que pueden crearse

Para ello simplemente se selecciona la herramienta deseada y en el panel **Crear (Tipo de borde)** se especifican los sólidos, superficies o puntos deseados. Una vez especificadas las opciones del borde, se pulsa en **Crear** y una pieza que contiene la curva de borde se añade a la estación, mostrándose en la **Ventana gráfica** y en el navegador **Diseño**. Si el borde es alrededor de una superficie o mediante puntos en un sólido, el origen local de la pieza de borde coincide con el origen del sólido o superficie. Si se trata de un borde generado mediante la intersección de dos sólidos, el origen local de la pieza de borde coincidirá con el sistema de coordenadas mundo de la estación. Posteriormente podrá cambiarse de posición, para que por ejemplo coincida con un vértice de la curva.

Creación de una articulación

Mediante la herramienta **Física del eje** pueden crearse diferentes tipos de articulaciones o ejes. Un eje especifica el tipo de conexión entre un cuerpo rígido y su entorno o entre dos cuerpos rígidos entre sí, estando estos bajo condiciones específicas.

Los ejes o articulaciones que pueden crearse son los mostrados en la siguiente Figura 3.52.

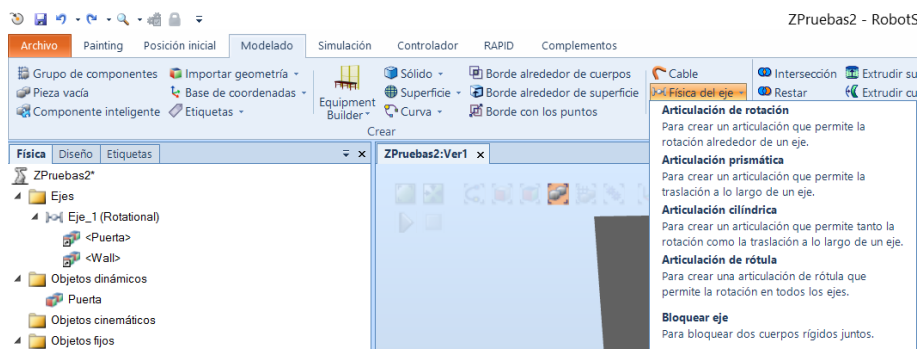


Figura 3.52 - Opciones para la creación de ejes o articulaciones

Al seleccionar la articulación deseada en este desplegable se abre el panel (**Tipo de articulación**), en el cual deben especificarse los cuerpos conectados mediante el eje, la posición del eje y la dirección del mismo. En la Figura 3.53 se muestran algunos ejemplos de este panel.

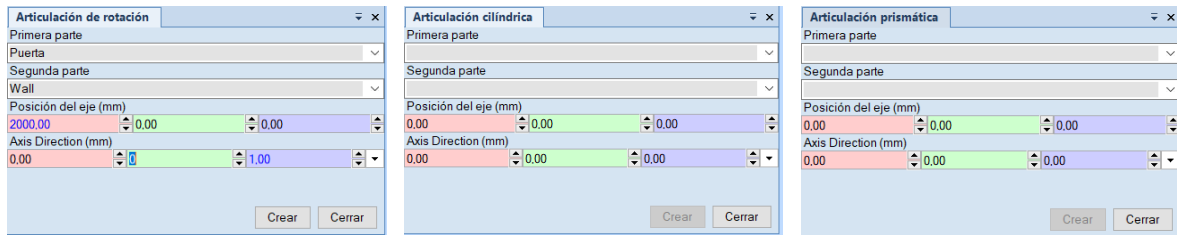


Figura 3.53 - Ejemplos del panel de configuración para la creación de distintas articulaciones

Una vez especificadas las características de la articulación, se pulsa en **Crear**, de forma que el eje se añade a la estación, mostrándose en el navegador **Física**. En la siguiente Figura 3.54 se muestra un ejemplo de una articulación de rotación entre el cuerpo sólido **Puerta** y otro cuerpo sólido **Wall**. Esta articulación permite la rotación alrededor del Eje_1 del cuerpo **Puerta** respecto del cuerpo **Wall**. Para ello **Wall** tiene que tener un comportamiento físico **Anclado** y **Puerta** tiene que tener un comportamiento físico **Dinámico**.

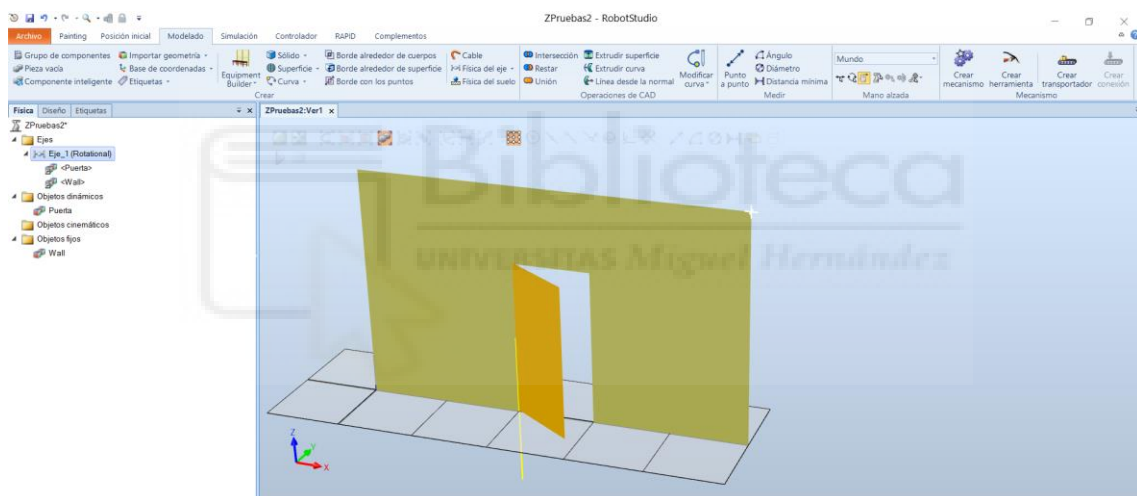


Figura 3.54 - Ejemplo de articulación de rotación

Creación de un sólido combinando otros dos

Mediante las herramientas de **Intersección**, **Restar** y **Unión** pueden crearse cuerpos sólidos más complejos a partir de dos cuerpos sólidos sencillos. Al seleccionar una de las herramientas aparece el panel (**Intersección**, **Restar** o **Unión**) en que se deben especificar los cuerpos sólidos, que no piezas, mediante los cuales crear el nuevo cuerpo sólido. (Figura 3.55)

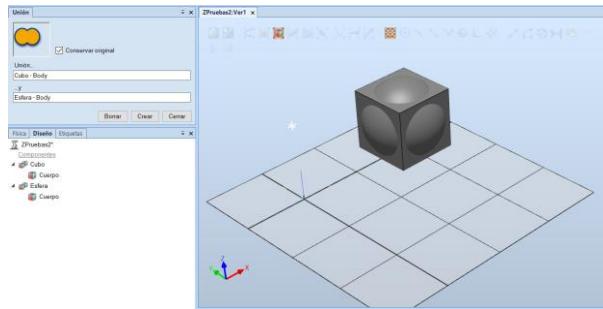


Figura 3.55 - Ejemplo del panel de la herramienta Unión

La herramienta **Intersección** crea un cuerpo sólido que consiste en la intersección de los dos cuerpos indicados, es decir, el volumen que comparte ambos cuerpos. En esta instrucción no es relevante el orden en el que se indican los cuerpos, se obtendrá el mismo resultado. (Figura 3.56)

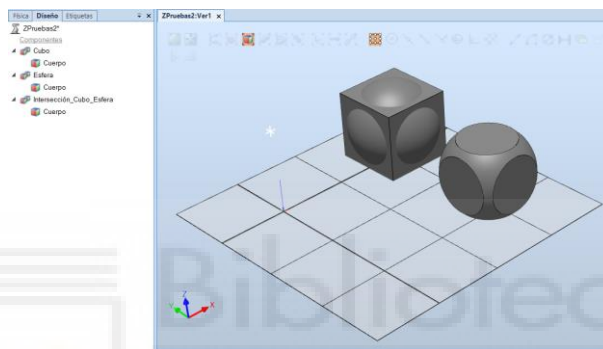


Figura 3.56 - Resultado de la intersección de los cuerpos Cubo y Esfera

La herramienta **Restar** crea un cuerpo sólido que consiste en la resta de los dos cuerpos indicados, es decir, elimina en un cuerpo el volumen del otro cuerpo. Por ese motivo en esta herramienta sí que es relevante el orden en el que se indican los cuerpos, ya que al cuerpo que se indica primero se le elimina el volumen del segundo. (Figura 3.57)

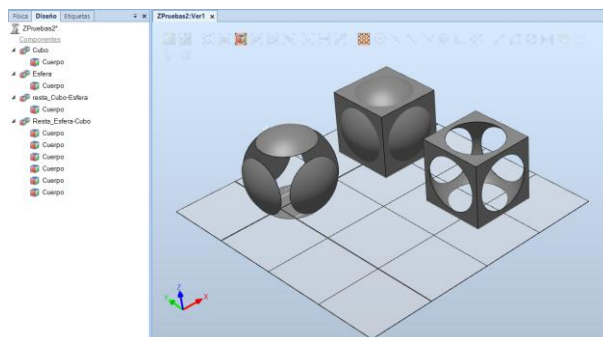


Figura 3.57 - Resultado de la resta de los cuerpos Cubo y Esfera

La herramienta **Unión** crea un cuerpo sólido que consiste en la unión de los cuerpos indicados, es decir, integra los volúmenes de los cuerpos para formar uno solo. En esta instrucción no es relevante el orden en el que se indican los cuerpos, se obtendrá el mismo resultado. (Figura 3.58)

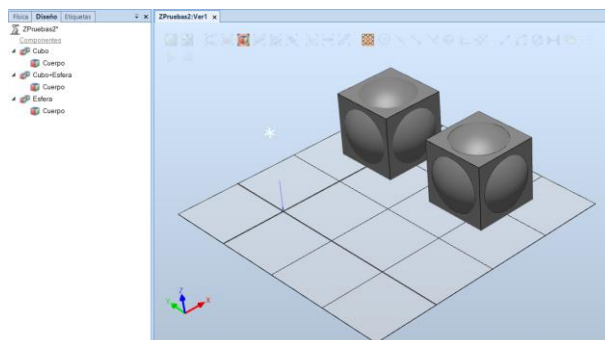


Figura 3.58 - Resultado de la unión de los cuerpos Cubo y Esfera

Creación de un sólido a partir de superficies o curvas

Mediante estas dos herramientas se pueden crear cuerpos sólidos o superficies a partir de superficies o curvas, teniéndose la posibilidad de obtener los primeros mediante superficies o si la curva es cerrada y obteniéndose superficies mediante curvas abiertas o cerradas.

Para ello el primer paso es crear la superficie o la curva de contorno del sólido o superficie, como se ha explicado anteriormente (Figura 3.59).

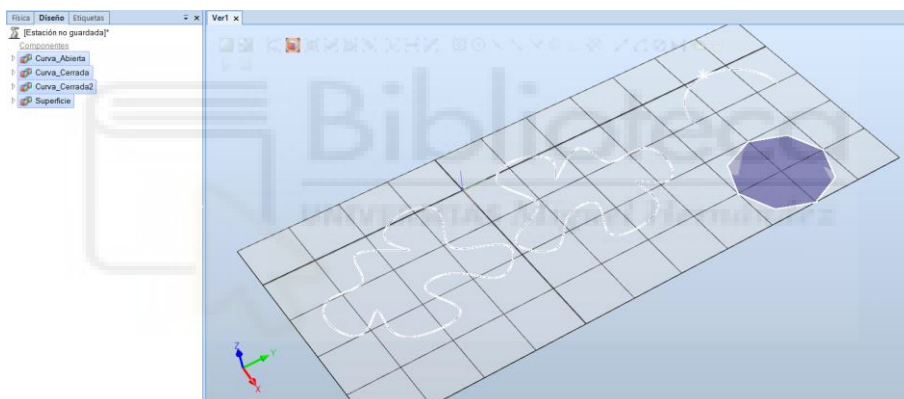


Figura 3.59 - Ejemplo de diferentes tipos de curvas/superficies

Una vez se tiene la superficie o curva, al seleccionar la herramienta **Extrudir superficie** se abre el panel **Extrudir superficie o curva** (Figura 3.60), en el cual se debe seleccionar la superficie o curva a extrudir, si se quiere extrudir siguiendo una recta o una curva y si se quiere crear un sólido, en el caso que sea posible, o una superficie.

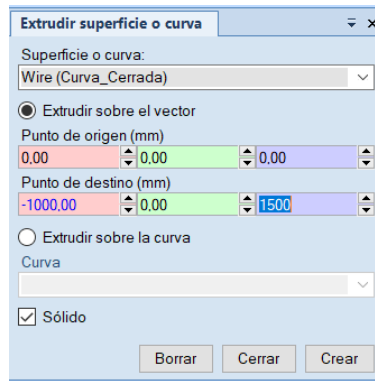


Figura 3.60 - Panel extrudir superficie o curva

Una vez especificadas las opciones, se pulsa en **Crear** y una pieza que contiene la superficie o el sólido se añade a la estación, mostrándose en la **Ventana gráfica** y en el navegador **Diseño**. El origen local, es decir el sistema de coordenadas del sólido o superficie, siempre estará situado coincidiendo con el sistema de coordenadas mundo de la estación. Posteriormente podrá cambiarse de posición. En la siguiente Figura 3.61 se observa un ejemplo del uso de estas herramientas.

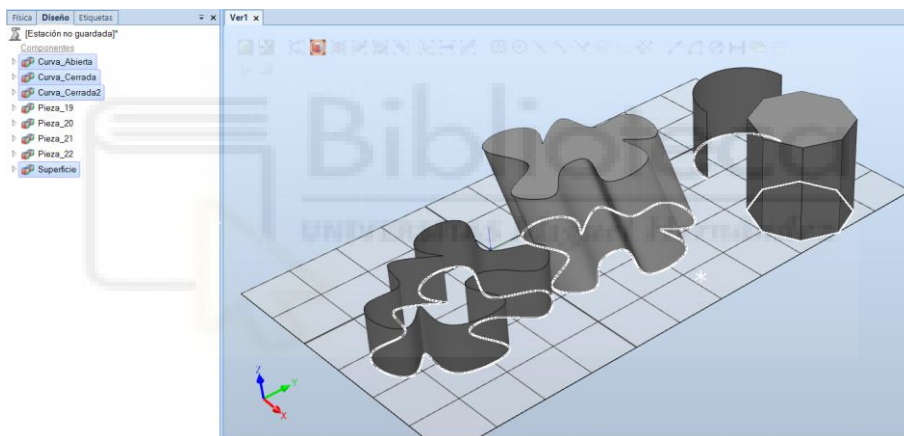


Figura 3.61 - Ejemplos de sólidos o superficies creados mediante la herramienta Extrudir superficie

3.1.9 Modificación de componentes en RobotStudio

Una vez se han creado o importado los componentes que forman la estación estos pueden ser editados en diferentes aspectos para conseguir el resultado requerido.

Los primeros aspectos que pueden modificarse de un componente son los que hacen referencia a su apariencia y su geometría. Para acceder al grupo de herramientas que permiten estas modificaciones se debe seleccionar el componente en cuestión mediante el botón derecho del ratón y seleccionar la opción **Modificar**, de forma que aparece un desplegable con las diferentes herramientas, como se muestra en la Figura 3.62.

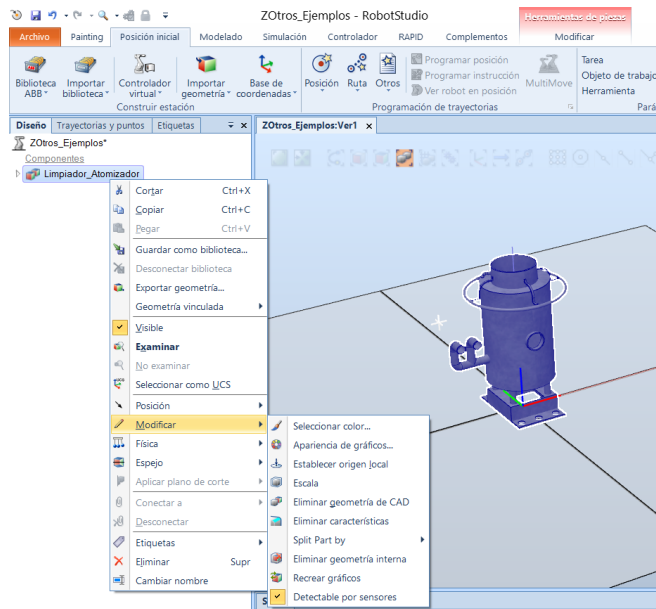


Figura 3.62 - Opciones de la herramienta modificar

También puede accederse a estas herramientas mediante la pestaña **Modificar**, que aparece al seleccionar el componente con el botón izquierdo del ratón. En esta pestaña en los grupos **Modificar** y **Reducir** se encuentran las herramientas para modificar estos aspectos. (Figura 3.63)

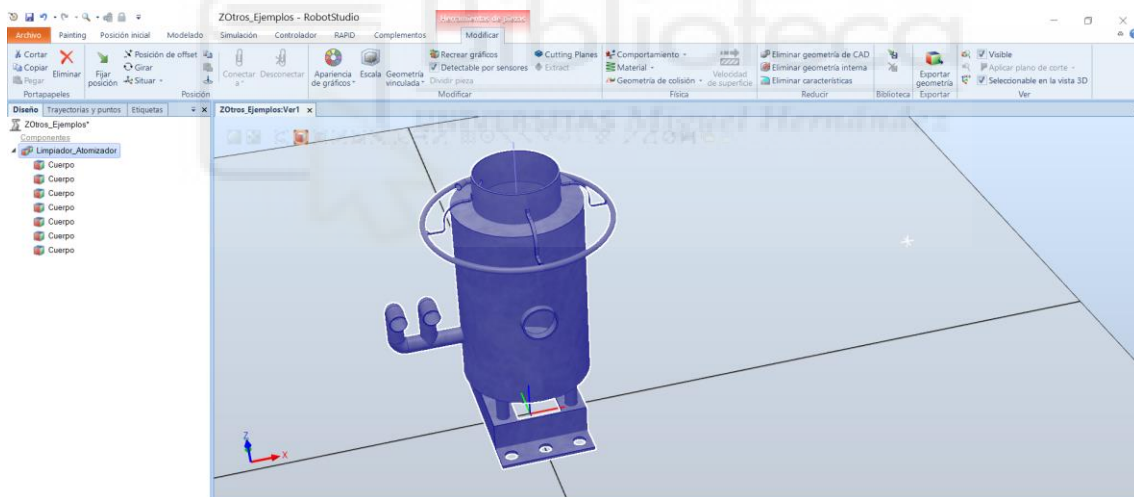


Figura 3.63 - Pestaña Modificar para un componente de la estación

- **Seleccionar color.** Esta herramienta permite cambiar el color del componente al completo. Mediante esta herramienta, también puede cambiarse el color de solo uno de los cuerpos o una de las caras que forman el componente, seleccionándola en la ventana grafica mediante el botón derecho del ratón. (Figura 3.64)

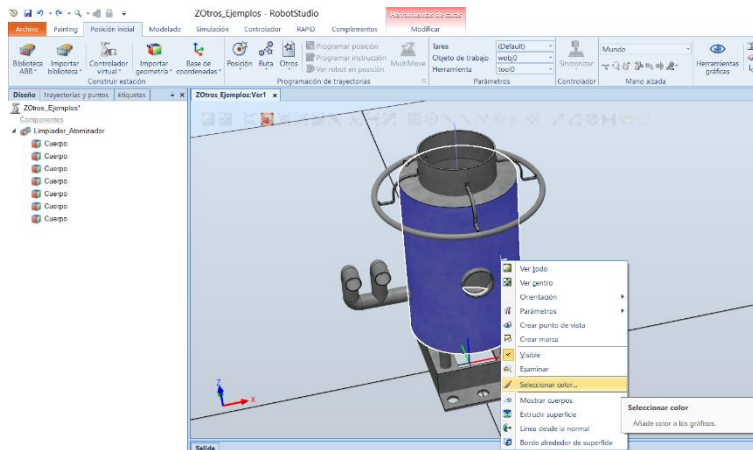


Figura 3.64 - Herramienta Seleccionar color

- **Apariencia de gráficos.** Permite modificar la apariencia del componente de una forma más completa que la herramienta anterior (Figura 3.65). Se pueden modificar aspectos como por ejemplo el color, la opacidad, el brillo, seleccionar una textura de una imagen para la apariencia del componente, propiedades de la superficie, propiedades de la textura, etc. Además RobotStudio ofrece una serie de materiales de apariencia predefinidos para modificar la apariencia del componente, como diferentes metales, hormigón, madera, piedra, vidrio, etc.

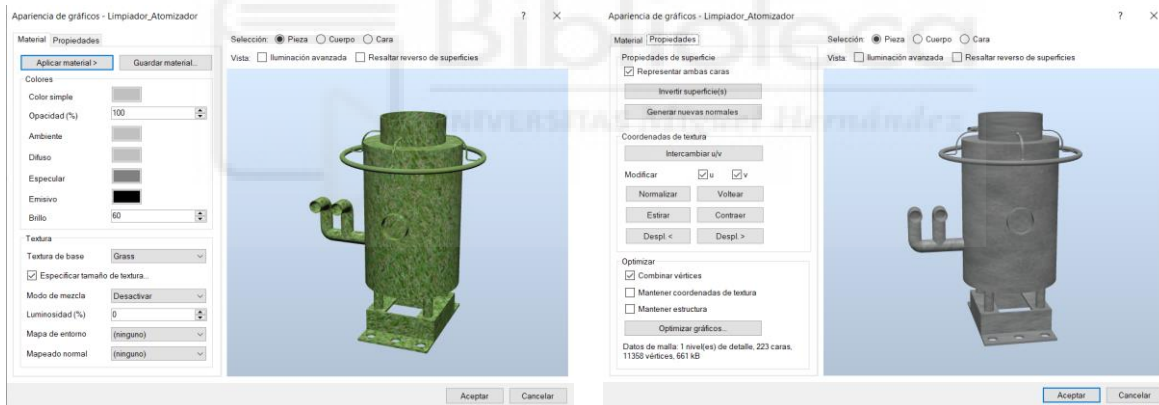


Figura 3.65 - Ejemplos de la herramienta Apariencia de gráficos

- **Establecer origen local.** Esta es una herramienta muy importante que se emplea para reponer el origen de coordenadas local del componente. Posteriormente se expondrá como se posicionan los componentes de la estación, para lo cual es necesario conocer donde se encuentra el origen de coordenadas local de los mismos. Al seleccionar esta herramienta se abre el panel **Establecer origen local**, en el cual deben especificarse las coordenadas de posición y orientación del origen de coordenadas local del componente. Las coordenadas de posición pueden especificarse numéricamente en el panel o puede indicarse un punto de la ventana gráfica de forma que se selecciona la posición de ese punto en el espacio.
- **Escala.** Esta herramienta permite escalar el componente según el factor de escala especificado.

- **Eliminar geometría de CAD.** Esta herramienta elimina los datos geométricos de un componente. Se pasa de tener una pieza con los distintos cuerpos que forman el componente a una sola pieza sin diferentes cuerpos.
- **Eliminar características.** Esta herramienta permite simplificar la geometría del modelo mediante la eliminación de mezclas, biselés y orificios según el tamaño especificado.
- **Split Part by.** Esta herramienta es específica del **Painting PowerPac** y permite dividir el componente en cuerpo o superficies más pequeñas. Dispone de tres opciones, **Body**, que divide el componente en cuerpos sólidos más pequeños, **Face**, que divide el componente en las diferentes superficies de las que está compuesto, y **Gap**, que realiza un mallado de las superficies y las divide en superficies triangulares más pequeñas. Esta herramienta resulta muy útil para eliminar o modificar partes más concretas de un componente.
- **Eliminar geometría interna.** Esta herramienta permite simplificar el modelo CAD mediante la eliminación de cuerpos y superficies internas que no son visibles desde el exterior del componente.
- **Recrear gráficos.** Esta herramienta permite seleccionar el nivel de detalle con el que se recreará la representación gráfica de la geometría CAD del componente.

Por otro lado también pueden modificarse aspectos del componente que hacen referencia a su física. Para acceder al grupo de herramientas que permiten estas modificaciones se debe seleccionar el componente en cuestión mediante el botón derecho del ratón y seleccionar la opción **Física**, de forma que aparece un desplegable con las diferentes herramientas, como se muestra en la Figura 3.66. Estas herramientas también aparecen en la pestaña **Modificar**, grupo **Física**.

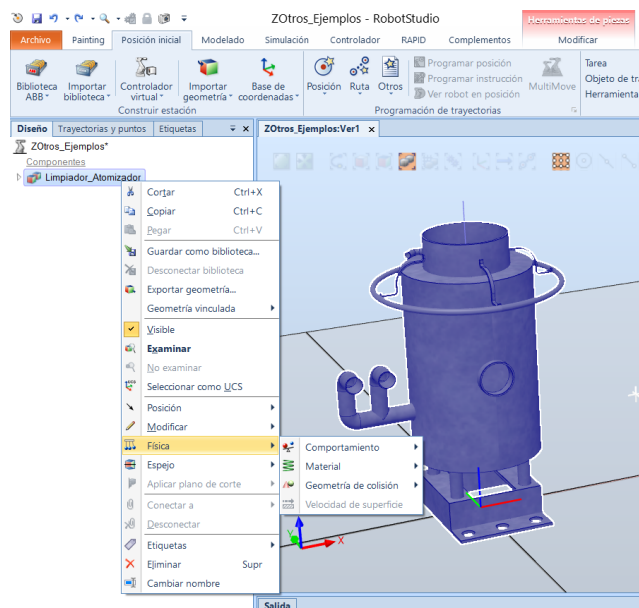


Figura 3.66 - Herramientas para la modificación de la física de una componente

- **Comportamiento.** Esta es una herramienta muy importante que permite definir cuál será el comportamiento físico del componente y su interacción con el resto de los componentes de la estación durante la simulación. Ciertas acciones, como la creación de una articulación o de un mecanismo, cambian automáticamente el comportamiento de los componentes involucrados.

Se tienen cuatro opciones de comportamiento:

1. **Inactivo.** Esta opción supone que el componente no participa en la simulación física, por lo que no interactúa con el resto de componentes y su posición permanece fija.
2. **Anclado.** Esta opción supone que el componente sí que interactúa con el resto de componentes de la estación pero se mantiene fijo en su posición.
3. **Cinemática.** Esta opción supone que el componente sí que interactúa con el resto de componentes de la estación y su movimiento está controlado por otros medios, es decir, RobotStudio controla el movimiento del componente, como por ejemplo un manipulador de robot.
4. **Dinámica.** Esta opción supone que el componente sí que interactúa con el resto de componentes de la estación y su movimiento se rige por las leyes de la física, no es controlado por RobotStudio.

- **Material.** Esta herramienta permite especificar el material del que está compuesto el componente. Las propiedades del material afectarán a su comportamiento durante la simulación. Esta opción no varía la apariencia gráfica del componente. RobotStudio proporciona una serie de materiales predefinidos, como acero, aluminio, plástico, hormigón, goma, etc., y también permite introducir cualquier material mediante definición de una serie de propiedades.

- **Geometría de colisión.** Esta herramienta se emplea para especificar cómo será la geometría del componente empleada para el cálculo de colisiones. Se tienen las siguientes opciones:

- **Calidad baja.** Mediante esta opción se crea una geometría más simple para el cálculo de las colisiones, similar a la original pero con un nivel de detalle bajo. (Figura 3.67)

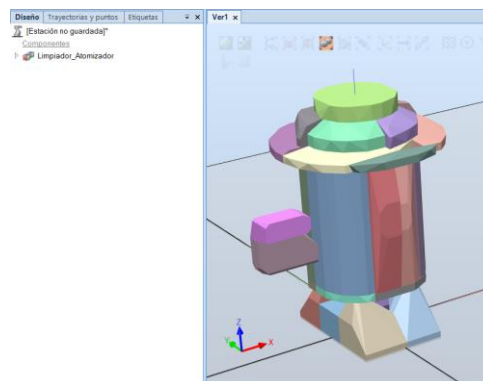


Figura 3.67 – Ejemplo de geometría de colisión de calidad baja

- **Calidad media.** Mediante esta opción se crea una geometría más simple para el cálculo de las colisiones, similar a la original pero con un nivel de detalle medio. (Figura 3.68)

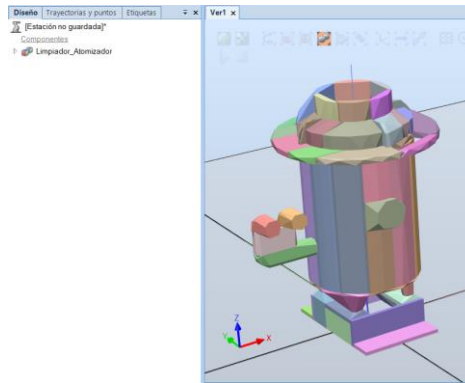


Figura 3.68 - Ejemplo de geometría de colisión de calidad media

- **Calidad alta.** Mediante esta opción se crea una geometría más simple para el cálculo de las colisiones, similar a la original pero con un nivel de detalle alto. (Figura 3.69)

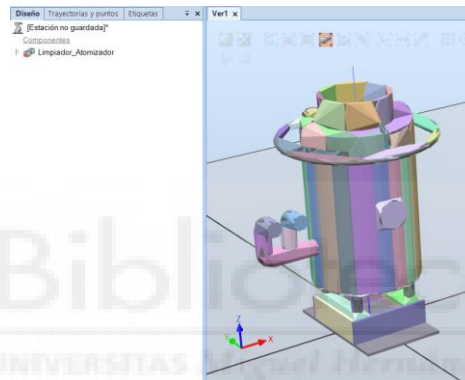


Figura 3.69 - Ejemplo de geometría de colisión de calidad alta

Estas tres opciones conllevan la creación de una nueva geometría lo que conlleva el consumo de una gran cantidad de recursos informáticos y tiempo. A mayor nivel de detalle mayor consumo de recursos y tiempo.

- **Usar geometría original.** Mediante esta opción se emplea la geometría CAD original del componente para el cálculo de colisiones. Si se emplea esta opción, no se requiere crear una nueva geometría de colisión por lo que no se consume una gran cantidad de recursos informáticos.

3.1.10 Descripción de los sistemas de coordenadas

Los sistemas de coordenadas se emplean para especificar la posición y orientación de un componente, objetivo, posición de robot, etc. en el espacio tridimensional de la estación mediante tres coordenadas X, Y y Z, y el uso de tres ángulos o cuadrantes.

Los diferentes sistemas de coordenadas que existen en RobotStudio están correlacionados jerárquicamente, es decir, el origen de cada sistema de coordenadas se establece como una posición en cada uno de sus antecesores en el orden jerárquico. Los manipuladores de robot en RobotStudio utilizan diferentes sistemas de coordenadas, cada uno de ellos adecuado para tipos concretos de movimientos o programaciones, siendo estos los siguientes:

Sistema de coordenadas del punto central de la herramienta (TCP)

Este sistema de coordenadas está situado en el punto central de la herramienta y define la posición de herramienta activa al situarse en los puntos objetivo programados. Es posible definir varios de estos sistemas de coordenadas para un mismo robot. Todos los robots poseen un sistema de coordenadas TCP predefinido, denominado **tool0**, posicionado en el centro de la brida de la muñeca del robot. (Figura 3.70)

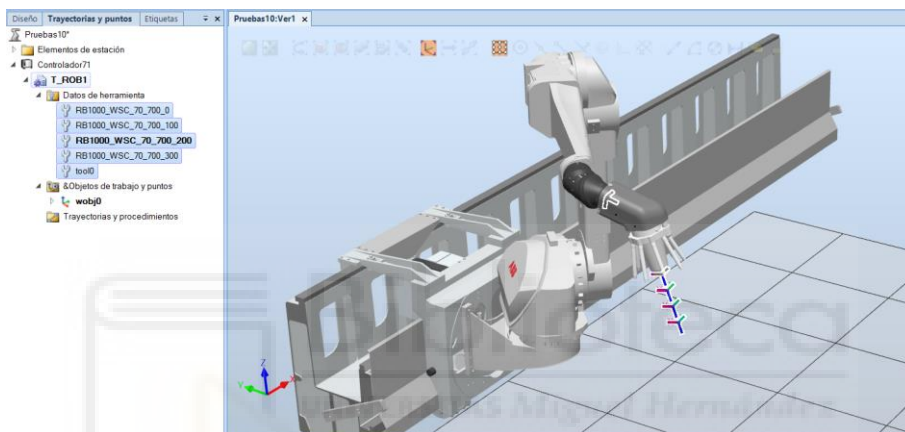


Figura 3.70 - Ejemplo de sistema de coordenadas del punto central de la herramienta

Sistema de coordenadas mundo de RobotStudio (RS-WCS)

Este sistema de coordenadas está situado en un punto fijo de la estación y representa a la totalidad de la misma. Se encuentra en la parte superior de la jerarquía de sistemas de coordenadas, es decir, el resto de sistemas de coordenadas están relacionados con y dependen de este. (Figura 3.71)

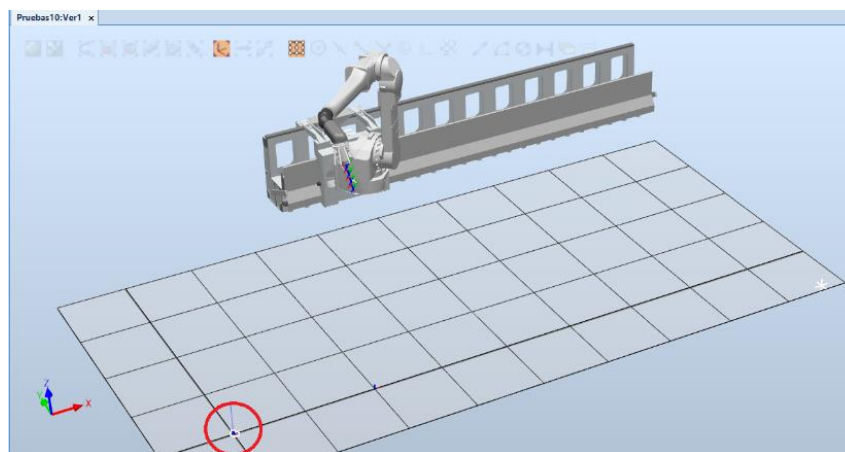


Figura 3.71 - Ejemplo de sistema de coordenadas mundo de RobotStudio (RS-WCS)

Sistema de coordenadas de la base (BF)

Cada uno de los robots presentes en la estación posee un sistema de coordenadas de la base que define su posición, estando siempre situado este en la base del robot. Este sistema de coordenadas también se denomina base de coordenadas de la base. (Figura 3.72)

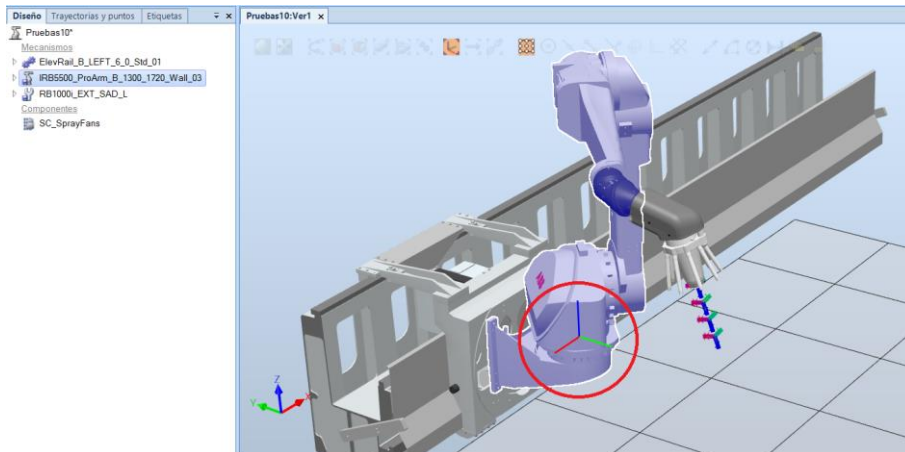


Figura 3.72 - Ejemplo de sistema de coordenadas de la base (BF)

Sistema de coordenadas de la tarea (TF)

Este sistema de coordenadas representa el origen del sistema de coordenadas mundo tal como está definido en el controlador de robot. Este sistema de coordenadas puede o no coincidir con el sistema de coordenadas de la base y/o con el sistema de coordenadas mundo de RobotStudio. Este sistema de coordenadas también se denomina base de coordenadas de la tarea. (Figura 3.73)

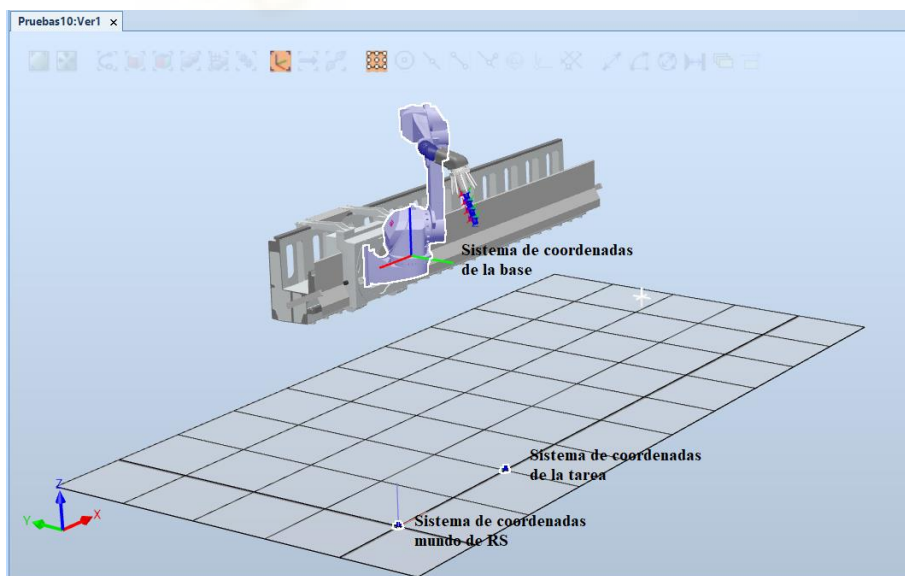


Figura 3.73 - Ejemplo de sistema de coordenadas de la tarea (TF)

En el caso de que solo se tenga un robot en la estación, el sistema de coordenadas de la tarea se corresponde con el sistema de coordenadas mundo del controlador. Si se tienen varios controladores en la estación, el sistema de coordenadas de la tarea permite el trabajo de los

robots en sistemas de coordenadas diferentes, es decir, los robots pueden situarse y trabajar de forma independiente entre sí mediante la definición de sistemas de coordenadas de tarea diferentes para cada robot.

Sistema de coordenadas del objeto de trabajo (Wobj)

Este sistema de coordenadas está compuesto a su vez por dos sistemas de coordenadas, la base de coordenadas del usuario y la base de coordenadas del objeto, siendo la segunda descendiente de la primera. Los sistemas de coordenadas del objeto de trabajo se emplean para referenciar en ellos los puntos objetivo (posiciones) programados. Si no se indica ningún objeto de trabajo al programar una posición, esta depende del objeto predeterminado **Wobj0**, que siempre coincide con el sistema de coordenadas de la tarea.

Este tipo de sistemas de coordenadas son muy útiles para reposicionar los objetivos programados, ya sea sobre una pieza de trabajo o en el espacio, simplemente ajustando la posición del sistema de coordenadas del objeto de trabajo en el que han sido programados. Esto puede hacerse automáticamente al mover la pieza de trabajo si previamente se ha conectado el sistema de coordenadas del objeto de trabajo a la pieza de trabajo, de forma que al mover la pieza de trabajo el sistema de coordenadas se mueve con ella. (Figura 3.74)

Dicho esto se sobreentiende que estos sistemas de coordenadas también se emplean para los movimientos coordinados, es decir, si un objeto de trabajo está fijado a una unidad mecánica, por ejemplo un transportador, y el sistema utiliza la opción de movimiento coordinado, el robot encontrará los objetivos en el objeto de trabajo aunque la unidad mecánica mueva el objeto de trabajo.

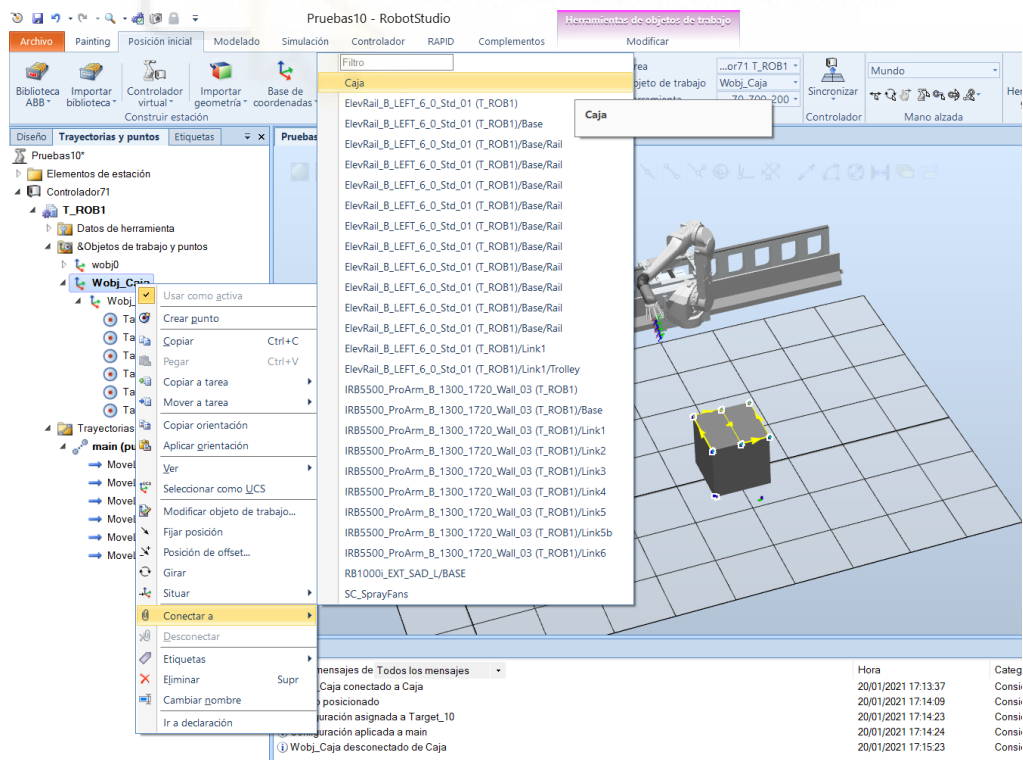


Figura 3.74 - Ejemplo de conexión de un sistema de coordenadas del objeto de trabajo a un componente de la estación

Para crear un sistema de coordenadas del objeto de trabajo se selecciona la pestaña **Posición Inicial**, grupo **Programación de trayectorias, Otros**, de forma que aparece el desplegable mostrado en la Figura 3.75, en el que se debe seleccionar la herramienta **Crear objeto de trabajo**.

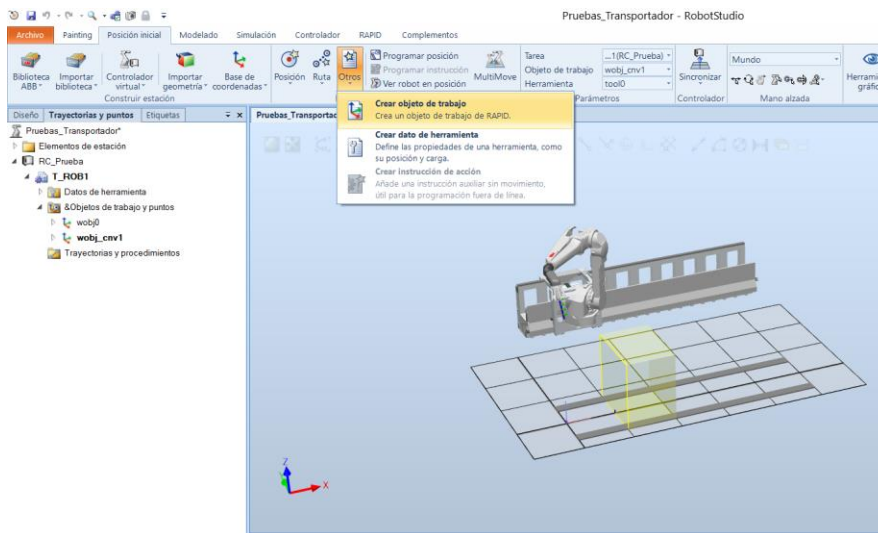


Figura 3.75 - Creación de un nuevo sistema de coordenadas del objeto de trabajo

Al seleccionarla aparece el panel **Crear objeto de trabajo** en el que deben especificarse todas las características del objeto de trabajo, como son el nombre, si esta sostenido por un robot, si se mueve mediante una unidad mecánica, la posición de los dos subsistemas de coordenadas que lo componen y la tarea y el módulo en el que se programan. (Figura 3.76)

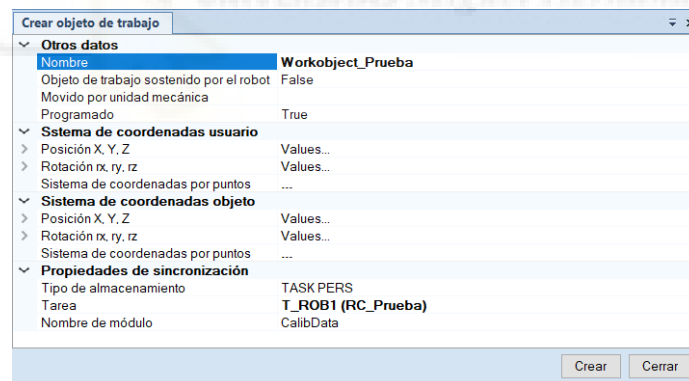


Figura 3.76 - Panel Crear objeto de trabajo

Para finalizar se pulsa **Crear** de forma que se crea el sistema de coordenadas del objeto de trabajo y este aparece en el navegador **Trayectorias y puntos**, dentro del controlador virtual y la tarea en el que se haya programado, dentro del apartado **Objetos de trabajo y puntos**.

Sistemas de coordenadas del usuario

Estos sistemas de coordenadas se emplean para crear puntos de referencia en la estación, por ejemplo para simplificar la programación.

Para crear un sistema de coordenadas del usuario se selecciona la pestaña **Posición Inicial**, grupo **Construir Estación, Base de coordenadas**, de forma que aparece el desplegable mostrado en la Figura 3.77, en el que se tiene dos herramientas para crear el sistema de coordenadas de usuario.

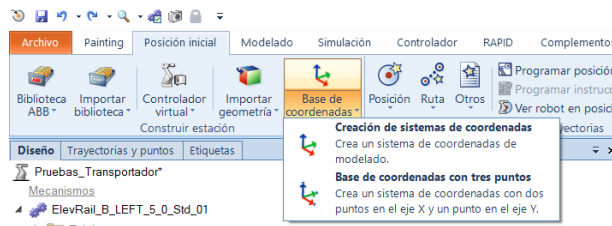


Figura 3.77 - Herramientas para crear un sistema de coordenadas de usuario

La primera de ellas se denomina **Creación de sistemas de coordenadas** y se emplea para crear un sistema de coordenadas en base a otro ya existente a una distancia y orientación especificadas.

La segunda opción se denomina **Base de coordenadas con tres puntos** y se emplea para crear un sistema de coordenadas mediante la especificación de puntos en el espacio.

3.1.11 Posicionamiento de robots y otros componentes de la estación

Una vez se han importado o creado los diferentes componentes que forman la estación estos deben posicionarse adecuadamente según lo requerido. Posicionar un objeto se define como ajustar su posición en X, Y y Z y su orientación.

Si todavía no se ha creado el controlador del robot, todos los componentes de la estación, ya sean robots, tracks, posicionadores, mecanismos u otros componentes, se posicionan de la misma forma.

En este caso, el primer paso para posicionar un elemento de la estación será conocer su origen local. En el caso de los robots, tracks y posicionares su origen local coincide con su sistema de coordenadas de la base. En el caso de un componente creado o importado se deberá observar donde se encuentra dicho origen local y en la mayoría de los casos reposicionarlo para que coincida con un vértice o algún punto del componente, permitiendo así un posicionamiento del mismo más sencillo y preciso. El reposicionamiento del origen local ya se ha expuesto en el apartado [3.1.9 Modificación de componentes en RobotStudio](#).

Una vez se conoce el origen local del elemento, este se debe seleccionar en el navegador **Diseño** mediante el botón derecho del ratón y a continuación se selecciona la opción **Posición**, de forma que aparece un desplegable con las diferentes herramientas de posicionamiento, como se muestra en la Figura 3.78.

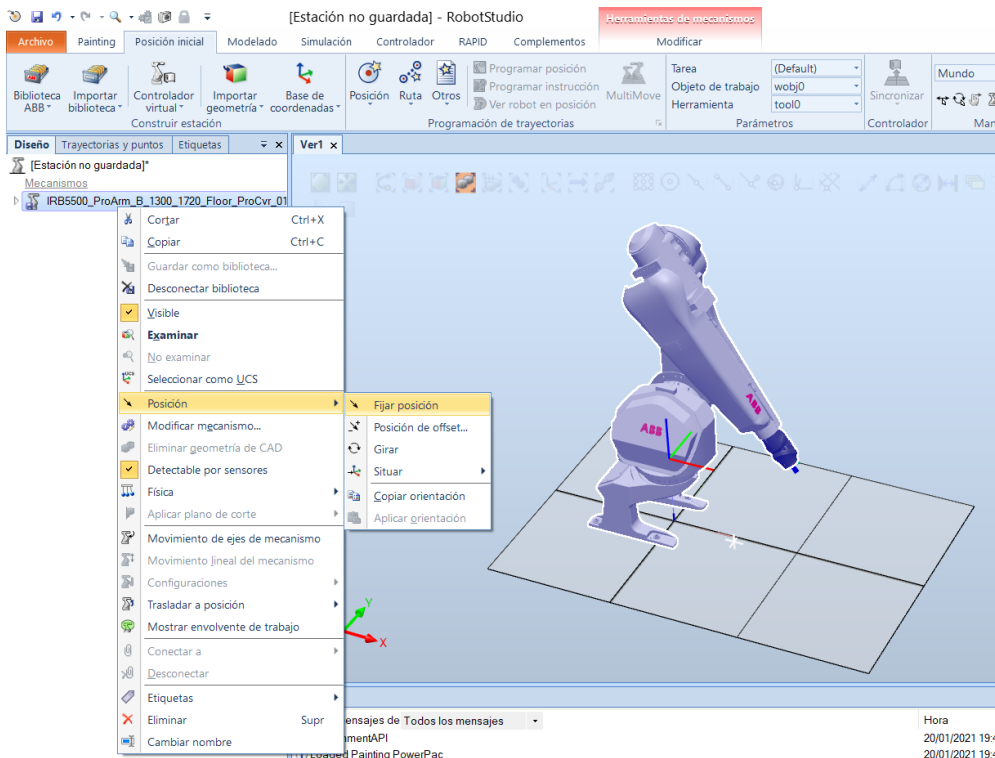


Figura 3.78 - Herramientas para el posicionamiento de un componente

La primera de las herramientas se denomina **Fijar posición** y permite posicionar el elemento respecto a los siguientes sistemas de coordenadas:

- **Mundo**. Se reposiciona el elemento respecto del sistema de coordenadas mundo de RobotStudio.
- **Objeto superior**. En el caso de que elemento en cuestión esté compuesto por varios cuerpos, cada uno de los cuales puede tener un origen local diferente, cada uno de estos cuerpos puede reposicionarse respecto de origen local del elemento completo.
- **Local**. El elemento se reposiciona respecto de su propio origen local.
- **UCS**. El elemento se reposiciona respecto de un sistema de coordenadas de usuario.

Al seleccionar esta opción aparece el panel **Definir posición**, mostrado en la Figura 3.79, en el cual debe especificarse el sistema de coordenadas respecto al que se reposicionará el objeto, su nueva posición en X, Y y Z y su orientación. La posición puede indicarse introduciendo los valores manualmente o puede seleccionarse un punto del espacio de forma que los valores de posición serán los de ese punto. Una vez se han especificado todos los aspectos se selecciona **Aplicar** y el objeto se posicionará con su origen local en la posición y orientación especificadas.

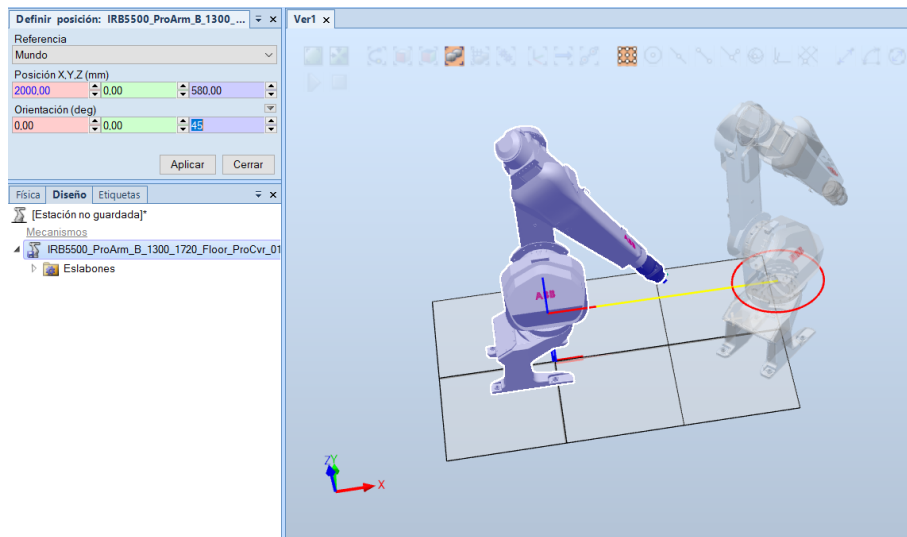


Figura 3.79 - Panel Definir posición

La siguiente opción se denomina **Posición de offset** y permite posicionar el elemento a una distancia especificada respecto de la posición actual y de uno de los sistemas de coordenadas explicados anteriormente. En la siguiente Figura 3.80 se observa que el robot se reposiciona a una distancia de 2000 mm en el eje X respecto de la actual y se gira 45° respecto al eje Z, obteniéndose así una posición con respecto al sistema de coordenadas mundo de 4000 mm en el eje X y 90° respecto al eje Z.

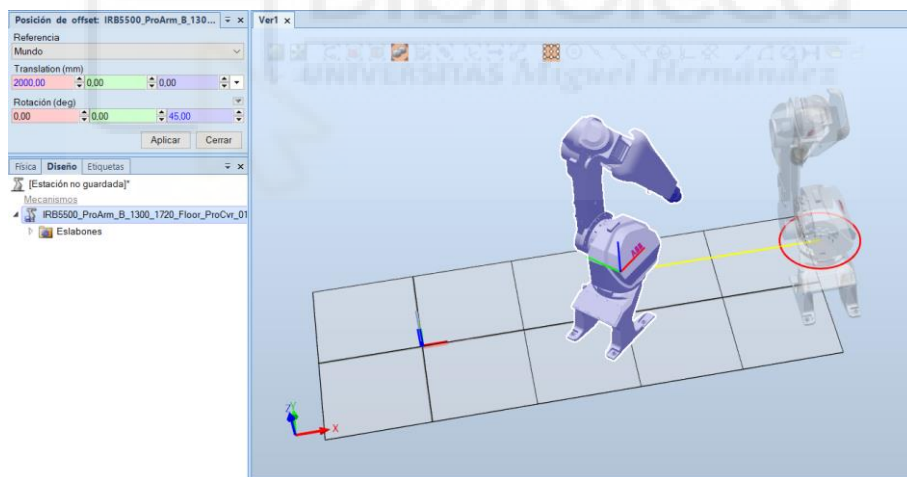


Figura 3.80 - Ejemplo de aplicación de la herramienta Posición de Offset

La siguiente opción se denomina **Girar** y permite rotar el elemento un ángulo especificado respecto a un eje de un sistema de coordenadas. (Figura 3.81)

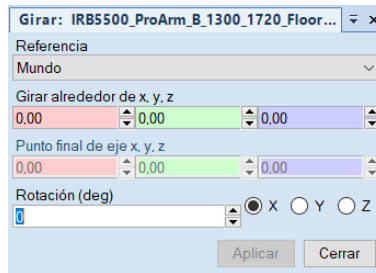


Figura 3.81 - Panel Girar

La última de las opciones se denomina **Situar** y permite posicionar el elemento mediante cinco opciones diferentes:

- **Un punto.** Se selecciona un punto origen del elemento y un punto destino donde se posicionará el punto origen del elemento, por lo tanto el elemento se mueve sin variar su orientación.
- **Dos puntos.** El elemento se reubicará para hacer coincidir el primer punto origen y destino y a continuación se reorientará para hacer coincidir la recta que forman los dos puntos origen con la recta que forman los dos puntos destino.
- **Tres puntos.** El elemento se reubicará para hacer coincidir el primer punto origen y destino y a continuación se reorientará para hacer coincidir el plano que forman los tres puntos origen con el plano que forman los tres puntos destino.
- **Base de coordenadas.** El elemento se reubicará para hacer coincidir su origen local, en posición y orientación, con la posición y orientación de un sistema de coordenadas especificado.
- **Dos bases de coordenadas.** El elemento se reubicará añadiendo al primer sistema de coordenadas seleccionado la posición del segundo sistema de coordenadas seleccionado, pero manteniendo su orientación.

Por otro lado existe la opción **Copiar orientación** que permite copiar únicamente la orientación de un elemento para después aplicársela a otro, teniendo así ambos exactamente la misma orientación.

También pueden posicionarse los elementos mediante las herramientas de movimiento a mano alzada. Estas herramientas permiten un rápido posicionado de los elementos pero con una precisión de posicionamiento baja. Para acceder a ellas se selecciona la pestaña **Posición Inicial**, grupo **Mano alzada** y en él se tienen tres herramientas para mover componentes a mano alzada. (Figura 3.82)

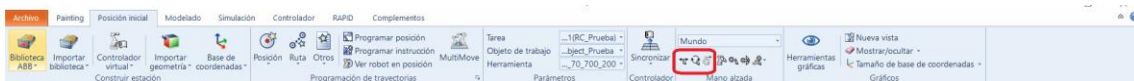


Figura 3.82 - Herramientas de posicionamiento a mano alzada

La primera de ellas se denomina **Mover** y permite desplazar linealmente el elemento en la dirección X, Y o Z deseada.

La segunda herramienta se denomina **Girar** y permite girar el elemento respecto de su origen local y del eje, X, Y o Z, seleccionado.

Por último se tiene la herramienta **Arrastrar**, que permite desplazar el elemento en todas las direcciones al mismo tiempo, siempre que este tenga un comportamiento físico dinámico.

También existen las opciones de posicionar el manipulador de robot sobre un track o un mecanismo definido por el usuario, la de posicionar la pieza de trabajo sobre un posicionador o un mecanismo definido por el usuario y la de posicionar cualquier elemento de la estación sobre otro elemento o conectarlo a otro elemento, de forma que ambos elementos se moverán conjuntamente. Para llevar a cabo esto se procede de la misma forma que para conectar una herramienta al robot, expuesta en el apartado [3.1.7 Importación del efector final del robot](#).

La primera opción consiste en seleccionar en el navegador **Diseño** con el botón derecho el robot o elemento de la estación deseado, seleccionar la opción **Conectar a** y seleccionar el elemento (track, posicionador, mecanismo, etc.) al que se desea conectar.

En la segunda opción, el robot o elemento de la estación se selecciona en el navegador **Diseño** y manteniendo pulsado el botón izquierdo del ratón sobre el robot o elemento, se arrastra hasta el elemento (track, posicionador, mecanismo, etc.) al que se desea conectar y una vez encima se suelta.

En ambos casos, a continuación aparece un cuadro de dialogo en el que se pregunta si se desea actualizar la posición del elemento. Si se selecciona **Sí**, el elemento a posicionar se colocará haciendo coincidir su origen local con el del elemento sobre el que se posiciona. Si se selecciona **No**, los elementos se conectarán, de forma que se moverán al unísono, pero cada uno mantendrá su posición.

Si se ha creado el controlador del robot, para reposicionar los robots, tracks o posicionadores controlados por el mismo se deben seguir los siguientes pasos para no obtener errores.

Por un lado puede reposicionarse mediante alguna de las opciones expuestas anteriormente y al seleccionar **Aplicar** se preguntará si se quiere mover también la base de coordenadas de la tarea (Figura 3.83). Si se indica que **Sí**, el robot y la base de coordenadas de la tarea se repositionan a la posición y orientación indicadas.

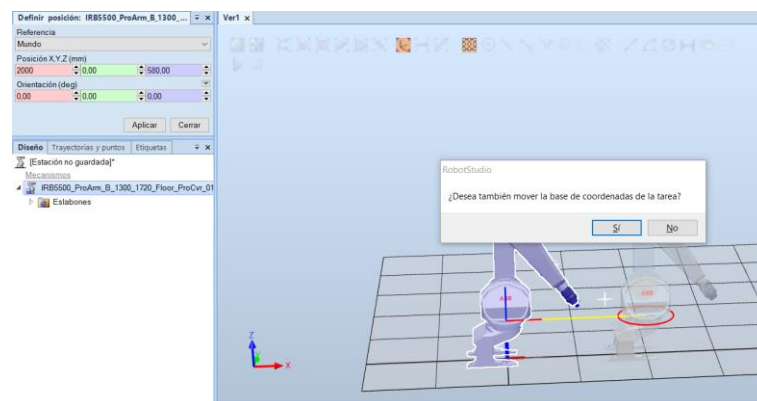


Figura 3.83 - Reposicionamiento de un robot una vez creado su controlador

Si se indica que **No**, solo se repositionará el robot manteniéndose la base de coordenadas de la tarea en su posición actual, lo que conllevará el reinicio del controlador para actualizar su

configuración y así actualizar la posición de la base de coordenadas de la base que se indica respecto de la posición de la base de coordenadas de la tarea.

Por otro lado, si únicamente se quiere repositonar el robot, pero no la base de coordenadas de la tarea, se selecciona la pestaña **Controlador**, grupo **Controlador virtual** y la herramienta **Configuración de movimiento**. Aparece el cuadro de dialogo **Configuración de movimiento**, mostrado en la Figura 3.84, en el cual debe seleccionar la unidad de robot y especificar la nueva posición y orientación de la base de coordenadas de la base. Esta nueva posición será respecto de la posición de la base de coordenadas de la tarea del controlador. Al seleccionar **Aceptar** se debe reiniciar el controlador y se debe aceptar el cuadro de dialogo que pregunta si se quiere actualizar la posición de la base de coordenadas del robot para que coincida con la indicada en el controlador, de forma que se repositona correctamente el robot.

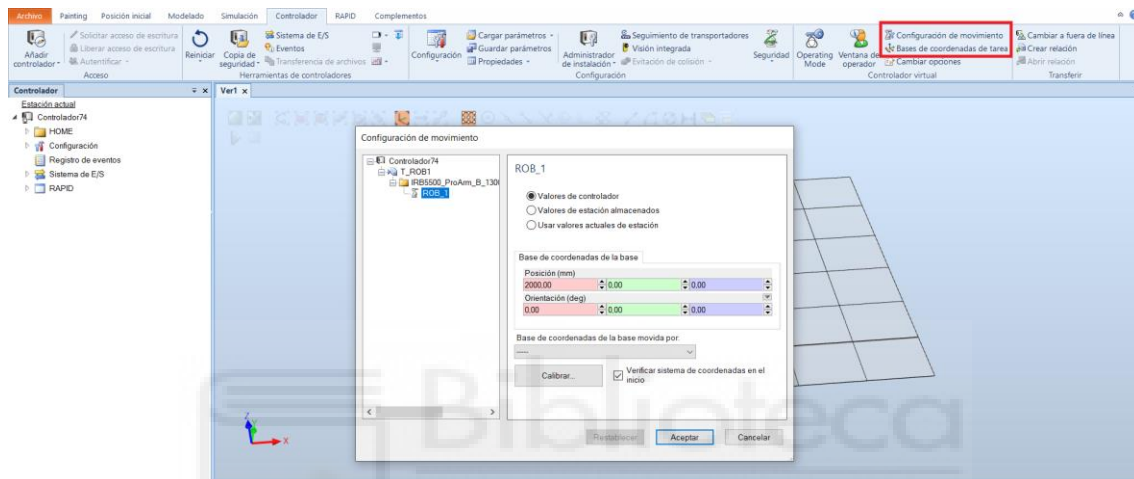


Figura 3.84 - Cuadro de dialogo Configuración de movimiento

Si se requiere repositonar la base de coordenadas de la tarea, se selecciona la pestaña **Controlador**, grupo **Controlador virtual** y la herramienta **Base de coordenadas de la tarea**. Aparece el panel **Modificar sistemas de coordenadas de tareas**, mostrado en la Figura 3.85, en el cual debe especificar la nueva posición y orientación del sistema de coordenadas de la tarea y respecto a que sistema de coordenadas se toma esa posición.

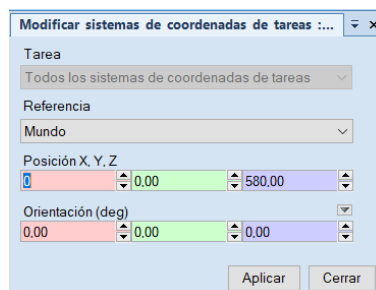


Figura 3.85 - Panel Modificar sistemas de coordenadas de tareas

3.1.12 Creación de un controlador virtual

Un controlador virtual es una copia virtual de un controlador IRC5 real que permite la ejecución local en un ordenador del mismo software empleado por un controlador real para controlar la ejecución de los robots, o lo que es lo mismo, para ejecutar el programa de RAPID, calcular los movimientos del robot y el manejo de las señales de E/S, consiguiéndose el mismo comportamiento de los robots fuera de línea que el que se consigue en línea. Los manipuladores de robot, tracks, posicionares y mecanismos presentes en la estación que no están conectados a ningún controlador no pueden ser programados.

Si no se ha añadido el controlador del robot en el momento de crear la estación, como se explica en el apartado [3.1.4 Creación de una estación](#), RobotStudio ofrece tres opciones diferentes para añadir un controlador virtual en una estación ya creada. Para acceder a cualquiera de las tres se debe seleccionar la pestaña **Posición inicial**, grupo **Construir estación** y la opción **Controlador virtual**, apareciendo el desplegable mostrado en la Figura 3.86 con las tres opciones.

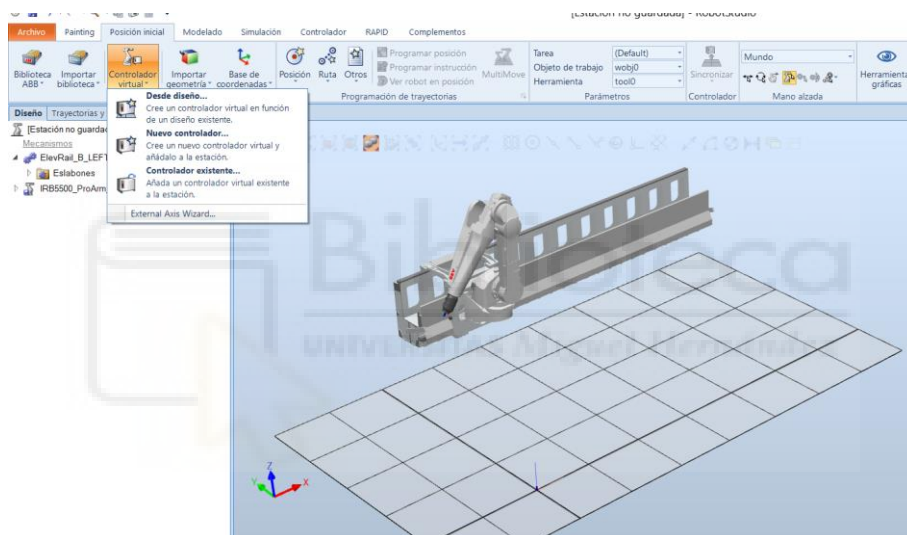


Figura 3.86 - Opciones para crear un controlador virtual de robot

La primera de las opciones se denomina **Desde diseño** y como su nombre indica permite crear un controlador virtual en función de un diseño de estación previamente creado, es decir, partiendo de los robots, tracks, posicionadores y mecanismos ya existentes en la estación. Si se emplea esta opción es muy importante que todos los mecanismos que vayan a ser controlados estén presentes en la estación antes de la creación del controlador virtual. Aunque es posible cambiar la posición de los mismos una vez creado el controlador virtual, es aconsejable que se posicionen correctamente antes de su creación, para evitar posibles errores. Además todos los mecanismos (robots, tracks, posicionadores, etc.) deben estar guardados como bibliotecas.

Al seleccionar esta opción se abre el cuadro de diálogo mostrado en la Figura 3.87, en el cual se debe indicar el nombre con el que se quiere denominar el controlador, la ubicación donde se guardará dicho controlador y la versión de RobotWare que se quiere utilizar.

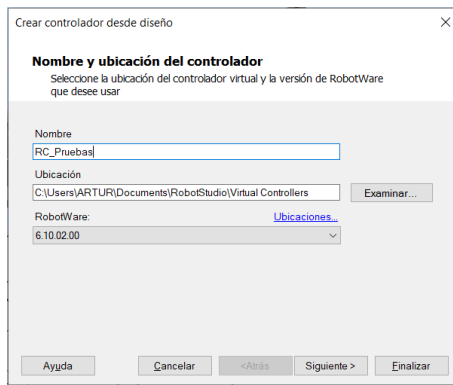


Figura 3.87 - Cuadro de dialogo Creación de controlador desde diseño

Se pulsa en **Siguiente** y se pasa a la siguiente ventana del cuadro de dialogo donde aparece el cuadro de **Mecanismos**, en el que se deben seleccionar los mecanismos a controlar. (Figura 3.88)

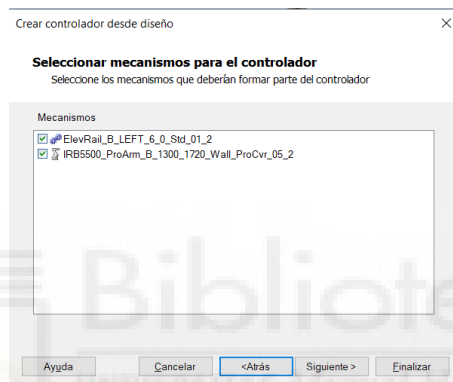


Figura 3.88 - Cuadro de dialogo Creación de controlador desde diseño - Mecanismos

Hecho esto, se pulsa en **Siguiente** y se pasa a la ventana **Tareas de controlador** del cuadro de dialogo (Figura 3.89). En esta ventana se solicita especificar la correlación de los mecanismos a una tarea de movimiento determinada. El usuario puede añadir o eliminar tareas y también puede mover de una tarea a otra los diferentes mecanismos. Si únicamente se ha seleccionado un mecanismo en la ventana anterior, esta ventana no se muestra.

En este paso existen una serie de reglas de obligado cumplimiento para poder crear el controlador:

- Únicamente está permitido un robot de TCP por tarea.
- El número máximo de tareas es de seis, pero solo pueden incluirse cuatro robots “normales” de TCP y éstos deben estar asignados a las cuatro primeras tareas. En el caso de robots de pintura solo puede incluirse un único robot.
- El número de tareas no puede ser superior al número de mecanismos.
- Si se incluye un robot de TCP y un eje externo (track, posicionador, etc.), inicialmente ambos se asignarán a la misma tarea. Sin embargo, es posible añadir una nueva tarea y asignar en esta el eje externo.
- Si se incluye más de un robot de TCP, inicialmente todos los ejes externos se asignarán a una tarea separada. Sin embargo, es posible trasladarlos a las otras tareas junto con los robots.

- El número de ejes externos en una tarea está limitado por el número de módulos de accionamiento disponibles. Normalmente en se trata de uno en el caso de los robots grandes, dos en los robots medianos y tres en los pequeños.

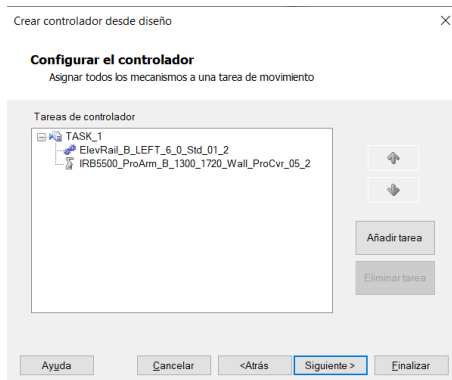


Figura 3.89 - Cuadro de dialogo Creación de controlador desde diseño - Tareas del controlador

A continuación se pulsa en **Siguiente** y se llega a la última ventana del cuadro de dialogo denominada **Opciones del controlador** (Figura 3.90). En esta ventana al seleccionar la opción **Opciones** se abre la ventana **Cambiar opciones**, en la que el usuario puede seleccionar las características que desea que posea el controlador del robot, como por ejemplo, coordinación de movimiento, supervisión de movimiento, opciones específicas del tipo de robot, etc. (Figura 3.91)

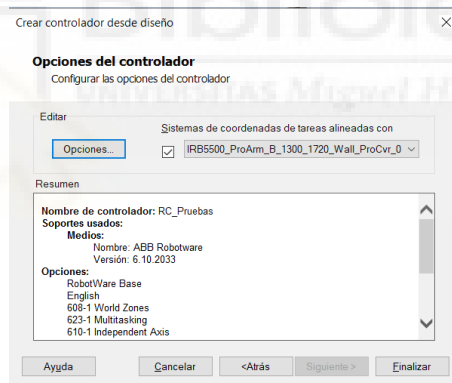


Figura 3.90 - Cuadro de dialogo Creación de controlador desde diseño - Opciones del controlador

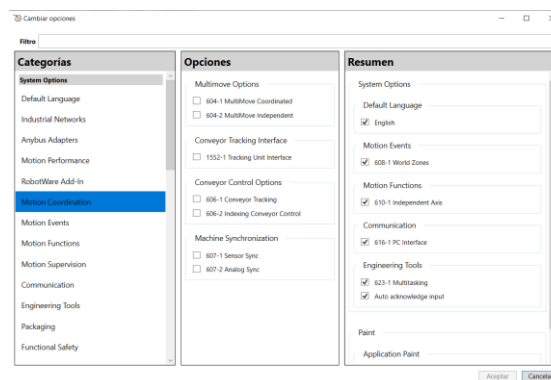


Figura 3.91 - Cuadro de dialogo Creación de controlador desde diseño - Cambiar opciones

Por último se aceptan las opciones especificadas y se selecciona **Finalizar** para crear el controlador virtual de los mecanismos especificados.

La segunda opción se denomina **Nuevo controlador** y permite crear un nuevo controlador para un único robot sin que este esté presente en la estación, este se especificará durante el proceso de creación del controlador y se añadirá a la estación junto con el controlador.

Al seleccionar esta opción se abre el cuadro de dialogo mostrado en la Figura 3.92, en el cual primeramente se debe indicar el nombre con el que se quiere denominar el controlador y la ubicación donde se guardará dicho controlador.

Si se selecciona la opción **Crear nuevo**, se debe especificar la versión de RobotWare que se quiere utilizar, el modelo del manipulador de robot que se controlará y si se quieren personalizar las opciones del controlador.

Si se selecciona la opción **Crear a partir de copia de seguridad**, se debe buscar y seleccionar el archivo de copia de seguridad de controlador necesario, la versión de RobotWare que se quiere utilizar y si se quiere restaurar en el nuevo controlador la copia de seguridad seleccionada.

En cualquiera de estos casos, en el apartado **Mecanismos** se debe especificar si se desea importar el robot de la biblioteca de RobotStudio o usar un robot presente en la estación.

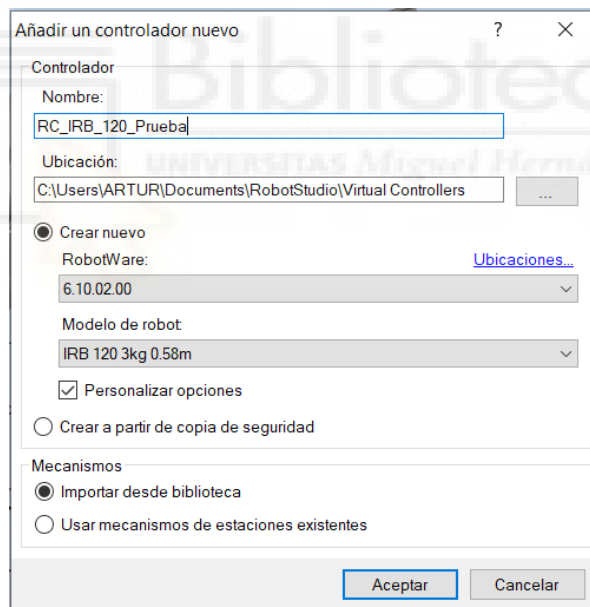


Figura 3.92 - Cuadro de dialogo Añadir un nuevo controlador

Para finalizar se pulsa **Aceptar** y se creará el controlador. Si existe más de una variante para el modelo de robot especificado, se abrirá un cuadro de dialogo para especificar la versión exacta del modelo de robot.

La tercera y última opción se denomina **Controlador existente** y permite añadir a la estación un controlador que se ha creado en otro momento, es decir, un controlado que ya existe.

Al seleccionar esta opción se abre el cuadro de dialogo mostrado en la Figura 3.93, en el cual primeramente se debe indicar la ubicación del controlador que se quiere importar. En el apartado **Opciones** se debe seleccionar si se restablece el controlador virtual antes de añadirlo a la estación, si se desean importar los mecanismos de la biblioteca de RobotStudio o usar mecanismos presentes en la estación y si se quiere sincronizar el programa de RAPID presente en el controlador con la estación.

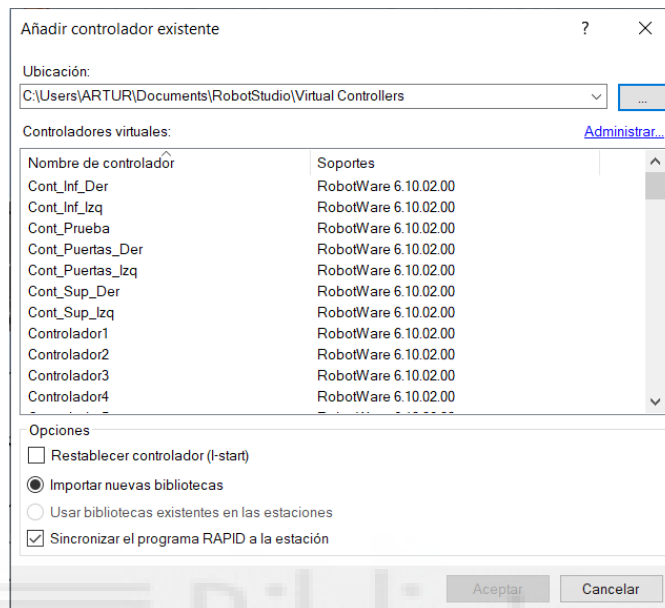


Figura 3.93 - Cuadro de dialogo Añadir controlador existente

Una vez especificado todo se pulsa **Aceptar** y se importa el controlador especificado.

3.1.13 Creación de transportadores y conexión con el controlador

Un transportador puede definirse como un dispositivo mecánico que permite el transporte continuo de la pieza de trabajo o un material de un punto a otro dentro de la estación de trabajo.

A partir de una geometría, más o menos compleja, RobotStudio permite simular en ella el desempeño de un mecanismo de transportador, es decir, basándose en una geometría RobotStudio permite crear un componente transportador que desplazará una o varias piezas de trabajo según unos parámetros especificados.

Para crear un transportador el primer paso será crear o importar la geometría que se utilizará para simular el transportador.

Una vez se tiene la geometría se accede a la pestaña **Modelado**, grupo **Mecanismo** y se selecciona la herramienta **Crear transportador**, abriéndose el panel **Crear transportador** (Figura 3.94). En este panel debe especificarse la posición y orientación del transportador respecto a un sistema de coordenadas. Por defecto se selecciona el sistema de coordenadas Local, que se refiere al origen local de la geometría del transportador. En el apartado **Transportador y**

geometría se debe seleccionar la geometría que formará el transportador y en el apartado **Tipo** se debe seleccionar el tipo de transportador, aunque solo se permite como única opción el tipo lineal. Finalmente debe indicarse la longitud del transportador y si debe generar automáticamente y continuamente copias del componente que se transporta.

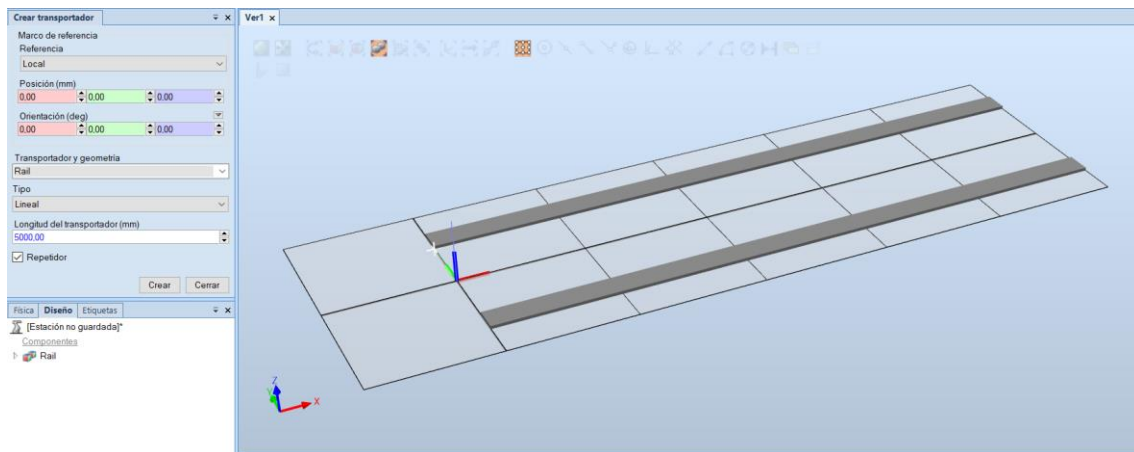


Figura 3.94 - Panel Crear transportador

Una vez se especifican las características, se pulsa **Crear**, de forma que se añade a la estación un mecanismo que contiene el transportador, mostrándose en la **Ventana gráfica** y en el navegador **Diseño**.

Teniendo creado el transportador, el siguiente paso será crear una conexión entre el controlador y el transportador, de forma que el robot puede realizar el seguimiento de la pieza de trabajo transportada. El seguimiento de transportadores es la función por la cual el robot sigue a un objeto de trabajo montado sobre un transportador en movimiento.

Para que el robot pueda realizar el seguimiento del transportador, además de la conexión, será necesario que en la creación del controlador se haya especificado la opción **606-1 Conveyor Tracking** de la categoría **Motion Coordination** (Figura 3.95). Al seleccionar esta opción RobotStudio informa que se debe seleccionar también una o varias de las opciones **1552-1 Tracking Unit Interface**, **709-1 DeviceNet Master/Slave** o **Conveyor Tracking on PIB**.

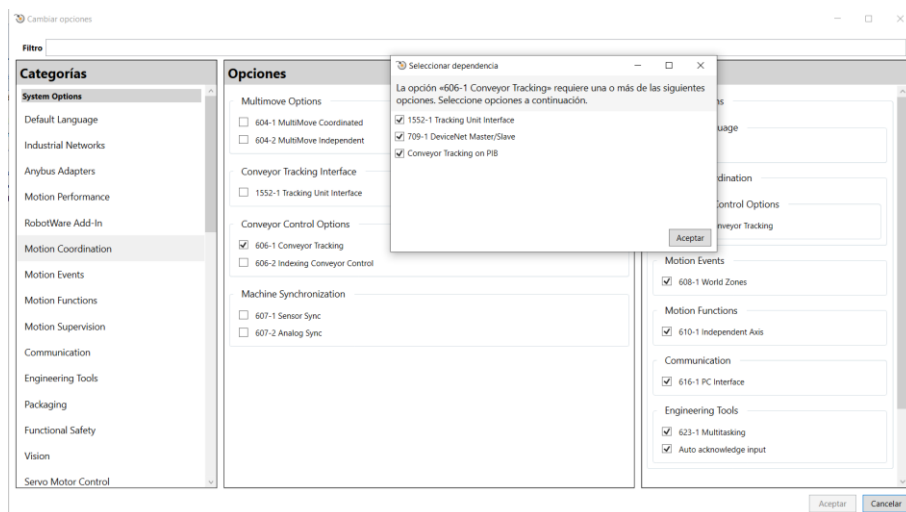


Figura 3.95 - Opciones del controlador - Conveyor Tracking

Para acceder al panel **Crear conexión** existen diferentes métodos (Figura 3.96). Se puede acceder desde la pestaña **Modelado**, grupo **Mecanismo**, seleccionando la herramienta **Crear conexión**. También se puede acceder desde el navegador **Diseño**, dentro del transportador creado, se selecciona mediante el botón derecho el apartado **Conexiones** y se indica **Crear conexión**. La última forma de acceder al panel es seleccionando en el navegador **Diseño**, mediante el botón izquierdo del ratón, el transportador, de forma que aparece una nueva pestaña **Modificar**, dentro de la cual en el grupo **Modificar** se encuentra la herramienta **Crear conexión**.

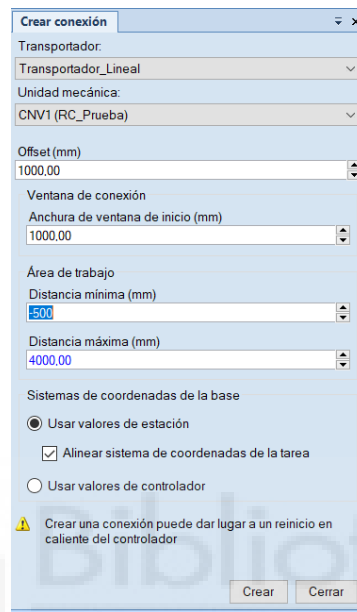


Figura 3.96 - Panel Crear conexión

Dentro del panel **Crear conexión** se debe especificar primeramente el transportador para el que se está creando la conexión y la unidad mecánica que lo mueve. A continuación debe especificarse la distancia de **Offset**, que define la ubicación de la base de coordenadas de la base de la unidad mecánica del transportador, respecto de la base de coordenadas de referencia del transportador. Los siguientes aspectos que deben especificarse hacen referencia a las distancias que controlan el seguimiento del objeto de trabajo. Todas estas distancias se miden respecto de la base de coordenadas de la base del transportador.

La primera de ellas es la **Anchura de la ventana de inicio**, que indica la distancia hasta la que es posible iniciar el seguimiento y conectarse al componente transportado. Se pueden localizar varios componentes dentro de la ventana de inicio simultáneamente pero el robot solo se conectará al primer objeto de trabajo, es decir, al primer componente que entre en la ventana de inicio. El resto de componentes que se encuentren al mismo tiempo dentro de la ventana de inicio no serán tenidos en cuenta, a no ser que el robot termine la tarea en el primer componente y suelte el objeto de trabajo conectado antes de que el segundo abandone la ventana de inicio.

La segunda distancia a especificar es la **Distancia mínima**, que indica la distancia más allá de la cual el seguimiento del objeto de trabajo debe anularse automáticamente si el transportador se mueve hacia atrás. Si el robot se ha conectado a un objeto de trabajo y el transportador se detiene y por alguna razón se mueve hacia atrás una distancia superior a la distancia mínima, el robot se desconectará del objeto.

Por último se debe especificar la **Distancia máxima**, que indica la distancia más allá de la cual el seguimiento del objeto de trabajo debe anularse automáticamente, si no se ha anulado en un punto anterior.

Finalmente se debe especificar una de las dos opciones para el **Sistema de coordenadas de la base**.

Seleccionando la primera, denominada **Usar valores de estación**, se actualiza el controlador virtual con los valores de diseño de la estación. Dentro de esta opción se puede seleccionar además la opción de **Alinear sistema de coordenadas de la tarea** para mover la base de coordenadas de la tarea al inicio de la conexión, es decir, para alinearla con la base de coordenadas de la base. La base de coordenadas de la base del transportador pasará a ser cero.

Seleccionando la segunda, denominada **Usar valores de controlador**, se actualiza el diseño de la estación y la base de coordenadas de la tarea para que concuerden con los valores del controlador virtual.

Pulsando **Crear** se realiza la creación de la conexión, que aparecerá en el apartado **Conexiones** del transportador y podrá visualizarse en la **Ventana gráfica**. (Figura 3.97)

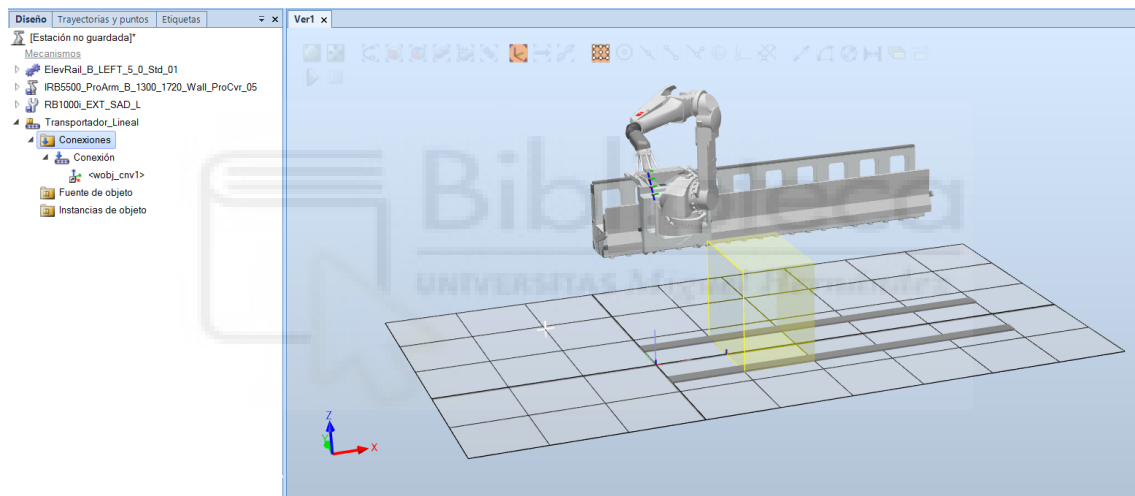


Figura 3.97 - Visualización de la conexión en un transportador

Al crear la conexión se crea también automáticamente un sistema de coordenadas del objeto de trabajo, denominado **wobj_cnv1**, que se conecta a la pieza de trabajo que entra en la ventana de inicio y se desplaza junto a ella. Si se desea que el robot realice el seguimiento de los puntos objetivo programados mientras se mueven en el transportador, estos deben programarse en el sistema de coordenadas del objeto de trabajo **wobj_cnv1**.

Para añadir un componente, presente en la estación, al transportador se selecciona con el botón derecho del ratón el apartado **Fuente de objeto** del transportador y se indica **Añadir objeto**, de forma que se abre el panel **Objeto del transportador**. En él se debe indicar el componente que se quiere añadir y la distancia de separación entre los sucesivos componentes creados de forma automática en el transportador. El componente añadido al transportador se situará haciendo coincidir su origen local con el origen del transportador por lo que si se requiere, se puede

especificar una distancia y/u orientación de compensación, es decir, la distancia y/u orientación a la que aparecerán los componentes en el transportador. (Figura 3.98)

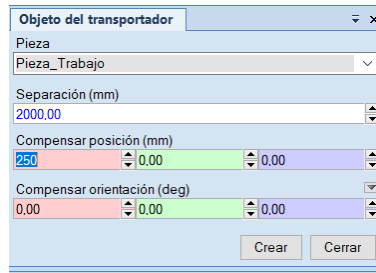


Figura 3.98 - Panel Objeto del transportador

Al añadir un objeto al transportador este desaparece del navegador **Diseño** y de la **Ventana gráfica** para pasar a estar ubicado en el apartado **Fuente de objeto** del transportador.

Finalmente existen una serie de aspectos que pueden modificarse en el transportador. Seleccionado el transportador con el botón derecho del ratón aparecen las diferentes herramientas que pueden emplearse con el transportador. La opción **Movimiento** abre el panel **Movimiento del transportador**, en el que se debe especificar la velocidad de movimiento de los objetos transportados y su aceleración. (Figura 3.99)

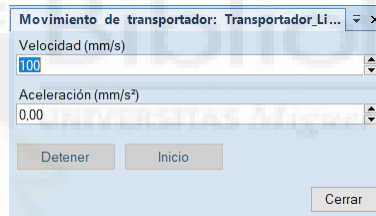


Figura 3.99 - Panel Movimiento del transportador

También se tiene la opción **Reset**, para restablecer los objetos presentes en el transportador y la opción **Borrar** que desvincula los objetos de trabajo, elimina los objetos presentes en el transportador y restablece el transportador a su posición de inicio.

3.1.14 Movimiento manual de mecanismos

Mover manualmente un mecanismo (robot, track, posicionador, etc.) permite colocarlo en la posición correcta para programar un punto objetivo, comprobar su alcance, comprobar posibles colisiones con otros elementos, etc., en definitiva ayuda a que la programación del robot sea más sencilla e intuitiva.

En RobotStudio existen dos opciones para mover manualmente un mecanismo. La primera de ellas es mediante las herramientas de mano alzada. Estas herramientas permiten mover de forma sencilla los diferentes eslabones de los mecanismos pero con una precisión baja. Para

acceder a ellas se selecciona la pestaña **Posición Inicial**, grupo **Mano alzada** y en él se tienen tres herramientas para mover mecanismos a mano alzada. (Figura 3.100)

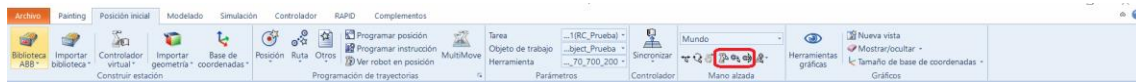


Figura 3.100 - Herramientas de mano alzada para mover mecanismos

La primera de ellas se denomina **Movimiento de eje** y permite mover los distintos ejes de un mecanismo. Para ello una vez seleccionada la herramienta, se selecciona en la ventana gráfica el eslabón del mecanismo que se quiere mover alrededor o a lo largo de su eje y se desplaza manualmente. Esta herramienta permite mover cualquier tipo de mecanismo.

La segunda herramienta se denomina **Movimiento lineal** y permite desplazar linealmente el TCP de la herramienta activa en la dirección X, Y o Z seleccionada, moviendo el resto de eslabones del robot de forma coordinada para realizar dicho movimiento lineal. Esta herramienta solo permite mover robots de TCP.

La última herramienta se denomina **Movimiento de reorientación** y permite reorientar el TCP de la herramienta activa rotándola alrededor del sistema de coordenadas del TCP y del eje seleccionado, moviendo el resto de eslabones del robot de forma coordinada para realizar dicho movimiento de reorientación. Esta herramienta solo permite mover robots de TCP.

La segunda opción se denomina **Movimiento de ejes de mecanismo** y se accede a ella seleccionándola en el desplegable que aparece al seleccionar con el botón derecho del ratón el mecanismo que se desea mover. (Figura 3.101)

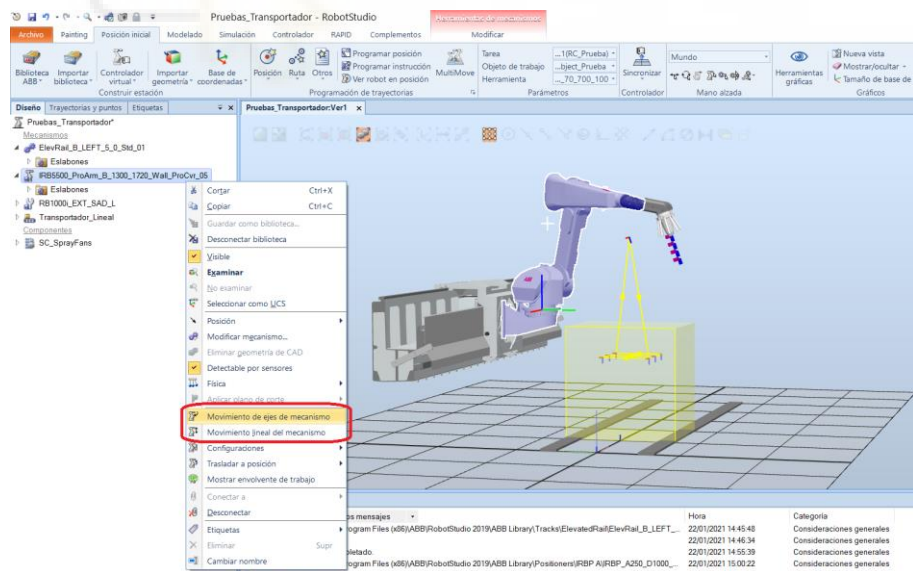


Figura 3.101 - Herramienta Movimiento de ejes de mecanismo

Al seleccionarla aparece el panel **Movimiento de ejes**, mostrado en la Figura 3.102, en el que aparece primeramente una barra de desplazamiento por cada eje del mecanismo, de forma que mediante su movimiento permite desplazar o rotar, según el tipo de eje, el eje al que hace referencia dentro de su rango de movimiento. En estas barras también puede especificarse

numéricamente la posición o el ángulo en el que colocar el eje, obteniendo así una precisión de movimiento perfecta.

También se muestra la posición del TCP del mecanismo. En el caso de que sea un robot de TCP, se muestra la posición del TCP de la herramienta activa. En el caso de que sea otro tipo de mecanismo, se muestra la posición de la base de coordenadas de la base. Si se tiene un robot montado sobre un track y se pretende mover solo el track, se mostrará la posición del TCP de la herramienta activa del robot.

Por último se debe indicar el paso con el que se desplazará el mecanismo en el caso de usar las flechas que aparecen al lado de cada barra de desplazamiento.

En el caso de un robot de TCP también se muestra la configuración en la que se encuentra. Además si este está montado sobre un track, se mostrara el identificador del mismo y una barra de desplazamiento para poder moverlo también. Se tiene también la opción de bloquear la posición del TCP, de forma que al mover el track se moverán todos los ejes del robot para mantener el TCP en esa posición.



Figura 3.102 - Panel Movimiento de ejes

Esta herramienta actúa de forma similar a la herramienta de mano alzada **Movimiento de eje**, pero de forma más completa y precisa.

Por último, únicamente para los mecanismos de robot de TCP, se tiene la herramienta denominada **Movimiento lineal del mecanismo** y se accede a ella seleccionándola en el desplegable que aparece al seleccionar con el botón derecho del ratón el mecanismo que se desea mover. (Figura 3.101)

Al seleccionarla aparece el panel **Movimiento lineal**, mostrado en la Figura 3.103, en el que se permite especificar la posición y orientación del TCP de la herramienta activa del robot, haciendo que todos los ejes del mismo se muevan para alcanzar, si es posible, esa posición y orientación del TCP. Estos valores pueden especificarse numéricamente o también pueden incrementarse o reducirse mediante las flechas que aparecen al lado de cada uno de ellos un valor igual al paso especificado. También se muestran la configuración del robot en esa posición y el sistema de coordenadas respecto al que se está indicando la misma.

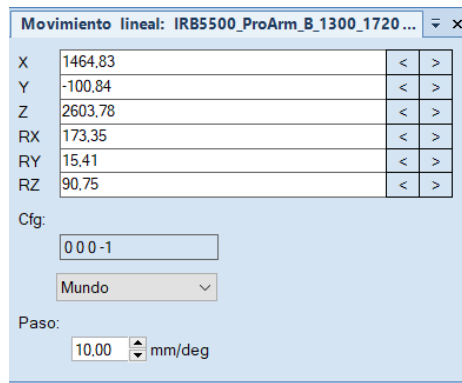


Figura 3.103 - Panel Movimiento lineal

Esta herramienta actúa de forma similar a las herramientas de mano alzada **Movimiento lineal** y **Movimiento de reorientación**, pero de forma más completa y precisa.

3.1.15 Creación de objetivos y trayectorias

Los puntos objetivo u objetivos son objetos de datos de RobotStudio que almacenan la posición y orientación de un punto especificado en el espacio tridimensional que el TCP de la herramienta del robot deberá alcanzar mediante alguno de los tipos de instrucciones de movimiento que pueden programarse dentro de una trayectoria.

El manipulador de robot puede alcanzar un mismo objetivo con diferentes configuraciones de sus ejes, es decir, con sus ejes posicionados de modo diferente. Esto se explica más detalladamente en el siguiente apartado [3.1.16 Orientaciones y configuraciones de los objetivos](#).

Así pues, un objetivo incluye valores de la posición del robot, la orientación de la herramienta, la configuración de los ejes del robot y la posición de los ejes externos adicionales. Este tipo de datos se explican con mayor detalle en el apartado [3.2.2 Datos de programa](#).

Una trayectoria, también denominado procedimiento al sincronizarse con RAPID, se puede definir como una secuencia de instrucciones de movimiento referidas a objetivos, que el controlador del robot deberá ejecutar para que el robot se desplace entre los diferentes objetivos y así completar la tarea requerida. Las trayectorias también pueden contener, además de instrucciones de movimiento, instrucciones de acción de diversos tipos y llamadas a otras trayectorias.

A continuación se expondrán las diferentes opciones que ofrece RobotStudio para realizar la programación de objetivos y trayectorias. Todas estas opciones se encuentran en la pestaña **Posición Inicial**, grupo **Programación de trayectorias**. (Figura 3.104)

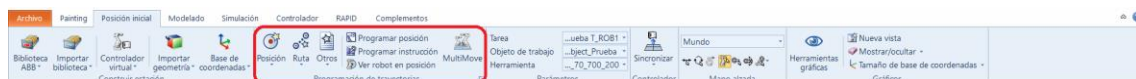


Figura 3.104 - Herramientas para la creación de objetivos y trayectorias

Antes de programar un objetivo se debe especificar en el grupo **Parámetros**, la tarea, el objeto de trabajo y la herramienta con la que se programa ese objetivo.

Para la programación de objetivos RobotStudio ofrece cuatro posibilidades diferentes. A las tres primeras de estas opciones se accede seleccionando la opción **Posición**, de forma que aparece un desplegable con estas opciones. (Figura 3.105)

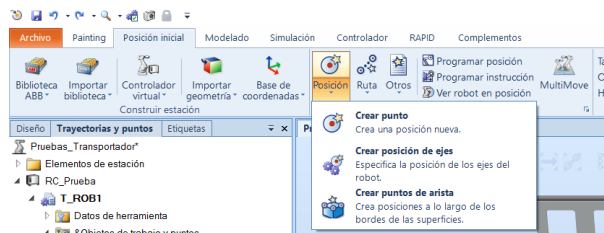


Figura 3.105 - Herramientas de la opción Posición para crear objetivos

La primera de ellas se denomina **Crear punto** y se emplea para crear objetivos especificando la posición y orientación de los mismos referenciándose en un sistema de coordenadas especificado. Al seleccionar esta herramienta se abre el panel **Crear punto** donde debe especificarse el sistema de coordenadas y la posición y orientación de cada punto. Estos parámetros pueden introducirse numéricamente o seleccionando un punto en la **Ventana gráfica**, de forma que la posición del objetivo será la posición de ese punto. Tras definir la posición de cada objetivo se pulsa en **Añadir** para ir añadiendo cada uno de los mismos. Una vez se tienen todos se pulsa **Crear**, de forma que los objetivos aparecen en el navegador **Trayectorias y puntos**, dentro del controlador, la tarea, el apartado **Objetos de trabajo y puntos** y el objeto de trabajo en el que se han programado. Los objetivos también se muestran en la **Ventana gráfica** mediante un sistema de coordenadas local en cada uno de ellos. (Figura 3.106)

El dato programado mediante esta herramienta será de tipo robtarget.

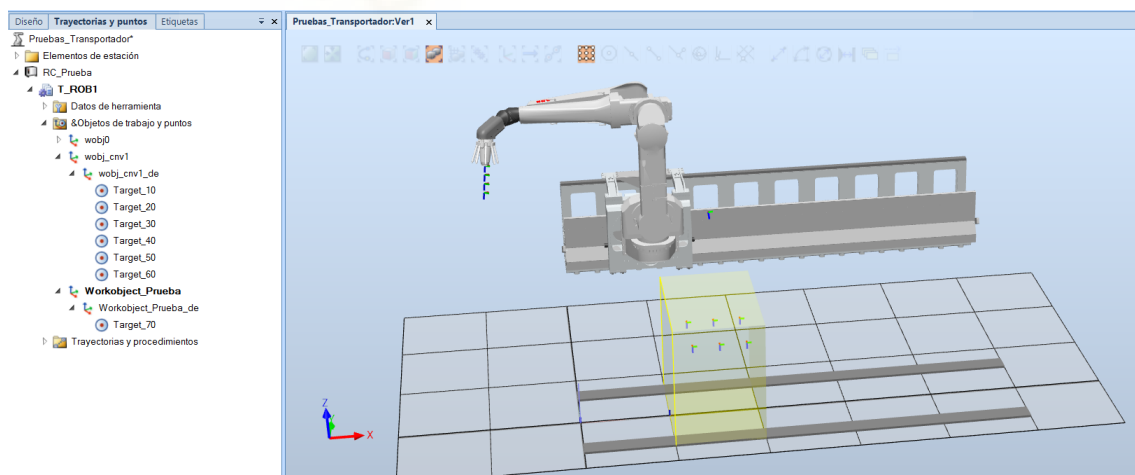


Figura 3.106 - Ejemplo de visualización de objetivos

La segunda de estas herramientas se denomina **Crear posición de ejes** y permite programar un objetivo especificando la posición de cada uno de los ejes del robot y, si los hay, de los ejes externos. Al seleccionar esta herramienta se abre el panel **Crear posición de ejes**, en el que se debe indicar el nombre del objetivo, los valores de los ejes del robot y del eje externo, el tipo de variable, y la tarea y el módulo en el que se programa. Cuando se tienen todos los datos se pulsa

Crear, de forma que los objetivos aparecen en el navegador **Trayectorias y puntos**, dentro del controlador, la tarea, el apartado **Objetos de trabajo y puntos** y **Posición de ejes**. Los objetivos también se muestran en la **Ventana gráfica** mediante un sistema de coordenadas local en cada uno de ellos. (Figura 3.107)

El dato programado mediante esta herramienta será de tipo jointtarget.

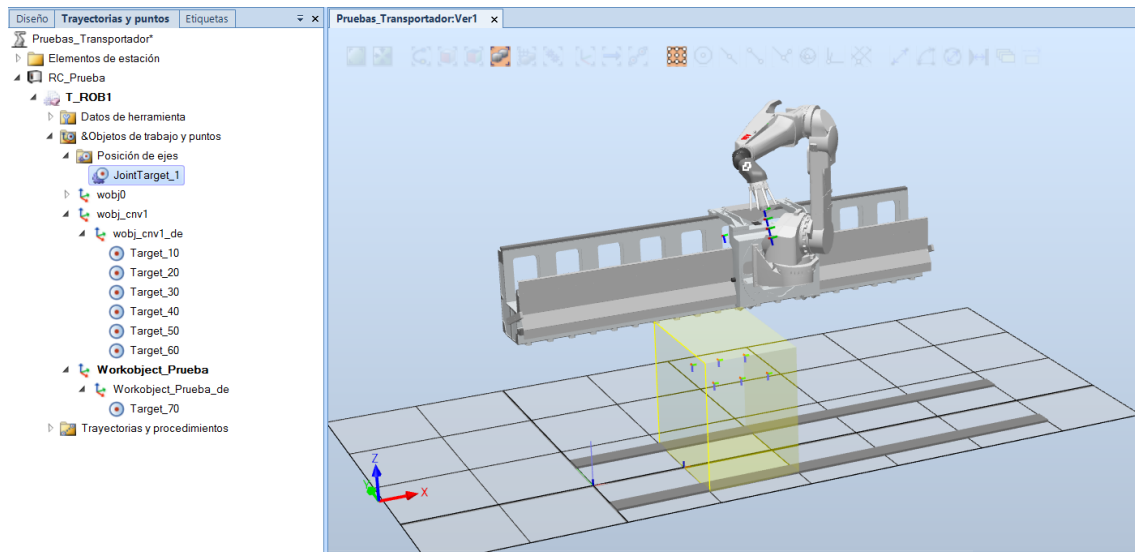


Figura 3.107 - Ejemplo de visualización de objetivos de ejes

La tercera de estas opciones se denomina **Crear puntos de arista** y permite crear objetivos a lo largo de una de las aristas de un componente. Al seleccionar esta herramienta se abre el panel **Puntos de arista** (Figura 3.108), en el que se debe indicar el offset vertical, el offset horizontal, el ángulo de aproximación y la superficie del componente donde se encuentra la arista. Una vez especificado esto, los objetivos se van seleccionando en la **Ventana gráfica** en alguna de las aristas de la superficie especificada. También puede especificarse el nombre del objetivo y la tarea y el objeto de trabajo donde se programarán. Una vez se tienen todos los objetivos se pulsa **Crear**, de forma que los objetivos aparecen en el navegador **Trayectorias y puntos**, dentro del controlador, la tarea, el apartado **Objetos de trabajo y puntos** y el objeto de trabajo en el que se han programado. Los objetivos también se muestran en la **Ventana gráfica** mediante un sistema de coordenadas local en cada uno de ellos. El dato programado mediante esta herramienta será de tipo robtarjet.

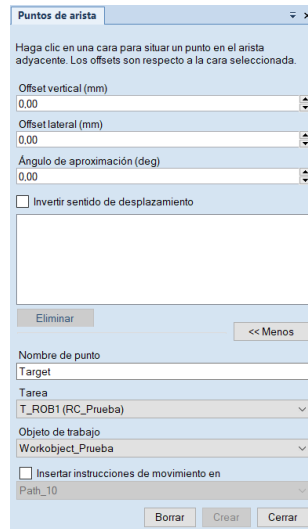


Figura 3.108 - Panel Puntos de arista

Por último se tiene la herramienta **Programar posición**, mediante la cual se programa un objetivo en la posición y orientación actual del TCP de la herramienta activa del robot. Para ello se posiciona el robot, y si lo hay el eje externo, para que la posición y orientación del TCP de la herramienta sea la deseada y se pulsa sobre la herramienta **Programar posición**. Si en la plantilla de instrucciones, que se explicará en el siguiente apartado, se tiene seleccionada una instrucción de movimiento lineal o circular (MoveL, MoveC, etc.) el objetivo que se programa será un objetivo de posición normal como los programados con la herramienta **Crear punto**. Si se tiene seleccionada una instrucción de movimiento de ejes (MoveAbsJ, MoveExtJ) se programará un objetivo de ejes, como los programados mediante la herramienta **Crear posición de ejes**. Los objetivos se programarán en la tarea, el objeto de trabajo y la herramienta especificados en el grupo **Parámetros**.

Una vez se tienen programados los objetivos estos se pueden modificar. Seleccionando el objetivo que se desea modificar con el botón derecho del ratón aparece el desplegable mostrado en la Figura 3.109.

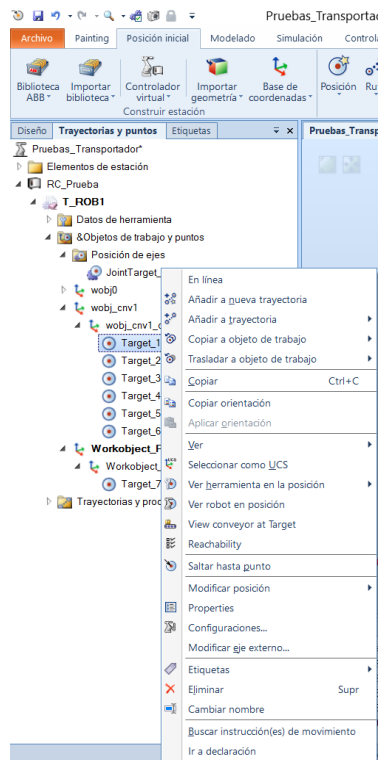


Figura 3.109 - Herramientas de modificación de objetivos

Mediante la herramienta **Cambiar nombre** se pueden renombrar los objetivos.

Mediante la herramienta **Copiar a objeto de trabajo** se puede copiar un objetivo presente en un objeto de trabajo a otro objeto de trabajo, pudiendo o no reposicionar el objetivo copiado según la posición del objeto de trabajo al que se copia.

Mediante la herramienta **Trasladar a objeto de trabajo** se puede mover un objetivo del objeto de trabajo en el que está programado a otro objeto de trabajo, pudiendo o no reposicionar el objetivo según la posición del objeto de trabajo al que se mueve.

La herramienta **Copiar orientación** permite almacenar la orientación del objetivo seleccionado, para después mediante la herramienta **Aplicar orientación** aplicársela a otros objetivos, haciendo que todos posean la misma orientación.

La opción **Ver** permite visualizar en la **Ventana gráfica** la herramienta que se especifique en el objetivo, para así comprobar si tiene la orientación requerida.

La opción **Ver robot en posición** permite visualizar en la **Ventana gráfica** el robot posicionado para que el TCP de la herramienta activa se encuentre en el objetivo. Si se selecciona con el botón izquierdo del ratón otro objetivo, la posición del robot cambia al nuevo objetivo. De esta forma puede visualizarse como se posicionarán los ejes del robot para alcanzar el objetivo.

Las opciones **View conveyor at target** y **Reachability** son propias del **Painting Powerpac**. La primera de ellas permite visualizar el objeto del transportador en la posición en la que estaba al programar el objetivo y la segunda permite comprobar si el robot puede alcanzar el objetivo.

La herramienta **Saltar hasta punto** permite visualizar en la **Ventana gráfica** el robot posicionado para que el TCP de la herramienta activa se encuentre en el objetivo. Si se selecciona otro

objetivo mediante el botón izquierdo del ratón, el robot no cambia de posición, como si pasaba con la opción **Ver robot en posición**.

Dentro de la opción **Modificar posición** se encuentran las herramientas mostradas en la siguiente Figura 3.110.

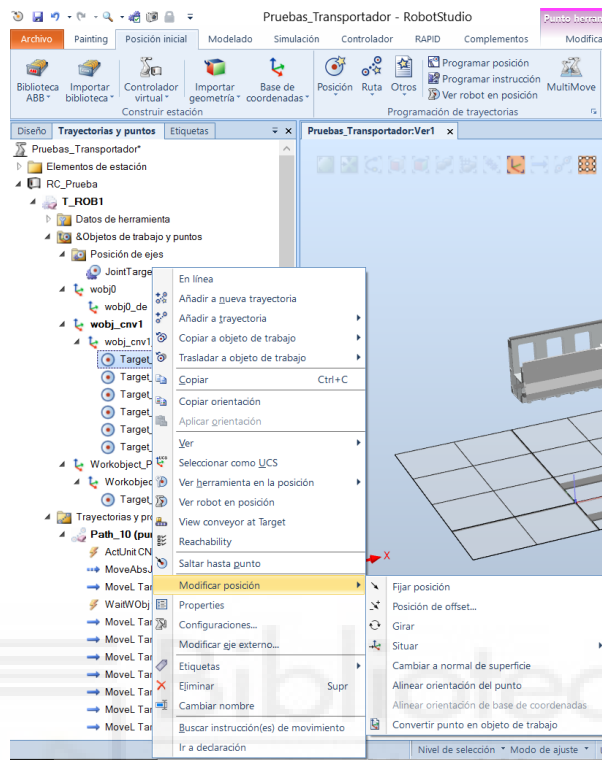


Figura 3.110 - Herramientas de la opción **Modificar posición**

Las cuatro primeras herramientas se emplean para modificar la posición y/u orientación del objetivo y funcionan de la misma forma que las herramientas con el mismo nombre explicadas en el apartado [3.1.11 Posicionamiento de robots y otros componentes de la estación](#), que se empleaban para reubicar los componentes de la estación. Entre estas herramientas la que posiblemente sea la más empleada es la herramienta **Girar**, ya que permite reorientar los objetivos de forma que al desplazarse el TCP de la herramienta del robot a estos, se posicione correctamente.

Mediante la herramienta **Cambiar a normal de superficie** se cambia la orientación del objetivo a la perpendicular a la superficie especificada.

La herramienta **Alinear orientación del punto** permite alinear la rotación alrededor de un eje del objetivo seleccionado sin afectar la rotación alrededor de los demás ejes.

La herramienta **Configuraciones** permite establecer la configuración de los ejes del robot para ese objetivo.

Por último, la herramienta **Modificar eje externo** se emplea para modificar la posición del eje externo en ese objetivo. Esto debe hacerse con precaución, ya que al modificar el eje externo puede que el objetivo ya no sea alcanzable por el robot.

Los objetivos también se pueden reubicar mediante las herramientas de mano alzada **Mover** y **Girar**, pero la precisión que se obtiene con estas es muy baja.

En cuanto a la creación y programación de trayectorias existen dos opciones posibles para crear la trayectoria, ambas aparecen al seleccionar la opción **Ruta**, como se muestra en la Figura 3.111.

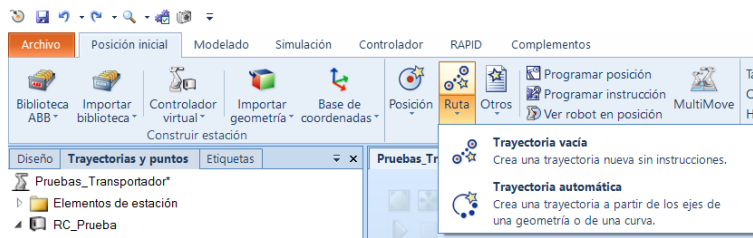


Figura 3.111 - Opción Ruta para la creación de trayectorias

La primera de ellas se denomina **Trayectoria vacía** y permite crear simplemente el procedimiento trayectoria, sin ninguna instrucción de movimiento u otro tipo dentro de la misma. Posteriormente se expondrá como incluir las instrucciones en la trayectoria.

Esta opción también se puede llevar a cabo seleccionado dentro del navegador **Trayectorias y puntos**, dentro del controlador y la tarea específica, el apartado **Trayectorias y procedimientos**, mediante el botón derecho del ratón, y seleccionando la opción **Crear trayectoria**. Las trayectorias creadas aparecerán en dicho apartado y se nombran automáticamente como Path_10, Path_20, etc., pudiéndose cambiar este nombre posteriormente.

La segunda opción se denomina **Trayectoria automática** y permite crear una trayectoria a la vez que se programan sus instrucciones de movimiento, y por tanto se crean sus objetivos. Al seleccionar esta opción aparece el panel **Trayectoria automática**, mostrado en la Figura 3.112.

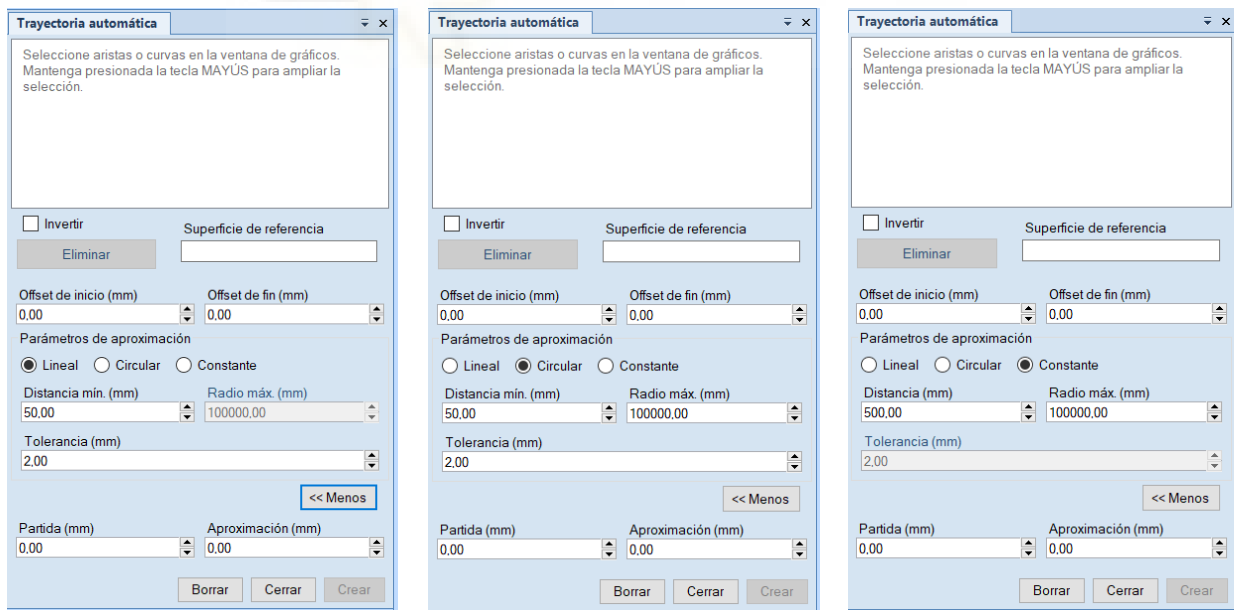


Figura 3.112 - Ejemplos del panel Trayectoria automática

El primer aspecto a seleccionar, en el apartado **Parámetros de aproximación**, es el tipo de trayectoria que se quiere crear. Si se selecciona **Lineal** se creará una trayectoria con instrucciones de movimiento lineal (MoveL) a lo largo de la arista o aristas de un componente,

de una curva o una recta seleccionada. La distancia entre los objetivos de la trayectoria será igual a la distancia entre los vértices de las aristas o en el caso de una curva será igual a la **Distancia mín.** especificada.

En el caso de seleccionar una trayectoria **Circular** se creará una trayectoria con una instrucción de movimiento lineal, de aproximación a la curva, e instrucciones de movimiento circular (MoveC) para realizar el movimiento a lo largo de la curva. En este caso se debe seleccionar una curva y la distancia entre los objetivos dependerá del tipo de curva. Si se trata de una circunferencia o un arco de circunferencia, la distancia entre objetivos será igual a la mitad de su longitud, es decir, se colocarán objetivos al inicio y al final del arco y otro en el punto medio del mismo. Si se trata de una curva tipo spline la distancia entre los objetivos será igual a la **Distancia mín.** especificada. Además también puede especificarse el radio máximo que puede tener la curva de la trayectoria.

Si se selecciona **Contante** se creará una trayectoria con instrucciones de movimiento lineal (MoveL) a lo largo de la arista o aristas de un componente, de una curva o una recta seleccionada, cuyos objetivos se dispondrán de forma equidistante a una distancia igual a la **Distancia mín.** especificada.

En cualquiera de los casos puede seleccionarse una distancia de **Offset de inicio** y de **Offset de fin**, es decir, la distancia desde el punto inicial a la que se coloca el primer objetivo y la distancia desde el punto final a la que se coloca el último objetivo.

También pueden especificarse las **distancias de Aproximación y de Partida**. Especificando estas distancias se crean objetivos a la distancia especificada del punto inicial y final respectivamente en dirección de su eje Z local.

Por último, la dirección de la trayectoria se puede cambiar de sentido seleccionando la opción **Invertir**.

Una vez se tienen todos los parámetros especificados se pulsa **Crear**, de forma que aparecerá una nueva trayectoria en el apartado **Trayectorias y procedimientos** del navegador **Trayectorias y puntos**, con las instrucciones de movimiento necesarias para realizar el movimiento entre los objetivos, y se crearán los objetivos en el objeto de trabajo activo.

La velocidad, la zona, la herramienta y el objeto de trabajo con las que se programarán las instrucciones de movimiento son las especificadas en la plantilla de instrucciones en el momento de crear la trayectoria.

Una vez se tiene una trayectoria creada, ya sea vacía o con instrucciones, se pueden añadir más instrucciones a la misma. En el caso de querer añadir instrucciones de movimiento, se seleccionan los objetivos a los que se desplazará el TCP del robot mediante las instrucciones de movimiento y se arrastran hasta la trayectoria en la que se quieren crear dichas instrucciones. Al soltarlos se crean instrucciones de movimiento, en función de la plantilla de instrucciones, para alcanzar los objetivos. Esto también puede hacerse seleccionando los objetivos y pulsando con el botón derecho del ratón encima de la selección. A continuación se selecciona la opción **Añadir a trayectoria** y se elige la trayectoria a la que añadir las instrucciones de movimiento para esos objetivos. También se tiene la opción de **Añadir a nueva trayectoria**, mediante la cual se crea una nueva trayectoria con las instrucciones de movimiento para los objetivos seleccionados.

También se pueden programar instrucciones de movimiento seleccionando con el botón derecho del ratón la trayectoria en la que se quiere incluir o una de las instrucciones dentro de la trayectoria y mediante la herramienta **Insert Move Instruction**. (Figura 3.113)

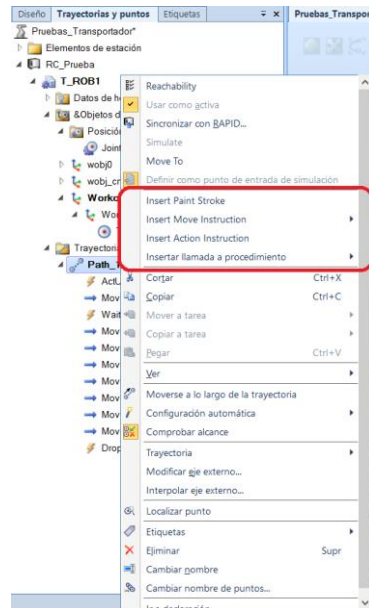


Figura 3.113 - Herramientas para insertar instrucciones en una trayectoria

Al seleccionarla se debe especificar el tipo de instrucción de movimiento y aparece el panel **Crear instrucción Move**, donde se debe especificar la posición y orientación para crear los objetivos de las instrucciones. Además puede especificarse el identificador con el que se crearán los objetivos, la tarea y el objeto de trabajo en el que se programarán y la trayectoria en la que se incluirán las instrucciones de movimiento. (Figura 3.114)

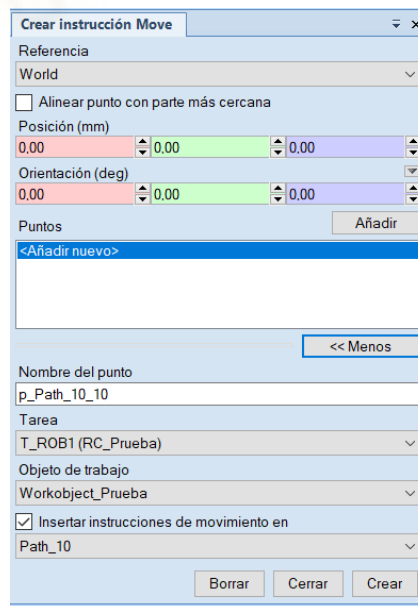


Figura 3.114 - Panel Crear instrucción Move

Por último se tiene la herramienta **Programar instrucción**, mediante la cual se programa un objetivo en la posición y orientación actual del TCP de la herramienta activa del robot y una

instrucción de movimiento, según la plantilla de instrucciones, para alcanzarlo. Para ello se posiciona el robot, y si lo hay el eje externo, para que la posición y orientación del TCP de la herramienta sea la deseada y se pulsa sobre la herramienta **Programar instrucción**. La instrucción de movimiento creada será la especificada en la plantilla de instrucciones y el objetivo dependerá también de esto, puesto que si se tiene seleccionada una instrucción de movimiento de ejes (MoveAbsJ, MoveExtJ) se programará un objetivo de ejes.

La instrucción se programará en la trayectoria seleccionada como punto de entrada de la simulación. La velocidad, la zona, la herramienta y el objeto de trabajo con las que se programará la instrucción de movimiento son las especificadas en la plantilla de instrucciones en el momento de crear la instrucción.

Si se quiere añadir una instrucción de acción esto se hace seleccionando con el botón derecho del ratón la trayectoria en la que se quiere incluir o una de las instrucciones dentro de la trayectoria y mediante la herramienta **Insert Action Instruction**, como se muestra en la Figura 3.113. También puede hacerse desde el la pestaña **Posición inicial**, grupo **Programación de trayectorias, Otros, Crear instrucción de acción**. Una instrucción de acción es un tipo de instrucción destinada a establecer parámetros, activar o desactivar equipos y funciones, etc., es decir, destinada a tareas diferentes al movimiento de los ejes del robot o los ejes externos.

Las instrucciones creadas, de cualquier tipo, se pueden repositionar dentro del orden de ejecución de la trayectoria seleccionándolas y arrastrándolas a otra posición.

Las trayectorias y las instrucciones también pueden modificarse y comprobar su correcto desempeño mediante las herramientas que aparecen al seleccionarlas con el botón derecho del ratón, como se muestra en la Figura 3.115.

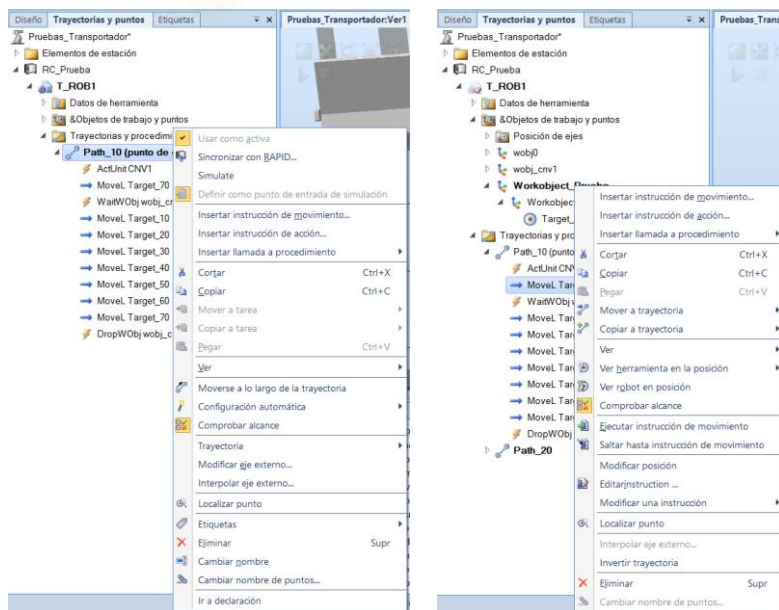


Figura 3.115 - Herramientas de modificación de trayectorias

En el caso de las trayectorias completas, las herramientas más relevantes son las siguientes.

- **Sincronizar con RAPID.** Esta herramienta permite sincronizar únicamente las instrucciones de la trayectoria seleccionada, y por tanto los objetivos a los que se hacen referencia en ella, con RAPID.
- **Definir como punto de entrada de simulación.** Esta herramienta se emplea para especificar la trayectoria seleccionada como punto de partida para la simulación.
- **Moverse a lo largo de la trayectoria.** Esta herramienta crea momentáneamente un módulo de programa, lo edita para incluir la trayectoria seleccionada, con sus instrucciones y objetivos, y ejecuta dicha trayectoria para comprobar el movimiento del robot antes de sincronizarla con RAPID. Cuando acaba el movimiento borra el módulo creado sin dejar ningún rastro en RAPID.
- **Configuración automática.** Mediante esta herramienta se optimizan las configuraciones del robot en los diferentes objetivos de la trayectoria en relación a su objetivo anterior. Existen dos variantes para esta herramienta. La primera de ellas se denomina **Instrucciones de movimiento lineal/circular** y se emplea para calcular configuraciones óptimas para las instrucciones de movimiento lineal y circular, pero mantiene las configuraciones para las instrucciones de movimiento de ejes. La segunda se denomina **Todas las instrucciones de movimiento** y como su nombre indica calcula configuraciones óptimas para todas instrucciones de movimiento de la trayectoria.
- **Comprobar alcance.** Esta herramienta permite comprobar automáticamente al programar una trayectoria si los objetivos a los que hacen referencia sus instrucciones de movimiento son alcanzables por el robot.
- **Trayectoria.** Dentro de esta opción se encuentran las herramientas mostradas en la Figura 3.116, empleadas para modificar la trayectoria. Las más relevantes son las siguientes:
 - **Interpolar trayectoria.** Mediante esta herramienta se reorientan los objetivos de una trayectoria para lograr una distribución más uniforme, es decir, la diferencia de orientación existente entre el objetivo inicial y el final se distribuye uniformemente entre los objetivos intermedios. Existen dos tipos de interpolación, lineal o absoluta. La interpolación lineal distribuye la diferencia de orientación uniformemente, en función de las posiciones de los objetivos a lo largo de la longitud de la trayectoria. La interpolación absoluta distribuye la diferencia de orientación uniformemente, en función del orden de los objetivos en la trayectoria.
 - **Reflejar trayectoria.** Mediante esta herramienta se replican simétricamente respecto de un plano espejo seleccionado todas las instrucciones de movimiento de la trayectoria, y por tanto sus objetivos, en una nueva trayectoria, que puede encontrarse en el mismo controlador o en otro controlador presente en la estación.
 - **Invertir trayectoria.** Esta herramienta cambia el orden de la secuencia de instrucciones de movimiento de la trayectoria, de forma que el robot avanza desde el último objetivo hacia el primero, relacionados con las instrucciones de movimiento.

- **Girar trayectoria.** Esta herramienta permite rotar la trayectoria completa, es decir, cambiar de posición todos los objetivos de la trayectoria, un ángulo especificado alrededor de un eje indicado. Al girar la trayectoria, los objetivos incluidos perderán sus configuraciones de ejes, si las tenían definidas.
- **Trasladar trayectoria.** Esta herramienta permite trasladar la trayectoria completa, es decir, cambiar de posición todos los objetivos de la trayectoria, una distancia especificada a lo largo de un eje indicado.

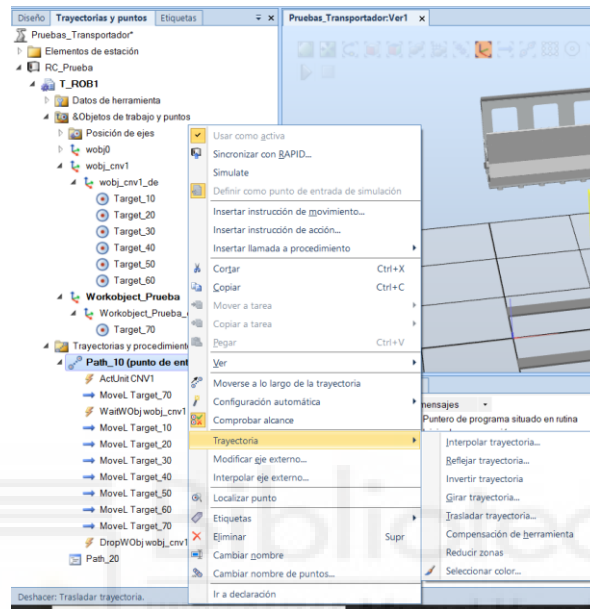


Figura 3.116 - Herramientas de la opción Trayectoria para modificar una trayectoria

- **Modificar eje externo.** Esta herramienta se emplea para modificar la posición del eje externo en todos los objetivos de esa trayectoria. Esto debe hacerse con precaución, ya que al modificar el eje externo puede que los objetivos ya no sean alcanzables por el robot.
- **Localizar punto.** Esta herramienta localiza y selecciona todos los objetivos a los que se refieren las instrucciones de movimiento de la trayectoria.

En el caso de instrucciones en particular, las herramientas más relevantes son las siguientes.

- **Mover a trayectoria.** Esta herramienta permite mover la instrucción de movimiento seleccionada de la trayectoria en la que está programada a otra trayectoria, desapareciendo de la primera y manteniendo el objetivo al que hace referencia en su posición. La instrucción se colocará al final de la lista de instrucciones de la trayectoria a la que se mueve.
- **Copiar a trayectoria.** Esta herramienta permite copiar la instrucción de movimiento seleccionada de la trayectoria en la que está programada a otra trayectoria, manteniendo la instrucción en la primera y preguntando si se desea crear un nuevo

objetivo copia del objetivo de la instrucción o se desea emplear el objetivo al que hace referencia la instrucción.

- **Ejecutar instrucción de movimiento.** Esta herramienta crea momentáneamente un módulo de programa, lo edita para incluir la instrucción seleccionada y el objetivo al que hace referencia, y ejecuta dicha instrucción para comprobar el movimiento del robot con las propiedades de movimiento especificadas. Cuando acaba el movimiento borra el módulo creado sin dejar ningún rastro en RAPID.
- **Saltar hasta instrucción de movimiento.** Esta herramienta permite visualizar en la ventana gráfica el robot posicionado para que el TCP de la herramienta activa se encuentre en el objetivo al que hace referencia la instrucción seleccionada. Si se selecciona otra instrucción mediante el botón izquierdo del ratón, el robot no cambia de posición, como si pasaba con la opción **Ver robot en posición**.
- **Modificar posición.** Esta herramienta se emplea para cambiar la posición y orientación del objetivo al que hace referencia la instrucción de movimiento seleccionada por la posición y orientación actual de TCP de la herramienta activa del robot.
- **Editar instrucción.** Esta es una herramienta muy importante que permite modificar los argumentos y propiedades editables de una instrucción de cualquier tipo. (Figura 3.117) En el caso de una instrucción de movimiento, permite cambiar el tipo de movimiento entre lineal y de ejes (MoveL o MoveJ), cambiar argumentos como la velocidad, la zona (precisión), la herramienta, etc. y también permite cambiar el tipo de instrucción de movimiento siempre que sean compatibles, como por ejemplo cambiar una instrucción MoveL por PaintL o MoveLDO.

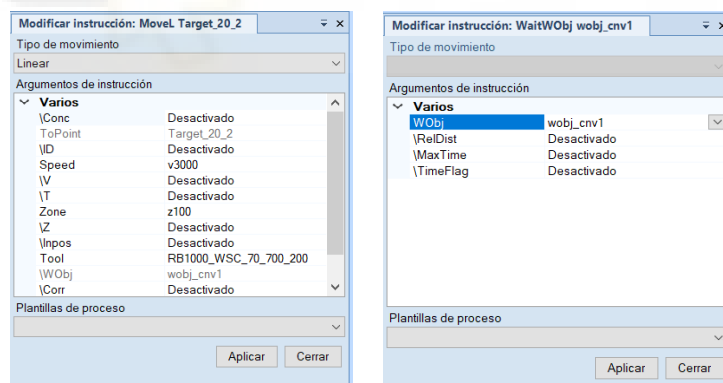


Figura 3.117 - Herramienta Editar instrucción

- **Modificar una instrucción.** Dentro de esta opción se encuentran las herramientas mostradas en la Figura 3.118, empleadas para modificar la trayectoria. Las más relevantes son las siguientes:
 - **Velocidad, Zona y Herramienta.** Estas tres herramientas permiten modificar rápidamente los argumentos de velocidad, zona o herramienta de la instrucción de movimiento seleccionada.
 - **Configuraciones.** Esta herramienta permite establecer la configuración de los ejes del robot para el objetivo al que hace referencia la instrucción de movimiento seleccionada.

- **Modificar eje externo.** Esta herramienta se emplea para modificar la posición del eje externo en el objetivo al que hace referencia la instrucción de movimiento seleccionada. Esto debe hacerse con precaución, ya que al modificar el eje externo puede que el objetivo ya no sea alcanzable por el robot.

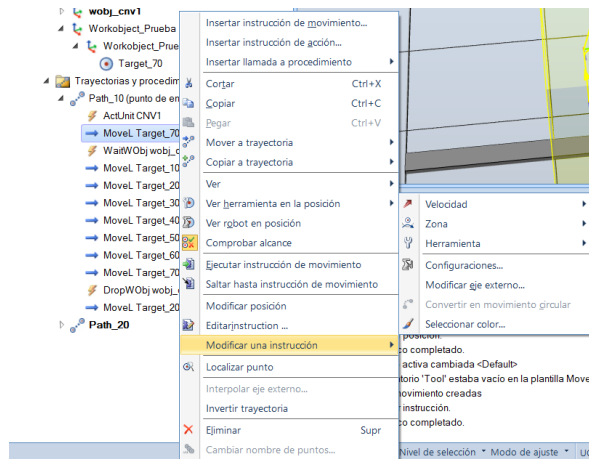


Figura 3.118 - Herramientas de la opción Modificar una instrucción

- **Localizar punto.** Esta herramienta localiza y selecciona el objetivo al que se hace referencia la instrucción de movimiento seleccionada.

Por último, cabe señalar que tanto los objetivos como las trayectorias, con todas sus instrucciones, pueden ser programados directamente en el editor de lenguaje RAPID según se explica en el apartado posterior [3.2 Programación básica en RAPID](#), aunque es más sencillo y práctico hacerlo de forma gráfica como se ha explicado en el presente apartado.

3.1.16 Orientaciones y configuraciones de los objetivos

Estos dos aspectos son de vital importancia en la programación de la estación robótica, ya que de ellos dependerá, en gran parte, que los objetivos sean alcanzables por el robot y por tanto se puedan ejecutar las diferentes trayectorias que forman la tarea a realizar.

Orientación

La orientación de un objetivo puede definirse como cada una de las posibles ubicaciones que puede tomar manteniendo fija su posición en X, Y y Z.

Al crear objetivos mediante la herramienta **Crear punto**, la orientación de estos será la especificada al crearlos. Si se crean objetivos mediante la herramienta **Crear posición de ejes**, la orientación del TCP de la herramienta quedará definida por las posiciones de ángulo especificadas para los ejes del robot, es decir, por su configuración. Si se crean objetivos

mediante la herramienta **Crear puntos de arista**, estos quedarán orientados perpendicularmente a la superficie donde se encuentra la arista. Si se crea un objetivo mediante la herramienta **Programar posición**, la orientación de este será igual a la orientación del TCP de la herramienta activa en el momento de programarlo.

Al crear trayectorias mediante la herramienta **Trayectoria automática**, se crean también los objetivos a los que se refieren las instrucciones de movimiento de la trayectoria. La orientación de estos objetivos dependerá de las características de las superficies y de las aristas o curvas circundantes. Si se crea una instrucción mediante la herramienta **Programar instrucción**, también se crea un objetivo al que hace referencia dicha instrucción, cuya orientación será igual a la orientación del TCP de la herramienta activa.

La orientación de un objetivo puede modificarse mediante las herramientas **Girar**, **Cambiar a normal de superficie** y **Alinear orientación del punto**, ya expuestas en el apartado anterior.

Otra opción interesante es emplear la herramienta **Copiar orientación**, para copiar la orientación de un objetivo y posteriormente emplear la herramienta **Aplicar orientación**, para aplicar la orientación copiada a uno o varios objetivos.

Configuración

La configuración del robot en cada objetivo puede definirse como la posición de los ejes del robot, es decir, el ángulo en el que se encuentra cada uno de sus ejes, para alcanzar dicho objetivo. A partir del objetivo programado el controlador calcula la posición de los ejes del robot para alcanzarlo, siendo común encontrar más de una configuración posible para un mismo objetivo. Por esta razón, cada uno de los objetivos tiene una configuración propia que especifica en qué cuadrante debe situarse cada eje del robot.

Las configuraciones de los ejes de los robots se designan con una serie de varios números enteros que determinan en qué cuadrante de una revolución completa se encuentran los ejes significativos.

Para los robots de 6 ejes con eslabones en serie se emplean los cuadrantes cf1, cf4, cf6 y cfx. Los números de cuadrante de cf1, cf4 y cf6 se refieren a los ejes 1, 4 y 6 del robot. cfx se emplea para seleccionar una de las ocho configuraciones posibles del robot, numeradas de 0 a 7, que indican como se posiciona respecto de las tres singularidades que existen dentro de su rango de trabajo. Estas configuraciones se muestran en la siguiente Tabla 3.1.

cfx	Centro de muñeca respecto del eje 1	Centro de muñeca respecto del brazo interior	Ángulo del eje 5
0	Delante de	Delante de	Positivo
1	Delante de	Delante de	Negativo
2	Delante de	Detrás	Positivo
3	Delante de	Detrás	Negativo
4	Detrás	Delante de	Positivo
5	Detrás	Delante de	Negativo
6	Detrás	Detrás	Positivo
7	Detrás	Detrás	Negativo

Tabla 3.1 - Configuraciones posibles para el cuadrante cfx para los robots de 6 ejes con eslabones en serie

Para los robots de 6 ejes con barra paralela solo se emplean los cuadrantes cf1, cf4 y cf6.

Para los robots de 4 ejes solo se utiliza el cuadrante cf6.

Para los robots de brazo paralelo solo se utiliza el cuadrante cf4.

Para los robots SCARA solo se emplean los cuadrantes cf1, cf4 y cfx. El número de cfx se utiliza para indicar el signo del ángulo del eje 2, siendo este igual a 1 si el ángulo es negativo, e igual a 0 si es positivo.

Para los robots de 7 ejes se emplean los cuadrantes cf1, cf4 y cf6, para los ejes 1, 4 y 6, respectivamente. cfx se emplea para seleccionar una de las 8 configuraciones de robot posibles similares al funcionamiento en otros robots. Estas configuraciones se muestran en la siguiente Tabla 3.2.

cfx	Ángulo del eje 2	Centro de muñeca respecto del brazo interior	Ángulo del eje 5
0	Positivo	Delante de	Positivo
1	Positivo	Delante de	Negativo
2	Positivo	Detrás	Positivo
3	Positivo	Detrás	Negativo
4	Negativo	Delante de	Positivo
5	Negativo	Delante de	Negativo
6	Negativo	Detrás	Positivo
7	Negativo	Detrás	Negativo

Tabla 3.2 - Configuraciones posibles para el cuadrante cfx para los robots de 7 ejes

Para los robots de pintura se emplean los cuadrantes cf1, cf4 y cf6, para los ejes 1, 4 y 6, respectivamente, y cfx para el eje 5.

Particularmente, para el robot de pintura IRB 5500 se emplean los cuadrantes cf1, cf4 y cf6, para los ejes 1, 4 y 6, respectivamente. El parámetro cfx contiene una combinación del número de cuadrante del eje 5 y las cuatro configuraciones posibles para los ejes 2 y 3. Si cfx contiene un valor con solo un número indica que el centro de la muñeca se encuentra por delante del eje 1, indicando su número la rotación del eje 5. Si cfx es igual a 0 indica que el eje 5 se encuentra entre 0° a 90°, en sentido positivo a partir de la posición cero, independientemente de donde se encuentre el punto central de muñeca. Si cfx contiene un valor con cuatro números indica que el centro de la muñeca se encuentra por detrás del eje 1. El primer número será 2, seguido de dos ceros y el cuarto número indica la rotación del eje 5. El signo de cfx dependerá de la rotación del eje 5.

El robot de pintura IRB 5350 tiene dos ejes de rotación y un eje lineal. En él se emplea el cuadrante cf1 para el eje de rotación 1 y el cuadrante cfx para el eje de rotación 2.

Los cuadrantes cf1, cf4, cf6, y en ocasiones cfx, se numeran de cero en adelante para una rotación positiva, es decir, en el sentido antihorario, y de -1 en adelante para la rotación negativa, es decir, en el sentido horario. El cuadrante 0 es el primer cuarto de revolución, es decir, de 0° a 90°, en sentido positivo a partir de la posición cero, el cuadrante 1 es el siguiente cuarto de revolución, es decir, de 90° a 180°, y así sucesivamente. El cuadrante -1 es el cuarto de revolución de 0° a -90°, en sentido negativo a partir de la posición cero, y así sucesivamente.

En la siguiente Figura 3.119 se muestran los números de los cuadrantes de configuración para los ejes 1, 4 o 6 de un robot de 6 ejes, cuya posición cero es recta.

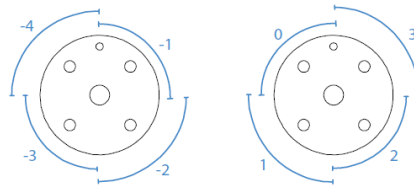


Figura 3.119 - Números de los cuadrantes de configuración para los ejes 1, 4 y 6

En el caso de un eje lineal, el número de cuadrante define el intervalo en metros en el que se encuentra el eje desde la posición neutral. El valor 0 significa una posición entre 0 y 1 metro, el valor 1 significa una posición entre 1 y 2 metros, y así sucesivamente. En el caso de los valores negativos, el valor -1 significa una posición entre -1 y 0 metros, y así sucesivamente.

En la siguiente Figura 3.120 se muestran los números de los cuadrantes de configuración para los ejes lineales.

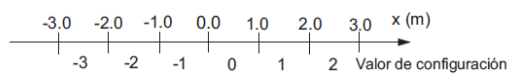


Figura 3.120 - Números de los cuadrantes de configuración para los ejes lineales

Un ejemplo de configuración para el robot industrial de pintura de seis ejes IRB 5500, empleado en el presente proyecto, es la siguiente:

[0 -1 2 1]

El primer número, 0, indica que el eje 1 del robot se encuentra posicionado en el primer cuadrante positivo, es decir, con un ángulo entre 0° y 90° .

El segundo número, -1, indica que el eje 4 del robot se encuentra posicionado en el primer cuadrante negativo, es decir, con un ángulo entre 0° y -90° .

El tercer número, 2, indica que el eje 6 del robot se encuentra posicionado en el tercer cuadrante positivo, es decir, con un ángulo entre 180° y 270° .

El cuarto número, 1, indica que el centro de la muñeca del robot se encuentra por delante del eje 1 y su eje 5 se encuentra posicionado en el primer cuadrante positivo, es decir, con un ángulo entre 90° y 180° .

Si un objetivo se programa mediante la herramienta **Programar posición**, expuesta en el apartado anterior, la configuración del robot en ese objetivo será la que tenía en el momento de programarlo. Los objetivos programados mediante la especificación de sus posiciones y orientaciones reciben el valor de configuración predeterminado [0 0 0 0], siendo muy probable que esta configuración no sea válida para que el robot alcance ese objetivo. Además aunque todos los objetivos de una trayectoria tengan definidas configuraciones alcanzables a priori, pueden ocurrir problemas al ejecutar la trayectoria si el robot no es capaz de pasar de una

configuración a la siguiente. Esto puede ocurrir en los casos en que un eje se desplaza más de 90 grados al realizar un movimiento lineal o circular.

Para solucionar estos posibles problemas la opción más sencilla es programar una configuración alcanzable a priori para cada uno de los objetivos de la trayectoria y verificar que efectivamente el robot puede moverse a lo largo de la misma.

Otra opción es desactivar el control de configuraciones, mediante las instrucciones **ConfJ \Off** o **ConfL \Off**, de forma que se omiten las configuraciones programadas y se deja que el controlador del robot elija las configuraciones durante la ejecución de la trayectoria. Esta puede parecer la solución a todos los problemas, pero debe programarse cuidadosamente ya puede provocar otros problemas.

Por último, si no existe ninguna configuración alcanzable para la posición y orientación del objetivo, este deberá ser reposicionado y/o reorientado, si la tarea lo permite, hasta una ubicación que permita una configuración alcanzable por el robot. Otra opción, ya que en algunos casos los objetivos tienen que tener una posición y orientación concretas, es añadir un eje externo que mueva los objetivos, es decir, que mueva la pieza de trabajo sobre la que están programados y a la que están conectados, o que mueva el robot para así tener un mayor rango de alcance.

Una forma sencilla y eficaz para evitar problemas, tanto si se tiene o no activado el control de configuraciones, es programar puntos intermedios de paso para guiar el movimiento del robot en desplazamientos o giros grandes de manera que el movimiento de cada eje sea inferior a 180° entre dos objetivos.

Para especificar las configuraciones de los objetivos existen dos opciones. La primera de ellas es especificar la configuración seleccionando el objetivo deseado con el botón derecho del ratón y empleando la herramienta **Configuraciones**. Al seleccionarla aparece el panel **Configuraciones**, mostrado en la Figura 3.121, en el que se indican las diferentes configuraciones alcanzables para la posición y orientación del objetivo. Al seleccionar cada una de ellas, en el apartado **Valores de eje** se muestran los valores de los ángulos de cada eje del robot. Se selecciona la que se crea adecuada y se pulsa **Aplicar**.

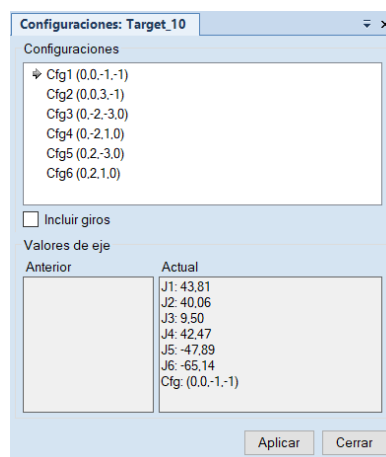


Figura 3.121 - Panel Configuraciones (desde objetivo)

La segunda opción es seleccionar dentro de la trayectoria la instrucción de movimiento referida al objetivo en cuestión mediante el botón derecho del ratón, seleccionar la opción **Modificar una instrucción** y la herramienta **Configuraciones**. Al seleccionarla aparece el panel **Configuraciones**, mostrado en la Figura 3.122, en el que se indican las diferentes configuraciones alcanzables para la posición y orientación del objetivo al que se refiere la instrucción de movimiento. Al seleccionar cada una de ellas, en el apartado **Valores de ejes – Actual** se muestran los valores de los ángulos de cada eje del robot para ese objetivo y esa configuración. En el apartado **Valores de ejes – Anterior** se muestran los valores de los ángulos de cada eje del robot para la configuración del objetivo anterior de la trayectoria. De esta forma se puede comparar la configuración del objetivo seleccionado con la del anterior y así programar configuraciones similares que no conlleven grandes desplazamientos de los ejes. Finalmente, se selecciona la que se crea adecuada y se pulsa **Aplicar**.

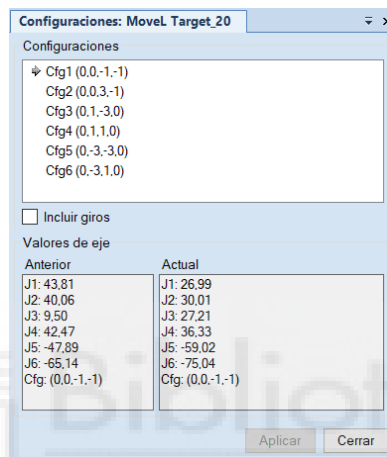


Figura 3.122 - Panel configuraciones (desde instrucción de movimiento)

Por otro lado, se puede optar por dejar que RobotStudio especifique las configuraciones de todos los objetivos presentes en una trayectoria. Esto se lleva a cabo seleccionando con el botón derecho del ratón la trayectoria deseada y seleccionando la opción **Configuración automática**. Mediante esta herramienta se optimizan las configuraciones del robot en los diferentes objetivos de la trayectoria en relación a su objetivo anterior. Existen dos variantes para esta herramienta. La primera de ellas se denomina **Instrucciones de movimiento lineal/circular** y se emplea para calcular configuraciones óptimas para las instrucciones de movimiento lineal y circular, pero mantiene las configuraciones para las instrucciones de movimiento de ejes. La segunda se denomina **Todas las instrucciones de movimiento** y como su nombre indica calcula configuraciones óptimas para todas instrucciones de movimiento de la trayectoria.

Esta herramienta es muy útil pero puede ocasionar problemas si se ejecutan varias trayectorias con muchas instrucciones de movimiento de forma consecutiva, ya que se pueden tener configuraciones muy diferentes entre los objetivos de una trayectoria y otra.

Como se ha mencionado anteriormente, una vez se tienen especificadas todas las configuraciones de los objetivos presentes en una trayectoria se debe verificar que efectivamente el robot puede moverse a lo largo de la misma. Para eso puede emplearse la herramienta **Moverse a lo largo de una trayectoria**, expuesta en el apartado anterior.

Como último aspecto en cuanto a las configuraciones, cabe señalar que durante la ejecución de un programa de robot, con una o varias trayectorias, es posible especificar si se quieren o no controlar las configuraciones de los ejes del robot. De forma predeterminada el control de la configuración está activado, pudiéndose activar también mediante las instrucciones **ConfL \On**, para controlar la configuración en los objetivos de las instrucciones de movimiento lineales o circulares, y **ConfJ \On**, para controlar la configuración en los objetivos de las instrucciones de movimiento de ejes. Si emplean las instrucciones, el control de las configuraciones se aplica desde la siguiente instrucción de movimiento ejecutada y hasta que se ejecute una nueva instrucción de control de configuración.

Mediante el control de las configuraciones se fuerza al robot a emplear las configuraciones especificadas para alcanzar los objetivos, deteniéndose la ejecución del programa en el momento que aparezca el riesgo de que la configuración programada para el siguiente objetivo no pueda alcanzarse desde la posición actual.

De esta forma se obtienen ciclos y movimientos predecibles, aunque en algunas situaciones, como por ejemplo cuando el robot se mueve hasta un objetivo desde una posición desconocida, el uso del control de configuraciones puede causar un problema de alcance. Si se emplea el control de configuraciones se debe asignar una configuración alcanzable para cada uno de los objetivos.

Si se desactiva el control de las configuraciones mediante las **ConfL \Off** y/o **ConfJ \Off**, las configuraciones especificadas en los objetivos no se tendrán en cuenta durante la ejecución, de forma que el robot utilizará la configuración más cercana a su configuración actual para alcanzar el objetivo. Esto puede causar configuraciones diferentes cada vez que se ejecuta un ciclo del programa, es decir, cada vez el robot regresa a la posición inicial tras completar todas las trayectorias de un ciclo, puede elegir una configuración diferente de la especificada.

En el caso de instrucciones de movimiento lineal o circular, esto puede derivar en que los ejes del robot se acercan cada vez más a sus límites de eje, provocando que el robot sea incapaz de alcanzar los objetivos.

En el caso de instrucciones de movimiento de ejes, esto puede derivar en movimientos de barrido impredecibles por parte del robot.

3.1.17 Plantillas de instrucciones

RobotStudio ofrece un conjunto de instrucciones, tanto de movimiento como de acción, predeterminadas al crear una nueva estación. Estas instrucciones se encuentran dentro de la plantilla de instrucciones, especificándose para cada una de ellas los valores de sus argumentos.

En la **Barra de estado** se muestra la instrucción de movimiento activa de la plantilla de instrucciones, junto con sus argumentos. La instrucción de movimiento y cada uno de los argumentos que se muestran se pueden modificar pulsando sobre ellos y seleccionando el

nuevo valor del argumento, de forma que ese será el tipo de instrucción de movimiento y sus argumentos en la próxima instrucción de movimiento que se programe. (Figura 3.123)

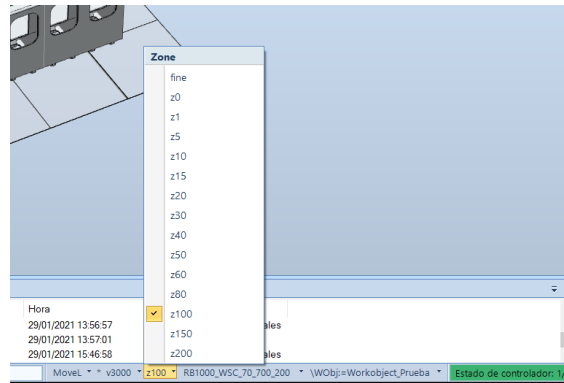


Figura 3.123 - Modificación de los argumentos de la instrucción de movimiento en la Barra de estado

Para acceder a la plantilla de instrucciones se debe pulsar sobre la instrucción de movimiento activa en la **Barra de estado** y seleccionar la opción **Editar plantillas de instrucciones**, como se muestra en la siguiente Figura 3.124.

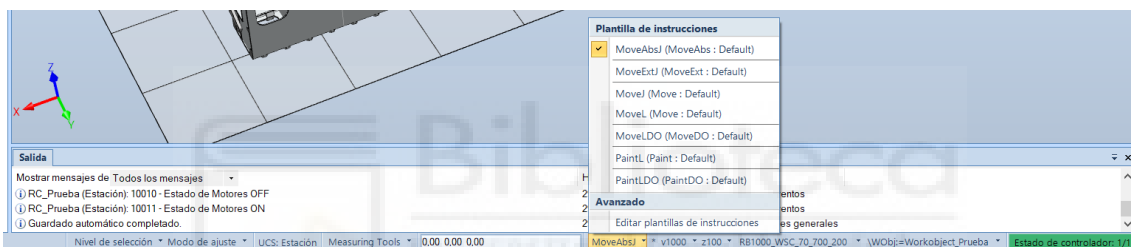


Figura 3.124 - Editar plantilla de instrucciones

La siguiente Figura 3.125 se muestra la plantilla de instrucciones con las instrucciones predeterminadas para cualquier estación.

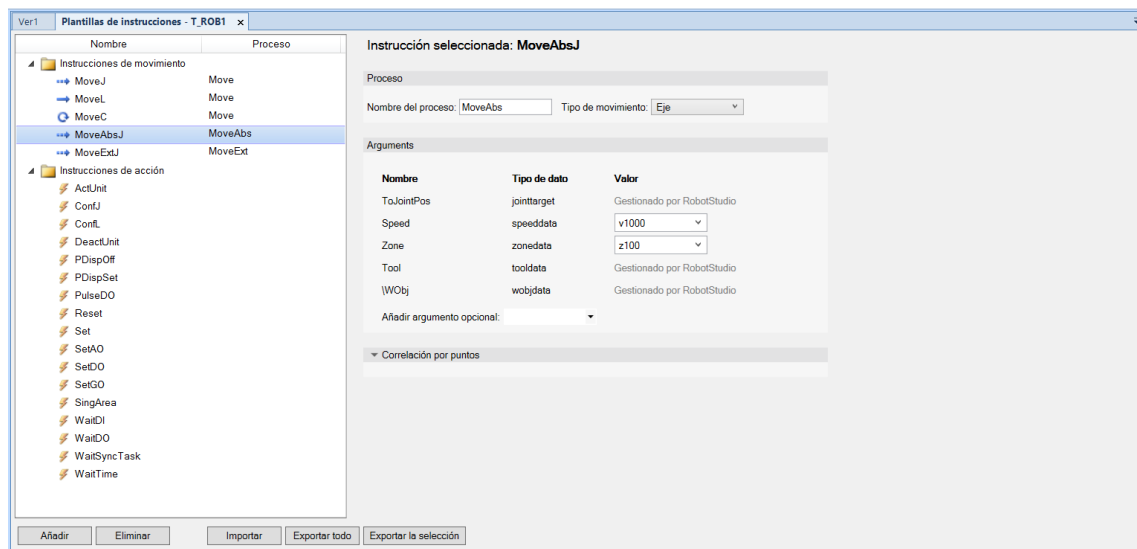


Figura 3.125 - Plantilla de instrucciones predeterminada

En la siguiente Figura 3.126 se muestra la plantilla de instrucciones con las instrucciones predeterminadas para un controlador con la opción **Conveyor tracking** y el **Painting PowerPac**, empleadas ambas en el presente proyecto.

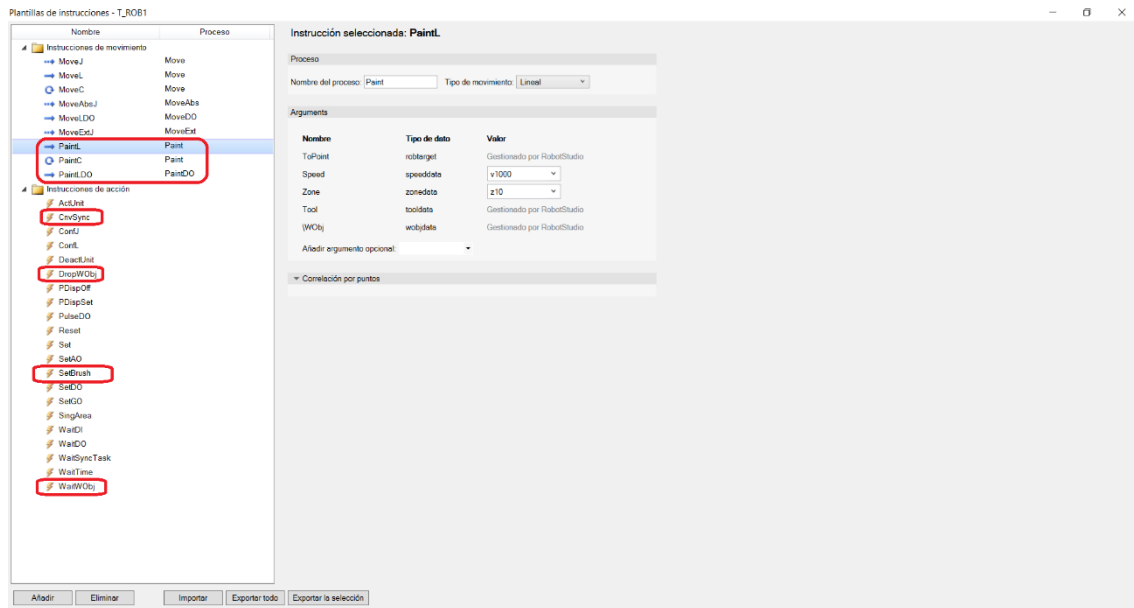


Figura 3.126 - Plantilla de instrucciones con la opción Conveyor Tracking activada en el controlador y el Painting

PowerPac

La mayoría de los argumentos predeterminados de las instrucciones pueden ser modificados, seleccionándolos y cambiando su valor, otros quedan gestionados por RobotStudio. Si se modifican dichos argumentos para una instrucción en la plantilla, la próxima vez que se programe la instrucción aparecerá con esos valores de argumentos. Además pueden añadirse otros argumentos opcionales de la instrucción. En el caso de las instrucciones de movimiento, puede variarse el tipo de movimiento, aunque esto no es aconsejable.

Únicamente podrán ser programadas gráficamente y visualizadas en las trayectorias las instrucciones presentes en la plantilla de instrucciones, por lo que si se requieren otras instrucciones, además de las predefinidas, se deberán añadir a la plantilla de instrucciones.

Para ello, se pulsa sobre la opción **Añadir**, de forma que aparece el cuadro de dialogo **Añadir instrucción**, mostrado en la Figura 3.127, se pulsa sobre la sección **Instrucción**, de forma que aparece un desplegable en el cual se debe seleccionar la instrucción a añadir a la plantilla de instrucciones y se pulsa **Añadir**.

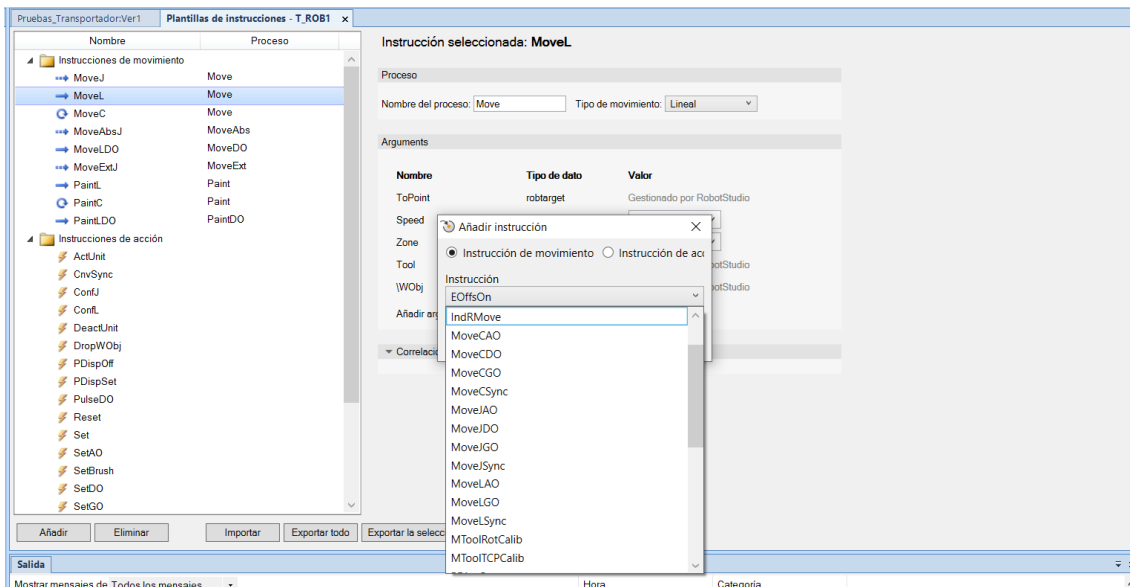


Figura 3.127 - Cuadro de diálogo Añadir instrucción

Una vez se han añadido todas las instrucciones deseadas a la plantilla, esta se puede guardar para importarla en otro controlador.

Existen dos opciones para realizar el guardado de la plantilla. La primera de ellas se lleva a cabo pulsando en la opción **Exportar todo**, de forma que se guarda la plantilla completa, es decir, la plantilla con todas las instrucciones que la componen. La segunda opción se lleva a cabo seleccionando las instrucciones de la plantilla que se desean guardar y pulsando **Exportar la selección**, de forma que solo se guardan como plantilla las instrucciones seleccionadas. Esta segunda opción se emplea para guardar plantillas formadas por grupos de instrucciones de aplicaciones concretas, como por ejemplo pintado, soldadura, dispensación, etc. Los archivos de plantilla se guardan en formato XML.

Para importar un plantilla simplemente se debe pulsar sobre opción **Importar**, de forma que se abre el explorador de archivos de Windows para buscar la plantilla a importar.

3.1.18 Sincronización del controlador virtual con RAPID

Una vez se han creado los objetivos y trayectorias requeridas para realizar la tarea, para que el controlador del robot pueda ejecutarlas estas deben ser sincronizadas con RAPID, es decir, se deben trasladar los objetivos, objetos de trabajo, herramientas y trayectorias, con todas sus instrucciones, programadas gráficamente a un programa de lenguaje de programación RAPID, creándose así el código del programa de RAPID a partir de los datos de la estación.

La sincronización con RAPID también se emplea para actualizar los posibles cambios realizados en los objetivos, objetos de trabajo, herramientas y/o trayectorias que ya se habían sincronizado anteriormente. Esta operación se debe realizar antes de ejecutar una simulación, guardar un

programa en archivo o de realizar una copia de seguridad de un controlador virtual si se quiere que los cambios realizados en la programación de la tarea queden reflejados.

Para llevar a cabo la sincronización con RAPID se deben seguir los siguientes pasos.

El primer paso es acceder al cuadro de dialogo **Sincronizar con RAPID**, mostrado en la siguiente Figura 3.128, para lo cual se tienen 4 opciones posibles. A la primera de ellas se accede seleccionando la pestaña **Posición inicial**, grupo **Controlador**, **Sincronizar** y **Sincronizar con RAPID**. Si se emplea esta opción se deberá seleccionar previamente, mediante el botón izquierdo del ratón, el controlador en el cual se sincronizarán todos los datos de la estación anteriormente mencionados, a través de la pestaña **Posición inicial**, en el navegador **Trayectorias y puntos**. Si solo se tiene un controlador esto no será necesario, ya que es el único controlador en el que sincronizar.

A la segunda opción se accede seleccionando la pestaña **RAPID**, grupo **Acceso**, herramienta **Sincronizar** y **Sincronizar con RAPID**. Para que sea posible seleccionar la herramienta **Sincronizar** previamente se deberá seleccionar en el navegador **Controlador**, mediante el botón izquierdo del ratón, el controlador que se quiere sincronizar.

Otra opción es seleccionar en la pestaña **Posición inicial**, navegador **Trayectorias y puntos**, el controlador en cuestión mediante el botón derecho del ratón y seleccionar la opción **Sincronizar con RAPID**.

Por último se puede acceder a dicho cuadro de dialogo seleccionando en la pestaña **Posición inicial**, navegador **Trayectorias y puntos**, dentro del controlador y la tarea, la trayectoria que se quiere sincronizar mediante el botón derecho del ratón y seleccionando la opción **Sincronizar con RAPID**.

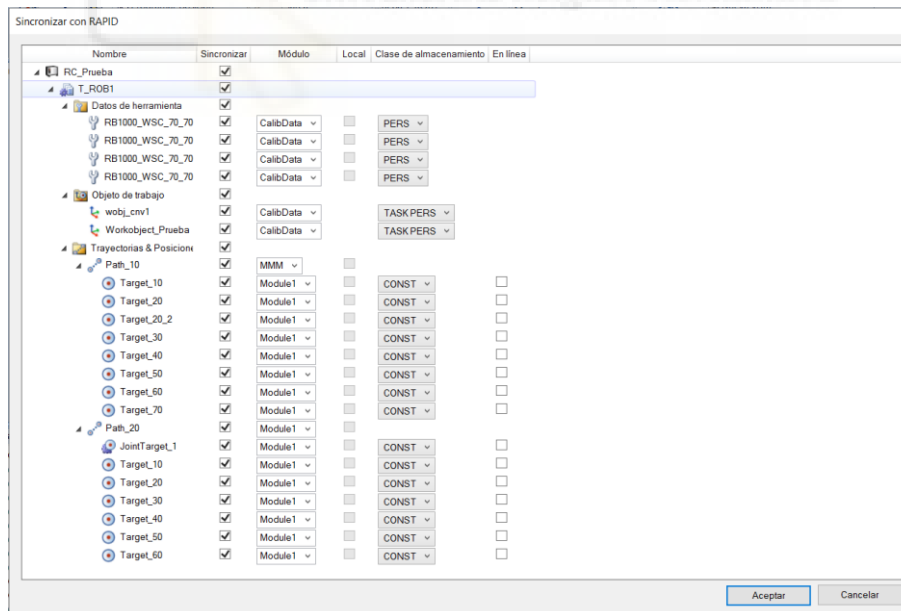


Figura 3.128 - Cuadro de dialogo Sincronizar con RAPID

Dentro de este cuadro de dialogo se deben seleccionar los elementos que se quieren sincronizar. Para cada elemento a sincronizar se puede elegir el módulo del controlador virtual en el que se sincronizará. Se puede elegir un módulo existente o escribir el nombre de un nuevo módulo, de

forma que al sincronizar se creará un nuevo módulo con ese nombre. Si es la primera vez que se sincroniza la estación y no existe ningún módulo se creará uno automáticamente.

También se puede especificar si los elementos se sincronizan de forma que se creen datos y procedimientos locales. Una vez se han sincronizado los elementos como locales o no locales (globales) al volver a realizar la sincronización este aspecto no se puede variar, se deberá hacer a través del editor de código de RAPID.

En lo referido a los datos de herramienta y los objetos de trabajo también puede especificarse el tipo de almacenamiento de la declaración de datos.

También puede especificarse si la declaración de datos debe declararse como en línea o con nombre. Si se selecciona en línea, los datos se declaran en la propia instrucción.

Una vez se tiene todo especificado se pulsa **Aceptar** de forma que se llevará a cabo la sincronización. Tras realizarse se mostrará en el editor de lenguaje de RAPID el código que hace referencia a los elementos sincronizados. También se muestra en la pestaña **RAPID**, navegador **Controlador**, dentro del controlador, la tarea y el módulo especificados, las diferentes trayectorias sincronizadas, ahora denominadas procedimientos. (Figura 3.129)

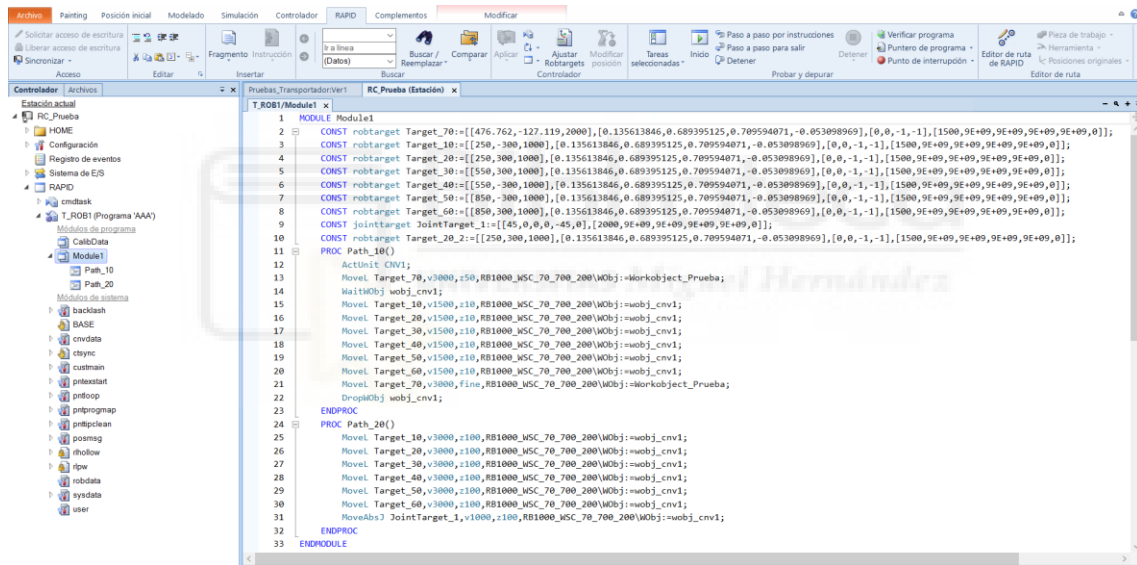


Figura 3.129 - Ejemplo de visualización de objetivos y trayectorias sincronizadas con RAPID

3.1.19 Lógica de estación

La lógica de estación es una característica de RobotStudio muy importante para la programación de la estación robótica. Permite especificar las conexiones y enlazamientos entre controladores de la estación, entre controladores y componentes inteligentes y/o entre componentes inteligentes. Las conexiones pueden ser entre señales de E/S de la propia estación, de los controladores y de los componentes inteligentes, y los enlazamientos entre propiedades de los componentes inteligentes.

RobotStudio ofrece dos métodos para crear estas conexiones y enlazamientos en la lógica de estación. El primero es crearlas de forma gráfica seleccionando la señal o propiedad en los bloques que representan los controladores y los componentes inteligentes de la estación y conectándola a otro bloque. El segundo método es crearlas especificando en un cuadro de dialogo el elemento y la señal o propiedad de origen y el elemento y la señal o propiedad de destino. El primero método ofrece una forma sencilla y visual de crear las conexiones, facilitando así la programación de la estación.

Además la lógica de estación permite la creación y modificación de componentes inteligentes. Los componentes inteligentes son objetos de RobotStudio, con o sin representación gráfica, con propiedades y lógica integradas que permiten simular componentes que no forman parte del controlador virtual. Los componentes inteligentes se detallan en el siguiente apartado.

Par acceder a la lógica de estación se debe seleccionar la pestaña **Simulación**, grupo **Configurar**, herramienta **Lógica de estación**, abriéndose así la venta **Lógica de estación**.

Esta ventana se compone a su vez por las siguientes cuatro pestañas.

Pestaña Componer

Esta pestaña a su vez se compone de tres apartados, como se muestra en la Figura 3.130.

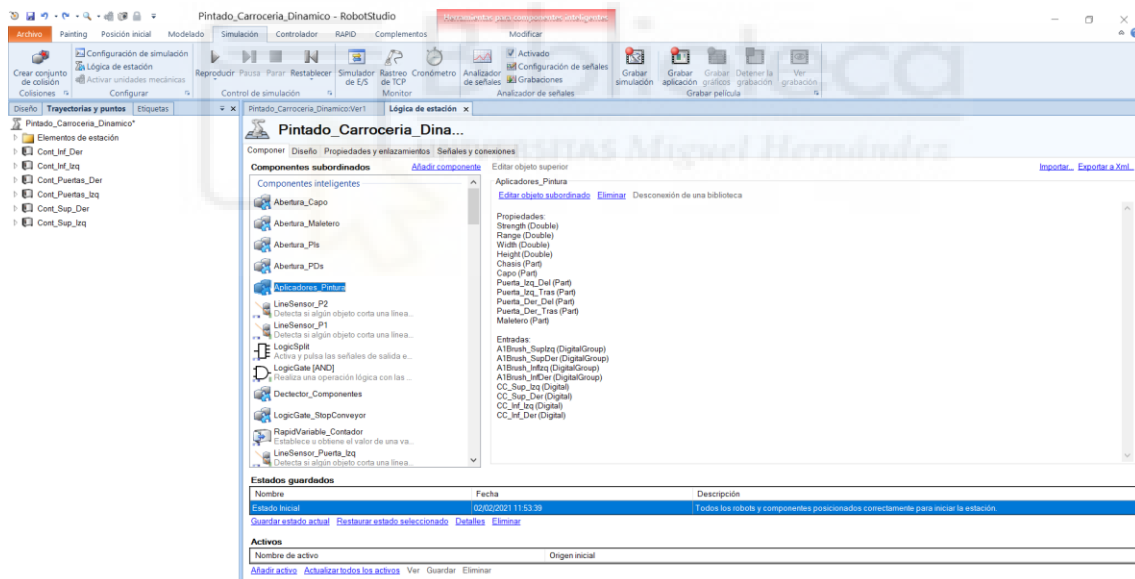


Figura 3.130 - Pestaña Componer de la Lógica de estación

El primero de ellos se denomina **Componentes subordinados** y en él se enumeran, en forma de lista, todos los componentes inteligentes y todos los componentes gráficos que forman parte de la estación. Seleccionando cualquier componente inteligente de esta lista, mediante el botón izquierdo del ratón, aparecerán sus propiedades y sus entradas y salidas a la derecha de la lista. En este apartado se tienen las siguientes opciones:

- **Añadir componente.** Mediante esta opción se pueden añadir a la estación componentes inteligentes predefinidos que ofrece RobotStudio, componentes inteligentes vacíos, un archivo de biblioteca de RobotStudio o un archivo que contenga

una geometría CAD. Al seleccionar esta opción aparece el desplegable mostrado en la Figura 3.131, en el cual se debe seleccionar el elemento a añadir a la estación. Los componentes inteligentes predefinidos están agrupados según la función que desempeñan.

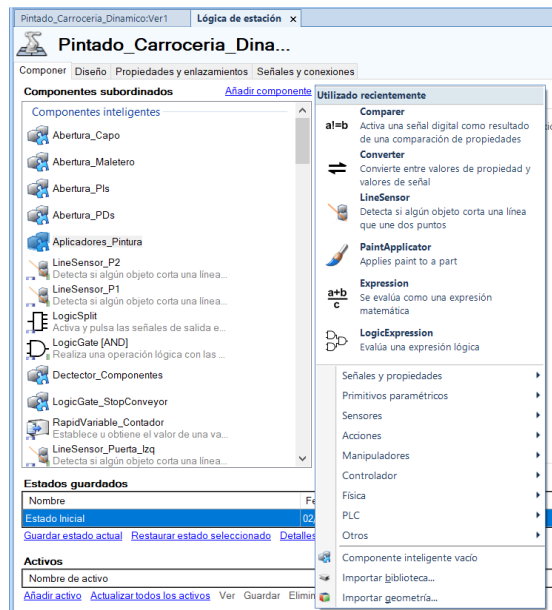


Figura 3.131 - Opciones de la herramienta Añadir componente

- **Exportar a XML.** Mediante esta opción se puede exportar y guardar en un archivo .rsxml el conjunto de componentes inteligentes definidos en la lógica de estación junto con sus propiedades.
- **Importar.** Permite importar un archivo de lógica de estación de tipo .rsxml previamente exportado mediante la opción anteriormente descrita.
- **Editar objeto superior.** Si se está editando un componente inteligente formado por otros componentes, mediante esta opción se cambia el editor al objeto superior, es decir, al componente inteligente que los contiene.

Por otro lado, si se selecciona mediante el botón derecho del ratón cualquier componente inteligente de los que aparecen en este apartado se mostrarán algunas de las siguientes opciones, dependiendo del tipo de componente.

- **Editar.** Si se selecciona un componente inteligente formado por otros componentes inteligentes dentro del mismo, esta opción permite cambiar el editor al nivel dentro del componente para poder editar los componentes inteligentes que lo forman, las propiedades, las señales, las conexiones y enlazamientos, etc.
- **Eliminar.** Mediante esta opción se elimina el componente seleccionado, ya sea un componente inteligente o un componente gráfico de la estación.
- **Mostrar en el navegador.** Esta opción indica si un componente, inteligente o gráfico, debe mostrarse en el navegador diseño.

- **Propiedades.** Si se ha seleccionado un componente inteligente, mediante esta opción se abre el panel **Propiedades** donde se pueden modificar las propiedades del componente inteligente.

El segundo apartado se denomina **Estados guardados** y permite guardar y restaurar posteriormente, mediante alguno de los estados guardados, el estado de la estación al completo o de alguno de sus componentes en particular. El estado de cualquier componente puede entenderse como su ubicación y sus propiedades y características en un momento determinado. En este apartado se tienen las siguientes opciones:

- **Guardar estado actual.** Mediante esta opción se abre el cuadro de dialogo **Guardar estado actual**, mostrado en la Figura 3.132, en el cual se debe especificar el nombre y descripción del estado, los componentes de la estación de los que se desean guardar su estado, pudiéndose seleccionar la estación completa, y los valores de estado que se desean almacenar (ubicación, valores de eje, valores de señales E/S, propiedades, etc.).

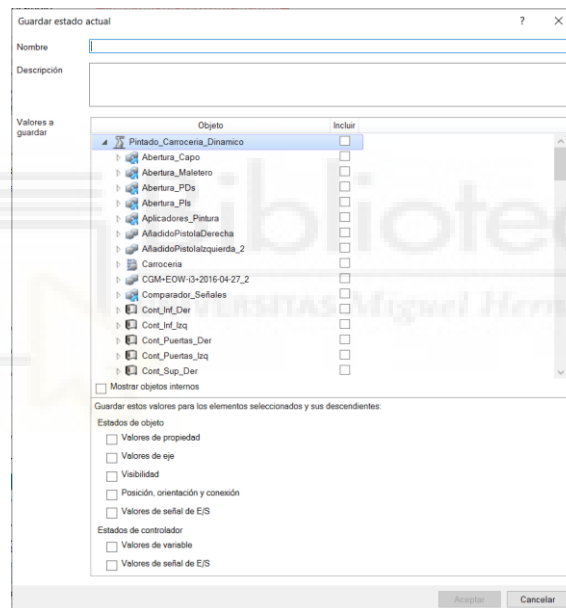


Figura 3.132 - Cuadro de dialogo Guardar estado actual

- **Restaurar estado seleccionado.** Mediante esta opción se devuelve el componente o componentes especificados en el estado a los valores de estado guardados.
- **Detalles.** Mediante esta opción se abre una ventana que muestra todos los detalles del estado guardado seleccionado.
- **Eliminar.** Mediante esta opción se elimina el estado seleccionado.

El tercer apartado se denomina **Activos** y permite examinar, seleccionar y guardar cualquier archivo como un activo.

Pestaña Diseño

En esta pestaña se muestran de forma gráfica, mediante representación de bloques, los componentes inteligentes y los controladores de robot presentes en la estación, como se muestra en la Figura 3.133. En cada uno de los bloques de componentes inteligentes se muestran sus propiedades y las entradas y salidas al bloque. En los bloques que representan los controladores virtuales se muestran las señales de E/S que se especifiquen. En esta ventana gráfica también se muestran las señales de entrada y salida de la estación, que no dependen de ningún controlador.

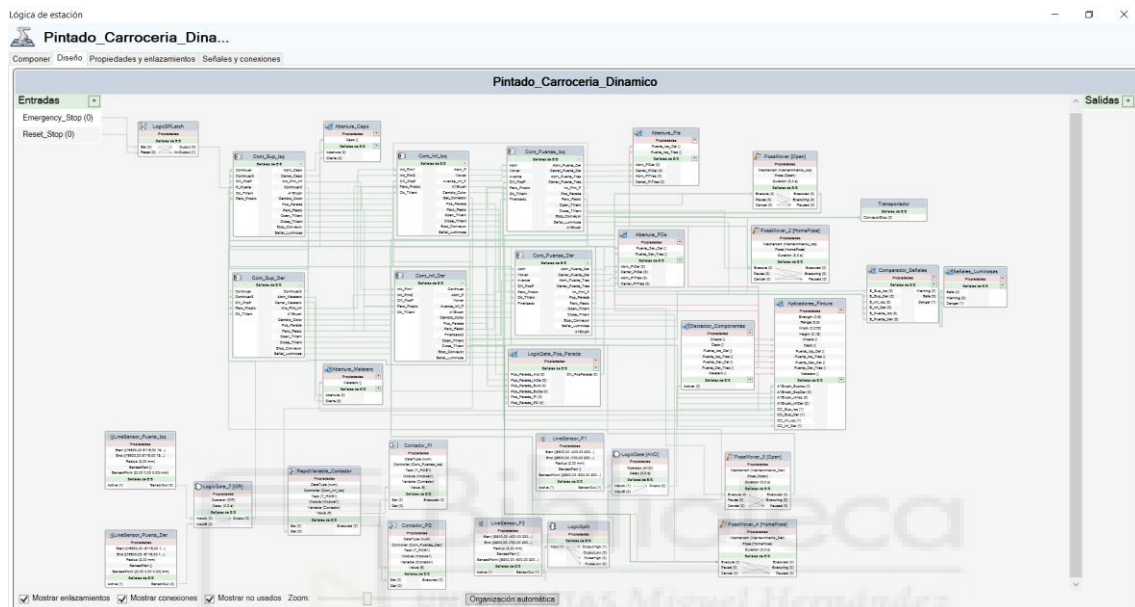


Figura 3.133 - Pestaña Diseño de la Lógica de estación

Como se ha indicado anteriormente esta ventana gráfica ofrece una de las opciones para crear de forma gráfica las conexiones y enlazamientos entre las señales de E/S y propiedades, respectivamente, de los diferentes componentes inteligentes y controladores.

Para crear una conexión o enlazamiento se debe seleccionar mediante el botón izquierdo del ratón la señal de E/S o propiedad a conectar (origen) y sin soltar el ratón, arrastrar hasta la señal o propiedad a la que se conecta (destino). Una vez posicionado el ratón encima de la señal de E/S o propiedad de destino se suelta el botón del ratón, de forma que se crea una línea entre las dos señales de E/S o las dos propiedades que indica de forma gráfica la conexión o enlazamiento. Los enlazamientos entre propiedades se muestran con una línea de color rojo y las conexiones entre señales de E/S se muestran con una línea de color verde, como puede verse en la anterior Figura 3.133.

Este método permite observar gráficamente todas las conexiones y enlazamientos de la estación al mismo tiempo, facilitando la creación de las mismas, ya que es un método visual muy intuitivo, y evitando posibles errores o ausencias en las conexiones o enlazamientos.

En esta pestaña también se puede crear señales de E/S de estación o si se está dentro de un componente señales de E/S del componente. En la parte superior izquierda y derecha de la ventana gráfica se muestran los listados de señales de entrada y salida, respectivamente. Al lado

de estos listados aparece un símbolo +, de forma que al pulsarlo abre el cuadro de dialogo **Añadir señales de E/S** mediante el cual se pueden crear dichas señales.

Otro aspecto a señalar de esta pestaña es que seleccionando mediante el botón derecho del ratón cualquier espacio vacío de la ventana gráfica aparece el desplegable de la Figura 3.131, del mismo modo que mediante la opción **Añadir componente** de la pestaña **Componer**, que permite añadir componentes a la estación.

Por último, en la parte final de la ventana gráfica se tienen las siguientes cuatro herramientas.

- **Mostrar enlazamientos.** Esta herramienta indica si se muestran o no los enlazamientos entre las propiedades de los componentes inteligentes.
- **Mostrar conexiones.** Esta herramienta indica si se muestran o no las conexiones entre las señales de E/S.
- **Mostrar no usados.** Esta herramienta indica si se muestran o no los componentes inteligentes añadidos pero que no se están empleando.
- **Zoom.** Esta herramienta permite acercar o alejar el zoom de la ventana gráfica para ver con mayor o menor detalle una zona de la misma.

Pestaña Propiedades y enlazamientos

En esta pestaña se muestran textualmente, en forma de tabla, los enlazamientos entre propiedades de los componentes inteligentes, como se muestra en la Figura 3.134. Esta tabla está formada por los siguientes campos:

- **Componentes de Origen.** Se refiere a los componentes que contienen las propiedades a conectar.
- **Propiedades de Origen.** Se refiere a las propiedades a conectar.
- **Componentes de Destino.** Se refiere a los componentes que contienen las propiedades o señales en las que se conectan las propiedades de origen.
- **Propiedades o señales de Destino.** Se refiere a las propiedades o señales en las que se conectan las propiedades de origen.

Cada fila o registro de la tabla hace referencia a un enlazamiento de propiedades.

Lógica de estación

Pintado_Carroceria_Dina...

Componentes Diseño Propiedades y enlaces Señales y conexiones

Enlazamientos de propiedad

Objeto de origen	Propiedad de origen	Objeto de destino	Señal o propiedad de destino
Detectador_Componentes	Chasis	Aplicadores_Pintura	Chasis
Detectador_Componentes	Capo	Aplicadores_Pintura	Capo
Detectador_Componentes	Puerta_Izq_Del	Aplicadores_Pintura	Puerta_Izq_Del
Detectador_Componentes	Puerta_Izq_Tras	Aplicadores_Pintura	Puerta_Izq_Tras
Detectador_Componentes	Puerta_Der_Del	Aplicadores_Pintura	Puerta_Der_Del
Detectador_Componentes	Puerta_Der_Tras	Aplicadores_Pintura	Puerta_Der_Tras
Detectador_Componentes	Maletero	Aplicadores_Pintura	Maletero
Detectador_Componentes	Capo	Abertura_Capo	Capo
Detectador_Componentes	Maletero	Abertura_Maletero	Maletero
Detectador_Componentes	Puerta_Izq_Del	Abertura_Pis	Puerta_Izq_Del
Detectador_Componentes	Puerta_Izq_Tras	Abertura_Pis	Puerta_Izq_Tras
Detectador_Componentes	Puerta_Der_Del	Abertura_PD	Puerta_Der_Del
Detectador_Componentes	Puerta_Der_Tras	Abertura_PD	Puerta_Der_Tras
RapidoVariable_Contador	Value	Contador_PI	Value
RapidoVariable_Contador	Value	Contador_PD	Value

Añade enlace Añadir enlace de expresión Editar Eliminar

Figura 3.134 - Pestaña Propiedades y enlazamientos

En esta pestaña se tienen las siguientes opciones:

- **Añadir enlaceamiento.** Al seleccionar esta opción se abre el cuadro de dialogo **Añadir enlaceamiento**, mediante el cual se puede crear un enlaceamiento de propiedades. En él se debe especificar el componente de origen, la propiedad de origen, el componente de destino y la propiedad o señal de destino. También debe especificarse si se permite el enlaceamiento cíclico, mediante el cual se puede definir dos veces la propiedad de destino en el mismo contexto, lo que de otro modo genera un error.
- **Añadir enlaceamiento de expresión.** Al seleccionar esta opción se abre el cuadro de dialogo **Añadir enlaceamiento de expresión**, mediante el cual se puede crear un enlaceamiento de propiedades en el que la propiedad de origen es una expresión matemática especificada. En él se debe especificar la expresión matemática que actúa como propiedad de origen, el componente de destino y la propiedad de destino, solo permitiéndose propiedades numéricas.
- **Editar.** Permite editar los componentes y propiedades de un enlaceamiento de propiedades ya creado.
- **Eliminar.** Permite eliminar un enlaceamiento de propiedades ya creado.

Los enlaceamientos creados de esta forma también se representan gráficamente en la pestaña **diseño**, igual que los creados gráficamente.

Pestaña Señales y conexiones

Esta pestaña se divide en dos apartados, **Señales de E/S** y **Conexiones de E/S**. En el primero se muestra de forma textual, en forma de tabla, las señales de entrada y salida de la estación o si

se está dentro de un componente inteligente, las señales de entrada y salida de dicho componente, como se muestra en la Figura 3.135.

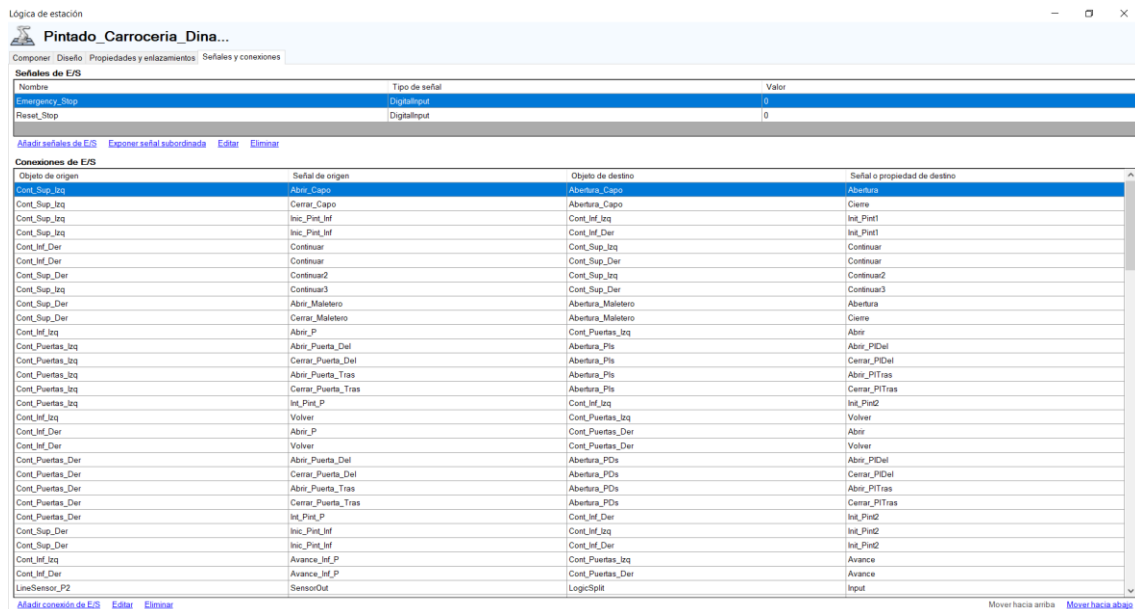


Figura 3.135 - Pestaña Señales y conexiones

Esta tabla está formada por los siguientes campos:

- **Nombre.** Indica el identificador de la señal.
- **Tipo de señal.** Especifica el tipo de señal, digital de entrada o salida, analógica de entrada o salida o digital de grupo de entrada o salida.
- **Valor.** Indica el valor de la señal.

Cada fila o registro de la tabla hace referencia a una señal de E/S.

En este apartado se tienen además las siguientes opciones:

- **Añadir señales de E/S.** Al seleccionar esta opción se abre el cuadro de dialogo **Añadir señales de E/S**, mediante el cual se puede crear una o varias señales de entrada o salida de la estación o dentro de un componente inteligente. En él se debe especificar el tipo de señal a crear, su nombre o identificador, el valor de la señal, si este valor se restablece de forma automática si se cambia durante la ejecución, el número de señales, si la propiedad se crea de forma oculta, es decir, no se mostrará en el **Editor de propiedades** y el **Simulador de E/S**, y por último se puede especificar si la señal es de solo lectura, es decir, no podrá ser modificada tras su creación en el **Editor de propiedades** y el **Simulador de E/S**.

Si se especifica la creación de más de una señal, se añaden a sus identificadores sufijos numéricos especificados en **Índice de inicio** y **Paso**.

En el caso de señales analógicas también debe especificarse su valor mínimo y su valor máximo.

- **Exponer señal subordinada.** Al seleccionar esta opción se abre el cuadro de dialogo **Exponer señal subordinada**, mediante el cual se puede añadir una nueva señal de E/S que está conectada a una señal de un objeto subordinado. En él deben especificarse el nombre de la señal, el objeto subordinado y la señal subordinada.

- **Editar.** Esta opción permite editar una señal ya creada. Solo es posible modificar el nombre de las señales y sus valores, estando todos los demás aspectos bloqueados.
- **Eliminar.** Esta opción permite eliminar una señal ya creada.

El segundo apartado de esta pestaña, **Conexiones de E/S**, muestra de forma textual, en forma de tabla, las conexiones entre las señales de E/S entre controladores de la estación, entre controladores y componentes inteligentes y/o entre componentes inteligentes. Esta tabla está formada por los siguientes campos:

- **Componentes de Origen.** Se refiere a los componentes que contienen las señales a conectar.
- **Señales de Origen.** Se refiere a las señales a conectar.
- **Componentes de Destino.** Se refiere a los componentes que contienen las señales o propiedades en las que se conectan las señales de origen.
- **Señales o propiedades de Destino.** Se refiere a las señales o propiedades en las que se conectan las señales de origen.

Cada fila o registro de la tabla hace referencia a una conexión de señales.

En este apartado se tienen además las siguientes opciones:

- **Añadir conexión de E/S.** Al seleccionar esta opción se abre el cuadro de dialogo **Añadir conexión de E/S**, mediante el cual se puede crear una conexión de señales. En él se debe especificar el componente de origen, la señal de origen, el componente de destino y la señal o propiedad de destino. También debe especificarse si se permite la conexión cíclica, mediante el cual se puede definir dos veces la señal de destino en el mismo contexto, lo que de otro modo genera un error.
- **Editar.** Esta opción permite editar una conexión entre señales ya creada.
- **Eliminar.** Esta opción permite eliminar una conexión entre señales ya creada.

Las conexiones creadas de esta forma también se representan gráficamente en la pestaña diseño, igual que las creadas gráficamente.

3.1.20 Componentes inteligentes

Los componentes inteligentes son objetos de RobotStudio con propiedades y lógica integradas, que pueden tener o no representación gráfica, empleados para simular componentes que no forman parte del controlador virtual. Estos componentes reaccionan de forma automática a determinados eventos durante la ejecución del programa de robot, como por ejemplo pasos de

tiempo de simulación, cambios en los valores de las señales de E/S o las propiedades, etc., de modo que se ejecutan y realizan la función específica para la que están destinados.

Existen dos tipos de componentes inteligentes, los componentes inteligentes básicos y los componentes inteligentes compuestos. RobotStudio proporciona 72 componentes inteligentes básicos predefinidos, destinados a realizar diferentes tareas relativamente sencillas, como por ejemplo movimientos sencillos de componentes, cálculos aritméticos, detección mediante sensores, tratamiento de señales, etc. Estos se agrupan según la función que desempeñan, como se muestra en la Figura 3.131, del apartado anterior. En este apartado se expondrán los componentes inteligentes básicos empleado en el presente proyecto.

Los componentes inteligentes compuestos son componentes inteligentes constituidos por diferentes componentes inteligentes básicos interconectados, con el fin de poder llevar a cabo tareas de mayor complejidad y específicas para la estación robótica que se programa.

El primer paso para crear un componente inteligente compuesto es crear un componente inteligente vacío, que posteriormente se editará para añadir los componentes inteligentes básicos que corresponda y sus conexiones y enlazamientos. Para crearlo existen tres opciones diferentes.

La primera de ellas se lleva a cabo seleccionando pestaña **Modelado**, grupo **Crear**, herramienta **Componente inteligente**. Este componente inteligente vacío se creará en el editor superior de la lógica de estación, no dentro de otro componente inteligente.

La segunda opción se lleva a cabo dentro de la lógica de estación, seleccionando en la pestaña **Componer** la opción **Añadir componente** y en el desplegable la herramienta **Componente inteligente vacío**.

La tercera también se lleva a cabo dentro de la lógica de estación pero en la pestaña **Diseño**, seleccionando mediante el botón derecho del ratón un espacio vacío de la ventana gráfica y en el desplegable la herramienta **Componente inteligente vacío**. La segunda y tercera opciones pueden llevarse a cabo en la lógica de la estación completa o dentro de la lógica de otro componente inteligente compuesto.

Una vez creado el componente inteligente se irán añadiendo componentes inteligentes básicos u otros componentes inteligentes compuestos dentro de este, se crearán señales de E/S propias, se realizarán las conexiones entre señales y los enlazamientos entre propiedades requeridos de la misma forma que se ha expuesto en el apartado anterior [3.1.19 Lógica de estación](#).

Los componentes inteligentes compuestos pueden guardarse como biblioteca de RobotStudio, de forma que puedan ser importados posteriormente en otras estaciones, seleccionándolos con el botón derecho del ratón, ya sea en el navegador **Diseño** o en la lógica de estación, y seleccionando la herramienta **Guardar como biblioteca**.

Para acceder a la lógica de un componente inteligente compuesto y así poder editarlo existen dos métodos. El primero de ellos es seleccionarlo, mediante el botón derecho del ratón, en el navegador **Diseño** o dentro de la lógica de estación y seleccionar la herramienta **Editar**. La segunda opción es seleccionarlo, mediante el botón izquierdo del ratón, en la lógica de estación, en la pestaña **Componer** y seleccionar la opción **Editar objeto subordinado**, que aparece en la descripción del mismo.

Todos los componentes inteligentes pueden protegerse para ocultar su estructura interna y evitar su edición. Para ello se selecciona el componente a proteger con el botón derecho del ratón, ya sea en el navegador **Diseño** o la lógica de estación, y se selecciona la herramienta **Protegido**. En el cuadro de dialogo que aparece puede establecerse una contraseña que se solicitará para desbloquearlo y así permitir su edición.

Si se quieren modificar las propiedades de un componente inteligente, básico o compuesto, se debe seleccionar el mismo, mediante el botón derecho del ratón, en el navegador **Diseño** o en la lógica de estación y seleccionar la opción **Propiedades**. De esta forma se abrirá el panel **Propiedades** en el cual podrán especificarse los nuevos valores de las mismas.

Los componentes inteligentes básicos empleados en el presente proyecto se exponen a continuación, agrupados según la función a la que están destinados.

Señales y propiedades

LogicGate

Este componente inteligente, mostrado en la siguiente Figura 3.136, realiza la función de uno de los tipos de puerta lógica que pueden especificarse en la propiedad **Operator**.

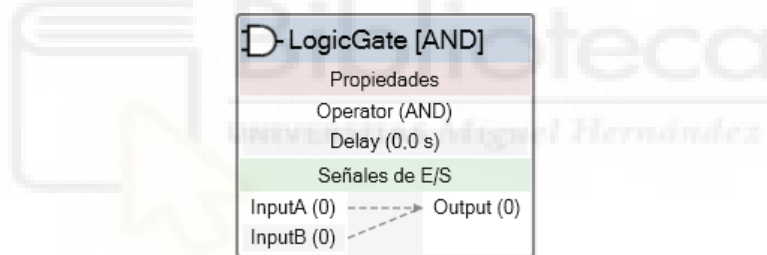


Figura 3.136 - Componente inteligente LogicGate

La señal de salida **Output** se activa en función de los valores de las dos señales de entrada **InputA** e **InputB** según el operador lógico **Operator**, con el retraso especificado en **Delay**. Este componente inteligente posee las siguientes señales y propiedades:

Señales

- **InputA**. Esta señal especifica la primera señal digital de entrada del componente.
- **InputB**. Esta señal especifica la segunda señal digital de entrada del componente.
- **Output**. Esta señal digital de salida especifica el resultado de la operación lógica.

Propiedades

- **Operator**. Esta propiedad especifica el operador lógico. Los operadores lógicos que pueden seleccionarse son AND, OR, XOR, NOT y NOP.
- **Delay**. Especifica el tiempo, en segundos, que se retrasará la activación de la señal Output.

LogicSplit

Este componente inteligente, mostrado en la siguiente Figura 3.137, cambia el valor de la salida **OutputHigh** al mismo valor que la entrada **Input** y el valor de la salida **OutputLow** al valor inverso de la entrada **Input**. Además, **PulseHigh** envía un impulso cuando **Input** tiene valor 1 y **PulseLow** envía un impulso cuando cambia a valor 0.

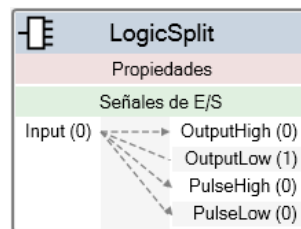


Figura 3.137 - Componente inteligente LogicSplit

Este componente inteligente posee las siguientes señales:

- **Input**. Esta señal especifica la señal digital de entrada del componente.
- **OutputHigh**. Esta señal digital de salida cambia a 1 si **Input** es igual a 1.
- **OutputLow**. Esta señal digital de salida cambia a 1 si **Input** es igual a 0.
- **PulseHigh**. Esta señal envía un impulso cuando la entrada **Input** cambia su valor a 1.
- **PulseLow**. Esta señal envía un impulso cuando la entrada **Input** cambia su valor a 0.

LogicSRLatch

Este componente inteligente, mostrado en la siguiente Figura 3.138, cambia el valor de la salida **Output** a 1 y el de la salida **InvOutput** a 0 cuando la entrada **Set** es igual a 1, manteniéndose estos valores de las salidas aunque el valor de **Set** pase a 0. Si el valor de **Reset** pasa a ser 1, el valor de **Output** cambia a 0 y el de **InvOutput** a 1, manteniéndose estos valores aunque **Reset** cambie de valor.

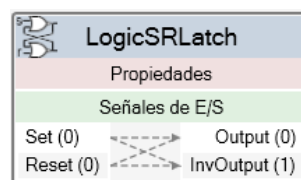


Figura 3.138 - Componente inteligente LogicSRLatch

Este componente inteligente posee las siguientes señales:

- **Set**. Si esta señal digital de entrada cambia a 1, **Output** cambia a 1 e **InvOutput** a 0.
- **Reset**. Si esta señal digital de entrada cambia a 1, **Output** cambia a 0 e **InvOutput** a 1.
- **Output**. Señal digital de salida cuyo valor depende de **Set** o **Reset**.
- **InvOutput**. Señal digital de salida cuyo valor depende de **Set** o **Reset**, teniendo siempre el valor inverso a **Output**.

Comparer

Este componente inteligente, mostrado en la siguiente Figura 3.139, se emplea para realizar una operación de comparación entre el valor de la propiedad **ValueA** y el de la propiedad **ValueB**. Si la condición de comparación se cumple **Output** pasa a tener valor 1, de lo contrario tiene valor 0. La condición de comparación se establece en la propiedad **Operator**.

a!=b Comparer	
Propiedades	
ValueA (0)	
Operator (==)	
ValueB (0)	
Señales de E/S	
	Output (0)

Figura 3.139 - Componente inteligente Comparer

Este componente inteligente posee las siguientes señales y propiedades:

Señales

- **Output.** El valor de esta señal digital de salida cambia a 1 al cumplirse la condición de comparación, de lo contrario tiene valor 0.

Propiedades

- **ValueA.** En esta propiedad se debe especificar el primer valor de la comparación.
- **Operator.** Esta propiedad especifica la condición de comparación. Se puede elegir entre estos operadores de comparación, ==, !=, >, >=, <, <=.
- **ValueB.** En esta propiedad se debe especificar el segundo valor de la comparación.

Sensores

LineSensor

Este componente inteligente, mostrado en la siguiente Figura 3.140, crea una recta con inicio en el punto indicado en la propiedad **Start**, final en el punto indicado en la propiedad **End** y grosor según el valor indicado en la propiedad **Radius**. Si la entrada digital **Active** tiene valor 1, el sensor se encontrará activo, detectando cualquier componente que intersecte la línea del sensor. En la propiedad **SensedPart** se indica el componente que se encuentra intersectando la recta. Si varios componentes intersectan la recta del sensor, en esta propiedad se indica el componente detectado más cercano al punto de inicio del sensor.

LineSensor	
Propiedades	
Start ([0,00 0,00 0,00] mm)	
End ([0,00 0,00 100,00] m...)	
Radius (0,00 mm)	
SensedPart ()	
SensedPoint ([0,00 0,00 0,00] mm)	
Señales de E/S	
Active (1)	SensorOut (0)

Figura 3.140 - Componente inteligente LineSensor

Este componente inteligente posee las siguientes señales y propiedades:

Señales

- **Active.** Esta señal digital de entrada indica si el sensor se encuentra activo (1) o no (0).
- **SensorOut.** Esta señal digital de salida cambia su valor a 1 cuando la recta del sensor es intersectada por algún componente.

Propiedades

- **Start.** En esta propiedad se indica el punto de inicio de la recta del sensor.
- **End.** En esta propiedad se indica el punto final de la recta del sensor.
- **Radius.** Mediante esta propiedad se indica el grosor de la línea del sensor.
- **SensedPart.** En esta propiedad se indica el componente que se encuentra intersectando la recta, es decir, el componente detectado. Si varios componentes intersectan la recta del sensor, en esta propiedad se indica el componente detectado más cercano al punto de inicio del sensor.
- **SensedPoint.** En esta propiedad se indica el punto del componente que está intersectando la recta, más cercano al punto de inicio de la recta del sensor.

Manipuladores

Rotator2

Este componente inteligente, mostrado en la siguiente Figura 3.141, se emplea para rotar el componente especificado un cierto ángulo empleando el tiempo que se indique. El componente rotará alrededor de un eje indica, con posición en un punto especificado y referenciado en el sistema de coordenadas indicado.

Rotator2	
Propiedades	
Object ()	
CenterPoint ([0,00 0,00 0,00] mm)	
Axis ([0,00 0,00 1000,00] ...)	
Angle (0,00 deg)	
Duration (0,0 s)	
Reference (Global)	
Señales de E/S	
Execute (0)	Executed (0)
	Executing (0)

Figura 3.141 - Componente inteligente Rotator2

Este componente inteligente posee las siguientes señales y propiedades:

Señales

- **Execute.** Al cambiar esta señal digital de entrada a 1 se inicia la rotación del componente.
- **Executed.** Esta señal digital de salida envía un impulso, es decir, cambia momentáneamente su valor a 1, cuando se ha terminado de ejecutar la rotación del componente.
- **Executing.** Esta señal digital de salida tiene valor 1 mientras se está ejecutando la rotación del componente, después pasa a tener valor 0.

Propiedades

- **Object.** En esta propiedad debe especificarse el componente a rotar.
- **CenterPoint.** Se debe especificar el punto alrededor del cual se rotará el componente. Este punto está referenciado con respecto a la propiedad **Reference**.
- **Axis.** En esta propiedad se debe indicar el eje alrededor del que se rotará el componente.
- **Angle.** En esta propiedad se debe indicar el ángulo que se rotará el componente.
- **Duration.** En esta propiedad se debe indicar el tiempo, en segundos, que se empleará para rotar el componente.
- **Reference.** En esta propiedad se debe especificar el sistema de coordenadas respecto del cual se toma el punto de giro. Este puede ser Global, Local o en función de otro objeto.

PoseMover

Este componente inteligente, mostrado en la siguiente Figura 3.142, se emplea para mover un mecanismo a la pose especificada empleando el tiempo indicado.

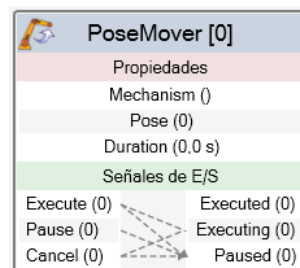


Figura 3.142 - Componente inteligente PoseMover

Este componente inteligente posee las siguientes señales y propiedades:

Señales

- **Execute.** Al cambiar esta señal digital de entrada a 1 se inicia el movimiento del mecanismo.
- **Pause.** Al cambiar esta señal digital de entrada a 1 se pausa el movimiento del mecanismo.
- **Cancel.** Al cambiar esta señal digital de entrada a 1 se cancela el movimiento del mecanismo.

- **Executed.** Esta señal digital de salida envía un impulso, es decir, cambia momentáneamente su valor a 1, cuando se ha alcanzado la pose indicada.
- **Executing.** Esta señal digital de salida tiene valor 1 mientras se está ejecutando el movimiento del mecanismo, después pasa a tener valor 0.
- **Paused.** Esta señal digital de salida tiene valor 1 mientras el movimiento del mecanismo se encuentra pausado, después pasa a tener valor 0.

Propiedades

- **Mechanism.** En esta propiedad se debe indicar el mecanismo a mover hasta la pose indicada.
- **Pose.** En esta propiedad se debe indicar el identificador de la pose hasta la que se requiere mover el mecanismo.
- **Duration.** En esta propiedad se debe indicar el tiempo, en segundos, que se empleará para trasladar el mecanismo hasta la pose indicada.

Controlador

RapidVariable

Este componente inteligente, mostrado en la siguiente Figura 3.143, se emplea para obtener o establecer el valor de una variable de RAPID desde la lógica de estación o del componente.

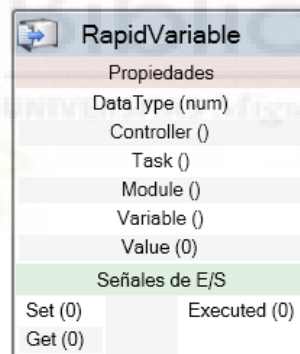


Figura 3.143 - Componente inteligente RapidVariable

Este componente inteligente posee las siguientes señales y propiedades:

Señales

- **Set.** Al activar esta señal digital de entrada, es decir, al cambiar su valor a 1, se establece el valor indicado en la propiedad **Value** para la variable especificada.
- **Get.** Al activar esta señal digital de entrada se obtiene el valor de la variable especificada, pasando a ser el valor de la propiedad **Value** el de esta.
- **Executed.** Esta señal digital de salida envía un impulso cuando se ha ejecutado el componente.

Propiedades

- **DataType.** En esta propiedad se debe especificar el tipo de dato de la variable a obtener/establecer.

- **Controller.** En esta propiedad se debe especificar el controlador en el que se encuentra la variable.
- **Task.** En esta propiedad se debe especificar la tarea de RAPID en la que se encuentra la variable.
- **Module.** En esta propiedad se debe especificar el módulo de RAPID en el que se encuentra la variable.
- **Variable.** En esta propiedad se debe especificar el identificador de la variable.
- **Value.** Si se quiere establecer el valor de la variable, en esta propiedad se debe especificar el valor que se le quiere dar. Si se quiere obtener el valor de una variable, el valor de esta propiedad pasará a ser el de la variable.

Otros

Highlighter

Este componente inteligente, mostrado en la siguiente Figura 3.144, se emplea para cambiar el color del componente indicado. El nuevo color del componente será una mezcla de su color original y del color especificado, según la opacidad especificada para este último.

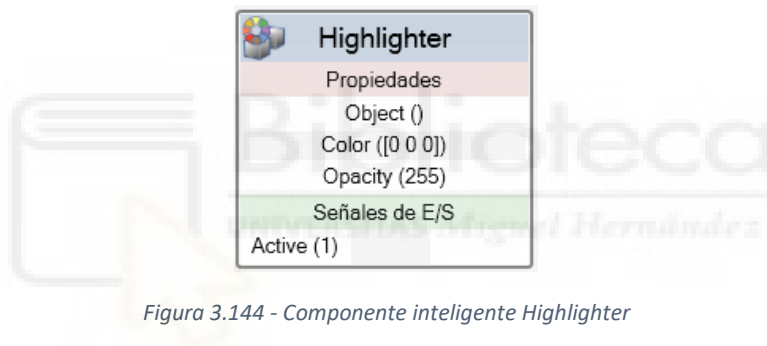


Figura 3.144 - Componente inteligente Highlighter

Este componente inteligente posee las siguientes señales y propiedades:

Señales

- **Active.** Al activar esta señal digital de entrada se cambia el color de componente. Al desactivar se restablece su color original.

Propiedades

- **Object.** En esta propiedad se debe especificar el componente al que se le cambiará el color.
- **Color.** En esta propiedad se deben especificar los valores RGB del color al que se quiere cambiar.
- **Opacity.** El valor de esta propiedad indica la opacidad del nuevo color sobre el original, a mayor opacidad más cambiará el color del componente al nuevo color. El rango de valor de esta propiedad es de 0 a 255.

LightControl

Este componente inteligente, mostrado en la siguiente Figura 3.145, se emplea para controlar una fuente de luz de la estación, pudiendo activarla/ desactivarla y variar sus diferentes propiedades.

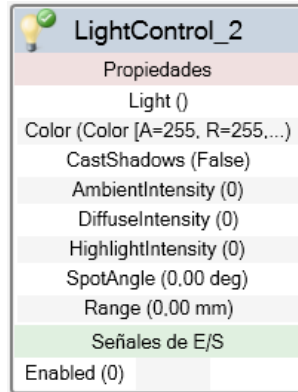


Figura 3.145 - Componente inteligente LightControl

Este componente inteligente posee las siguientes señales y propiedades:

Señales

- **Enabled.** Al activar esta señal digital de entrada se activa la fuente de luz especificada con las propiedades indicadas. Al desactivar esta señal se desactiva la fuente de luz.

Propiedades

- **Light.** En esta propiedad se debe especificar la fuente de luz a controlar.
- **Color.** Esta propiedad indica el color de la fuente de luz.
- **CastShadows.** Esta propiedad indica si se la luz proyectará las sombras de los objetos.
- **AmbientIntensity.** El valor de esta propiedad indica cómo será reflejada la luz ambiental especificada por las superficies de los componentes de la estación.
- **DiffuseIntensity.** El valor de esta propiedad indica cómo será reflejada la luz especificada por las superficies de los componentes de la estación.
- **HighlightIntensity.** El valor de esta propiedad indica la intensidad de resalte de la fuente de luz sobre las superficies de los componentes de la estación.
- **SpotAngle.** Esta propiedad indica el ángulo de apertura de una fuente de luz focal. El rango de valor de esta propiedad es de 5 a 135.
- **Range.** Esta propiedad indica la distancia hasta la que la fuente de luz será capaz de iluminar.

ColorTable

Este componente inteligente, mostrado en la siguiente Figura 3.146, permite crear una lista de valores RGB de colores. El número de colores presentes en la lista se especifica en la propiedad **NumColors**.

ColorTable	
Propiedades	
NumColors	(2)
SelectedColorIndex	(0)
SelectedColor	(Color [A=255, R=0, G...])
Color0	(Color [A=255, R=0, G...])
Color1	(Color [A=255, R=255,...])
Señales de E/S	

Figura 3.146 - Componente inteligente ColorTable

Este componente inteligente posee las siguientes propiedades:

Propiedades

- **NumColors.** En esta propiedad debe indicarse el número de colores que formarán la lista.
- **SelectedColorIndex.** En esta propiedad se especifica la posición o número de color de la lista seleccionado.
- **SelectedColor.** En esta propiedad se especifican los valores RGB del color seleccionado de la lista.
- **ColorX.** Cada uno de los colores de la lista se nombran mediante según Color + Índice de la lista, Color0, Color1, Color2, ..., Color(NumColors-1). En cada una estas propiedades se debe especificar el color de la misma.

PaintApplicator

Este componente inteligente, mostrado en la siguiente Figura 3.147, se emplea para simular la aplicación de pintura sobre un componente indicado. Este componente inteligente tiene representación gráfica en forma de cono, para simular el patrón de pintura que emite un pulverizador o un atomizador. El color del cono, y por tanto de la pintura a aplicar, se indica en la propiedad **Color**. La altura de este cono y las diagonales menor y mayor de su base se especifican en las propiedades **Range**, **Width** y **Height** respectivamente. Para aplicar la pintura sobre el componente, el componente inteligente se conecta y posiciona en la herramienta del robot, normalmente un pulverizador o un atomizador aunque la herramienta en sí no afecta al funcionamiento del componente inteligente. Al pasar el cono, que simular el patrón de pintura, sobre el componente a pintar, la zona del cono que intersecta con el componente se colorea del color especificado. Al detenerse la simulación el componente permanece pintado, pero al reiniciarla el pintado se resetea, de forma que el componente vuelve a tener su color original.

Uno de los problemas que presenta este componente inteligente es que no funciona, es decir, no simula la aplicación de pintura, sobre componentes que se mueven sobre un transportador.

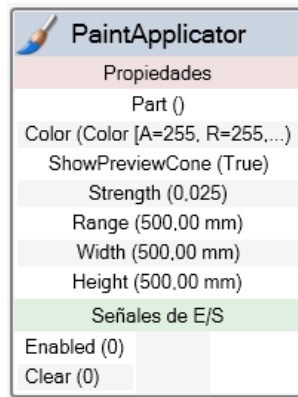


Figura 3.147 - Componente inteligente PaintApplicator

Este componente inteligente posee las siguientes señales y propiedades:

Señales

- **Enabled.** Al activar esta señal digital de entrada se activa el componente inteligente y por tanto la simulación de la aplicación de pintura. Al desactivar esta señal se desactiva la simulación de la aplicación de pintura.
- **Clear.** Al activar esta señal digital de entrada se borra la pintura aplicada, recuperando el componente su color original.

Propiedades

- **Part.** En esta propiedad se debe especificar el componente sobre el que se aplicará la pintura.
- **Color.** En esta propiedad se especifica el color de la pintura.
- **ShowPreviewCone.** Si esta propiedad tiene el valor TRUE (1) se mostrará la representación gráfica en forma de cono del componente durante la simulación. Si tiene valor FALSE (0) no se mostrará.
- **Strength.** Esta propiedad indica la opacidad con la que se mostrará la pintura aplicada sobre el componente. El valor de esta propiedad se expresa en tanto por uno, es decir de 0 a 1. Esta propiedad puede entenderse como la fuerza con la que se depositará la pintura sobre el componente, de forma que a mayor fuerza mayor será la opacidad del color aplicado y menos se verá el color base del componente.
- **Range.** El valor de esta propiedad indica la altura del cono de la representación gráfica.
- **Width.** El valor de esta propiedad indica la mitad de la longitud de la diagonal menor de la base del cono.
- **Height.** El valor de esta propiedad indica la mitad de la longitud de la diagonal mayor de la base del cono.

3.1.21 Señales de E/S del sistema de robot

Las señales de E/S reales del sistema de robot son representaciones de software lógicas de las entradas y salidas que se encuentran en un dispositivo de E/S, conectado a una red industrial dentro del propio sistema. También pueden tenerse señales de E/S simuladas, siendo estas representaciones de software lógicas de las entradas y salidas del sistema de robot que solo existen en la simulación de la estación, es decir, no tienen representación en ningún dispositivo de E/S. El número máximo de posibles señales de E/S definidas por el usuario en un sistema de robot es de 12000 señales. Los controladores de robot disponen de una serie de señales de E/S predefinidas, variando su número en función de las opciones seleccionadas al crear el controlador.

Las señales de E/S son un aspecto de gran importancia en la programación de una estación robótica, empleándose para transmitir información entre los componentes de la estación y controlar los diferentes eventos que pueden ocurrir durante la ejecución del programa, como controlar dispositivos de la estación, activar y/o leer datos de sensores, etc.

Para crear una nueva señal de E/S se deberá acceder a la pestaña **Controlador**, seleccionar en el navegador **Controlador** el controlador en el que se quiere crear la señal, grupo **Configuración**, herramienta **Configuración** y en el desplegable que aparece seleccionar la opción **I/O System**.

Al hacer esto aparece una nueva ventana llamada **Configuración – I/O System**, en la cual se muestran los diferentes parámetros del sistema para los dispositivos y señales de E/S. Seleccionando en la sección **Tipo**, de esta ventana, la opción **Signal** aparecerán todas las señales de E/S presentes en el controlador de robot, como se muestra en la Figura 3.148.

Para crear una nueva señal se selecciona esta opción **Signal** mediante el botón derecho del ratón, de forma que aparece la opción **Nuevo Signal**. También se puede acceder a esta opción seleccionando cualquiera de las señales existentes mediante el botón derecho del ratón. En este caso aparecerán además de la opción **Nuevo Signal**, las opciones **Editar Signal**, **Copiar Signal** y **Eliminar Signal**. Mediante la opción **Editar Signal** se pueden editar los parámetros de una señal existente. La opción **Copiar Signal** permite crear una nueva señal cuyos parámetros serán los mismos que los de la señal seleccionada. Y por último la opción **Eliminar Signal** permite eliminar una señal.

Tipo	Name	Type of Signal	Assigned to Device	Signal Identification Label	Device Mapping	Category	Access Level	Default Value	Filter Time Passive (ms)
Access Level	A1BellRotating	Digital Input			N/D		All	0	0
Cross Connection	A1Brush	Group Output			N/D		All	0	N/D
Device Trust Level	A1BrushTable	Analog Input			N/D		All	0	N/D
DeviceNet Command	A1BrushTabLimEn	Digital Output			N/D		All	0	N/D
DeviceNet Device	A1BrushTabNdx	Group Output			N/D		All	0	N/D
DeviceNet Internal Device	A1Enable	Digital Output			N/D		All	0	N/D
EtherNet/IP Command	A1Error	Digital Output			N/D		All	0	N/D
EtherNet/IP Device	A1Fluid	Analog Output			N/D		All	0	N/D
ICI Device	A1FluidAccu	Analog Output			N/D		All	0	N/D
Industrial Network	A1FluidEn	Digital Output			N/D		All	0	N/D
Robt/CI Device	A1FluidPITO	Digital Output			N/D		All	0	N/D
Robt/CI Server	A1FluidPIWS	Digital Output			N/D		All	0	N/D
Route	A1HVEen	Digital Output			N/D		All	0	N/D
Signal	A1HVErrNo	Group Output			N/D		All	0	N/D
Signal Safe	A1Material	Group Output			N/D		All	0	N/D
System Input	A1NewObjStro	Digital Output			N/D		Default	0	N/D
System Output	AS1	Digital Input	PANEL	Automatic Stop chain(X5:11 to X5:6) and (X5:9 to X5:1)	13	safety	ReadOnly	0	0
	AS2	Digital Input	PANEL	Automatic Stop chain backup(X5:5 to X5:6) and (X5:3 to X5:1)	14	safety	ReadOnly	0	0
	AUTO1	Digital Input	PANEL	Automatic Mode(X9:6)	5	safety	ReadOnly	0	0
	AUTO2	Digital Input	PANEL	Automatic Mode backup(X9:2)	6	safety	ReadOnly	0	0
	AutoModeOn	Digital Output			N/D		Default	0	N/D
	Ax2BrakeRelDI	Digital Input	PIB		Mac25:Digin14		Default	0	0
	Ax2BrakeRelDO	Digital Output	PIB		Man1 Brake7		Default	0	N/D
	BellRotating	Digital Output			N/D		Default	0	N/D
	c1CntFromEnc	Group Input	CTM1		Sync1Enc1.Value		Default	0	0
	c1CntToEnc	Group Output			N/D		Default	0	N/D
	c1CntToEncStr	Digital Output			N/D		All	0	N/D
	c1Connected	Digital Output			N/D		Default	0	N/D
	c1Counts	Group Input	CTM1		Enc1 position		Default	0	0
	c1CountsPerSec	Analog Input	CTM1		Enc1 velocity		Default	0	N/D
	c1DropWObj	Digital Output			N/D		All	0	N/D
	c1NewObjStrobe	Digital Output			N/D		All	0	N/D
	c1NullSpeed	Digital Output			N/D		Default	0	N/D
	c1ObjactInQ	Group Output			N/D		Default	0	N/D
	c1PassSw	Digital Output			N/D		Default	0	N/D
	c1Postn.JobQ	Digital Output			N/D		Default	1	N/D
	c1Position	Analog Output			N/D		Default	0	N/D

Figura 3.148 - Ventana llamada Configuración – I/O System

Al seleccionar la opción **Nuevo Signal** aparece el cuadro de dialogo **Instancia editor**, en el que se deberán especificar los parámetros de sistema específicos de la señal de E/S, empleados para controlar el comportamiento de la misma. En la siguiente Figura 3.149 se muestra este cuadro de dialogo con los parámetros comunes a todos los tipos de señal.

Figura 3.149 - Cuadro de dialogo Instancia editor con los parámetros comunes a todas las señales

El parámetro **Name** especifica el identificador de la señal. Este será una cadena de caracteres con un máximo de 32. El nombre de la señal debe ser único, considerándose dos nombres iguales aunque varíen las mayúsculas y minúsculas.

El parámetro **Type of Signal** especifica el tipo de señal que se creará. El tipo de señal especifica el comportamiento de la señal de E/S y la forma en que se representa e interpreta. Algunos de los parámetros de la configuración dependen del tipo de señal de E/S, por lo que es recomendable que este parámetro se asigne primero, mostrándose así todos los parámetros para ese tipo de señal. Los tipos de señales que pueden crearse son los siguientes:

- Digital Input
- Digital Output
- Analog Input
- Analog Output
- Group Input
- Group Output

El parámetro **Assigned to Device** especifica a qué dispositivo de E/S se encuentra conectada la señal. Si la señal no está conectada a ningún dispositivo se trata de una señal simulada.

El parámetro **Signal Identification Label** es opcional y se emplea para asignar a la señal una etiqueta de texto, es decir, un breve comentario acerca de la misma de máximo 80 caracteres.

El parámetro **Category** es opcional y se emplea para asignar una categoría a la señal, haciendo posible un posterior filtrado y clasificación de las señales en base a estas categorías.

El parámetro **Access Level** especifica quien puede tener acceso de escritura a la señal. Las opciones para este parámetro son:

- **All.** Todos los clientes, locales y remotos, tienen acceso de escritura.
- **Default.** Únicamente se tiene acceso de escritura desde instrucciones de RAPID y clientes locales.
- **ReadOnly.** Ningún cliente tiene acceso de escritura, solo pueden ser leídas.

Una vez se han especificados todos los parámetros de la señal se pulsa **Aceptar**, creándose así la señal y apareciendo en la ventana. Para que las señales creadas aparezcan y se tengan en cuenta en el controlador, este se debe reiniciar. Para esto se selecciona la pestaña **Controlador**, grupo **Herramientas de controladores**, herramienta **Reiniciar** y en el desplegable que aparece se selecciona la opción **Reiniciar (Arranque en caliente)**.

3.1.22 Simulación de la estación

La simulación de la estación puede definirse como la ejecución de un programa de robot en el controlador virtual, permitiendo observar en la ventana gráfica el movimiento del robot y del resto de componentes de la estación y como interactuarían entre sí. De esta forma se puede realizar una prueba exhaustiva del programa de robot previa programación en la estación real, pudiendo así perfeccionarlo y aumentando la seguridad del equipo y del personal.

Configuración de la simulación

Antes de ejecutar la simulación de la estación se deben configurar los parámetros de la misma. Para esto se accede a la pestaña **Simulación**, grupo **Configurar** y se selecciona la herramienta **Configuración de simulación**, de forma que aparece la ventana **Configuración de simulación**, mostrada en la siguiente Figura 3.150.

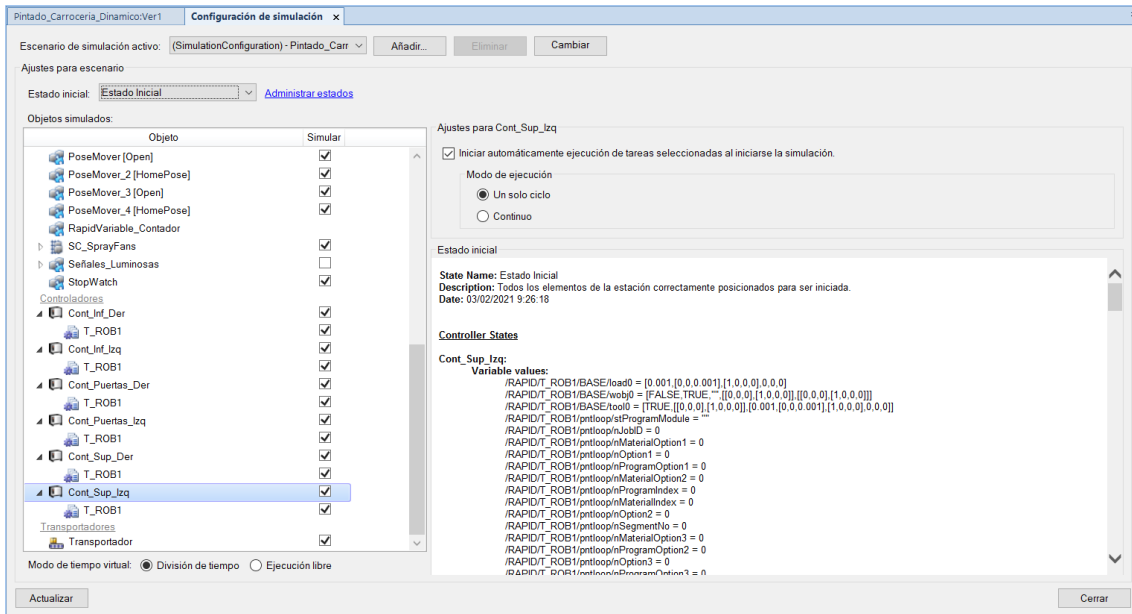


Figura 3.150 - Ventana Configuración de simulación

En la sección **Escenario de simulación activo** de esta ventana se puede seleccionar, editar o añadir un escenario de simulación. Se entiende por escenario de simulación la configuración completa del resto de parámetros de esta ventana relativos a la simulación, de forma que se pueden guardar diferentes configuraciones para diferentes situaciones de ejecución.

En la sección **Estado inicial** se puede especificar un estado guardado en la lógica de estación, como se explica en el apartado [3.1.19 Lógica de estación](#), para comenzar la simulación. De esta forma la simulación siempre se iniciará con los componentes de la estación configurados del mismo modo. Si se selecciona la opción **Administrar estados** se abre la ventana **Lógica de estación**, para poder gestionar los estados guardados.

En la sección **Objetos simulados** se muestran todos los componentes de la estación que puede formar parte de la simulación, como controladores virtuales, componentes inteligentes, transportadores, etc. En esta sección se pueden seleccionar cuales de estos componentes se quiere que participen en la simulación.

En la sección **Modo de tiempo virtual** se debe especificar el modo en el que RobotStudio gestionará el tiempo de la simulación. Se tienen las siguientes dos opciones:

- **Modo de división de tiempo.** Para la simulación de una estación con varios controladores, en este modo, RobotStudio gestiona el tiempo de forma que se consiga una correcta sincronización de los controladores. Cada controlador recibe una división de tiempo individual para ejecución. Cuando todos los controladores han terminado la ejecución de su división de tiempo asignada, RobotStudio avanza hasta la siguiente división de tiempo.
- **Modo de ejecución libre.** En este modo los controladores funcionan de forma independiente entre sí. El tiempo de ciclo será preciso pero la sincronización no lo será.

Por último si se selecciona un controlador, en la sección **Objetos simulados**, aparecen las opciones de simulación del mismo. Se puede seleccionar si se quiere que se inicie

automáticamente la ejecución de las tareas del controlador al iniciarse la simulación y si la ejecución del programa de robot solo se realizará una vez o si se realizará en bucle.

Si se selecciona una tarea dentro de un controlador aparece la opción de seleccionar el procedimiento que se tomará como punto de entrada de la simulación, es decir, el procedimiento de RAPID en el que se inicia la ejecución del programa.

Control de la simulación

Una vez se ha configurado la simulación, esta puede ser ejecutada, controlándola mediante las siguientes herramientas del grupo **Control** de simulación de la pestaña **Simulación**.

- **Reproducir.** Al seleccionar esta opción comienza la simulación, ejecutándose según la configuración de simulación establecida previamente. Si se ha pausado la simulación, esta opción pasa a denominarse **Reanudar** y seleccionándola se reanuda la simulación.
- **Pausa.** Durante la ejecución de una simulación esta opción permite pausarla. Una vez se ha pausado la simulación esta opción pasa a denominarse **Paso**, de forma que cada vez que se selecciona la simulación avanza un instante de tiempo especificado, pero no se reanuda.
- **Parar.** Seleccionando esta opción se detiene la ejecución de la simulación.
- **Restablecer.** Una vez detenida la simulación, esta opción permite restablecer la configuración de los componentes de la estación según el estado especificado. Si se tiene más de un estado guardado se deberá seleccionar el que se empleará para restablecer la estación. Esta herramienta también tiene la opción de **Guardar estado actual**, que se emplea de la misma forma que la herramienta presente en la lógica de estación.

Simulación de señales de E/S

El simulador de señales de E/S permite ver y editar las señales de entrada y salida de los controladores, la estación y los componentes inteligentes durante la simulación de la misma, es decir, se puede observar cómo varía el valor de las señales y también modificar su valor si es preciso, siempre que lo permita su nivel de acceso, a lo largo de la simulación de la estación. Esta herramienta se encuentra en la pestaña **Simulación**, grupo **Monitor**. La siguiente Figura 3.151 muestra un ejemplo de las señales de E/S de un controlador, de la estación y de un componente inteligente.

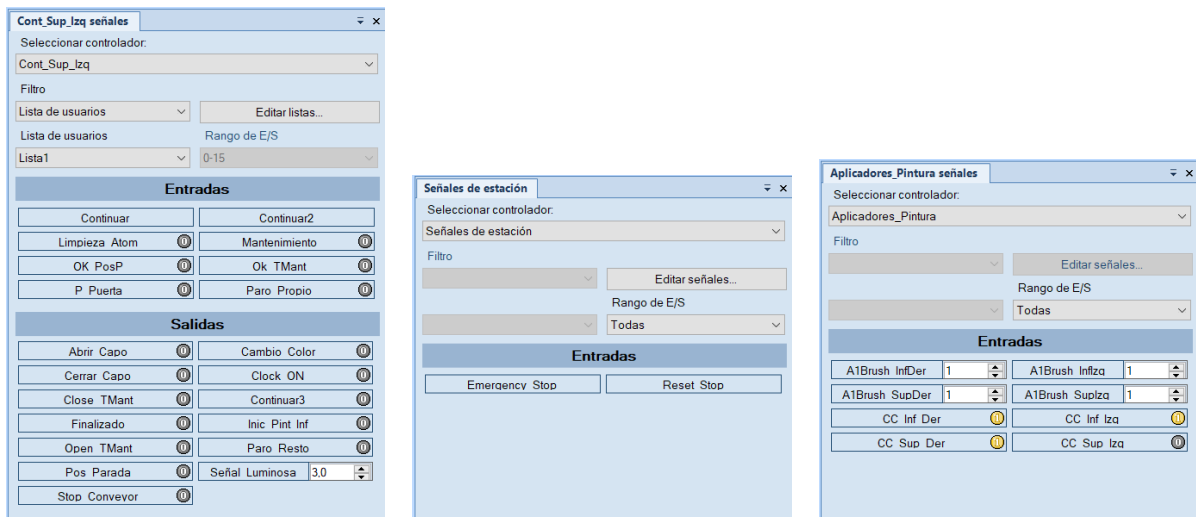







Figura 3.151 - Ejemplo de señales de E/S de un controlador, de una estación y de un componente inteligente

Al seleccionar esta herramienta se abre el panel **Señales**. En la sección **Seleccionar controlador** se puede seleccionar entre la estación, un controlador o un componente inteligente presente en la estación, para así ver sus señales de E/S. Estas señales se mostrarán clasificadas entre **Entradas** y **Salidas**, en grupos de 16 señales. Si se tienen más de 16 señales, en la sección **Rango de E/S** se puede cambiar el rango de señales mostrado o elegir mostrar todas las señales.

Si se trata de una señal digital, al lado de la misma aparece uno de los siguientes símbolos indicando su valor. El símbolo  indica que la señal tiene valor TRUE (1), mientras que el símbolo  indica que la señal tiene valor FALSE (0). Si aparece un asa sobre el símbolo , esto indica que la señal es una conexión cruzada. Si aparece un número -1 sobre el símbolo , esto indica que la señal se encuentra invertida. Para cambiar el valor de señal simplemente se debe pulsar sobre ella, observándose como cambia el símbolo.

Si se trata de una señal analógica, al lado la misma aparece el cuadro de valor  indicando el valor de la misma. Para cambiar su valor simplemente se debe introducir el nuevo valor en el cuadro.

Si se selecciona un controlador de la estación, en la sección **Filtro** se pueden aplicar filtros de forma que sea más sencillo observar las señales que se requiere. Las señales pueden filtrarse por:

- **Dispositivo.** Este filtro muestra las señales programadas en el dispositivo especificado en la sección Especificación de filtro, que en este caso se denomina Dispositivo.
- **Grupo.** Este filtro muestra una señal de entrada o salida de grupo. Se puede seleccionar el grupo en la sección Especificación de filtro, que en este caso se denomina Grupo.
- **Lista de usuarios.** Este filtro muestra las señales que han sido añadidas previamente a una lista. Se puede seleccionar la lista en la sección Especificación de filtro, que en este caso se denomina Lista de usuarios. Para crear o editar una lista se debe seleccionar la opción Editar listas. Este filtro es muy útil ya que permite visualizar únicamente las señales de la lista, que previamente se han incluido por ser las que se quiere analizar.
- **Entradas digitales.** Este filtro muestra las señales digitales de entrada.
- **Salidas digitales.** Este filtro muestra las señales digitales de salida.

- **Entradas analógicas.** Este filtro muestra las señales analógicas de entrada.
- **Salidas analógicas.** Este filtro muestra las señales analógicas de salida.

Medición del tiempo de simulación

Para realizar la medición del tiempo de simulación RobotStudio ofrece una herramienta denominada **Cronómetro** (Figura 3.152), situada en la pestaña **Simulación**, grupo **Monitor**. Esta herramienta permite medir el tiempo transcurrido entre dos puntos de disparo del programa de robot, denominados disparador de inicio y disparador de fin.

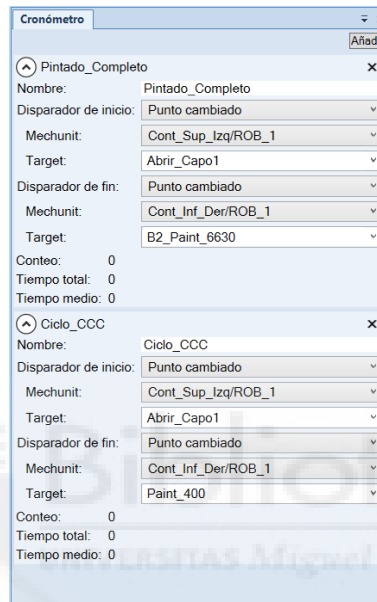


Figura 3.152 - Panel Cronómetro

El primer paso será añadir un nuevo cronómetro, para lo cual en el panel **Cronómetro** se debe seleccionar la opción **Añadir**. Una vez creado, el primer aspecto que puede especificarse es el nombre del cronómetro. Tras esto se deben especificar los disparadores de inicio y fin.

Para el disparador de inicio se pueden seleccionar como punto de disparo los siguientes:

- **Inicio de simulación.** El cronómetro se inicia al iniciarse la simulación, es decir, al pulsar la opción **Reproducir** del control de la simulación.
- **Punto cambiado.** El cronómetro se inicia cuando el robot especificado alcanza el punto objetivo programado indicado. Si se selecciona esta opción, se debe especificar el controlador del robot y el punto objetivo que actuará como punto de disparo de inicio.
- **Valor de E/S.** El cronómetro se inicia cuando la señal especificada cambia al valor indicado. Si se selecciona esta opción se debe especificar el controlador o componente inteligente en el que se encuentra la señal, la señal de entrada o salida en cuestión y el valor de la señal que actuará como disparador de inicio.

Para el disparador de fin se pueden seleccionar como punto de disparo los siguientes:

- **Paro de simulación.** El cronómetro se detiene al interrumpirse la simulación, es decir, al pulsar la opción **Parar** del control de la simulación o terminarse la ejecución del programa de robot.
- **Punto cambiado.** El cronómetro se detiene cuando el robot especificado alcanza el punto objetivo programado indicado. Si se selecciona esta opción, se debe especificar el controlador del robot y el punto objetivo que actuará como punto de disparo de fin.
- **Valor de E/S.** El cronómetro se detiene cuando la señal especificada cambia al valor indicado. Si se selecciona esta opción se debe especificar el controlador o componente inteligente en el que se encuentra la señal, la señal de entrada o salida en cuestión y el valor de la señal que actuará como disparador de fin.

En la sección **Conteo** del cronómetro se indicará el número de veces que se ha iniciado el mismo. En la sección **Tiempo total** se indicará la suma del tiempo de todos los ciclos del cronómetro. Por último, en la sección **Tiempo medio** se indica el tiempo medio empleado, es decir, el **Tiempo total** dividido por el **Conteo**.

3.1.23 Detección de colisiones

RobotStudio proporciona dos herramientas en cuanto a la detección de colisiones se refiere. La primera de ellas se denomina **Collision Avoidance**. Esta herramienta monitoriza la geometría del robot y su envolvente de trabajo para evitar que el robot colisione con algún otro elemento de la estación. Para ello los elementos de la estación con los que no se quiere que el robot colisione se deben incluir en la configuración de la herramienta. Al ejecutar la simulación, y por tanto el programa de robot, si existe alguna colisión se muestra resaltando los componentes en la ventana gráfica antes de que ocurra. Para poder emplear esta herramienta, al crear el controlador se debe seleccionar la opción **Detección de colisiones** en las **Opciones del controlador**. No se expondrá con más detalle esta herramienta ya que, a pesar de ser de gran utilidad, no es posible seleccionarla en los controladores de robots de pintura, y todos los robots del presente trabajo son de ese tipo.

La segunda de las herramientas para la detección de colisiones se denomina **Colisiones** y se emplea para detectar y registrar las colisiones que se producen entre los componentes de la estación. Esta herramienta resulta muy útil para detectar las colisiones entre los componentes de la estación en la fase de programación, permitiendo así corregir el programa para eliminarlas, y por tanto evitar colisiones reales en la estación robótica real. También permite detectar las situaciones en las que casi se produce una colisión, es decir, las situaciones en las que los componentes especificados se encuentran a una distancia entre ellos inferior a la distancia especificada de casi colisión. Al detectar una casi colisión o una colisión la simulación se pausa para poder observar donde y como se ha producido.

Esta herramienta consume una gran cantidad de recursos informáticos para realizar los cálculos de detección de colisiones, por lo que se deben incluir únicamente los componentes que potencialmente puedan colisionar durante la simulación.

El primer paso para emplear esta herramienta es crear un conjunto de colisión. Para ello se debe acceder a la pestaña **Simulación**, grupo **Colisiones** y seleccionar la herramienta **Crear conjunto de colisión**. Un conjunto de colisión se compone de dos conjuntos de objetos, Objetos A y Objetos B, en los que se añaden los componentes de la estación entre los que se realizará la comprobación de las colisiones, es decir, se comprobará si alguno de los componentes del conjunto A colisiona con alguno de los componentes del conjunto B. Si se produce una colisión esta se representa en la ventana gráfica resaltando los componentes en un color especificado y se registra en la **Ventana de salida**.

Una vez creado el conjunto de colisión este aparece en el navegador **Diseño**. A continuación se deben agregar los componentes a los conjuntos de objetos A y B. Para esto, se selecciona el componente deseado en el navegador **Diseño**, mediante el botón izquierdo del ratón, y se arrastra hasta el conjunto de objetos en el que se quiera incluir.

Una vez se han añadido los componentes a los conjuntos de objetos se debe configurar el conjunto de colisión. Para esto se selecciona en el navegador **Diseño** el conjunto de colisión a configurar, mediante el botón derecho del ratón, y se selecciona la opción **Modificar conjunto de colisión**, abriéndose el panel **Modificar conjunto de colisión**, mostrado en la siguiente Figura 3.153.

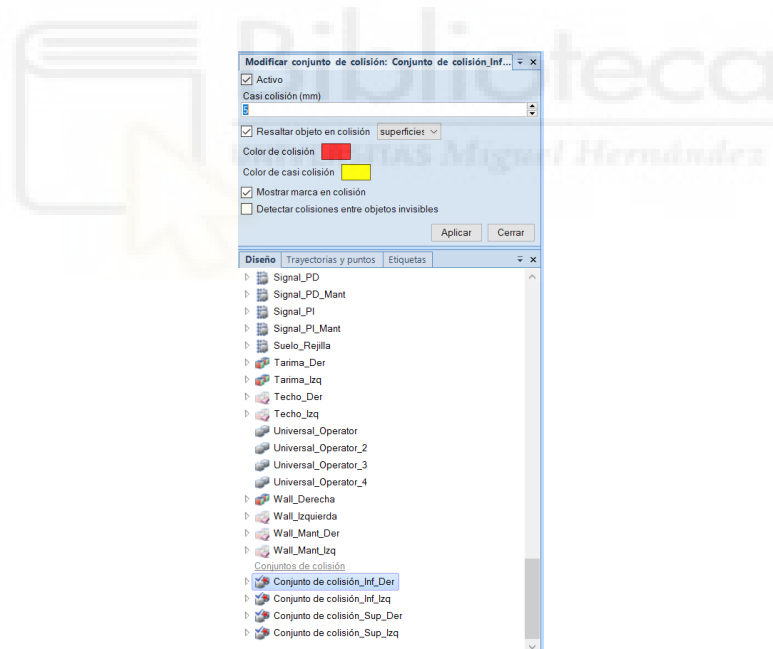


Figura 3.153 - Panel Modificar conjunto de colisión

El primer aspecto a indicar en este panel de configuración es si el conjunto de colisión está activado. Seguidamente se debe especificar la distancia de casi colisión, es decir, la distancia por debajo de la cual si se aproximan dos componentes, uno del conjunto A y otro del conjunto B, se detectará una casi colisión. También debe especificarse si al detectarse una casi colisión o una colisión se resaltarán los componentes en la ventana gráfica, en caso afirmativo, que parte de los componentes se resaltaré (pieza, cuerpo o superficie) y de qué color se hará. Se debe indicar también si se quiere que al detectarse una casi colisión o colisión se muestre una etiqueta en la

ventana gráfica con los nombres de los componentes que han colisionado. Por último se debe indicar si se quiere detectar colisiones entre componentes aunque la visualización de estos se haya cambiado a invisible. En la siguiente Figura 3.154 se muestra un ejemplo de como se muestra en la ventana gráfica y en la ventana de salida una casi colisión y una colisión.

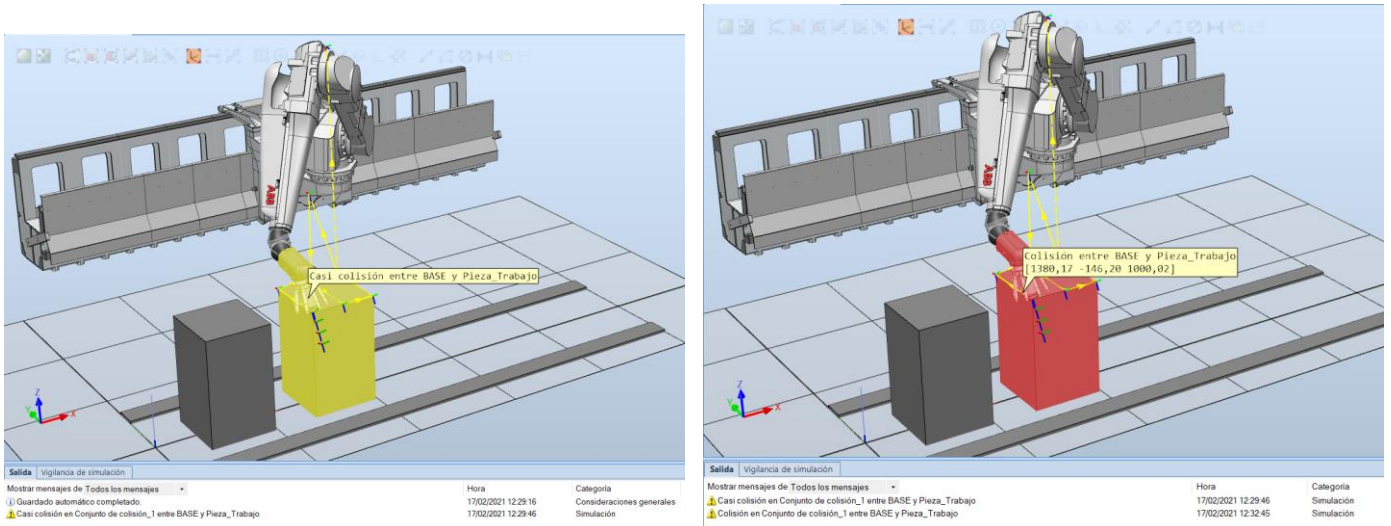


Figura 3.154 - Visualización de un caso de casi colisión y un caso de colisión

Es recomendable cambiar la geometría de colisión de los componentes agregados a los conjuntos de colisión a la opción **Usar geometría original**, como se explica en el apartado [3.1.9 Modificación de componentes en RobotStudio](#).

También deben configurarse las opciones de simulación de colisiones de RobotStudio. Para esto se debe acceder a la pestaña **Archivo, Opciones** y dentro del cuadro de dialogo que aparece, sección **Simulación, Colisiones**, como se muestra en la Figura 3.155.

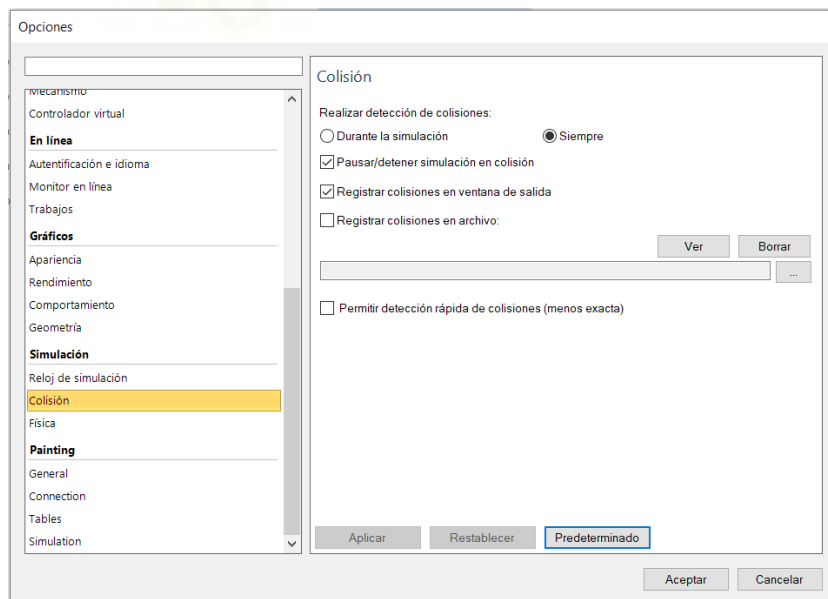


Figura 3.155 – Configuración de las opciones de simulación de colisiones de RobotStudio

En este cuadro de dialogo se tienen las siguientes opciones:

- **Realizar detección de colisiones.** Se debe especificar si se desea que la detección de colisiones se realice únicamente durante la simulación o siempre.
- **Pausar/detener simulación en colisión.** Se debe indicar si se desea que la simulación se pause al detectar una colisión.
- **Registrar colisiones en ventana de salida.** Se debe indicar si las colisiones se registran en la ventana de salida de RobotStudio.
- **Registrar colisiones en archivo.** Se debe indicar si se desea que las colisiones queden registradas en un archivo.
- **Permitir detección rápida de colisiones.** Activando esta casilla se realizará la detección de colisiones entre rectángulos geométricos delimitadores en lugar de triángulos geométricos. Esta opción aumenta el rendimiento pero puede causar detecciones de colisiones falsas, ya que los triángulos geométricos son la geometría real y los rectángulos delimitadores tienen siempre mayor tamaño.

3.1.24 Complementos y PowerPacs

El software RobotStudio ofrece una serie de módulos complementarios al programa base denominados PowerPacs. Cada uno de estos módulos o PowerPac está destinado específicamente a ampliar la potencia del software en una tarea determinada, como la soldadura por arco, el mecanizado, el pintado, etc.

El Painting PowerPac, mostrado en la Figura 3.156, es un módulo complementario de RobotStudio que se utiliza para crear y editar los programas de pintura. Proporciona funciones para conectar, obtener y transferir archivos a los robots en la estación real. Ofrece una forma intuitiva de gestionar los programas de pintura y los modelos CAD de las piezas. Se pueden crear, modificar y probar programas nuevos y existentes con funciones de programación avanzadas.

La copia de programas entre los robots en una estación de pintura ahorra mucho esfuerzo y garantiza una sincronización adecuada. Esto, junto con la vista 3D mejorada, permite crear y editar eficazmente las instrucciones de pintura.

Esto permite utilizar el software como herramienta en la estación real para editar los programas sin detener la producción en curso.

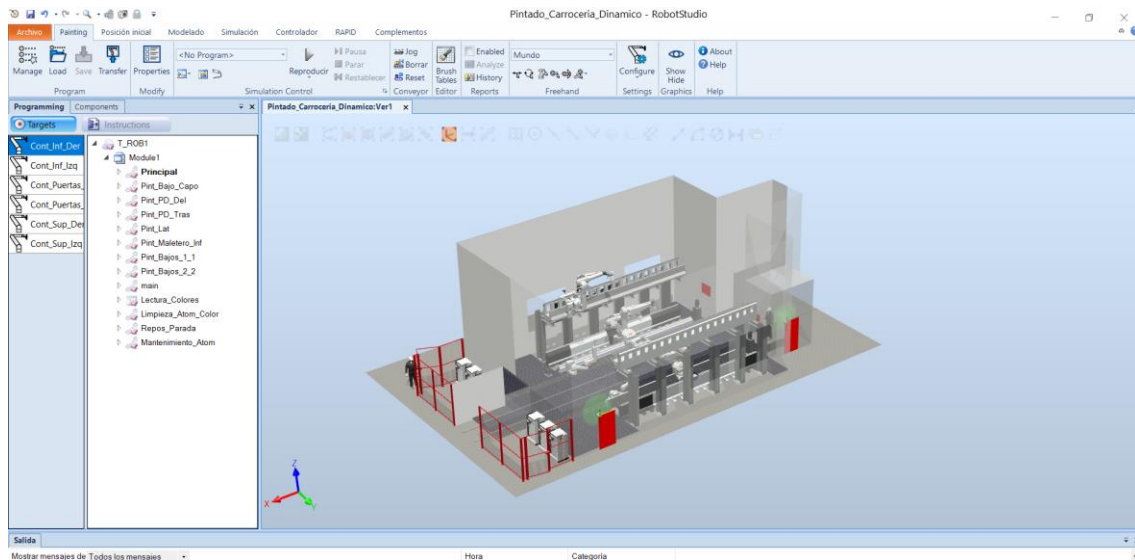


Figura 3.156 - Painting PowerPac

El Painting PowerPac puede emplearse para controlar una serie de funciones, como por ejemplo las siguientes:

- Recuperación de los programas de pintura de los robots de la planta en RobotStudio.
- Edición de la trayectoria de pintado.
- Verificación de la alcanzabilidad de la posición del objetivo y de los puntos de disparo (SetBrush).
- Establecimiento y edición de puntos de disparo.
- Simulación de programas de pintura
- Transferencia de archivos a los robots
- Habilitar el acceso a las funciones basado en el inicio de sesión del usuario

El Painting PowerPac es una herramienta de programación que se utiliza en las líneas de pintura para optimizar los programas existentes o en modo offline para crear programas para nuevos modelos. Los siguientes pasos indican el flujo de trabajo para trabajar con el programa de pintado utilizando RobotStudio y el Painting PowerPac:

1. Preparación de la estación de RobotStudio

Una estación robótica de RobotStudio consta de uno o varios robots, transportadores y equipos periféricos. La estación debe coincidir estrechamente con las células de pintura reales.

2. Preparación de la pieza de trabajo

Se pueden seguir los siguientes pasos para preparar el modelo CAD del objeto a pintar. El modelo CAD del objeto se denominará pieza de trabajo.

1. Se importa el modelo de CAD en una estación de RobotStudio.
2. Se reducen los datos del modelo de CAD, mediante, por ejemplo, la herramienta **Split Part by** de este PowerPac, expuesta en el apartado [3.1.9 Modificación de componentes en RobotStudio](#).

3. Se crean los paneles de pintado de la pieza de trabajo. En el presente proyecto se intentó llevar a cabo este proceso, pero se desistió ya que la geometría de la carrocería es muy compleja.
4. Se coloca la pieza de trabajo en la estación de pintura para prepararla para la programación.
5. Se identifican los conjuntos de colisión.
6. Se guarda la pieza de trabajo como un archivo de biblioteca de RobotStudio.

3. Preparación de los programas

Pueden seguirse los siguientes pasos para crear un programa de pintura sobre una pieza de trabajo.

1. Se crea un nuevo programa desde cero o a partir de un programa existente.
2. Se asocia la biblioteca de piezas de trabajo y los objetos de trabajo con el programa.
3. Se carga el programa en RobotStudio.
4. Se crean los objetivos y trayectorias de pintura sobre la pieza.
5. Se define las tablas de pinceles.

4. Simulación de la célula robótica de pintura

La simulación se emplea para confirmar la correcta ejecución de los programas de robot de cada uno de los controladores de los robots de la estación. Los programas pueden ajustarse en cuanto a velocidad, eficacia y tiempo de ciclo.

5. Realización de los análisis necesarios y se obtienen los informes

El consumo de pintura estimado y los resultados de la simulación están disponibles después de la simulación.

6. Transferencia de los programas de robot de las estaciones virtuales a la estación real

Una vez se ha finalizado el programa de pintado, este podrá ser transferido y probado en la línea de producción real.

7. Edición de los programas de pintura en la estación real

Una vez transferidos a la estación real, los programas de pintado pueden ser editados sin detener la producción de la línea.

Durante el desarrollo del presente proyecto se intentó emplear el Painting PowerPac para mejorarlo, pero la información que se detalla en el único manual existente no es suficiente para aprender a utilizarlo desde cero, por lo que solo se han empleado algunas características del mismo como por ejemplo la herramienta **Split Part by**, nombrada anteriormente, y la herramienta **Insert Paint Stroke**, expuesta en el apartado [4.3.7 Creación de puntos objetivo y programación de trayectorias, procedimientos y rutinas](#).

En el presente proyecto también se ha empleado el complemento **Equipment Builder**, descargado mediante la herramienta **RobotApps**, la cual ofrece una serie de complementos, modelos CAD, componentes inteligentes, etc. creados por usuarios, que pueden descargarse de forma gratuita.

Este completo permite crear de una manera sencilla e intuitiva vallas, paredes, transportadores, pallets y herramientas de vacío, como se muestra en la Figura 3.157.

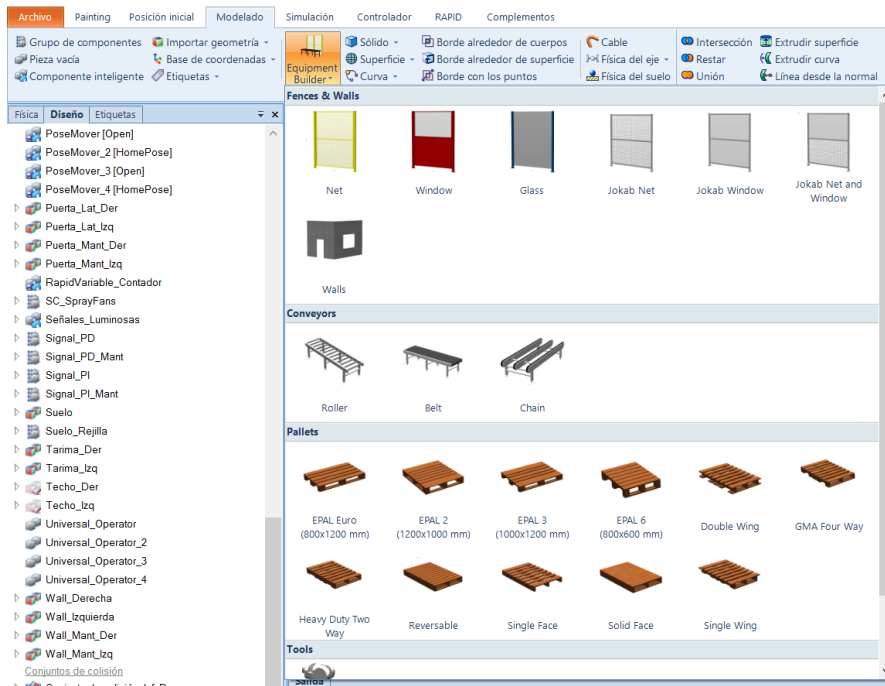


Figura 3.157 - Complemento Equipment Builder

Estos componentes se crean especificando sus dimensiones y características en el panel que se abre al seleccionar uno de ellos. (Figura 3.158)

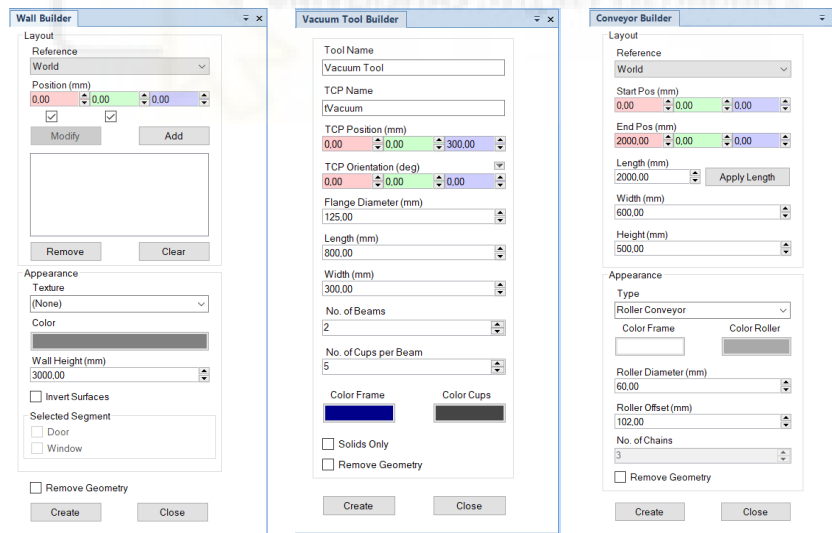


Figura 3.158 - Ejemplo de los paneles del complemento Equipment Builder

3.2 Programación básica en RAPID

Un robot industrial puede definirse como un manipulador multifuncional reprogramable, de forma que permite una adaptación rápida y económica a diferentes situaciones o aplicaciones.

La programación de un robot puede definirse como el procedimiento mediante el cual el usuario le comunica al robot la secuencia de acciones que debe realizar de forma automática para llevar a cabo la tarea requerida. La mayoría de estas acciones consisten en moverse entre puntos del espacio declarados, lectura/escritura de señales y manipulación de objetos del entorno de trabajo.

Se entiende de esta forma que existe una relación directa entre las características y posibilidades del sistema de programación y las del robot en sí mismo.

Cada fabricante del sector robótico ha desarrollado su propio lenguaje de programación, válido únicamente para el control de sus propios robots.

El lenguaje RAPID (Robotics Application Programming Interactive Dialogue), desarrollado por la empresa ABB en 1994 para reemplazar el lenguaje de programación ARLA, es un lenguaje de programación textual de alto nivel altamente estructurado.

Las características más importantes de este lenguaje son la utilización de procedimientos, funciones y rutinas TRAP, estas últimas para la gestión de interrupciones, el uso de rutinas parametrizables, la estructura completamente modular de los programas, la posibilidad de declarar rutinas y datos como locales o globales y el uso de estructuras predefinidas para especificar la configuración del robot y las características de la herramienta.

Un programa escrito en RAPID está compuesto por un conjunto de instrucciones que describen la actividad del robot. Se trata de instrucciones específicas para los distintos comandos que ejecutará el robot. Cada instrucción cuentan con un conjunto de argumentos asociados que definen completamente como se ejecutará dicha instrucción.

A continuación se expondrán en profundidad los aspectos más importantes del lenguaje RAPID para comprender correctamente la programación del presente proyecto.

3.2.1 Elementos básicos del lenguaje

Identificadores

Mediante los identificadores se nombran los módulos, rutinas, datos y etiquetas. Existen una serie de directrices a seguir para su correcta escritura:

- El primer carácter del identificador debe ser una letra.
- El resto de caracteres que lo forman únicamente pueden ser letras, dígitos o caracteres de subrayado (_).
- La longitud máxima de cualquier identificador no puede superar los 32 caracteres.

- Dos identificados con los mismos caracteres, aunque diferentes mayúsculas y minúsculas, serán considerados iguales.
- Las palabras expuestas en la siguiente Tabla 3.3 hacen referencia a elementos propios del lenguaje de RAPID, y por lo tanto son palabras reservadas que no podrán utilizarse como identificadores.

ALIAS	AND	BACKWARD	CASE
CONNECT	CONST	DEFAULT	DIV
DO	ELSE	ELSEIF	ENDFOR
ENDFUNC	ENDIF	ENDMODULE	ENDPROC
ENDRECORD	ENDTEST	ENDTRAP	ENDWHILE
ERROR	EXIT	FALSE	FOR
FROM	FUNC	GOTO	IF
INOUT	LOCAL	MOD	MODULE
NOSTEPIN	NOT	NOVIEW	OR
PERS	PROC	RAISE	READONLY
RECORD	RETRY	RETURN	STEP
SYSMODULE	TEST	THEN	TO
TRAP	TRUE	TRYNEXT	UNDO
VAR	VIEWONLY	WHILE	WITH
XOR			

Tabla 3.3 - Palabras reservadas que hacen referencia a elementos propios del lenguaje de RAPID

- Además de estas, existen también palabras reservadas para tipos de datos, datos de sistema, instrucciones y funciones.

Valores

En el lenguaje RAPID existen tres tipos de valores básicos:

- **Valores numéricos.** Estos se pueden expresar como un número entero o un número decimal. También puede expresarse de forma exponencial. Por ejemplo, 5, -10, 5.369, -74E5.
- **Valores lógicos o booleanos.** Estos pueden expresarse como TRUE o FALSE.
- **Valores de cadena de caracteres.** Estos pueden expresarse como secuencias de caracteres, caracteres de control y/o códigos de caracteres. Las cadenas de caracteres siempre deben escribirse entrecomilladas. La longitud máxima de una cadena de caracteres es de 80 caracteres. Por ejemplo, "Programa finalizado con éxito".

Basándose en estos tres tipos de valores básicos, realizando operaciones con ellos y combinándolos, se obtendrá una gran variedad de tipos de datos más complejos.

Comentarios

Los comentarios se emplean para realizar anotaciones o aclaraciones a lo largo de un programa, sin afectar de ninguna forma a la ejecución del mismo. Un comentario se inicia con un signo de exclamación (!) y termina con un salto de línea.

Expresiones

Las expresiones son evaluaciones de combinaciones de valores, operadores y/o llamadas a funciones que devuelven un valor. En el lenguaje RAPID existen varios tipos de expresiones.

- **Expresiones aritméticas.** Estas expresiones se emplean para evaluar valores numéricos. En la siguiente Tabla 3.4 se exponen las diferentes expresiones aritméticas, junto con los operadores aritméticos, que pueden emplearse en RAPID.

Operador	Funcionamiento	Tipos de operandos	Tipo de resultado
+	suma	num + num	num
+	suma	dnum + num	dnum
+	suma unitaria; conservar el signo	+num o +dnum o +pos	igual
+	suma de vectores	pos + pos	pos
-	resta	num - num	num
-	resta	dnum - dnum	dnum
-	resta unitaria; cambio de signo	-num o -pos	igual
-	resta unitaria; cambio de signo	-num o -dnum o -pos	igual
-	resta de vectores	pos - pos	pos
*	multiplicación	num * num	num
*	multiplicación	dnum * dnum	dnum
*	multiplicación de vectores escalares	num * pos o pos * num	pos
*	producto de vectores	pos * pos	pos
*	vinculación de rotaciones	orient * orient	orient
/	división	num / num	num
/	división	dnum / dnum	dnum
DIV	división entera	num DIV num	num
DIV	división entera	dnum DIV dnum	dnum
MOD	módulo de entero; resto	num MOD num	num
MOD	módulo de entero; resto	dnum MOD dnum	dnum

Tabla 3.4 - Operadores y expresiones aritméticas en RAPID

- **Expresiones lógicas.** Estas expresiones se emplean para evaluar valores lógicos. En la siguiente Tabla 3.5 se exponen las diferentes expresiones lógicas, junto con los operadores lógicos, que pueden emplearse en RAPID.

Operador	Funcionamiento	Tipos de operandos	Tipo de resultado
<	menor que	num < num	bool
<	menor que	dnum < dnum	bool
<=	menor que o igual a	num <= num	bool
<=	menor que o igual a	dnum <= dnum	bool
=	igual a	cualquiera = cualquiera	bool
>=	mayor que o igual a	num >= num	bool
>=	mayor que o igual a	dnum >= dnum	bool
>	mayor que	num > num	bool
>	mayor que o igual a	dnum > dnum	bool
<>	distinto de	cualquiera <> cualquiera	bool
AND	y	bool AND bool	bool
XOR	OR exclusivo	bool XOR bool	bool
OR	O bien	bool OR bool	bool
NOT	no unitario; negación	NOT bool	bool

Tabla 3.5 - Operadores y expresiones lógicas en RAPID

- **Expresiones de cadena de caracteres.** Estas expresiones se emplean para realizar operaciones con cadenas de caracteres. En RAPID solo existe la concatenación de cadenas de caracteres, empleando el operador +. Por ejemplo, "CADENA" + " " + "COMPLETA" devuelve el resultado "CADENA COMPLETA".

3.2.2 Datos de programa

En el lenguaje RAPID la información de los diferentes aspectos que conforman los programas se almacena en una gran variedad de datos de programa, según los distintos tipos de información que expresan. En RAPID no existe un número límite de datos declarados, ya que estos son creados por el usuario y se les asigna el nombre o identificador deseado, teniendo así un número de posibilidades infinitas, solo limitadas por la memoria disponible en el sistema.

La gran variedad de datos de los que dispone el lenguaje RAPID pueden clasificarse según diferentes aspectos. Una primera clasificación puede hacerse atendiendo a su descomposición, teniendo así:

- **Datos atómicos.** Estos datos no están definidos partiendo de ningún otro tipo y no pueden ser divididos en distintas partes o distintos componentes. Por ejemplo los datos *num*, como 9, 3.5, -2E2.
- **Datos de registro.** Se trata de datos compuestos con componentes con nombre y ordenados. Por ejemplos los datos *pos*, que expresan las coordenadas de una posición: X, Y y Z, como *pos1 := [20,15,0]* o *pos1 := [PosX,PosY,PosZ]*, estando PosX, PosY y PosZ definidas previamente.

- **Datos alias.** Este tipo de datos son equivalentes a otros tipos. Los datos con los mismos tipos de alias pueden usarse unos en sustitución de otros. Por ejemplo *tpnum* es un tipo de dato de alias de *num*.

Otra clasificación posible de los diferentes tipos de datos es atendiendo a su valor, teniendo así los siguientes tipos de datos:

- **Datos con valor.** Este tipo de datos representa alguna forma de valor, y por lo tanto pueden emplearse en operaciones basadas en valores, como inicialización, asignación, comprobaciones de igualdad y de diferencia, etc. Por ejemplo los datos *num*.
- **Datos de semivalor.** Estos datos pueden emplearse en operaciones basadas en valores, excepto la inicialización y la asignación. Los tipos de datos de señal *signalxx* (*signalai*, *signaldi*, *signalgi*, *signalao*, *signaldo*, *signalgo*) son del tipo de datos de semivalor.
- **Datos sin valor.** Los datos sin valor no pueden emplearse en operaciones basadas en valores. Por ejemplo el tipo de dato *iodev*.

En la descripción de un tipo de datos sólo se especifica cuando es un tipo de datos de semivalor o sin valor.

Excepto para los datos predefinidos y las variables de un bucle, es necesario declarar todos los datos que forman parte de los programas. Al declarar un dato se le asigna un identificador y se define su variabilidad y su ámbito.

El ámbito de un dato es el área dentro de la cual es visible dicho dato. Primeramente se puede diferenciar entre ámbito local y ámbito global. Si se añade la directiva LOCAL delante de la declaración de un dato, el ámbito de aplicación de este pasa a ser únicamente el módulo dentro del que se ha declarado. Por el contrario, si al declarar un dato no se añade dicha directiva opcional el dato tendrá un ámbito global, siendo su ámbito de aplicación el de todos los módulos del sistema. La directiva LOCAL sólo puede usarse en el nivel de módulo, no dentro de una rutina. Un ejemplo de declaración de datos de tipo *num* variables con ámbito local y global es el siguiente:

```
LOCAL VAR num local_variable;
```

```
VAR num global_variable;
```

Por otro lado se puede diferenciar entre ámbito de programa y ámbito de rutina. Todo dato declarado fuera de una rutina se denomina dato de programa y se le aplica las siguientes reglas de ámbito:

- Como se ha mencionado anteriormente, el ámbito de un dato de programa predefinido o global puede incluir cualquier módulo. Mientras que el ámbito de un dato de programa local comprende el módulo en el que se declara.
- Dentro de su ámbito, si un dato de programa local tiene el mismo identificador que un dato o rutina global (incluidas las instrucciones, las rutinas predefinidas y los datos) estos últimos no serán visibles, es decir, si tienen el mismo identificador, dentro del ámbito local solo serán visibles los datos locales.
- No está permitido que los datos de programa tengan el mismo nombre que otros datos o rutinas del mismo módulo.
- No está permitido que los datos de programa globales tengan el mismo nombre que otros datos globales o rutinas de otros módulos.

Los datos que se declaran dentro de una rutina se denominan datos de rutina. Al igual que a los datos de programa también se les aplican unas reglas de ámbito:

- El ámbito de un dato de rutina únicamente comprende la rutina en la que se ha declarado.
- Dentro de su ámbito, si un dato de rutina posee el mismo identificador que un dato o rutina global (incluidas las instrucciones, las rutinas predefinidas y los datos) estos últimos quedarán ocultos.
- No está permitido que los datos de rutina tengan el mismo nombre que otros datos o etiquetas de la misma rutina.

Para comprender mejor estas reglas sobre los diferentes ámbitos de los datos se muestra el siguiente ejemplo de la Figura 3.159.

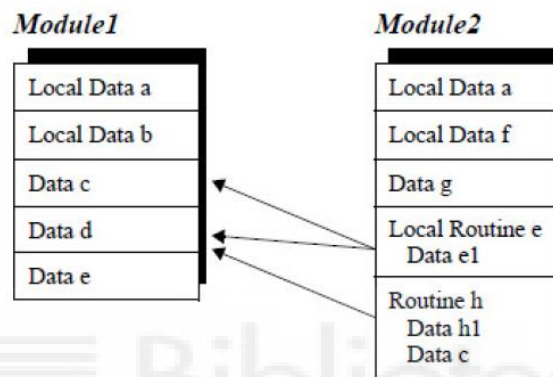


Figura 3.159 - Ejemplo sobre las reglas de los diferentes ámbitos de los datos

En el Module1 se tiene los datos de programa locales a y b, solo visibles en dicho módulo, y los datos de programa globales c, d y e, visibles tanto en el Module1 como en el Module2. En el Module2 se tienen los datos de programa locales a y f, solo visibles en dicho módulo, el dato de programa global g, visible en ambos módulos, la rutina local e, dentro de la cual se encuentra el dato de rutina e1, y la rutina global h, dentro de la cual se encuentran los datos de rutina h1 y c.

Expuesto esto, y siguiendo las reglas expuestas anteriormente, desde la rutina e, estarán disponibles, o dicho de otro modo, serán visibles los datos c y d del Module1 y los datos a, f, g y e1 del Module2. Quedando no disponibles los datos a y b del Module1 por ser locales, el dato e del Module1 por tener el mismo identificador que la rutina y los datos h1 y c del Module2 por ser datos de rutina.

Desde la rutina h, estará visible solamente el dato d del Module1 y los datos a, f, g, h1 y c del Module2. Quedando no disponibles los datos a y b del Module1 por ser locales, el dato e del Module1 por tener el mismo identificador que una rutina del módulo al que pertenece, el dato c por tener el mismo identificador que un dato dentro de la propia rutina y el dato e1 por ser un dato de rutina.

Según la variabilidad con la que se definen los datos estos se pueden clasificar en tres grupos:

- **Variables (VAR).** Las variables contienen valores de datos. La asignación de nuevos valores a las variables puede realizarse durante la ejecución del programa. Al detener el

programa y reanudarlo, la variable conserva su valor actual, pero si se mueve el puntero de programa a Main el valor del dato de la variable se reinicia, no conservándose el actual.

Las variables se declaran con la palabra clave VAR, seguida del tipo de dato y del identificador. Por ejemplo, VAR num Color, definiéndose así una nueva variable llamada Color de tipo de dato de valor numérico.

Las variables pueden declararse como variables locales o globales y tanto en el ámbito de programa y como en el ámbito de rutina.

Es posible asignar un formato de matriz, de grado 1, 2 o 3, a variables de cualquier tipo, especificando las dimensiones en la declaración. Por ejemplo, VAR pos Pos_Caja{10,20}.

Las variables pueden inicializarse mediante un valor básico, por ejemplo 2, TRUE o "Completado", según el tipo de dato, o una expresión constante, por ejemplo, a + b + 5, siempre que hayan sido declaradas previamente a y b.

En el caso de no inicializarse la variable con ningún valor o expresión, se establece un valor de 0 en el caso de dato tipo num, un valor de FALSE en el caso de un dato de tipo bool o una cadena de caracteres vacía "" en el caso de un dato de tipo string.

Para asignar valores a una variable, ya sea un valor de inicialización o un nuevo valor dentro de la ejecución del programa se emplea la instrucción :=, mostrándose esto en los siguientes ejemplos:

VAR num Color := 25;	Se inicializa con un valor de 25 el dato de tipo num variable global Color.
Color := 3;	Se cambia el valor del dato Color a 3 dentro del programa.
LOCAL VAR string Nombre := "Arturo";	Se inicializa el dato Nombre de tipo string variable local con la cadena de caracteres "Arturo".
LOCAL VAR string Apellido := "Lopez";	Se inicializa el dato Apellido de tipo string variable local con la cadena de caracteres "Lopez".
LOCAL VAR string NombreCompleto;	Se declara el dato NombreCompleto de tipo string variable local vacío, sin valor inicial.
NombreCompleto := Nombre + " " + Apellido;	Se cambia el valor del dato NombreCompleto a "Arturo Lopez" dentro de la ejecución del programa

- **Persistentes (PERS).** Pueden describirse como variables persistentes. Se trata, por tanto, de variables prácticamente iguales que las variables normales, salvo por la diferencia de que la actualización del valor de una variable persistente, a lo largo de la ejecución del programa, provoca la actualización del valor de inicialización de la declaración de dicha variable, es decir, la variable persistente recuerda el valor actual que se le haya dado, incluso si el programa es detenido y se mueve el puntero de programa a Main. Se debe tener en cuenta que debido a aspectos de rendimiento, esta actualización no tiene lugar durante la ejecución del programa, si no que el valor inicial se actualiza al guardar el módulo o también al editar el programa.

Las variables persistentes se declaran con la palabra clave PERS, seguida del tipo de dato y del identificador. Por ejemplo, PERS num Color, definiéndose así una nueva variable persistente llamada Color de tipo de dato de valor numérico.

Estas variables solo pueden ser declaradas en el ámbito de programa, no permitiéndose su declaración en el ámbito de rutina. Por otro lado estas variables persistentes pueden declararse como globales o como locales.

Como en el caso de las variables normales, es posible asignar un formato de matriz, de grado 1, 2 o 3, especificando las dimensiones en la declaración.

En el caso de las variables persistentes globales, estas pueden ser inicializadas con un valor o este puede omitirse. En el caso de que se desee inicializar una variable persistente, el valor de inicialización debe ser un solo valor, es decir, sin referencias a datos ni operandos, o un solo agregado cuyos miembros sean valores simples o agregados simples.

Como en el caso de las variables normales, para asignar valores a una variable persistente, ya sea un valor de inicialización o un nuevo valor dentro de la ejecución del programa se emplea la instrucción := . A continuación se muestran algunos ejemplos:

PERS num Color := 25;	Se inicializa con un valor de 25 el dato de tipo num variable persistente global Color.
Color := 3;	Se cambia el valor del dato Color a 3 dentro del programa, por lo que cuando se reinicie el programa el valor de Color será 3 en lugar de 25.
PERS string Año := "2021";	Se inicializa el dato Año de tipo string variable persistente global con la cadena de caracteres "2021".
PERS string Mes := "Enero";	Se inicializa el dato Mes de tipo string variable persistente global con la cadena de caracteres "Enero".
PERS string Mes_y_Año;	Se declara el dato Mes_y_Año de tipo string variable persistente global vacío, sin valor inicial.
Mes_y_Año := Mes + "del" + Año;	Se cambia el valor del dato Mes_y_Año a "Enero del 2021" dentro de la ejecución del programa, de forma que cuando se reinicie el programa este sea su valor inicial.

- **Constantes (CONST).** Las constantes representan valores, como las variables, pero en este caso fijos, es decir, las constantes deben inicializarse con un valor en el momento de la declaración y no es posible cambiar este valor posteriormente. Así es posible asegurarse completamente de que el valor del dato de la constante permanece inalterable a lo largo de la ejecución del programa.

Las constantes se declaran con la palabra clave CONST, seguida del tipo de dato, del identificador y de la asignación de un valor. Por ejemplo, CONST num Color := 3, definiéndose así una nueva constante llamada Color de tipo de dato de valor numérico cuyo valor es igual a 3.

Al igual que en el caso de las variables, es posible asignar un formato de matriz, de grado 1, 2 o 3, especificando las dimensiones en la declaración.

Como en el caso de las variables para asignar el valor de inicialización a una constante se emplea la instrucción := . A continuación se muestran algunos ejemplos:

CONST num Masa_H1 := 5;	Se inicializa con un valor de 5 el dato de tipo num constante global Masa_H1.
-------------------------	---

CONST num Masa_H2 := 10;	Se inicializa con un valor de 10 el dato de tipo num constante global Masa_H2.
CONST num MH1_H2 := Masa_H1 + Masa_H2;	Se inicializa el dato MH1_H2 de tipo num constante global con el valor de la suma de las constantes Masa_H1 y Masa_H2.

En el lenguaje de RAPID existen 105 tipos de datos diferentes. A continuación se realizará una breve explicación de los tipos de datos empleados en el presente proyecto.

num

Este tipo de dato se emplea con valores numéricos. Los valores que pueden ser asignados a este tipo de dato son los que ya se han expuesto para los valores numéricos en el apartado anterior [3.2.1 Elementos básicos del lenguaje](#). Los enteros entre -8388607 y +8388608 se almacenan siempre como enteros exactos. Los números con decimales son sólo números aproximados y, por tanto, no deben utilizarse en comparaciones de tipo igual a (=) ni distinto de (<>). En el caso de las divisiones y las operaciones que utilizan números con decimales, el resultado también será un número con decimales, es decir, no un entero exacto. El sistema cuenta con ciertos datos de tipo *num* predefinidos. Por ejemplo, se define la constante pi (π).

Por ejemplo:

```
VAR num Color;
```

```
VAR num Secuencia_Colores{100};
```



string

Este tipo de dato se emplea con cadenas de caracteres. Los valores que pueden ser asignados a este tipo de dato son los que ya se han expuesto para los valores de cadena de caracteres en el apartado anterior [3.2.1 Elementos básicos del lenguaje](#). Si desea incluir comillas o una barra invertida dentro de la cadena de caracteres, estos caracteres se deben escribir dos veces. Las cadenas de caracteres pueden contener cualquiera de los caracteres especificados en la norma ISO 8859-1 (Latin-1), además de caracteres de control (caracteres ajenos al estándar ISO 8859-1 (Latin-1) y con códigos numéricos de 0 a 255). El sistema dispone de varias constantes de cadena de caracteres predefinidas, que pueden usarse junto con las funciones para cadenas de caracteres:

STR_DIGIT, STR_UPPER, STR_LOWER, STR_WHITE, Letra "eth" islandesa, letra Y con acento agudo, letra "thorn" islandesa y las constantes siguientes, que ya están definidas en el sistema:

```
diskhome := "HOME:";
```

```
ram1disk := "HOME:";
```

```
disktemp := "TEMP:";
```

```
flp1 := "flp1:";
```

```
stSpace := " ";
```

```
stEmpty := "";
```

Por ejemplo:

```
VAR string date;
```

iodev

Este tipo de dato se emplea para dispositivos de E/S y archivos. Los datos de tipo *iodev* contienen una referencia a un archivo o a dispositivos de E/S. Pueden conectarse a la unidad física mediante la instrucción **Open** y utilizarse a continuación para operaciones de lectura y escritura. Este tipo de datos es un tipo de dato sin valor.

Por ejemplo:

```
VAR iodev infile;
```

jointtarget

Este tipo de dato se emplea para definir las posiciones individuales de los distintos ejes, tanto de los del robot como de los externos. *jointtarget* se utiliza para definir la posición a la que se moverán los ejes del robot y los ejes externos al ejecutar una instrucción *MoveAbsJ*.

Se trata por tanto de un tipo de dato de registro compuesto por los siguientes componentes:

- **robax (robot axes)**. Este a su vez es un tipo de dato *robjoint*. Especifica las posiciones de eje de los ejes del robot, en grados. La posición del eje se define como la rotación en grados del eje (brazo) correspondiente en sentido positivo o negativo a partir de la posición de calibración del eje.
- **extax (external axes)**. Este a su vez es un tipo de dato *extjoint*. Especifica la posición de los ejes externos, para cada eje independiente (*eax_a*, *eax_b* ... *eax_f*). Para los ejes de rotación, la posición se define como la rotación en grados de la posición de calibración. Para los ejes lineales, la posición se define como la distancia en mm existente respecto de la posición de calibración. El valor 9E9 se utiliza para definir los ejes que no están conectados. En el caso de los ejes que tengan datos de posición válidos pero que no estén activados no se realiza ningún movimiento pero tampoco se genera ningún error. Si algún eje externo está funcionando en el modo independiente y el robot y sus ejes externos deben realizar algún nuevo movimiento, los datos de posición del eje externo en el modo independiente no deben ser 9E9 sino algún valor arbitrario (no utilizado por el sistema).

Por ejemplo:

```
CONST jointtarget Pos_Inicial2 := [[0,20,-20,0,-45,0],[6750,9E+09,9E+09,9E+09,9E+09,0]];
```

robtarget

Este tipo de dato se emplea para definir la posición del robot y de los ejes adicionales. Los datos de posición se utilizan en las instrucciones de movimiento para indicar la posición hacia la que deben desplazarse los ejes del robot y los ejes adicionales. Debido a que el robot puede alcanzar una misma posición con métodos diferentes, también se especifica la configuración de los ejes. De esta forma, se definen los valores de los ejes si por algún motivo resultan ambiguos. La posición se define partiendo del sistema de coordenadas del objeto de trabajo, incluidos los posibles desplazamientos de programa. Si no se ha especificado ningún objeto de trabajo, se utiliza el sistema de coordenadas mundo.

Si la posición se programa con un objeto de trabajo distinto del utilizado en la instrucción, el robot no se moverá de la forma esperada. Debe asegurarse que se está usando el mismo objeto de trabajo que el utilizado al programar las instrucciones de movimiento. Un uso incorrecto puede causar accidentes o daños en el robot o en otros equipos.

Se trata por tanto de un tipo de dato de registro compuesto por los siguientes componentes:

- **trans (translation)**. Este a su vez es un tipo de dato *pos*. Especifica la posición (x, y, z) del punto central de la herramienta, expresado en mm.
- **rot (rotation)**. Este a su vez es un tipo de dato *orient*. Especifica la orientación de la herramienta, expresada en forma de un cuaternio (q1, q2, q3 y q4).
- **robconf (robot configuration)**. Este a su vez es un tipo de dato *confdata*. Especifica la configuración de ejes del robot (cf1, cf4, cf6 y cfx). Esto se define en forma del cuarto de revolución actual de los ejes 1, 4 y 6. El primer cuarto de revolución positivo, de 0 a 90° se define como 0. El significado del componente cfx depende del tipo de robot.
- **extax (external axes)**. Este a su vez es un tipo de dato *extjoint*. Especifica la posición de los ejes externos, para cada eje independiente (eax_a, eax_b ... eax_f). Para los ejes de rotación, la posición se define como la rotación en grados de la posición de calibración. Para los ejes lineales, la posición se define como la distancia en mm existente respecto de la posición de calibración. El valor 9E9 se utiliza para definir los ejes que no están conectados. En el caso de los ejes que tengan datos de posición válidos pero que no estén activados no se realiza ningún movimiento pero tampoco se genera ningún error. Si algún eje externo está funcionando en el modo independiente y el robot y sus ejes externos deben realizar algún nuevo movimiento, los datos de posición del eje externo en el modo independiente no deben ser 9E9 sino algún valor arbitrario (no utilizado por el sistema).

Por ejemplo:

```
LOCAL          CONST          robtarget          Paint_10:=[[3650,3.292,1125],[0.720581562,0.02943484,-  
0.692071672,0.030538552],[0,0,0,0],[3750,9E+09,9E+09,9E+09,0]];
```

```
CONST          robtarget          Aprox_Bajo_Capo:=[[4400,3.292,1125],[0.720581562,0.02943484,-  
0.692071672,0.030538552],[0,0,0,0],[3750,9E+09,9E+09,9E+09,0]];
```

3.2.3 Módulos

Los programas de RAPID se dividen en módulos, dentro de los cuales se encuentran los datos de programa y las diferentes rutinas. Los programas simples suelen utilizar un único módulo. Sin embargo, en entornos de programación más complejos, algunos procedimientos estándar utilizados por muchos programas diferentes pueden ser colocados en un módulo separado.

Existen dos tipos de módulos, los módulos de programa y los módulos de sistema, Figura 3.160.

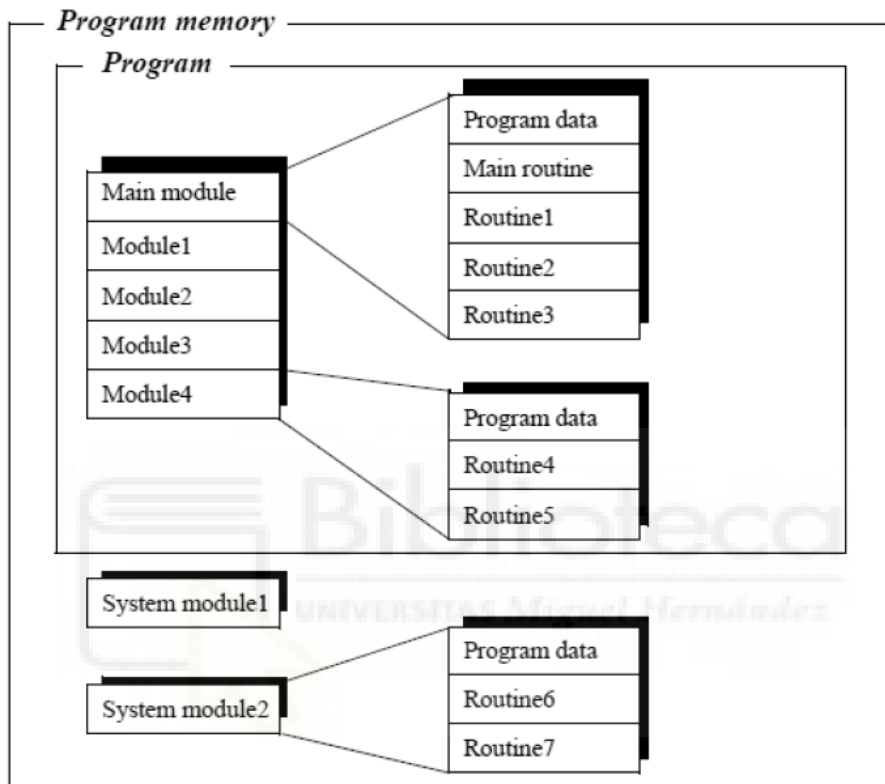


Figura 3.160 - Tipos de módulos de RAPID

Módulos de programa

Los módulos de programa están compuestos por datos de programa y una o varias rutinas diferentes. Es posible guardar cada módulo por separado o el programa en su totalidad. Los módulos de programa se guardan con la extensión de archivo .mod, por ejemplo, Module1.mod.

El programa puede contener muchos módulos, pero sólo uno de ellos puede contener el procedimiento de entrada, un procedimiento global denominado Main. La ejecución del programa implica la ejecución del procedimiento de entrada.

Para el controlador del robot no existe ninguna diferencia entre que el programa esté compuesto de uno o de varios módulos. El motivo de utilizar distintos módulos de programa es únicamente facilitar a los programadores la comprensión del programa y la reutilización de código.

Módulos de sistema

Los módulos de sistema se emplean para definir datos y rutinas comunes y específicos del sistema, como por ejemplo, herramientas. Los datos y procedimientos que deban permanecer en el sistema incluso al cambiar de programa deben ubicarse en un módulo de sistema.

Estos módulos no se incluyen al guardarse un programa, lo que significa que cualquier cambio que se haga en un módulo de sistema afectará a todos los programas almacenados actualmente en la memoria de programas o que se carguen posteriormente en ella. Los módulos de sistema se guardan con la extensión de archivo .sys, por ejemplo, ctsync.sys.

Mediante la declaración de un módulo se especifica el identificador y los atributos del mismo. Los atributos de un módulo solo pueden añadirse fuera de línea. No está permitido que un módulo tenga el mismo identificador que otro módulo o una rutina o un dato globales. Los atributos presentes en un módulo son los siguientes:

- **Tipo de módulo.** Se debe seleccionar si se quiere especificar un módulo de programa o un módulo de sistema. En la versión de RobotStudio empleada en el presente proyecto esto no se considera un atributo como tal, pero en versiones anteriores si existía el atributo SYSMODULE, de forma que si se especificaba se trataba de un módulo de sistema, y si no, se trataba de un módulo de programa.
- **NOSTEPIN.** Este atributo implica que no se permite la activación del módulo durante la ejecución paso a paso del programa.
- **VIEWONLY.** Implica que es un módulo de solo lectura, es decir, no se permite la modificación del mismo.
- **READONLY.** Implica que no se permite la modificación del módulo, pero sí la eliminación del atributo.
- **NOVIEW.** Implica que no se permite la visualización del módulo, sino solamente su ejecución.

En la Figura 3.161 puede verse el cuadro de dialogo para la declaración de un nuevo módulo.

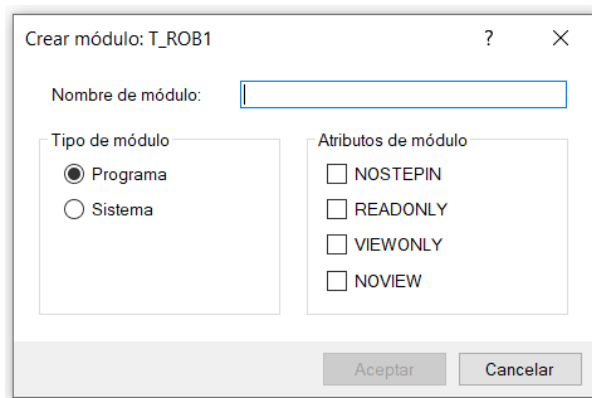


Figura 3.161 - Cuadro de dialogo Crear módulo

3.2.4 Rutinas

Una rutina (Figura 3.162) puede definirse como la parte de un programa que contiene la declaración de un conjunto de instrucciones y datos con el fin de realizar una tarea o función concreta. Además de esto, las rutinas pueden contener también un gestor de ejecución hacia atrás (sólo en el caso de los procedimientos), un gestor de errores y un gestor de deshacer.

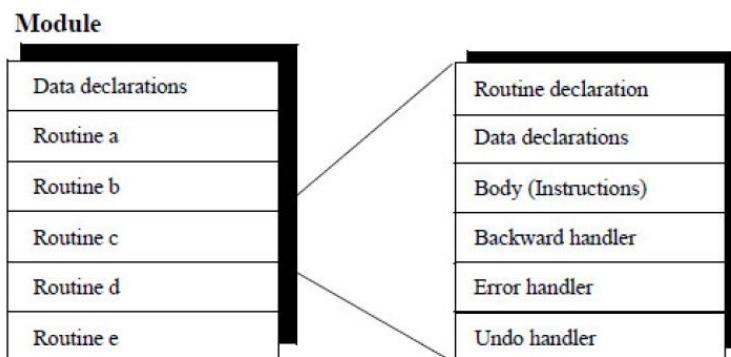


Figura 3.162 - Rutinas de RAPID

En el lenguaje de programación RAPID existen tres tipos de rutinas, también denominados subprogramas: procedimientos, funciones y rutinas TRAP.

El ámbito de una rutina es el área dentro de la cual es visible dicha rutina. Como en el caso de los datos de programa, se diferencia entre ámbito local y ámbito global. Si se añade la directiva LOCAL delante de la declaración de una rutina, el ámbito de aplicación de esta pasa a ser únicamente el módulo dentro del que se ha declarado. Por el contrario, si al declarar una rutina no se añade dicha directiva opcional la rutina tendrá un ámbito global, siendo su ámbito de aplicación el de todos los módulos del sistema. A las rutinas se les aplican las siguientes reglas de ámbito:

- Como ya se ha mencionado, el ámbito de una rutina global puede incluir cualquier módulo del programa, mientras que el ámbito de una rutina local solo comprende el módulo en el que se ha declarado.
- Dentro de su ámbito, si una rutina local tiene el mismo identificador que una rutina global o un dato, estos últimos no serán visibles, es decir, si tienen el mismo identificador, dentro del módulo en el que se ha declarado la rutina solo será visible con ese identificador, dicha rutina local.
- Dentro de su ámbito, las rutinas suponen la ocultación de instrucciones, rutinas predefinidas y datos que tengan el mismo identificador.
- No está permitido que una rutina tenga el mismo identificador que otra rutina, otro dato u otro tipo de dato del mismo módulo.
- No está permitido que una rutina global tenga el mismo identificador que un módulo o una rutina global, un dato global o un tipo de dato global de otro módulo.

Los parámetros de la declaración de una rutina establecen los argumentos que deben/pueden entregarse al llamar a la rutina. Todos los parámetros se declaran entre paréntesis a continuación del nombre de la rutina. Esta declaración contiene el tipo de dato, el tipo de parámetro y el identificador de cada parámetro. El parámetro recibe su valor de la llamada a la rutina y actúa como una variable dentro de la misma. Los parámetros controlan el comportamiento de la rutina y permiten utilizar una misma rutina para distintas operaciones. Al llamar a una rutina, el orden de los parámetros es importante para asignar el valor correcto al parámetro correcto.

Hay cuatro tipos de parámetros:

- **Parámetros normales.** Estos parámetros se utilizan sólo como datos de entrada y se procesan como variables de la rutina. La modificación del contenido de esta variable en la rutina no cambia el valor del argumento correspondiente.
- **INOUT.** Especifica que el argumento correspondiente debe ser una variable (un entero, un elemento o un componente) o una variable persistente entera cuyo valor puede ser modificado por la rutina.
- **VAR.** Especifica que el argumento correspondiente debe ser una variable (un entero, un elemento o un componente) cuyo valor puede ser modificado por la rutina.
- **PERS.** Especifica que el argumento correspondiente debe ser una variable persistente entera cuyo valor puede ser modificado por la rutina.

Por ejemplo:

```
PROC routine1 (num in_par, INOUT num inout_par, VAR num var_par, PERS num pers_par)
```

La actualización de parámetros de tipo INOUT, VAR o PERS, implica la actualización de los argumentos en sí, lo que da la posibilidad de utilizar argumentos para devolver valores a la rutina desde la que se le llama.

Un parámetro puede ser opcional, lo cual implica que puede ser omitido de la lista de argumentos de la llamada a la rutina. Esto se indica precediendo al parámetro por una barra invertida (\).

Por ejemplo:

```
PROC routine2 (num required_par \num optional_par)
```

En RAPID no se permite el anidamiento de las declaraciones de rutinas, es decir, no se permite declarar una rutina dentro de otra rutina.

Procedimientos (PROC)

Los procedimientos no devuelven ningún valor y se utilizan en el contexto de las instrucciones.

Los procedimientos se declaran con la palabra clave PROC seguida del identificador del procedimiento, los parámetros del mismo y el código de programa, es decir la sucesión de

instrucciones, que debe ser ejecutado por el procedimiento. En el caso de que se declare un procedimiento sin parámetros, este se declara mediante la palabra clave PROC seguida de unos paréntesis (). Por ejemplo:

```
PROC Procedimiento1(num Carga, string Material, VAR num Total)
```

```
PROC Procedimiento2()
```

La ejecución de un procedimiento finaliza explícitamente con una instrucción RETURN o implícitamente cuando se alcanza el final del procedimiento, declarado mediante la palabra clave ENDPROC, BACKWARD, ERROR o UNDO.

Como ya se explicó en el apartado [3.2.2 Datos de programa](#), los datos, de cualquier tipo, declarados dentro de un procedimiento solo son visibles en dicho procedimiento.

La llamada a un procedimiento se debe realizar desde otro procedimiento, salvo en el caso del procedimiento main, al que se llama directamente cuando se inicia el programa. En la llamada al procedimiento no se usan paréntesis. Por ejemplo:

```
PROC main()
```

```
    PROC Procedimiento1 5, "Acero", 1;
```

```
ENDPROC
```

Funciones (FUNC)

Las funciones devuelven un valor de un tipo concreto y se utilizan en el contexto de las expresiones. Las funciones pueden devolver valores de cualquier tipo de datos, excepto matrices. En todos los lugares en los que sea posible utilizar un valor, también puede utilizarse una función que devuelva un valor del mismo tipo de dato. El valor devuelto por una función puede asignarse a una variable.

La finalidad de las funciones es la de simplificar el código, sustituyendo varias líneas de código con sentencias complejas, usadas habitualmente, por una sola línea de llamada a la función.

Por ejemplo, en el siguiente código la función Offs (Point XOffset YOffset ZOffset):

```
p20 := Offs(p10, 100, 200, 300);
```

Puede reemplazar al siguiente código más largo y complejo:

```
p20 := p10;
```

```
p20.trans.x := p20.trans.x + 100;
```

```
p20.trans.y := p20.trans.y + 200;
```

```
p20.trans.z := p20.trans.z + 300;
```

En el lenguaje RAPID existen 216 funciones predefinidas. Por otro lado, si el usuario quiere declarar una función se debe utilizar la palabra clave FUNC seguida del tipo de dato que se desea que devuelva, el identificador de la función, los parámetros de la misma y el código de programa,

es decir la sucesión de instrucciones, que debe ser ejecutado por la función para obtener el valor de retorno. En el caso de que se declare una función sin parámetros, esta se declara mediante la palabra clave FUNC seguida de unos paréntesis ().

La evaluación de una función finaliza con la instrucción RETURN seguida del identificador del valor que se desea que devuelva la función. El valor que se indica que se devuelva debe ser del mismo tipo de dato que el tipo de dato que se ha especificado en la declaración de la función. El código de una función se finaliza con la palabra clave ENDFUNC. Por ejemplo:

```
FUNC num veclen (pos vector)
```

```
    RETURN Sqrt(Pow(vector.x,2)+Pow(vector.y,2)+Pow(vector.z,2));
```

```
ENDFUNC
```

Como en cualquier rutina, los datos, de cualquier tipo, declarados dentro de una función solo son visibles en dicha función. La llamada a una función se debe realizar desde dentro de un procedimiento.

Rutinas TRAP

Las rutinas TRAP proporcionan una forma de responder a las interrupciones. Estas rutinas pueden asociarse a una interrupción determinada. Cuando se produce dicha interrupción, se ejecuta automáticamente la rutina TRAP correspondiente. Las rutinas TRAP no pueden ejecutarse explícitamente desde el programa.

Las rutinas TRAP se declaran con la palabra clave TRAP seguida del identificador de la misma y el código de programa, es decir la sucesión de instrucciones, que debe ser ejecutado por la rutina TRAP para responder a la interrupción que ha provocado su ejecución. Al inicio de la ejecución de la rutina TRAP, el programa abandona su nivel de ejecución base.

La ejecución de una rutina TRAP finaliza explícitamente con la instrucción RETURN o implícitamente cuando se alcanza el final de la rutina TRAP, indicado mediante la palabra clave ENDTRAP, ERROR o UNDO. La ejecución continúa en el punto de programa en el que se produjo la interrupción. Un ejemplo muy simple sería:

```
TRAP feeder_empty
```

```
    wait_feeder;
```

```
    RETURN;
```

```
ENDTRAP
```

A continuación se expondrán las funciones predefinidas empleadas en el presente proyecto.

Funciones de posición

Offs

Offs (Point XOffset YOffset ZOffset)

Esta función se emplea para añadir un offset, es decir un desplazamiento adicional, en el sistema de coordenadas de objeto a una posición de robot.

Los argumentos de esta función son los siguientes:

Point (robtargt). Especifica los datos de la posición a la que se le añade el offset.

XOffset (num). Especifica el desplazamiento en mm en la dirección del eje x del sistema de coordenadas de objeto.

YOffset (num). Especifica el desplazamiento en mm en la dirección del eje y del sistema de coordenadas de objeto.

ZOffset (num). Especifica el desplazamiento en mm en la dirección del eje z del sistema de coordenadas de objeto.

El valor de retorno de esta función será un dato de tipo *robtargt*, indicando los datos de la posición desplazada.

Por ejemplo:

```
MoveL Offs(Punt_Paro,150,0,-40),v1500,fine,RB1000_WSC_70_700_200\WObj:=WO_In_Int;
```

CRobT

CRobT ([\TaskRef][\TaskName] [\Tool] [\WObj])

Esta función se emplea para leer la posición actual de los ejes del robot y los ejes externos, es decir, el *robtargt* actual.

Los argumentos de esta función son los siguientes:

[\TaskRef] (taskid). Este argumento opcional especifica la identidad de la tarea de programa desde la cual debe leerse *robtargt*. La identificación de la tarea será "nombre_tarea"+"ID". Por ejemplo, para la tarea T_ROB1 la identificación de la tarea es T_ROB1Id.

[\TaskName] (string). Este argumento opcional especifica el nombre de la tarea de programa desde la cual debe leerse *robtargt*. Por ejemplo, para T_ROB1 sería "T_ROB1".

Si no se utiliza ninguno de los argumentos, ni **[\TaskRef]** ni **[\TaskName]**, se selecciona por defecto la tarea actual.

[\Tool] (tooldata). Este argumento opcional especifica la herramienta utilizada para calcular la posición actual del robot. Si se omite se utiliza la herramienta activa actualmente.

[\WObj] (wobjdata). Este argumento opcional especifica el objeto de trabajo, es decir, el sistema de coordenadas con el que está relacionada la posición actual del robot devuelta por la función. Si se omite se utiliza el objeto de trabajo activo actualmente.

El valor de retorno de esta función será un dato de tipo *robtarget*, indicando la posición (x, y, z), la orientación (q1 ... q4) y la configuración de los ejes del robot, y la posición de los ejes externos teniendo en cuenta la herramienta especificada, el objeto de trabajo y el sistema de coordenadas activo.

Por ejemplo:

```
Punt_Paro:=CRobT(\Tool:=RB1000_WSC_70_700_200\WObj:=WO_In_Int);
```

CJointT

```
CJointT([\TaskRef][\TaskName])
```

Esta función se emplea para leer los ángulos actuales de los ejes del robot y de los ejes externos, es decir, el *jointtarget* actual.

Los argumentos de esta función son los siguientes:

[\TaskRef] (taskid). Este argumento opcional especifica la identidad de la tarea de programa desde la cual debe leerse *jointtarget*. La identificación de la tarea será "nombre_tarea"+"ID".

[\TaskName] (string). Este argumento opcional especifica el nombre de la tarea de programa desde la cual debe leerse *jointtarget*. Por ejemplo, para T_ROB1 sería "T_ROB1".

Si no se utiliza ninguno de los argumentos, ni *\TaskRef* ni *\TaskName*, se selecciona por defecto la tarea actual.

El valor de retorno de esta función será un dato de tipo *jointtarget*, indicando los ángulos actuales, en grados, de los ejes del robot en el lado del brazo y los valores actuales de los ejes externos, en mm para los ejes lineales y en grados para los ejes de rotación. Estos valores están relacionados con la posición de calibración de los ejes.

Por ejemplo:

```
jointPunt_Paro:=CJointT(\TaskRef:=T_ROB1Id);
```

Función de velocidad

MaxRobSpeed

```
MaxRobSpeed()
```

Esta función se emplea para devolver el límite de velocidad máxima establecido para el TCP del robot.

Esta función no posee ningún argumento.

El valor de retorno de esta función será un dato de tipo *num*, indicando la velocidad máxima que se permite alcanzar al TCP (*vmax*), en mm/s, para el robot utilizado y valores normales prácticos para el TCP.

Si desea cambiarse el valor del límite de velocidad máxima del TCP del robot, esto puede hacer desde la pestaña **Controlador**, seleccionando el controlador del robot, grupo **Configuración**, configuración, **Motion** y tipo **Motion Planner**, como se indica en la siguiente Figura 3.163.

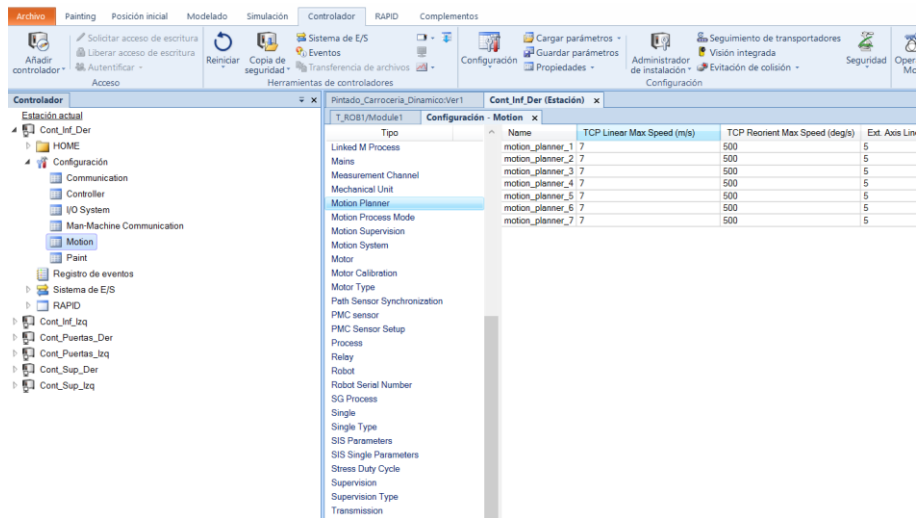


Figura 3.163 - Cambio del valor del límite de velocidad máxima del TCP del robot

Por ejemplo:

```
VelSet 100,MaxRobSpeed();
```

Lectura de la hora y la fecha actuales

CDate

CDate()

Esta función se emplea para leer la fecha actual del sistema.

Esta función no posee ningún argumento.

El valor de retorno de esta función será un dato de tipo *string*, indicando la fecha actual en una cadena de caracteres. El formato de fecha estándar es “año-mes-día”, por ejemplo “2021-01-01”.

Por ejemplo:

```
date:=CDate();
```

Lectura o escritura a través de un canal serie o un archivo alfanumérico

ReadNum

ReadNum (IODevice [\Delim] [\Time])

Esta función se emplea para leer un valor numérico de un archivo o un canal serie alfanumérico.

Los argumentos de esta función son los siguientes:

IODevice (iodev). Este argumento especifica la referencia del archivo o del canal serie que debe leerse.

[\Delim] (string). Este argumento opcional especifica una cadena de caracteres que contiene los delimitadores que deben emplearse para interpretar las líneas del archivo o del canal serie. Si no se utiliza este argumento, el archivo se lee según las líneas del mismo y el único delimitador que se tiene en cuenta es el carácter de salto de línea "\0A". Si se emplea este argumento, el carácter especificado se utilizará para determinar qué parte de la línea es significativa. Además el sistema de control añade siempre los caracteres de retorno de carro (\0D) y salto de línea (\0A) a los ya especificados por el usuario. Por ejemplo, para indicar que se lea el valor hasta que se encuentre un espacio el argumento sería el siguiente `\Delim:=" "`.

[\Time] (num). Este argumento opcional especifica el tiempo máximo, en segundos, del que dispone el sistema para realizar la operación de lectura. Si no se especifica este argumento, el tiempo máximo es de 60 segundos. Para esperar ininterrumpidamente, se debe emplear como argumento la constante predefinida `WAIT_MAX`. Si se agota el tiempo especificado antes de que se complete la operación de lectura, se llama al gestor de errores con el código de error `ERR_DEV_MAXTIME` o se detiene la ejecución.

El valor de retorno de esta función será un dato de tipo *num*, indicando el valor numérico leído de un archivo o un canal serie especificado. Si el archivo está vacío o se ha alcanzado el final del archivo, se devuelve un valor numérico mayor que el valor predeterminado `EOF_NUM`, 9.998E36.

Por ejemplo:

```
Lectura_Color:=ReadNum(infile\Delim:=" ");
```

Conversión de datos

NumToStr

NumToStr (Val Dec [\Exp])

Esta función se emplea para convertir un valor numérico en un valor de cadena de caracteres.

Los argumentos de esta función son los siguientes:

Val (num). Este argumento especifica el valor numérico que se desea convertir.

Dec (num). Este argumento especifica la cantidad de decimales del valor numérico que se convertirán en cadena de caracteres. Este argumento no debe ser negativo ni mayor que la precisión disponible para los valores numéricos.

[\Exp] (switch). Este argumento opcional especifica si el valor de retorno se muestra como un exponente.

El valor de retorno de esta función será un dato de tipo *string*, indicando una cadena de caracteres con el valor numérico especificado.

Por ejemplo:

Num_CE :=NumToStr(Contador,0);

3.2.5 Instrucciones

Las instrucciones podrían definirse como procedimientos predefinidos en RAPID, es decir, se emplean instrucciones para llevar a cabo una tarea concreta, previo ingreso de los valores de sus argumentos.

Las instrucciones se ejecutan secuencialmente a no ser que una instrucción de flujo del programa, una interrupción o un error hagan que la ejecución continúe en otro lugar del programa o se detenga.

La mayoría de las instrucciones deben terminar con un punto y coma (;). Las etiquetas deben terminar con dos puntos (:). Algunas instrucciones de control del flujo del programa pueden contener otras instrucciones y terminan con palabras clave determinadas. Estas se definirán posteriormente.

Todas las instrucciones se congregan en grupos específicos como se muestra en la siguiente Figura 3.164. En RobotStudio, se puede acceder a este menú desde la pestaña de **RAPID**, grupo **Insertar, Instrucción**.

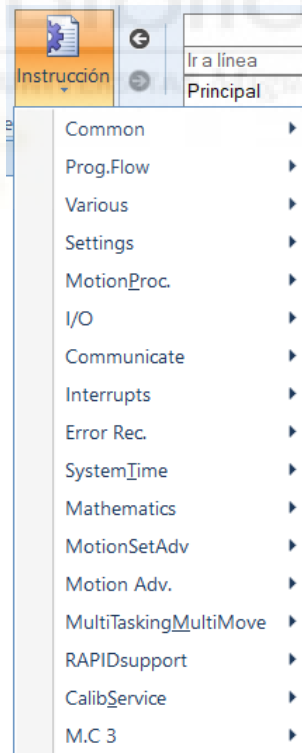


Figura 3.164 - Grupos de instrucciones

En las diferentes secciones de la Figura 3.164 se pueden encontrar solo instrucciones o instrucciones y funciones.

En el lenguaje de RAPID existen 367 instrucciones diferentes, por lo que resultaría ilógico realizar la descripción detallada de todas ellas en el presente proyecto. En su defecto, se describirán las instrucciones consideradas más relevantes, y algunas no tan utilizadas que sí se han empleado en el presente proyecto.

Previa a su descripción cabe señalar que los argumentos de las instrucciones que se indican entre corchetes [] son argumentos opcionales, mientras que los que se indican simplemente por su nombre son obligatorios.

3.2.5.1 Control del flujo del programa

Estas instrucciones se emplean cuando se requiere interrumpir el flujo secuencial normal de ejecución de instrucciones de los programas de RAPID, para llevar a cabo otra serie de instrucciones, de forma que pueda hacerse frente a las distintas situaciones que puedan ocurrir durante la ejecución del programa.

El flujo del programa se puede controlar de diferentes maneras. La primera de ellas sería llamar a otra rutina diferente, de forma que la ejecución del programa pasa a dicha rutina, y una vez ejecutada la totalidad de la misma, se continúa la ejecución del programa desde la siguiente instrucción de la rutina desde donde se la llamó. Para ello se tienen las siguientes instrucciones:

ProcCall

Procedure { Argument }

Esta instrucción es una llamada normal a un procedimiento. No se tiene que escribir como tal la instrucción ProcCall, sino simplemente llamar al procedimiento mediante su identificador y, en el caso que los haya, sus parámetros.

Por ejemplo:

Limpieza_Atom_Color;

Upper_Paint 50, lowspeed;

CallByVar

CallByVar Name Number

Esta instrucción se emplea para llamar a procedimientos que tienen nombres específicos a través de una variable.

Sólo puede usarse para llamar a procedimientos que no utilizan parámetros. No puede usarse para llamar a procedimientos de tipo LOCAL. La ejecución de esta instrucción requiere un poco más de tiempo que la ejecución de una llamada normal a un procedimiento.

Los argumentos de esta instrucción son:

Name (string). Se trata de la parte de cadena de caracteres del identificador.

Number (num). Se trata del valor numérico del número del procedimiento. El valor debe ser número un entero positivo.

Por ejemplo, de esta forma podría llamarse al "proc2":

```
reg1 := 2;  
CallByVar "proc", reg1;
```

RETURN

RETURN o RETURN value

Esta instrucción se emplea para finalizar la ejecución de una rutina. El resultado de la ejecución de la esta instrucción varía en función del tipo de rutina en el que se ejecuta.

Si se trata del procedimiento principal (Main), si el programa tiene el modo de ejecución de un solo ciclo, se detiene. Si tiene el modo de ejecución continuo, la ejecución del programa continúa con la primera instrucción de dicho procedimiento principal.

Si se trata de otro procedimiento, la ejecución del programa continúa con la instrucción que sigue a la llamada del procedimiento en el que se ha ejecutado RETURN.

Si se trata de una función, el valor también se devuelve.

Si se trata de una rutina TRAP, la ejecución del programa continúa en el punto en el que se produjo la interrupción.

Por ejemplo:

```
PROC errormessage()  
    IF di1=1 THEN  
        RETURN;  
    ENDIF  
    TPWrite "Error";  
ENDPROC
```

```
FUNC num abs_value(num value)  
    IF value<0 THEN  
        RETURN -value;  
    ELSE  
        RETURN value;  
    ENDIF  
ENDFUNC
```


La segunda opción es la de ejecutar unas instrucciones u otras dependiendo de si se cumple o no una determinada condición. Para ello se tienen las siguientes instrucciones:

IF compacto

IF Condition ...

Esta instrucción se emplea cuando solo es necesario ejecutar una instrucción si se cumple una determinada condición.

Los argumentos de esta instrucción son:

Condition (bool). Especifica la condición que debe cumplirse para que se ejecute la instrucción.

Por ejemplo:

IF contador > 25 Set Continuar;

IF

IF Condition THEN ...

{ELSEIF Condition THEN ...}

[ELSE ...]

ENDIF

Se emplea cuando es necesario ejecutar una serie de instrucciones diferentes en función de si se cumple o no una determinada condición.

Las condiciones se comprueban una tras otra hasta que una de ellas se cumple, continuando la ejecución del programa con las instrucciones asociadas a dicha condición. Si no se cumple ninguna de las condiciones, la ejecución del programa continúa con las instrucciones que aparecen a continuación de ELSE. Si se cumple más de una condición, sólo se ejecutan las instrucciones asociadas con la primera de las condiciones que se cumple.

Los argumentos de esta instrucción son:

Condition (bool). Especifica la condición que debe cumplirse para que se ejecute la instrucción.

Por ejemplo:

IF Finalizado=1 AND Color<>Secuencia_Colores{i+1} AND Secuencia_Colores{i+1}<>0 THEN

MoveAbsJ Pos_Int10,v2500,z50,tool0\WObj:=WO_In_Int;

Limpieza_Atom_Color;

N_Proc:="main";

MoveAbsJ Pos_Int10,v2500,z50,tool0\WObj:=WO_In_Int;

MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;

ELSE

MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;

ENDIF

TEST

TEST Test data {CASE Test value {, Test value}: ...}

[DEFAULT: ...]

ENDTEST

Esta instrucción se emplea cuando es necesario ejecutar instrucciones diferentes en función del valor de una expresión o un dato. Si la cantidad de alternativas no es muy alta, también puede utilizarse la instrucción IF explicada anteriormente. Los datos de test se comparan con los valores de test de la primera condición CASE, si el resultado es TRUE, se ejecutan las instrucciones de esa condición. Si no lo es, se comprueba la siguiente y así sucesivamente. Si no es verdadera ninguna comprobación, se ejecutan las instrucciones especificadas en DEFAULT, en el caso de que se haya especificado DEFAULT.

Los argumentos de esta instrucción son:

Test data (All). Especifica el dato o la expresión con el que se comparará el valor de test.

Test value (Mismo tipo que Test data). Especifica el valor que debe tener el dato de test para que se ejecuten las instrucciones asociadas a una determinada condición CASE.

Por ejemplo:

TEST reg1

CASE 1,2,3 :

routine1;

CASE 4 :

routine2;

DEFAULT :

TPWrite "Illegal choice";

Stop;

ENDTEST

La tercera es repetir una serie de instrucciones un número determinado de veces o hasta que se cumple una determinada condición. Para ello se tienen las siguientes instrucciones:

FOR

FOR Loop counter FROM Start value TO End value [STEP Step value] DO ...

ENDFOR

Esta instrucción se emplea cuando es necesario repetir una o varias instrucciones un número determinado de veces.

El contador del bucle, un dato de tipo *num*, sólo está disponible dentro del bucle, suponiendo esto la ocultación de otros datos y rutinas que tengan el mismo identificador. El contador solo puede ser leído, no actualizado, por las instrucciones dentro del bucle. No es posible utilizar valores decimales para los valores de inicio, final o paso en combinación con condiciones de finalización exactas para el bucle FOR.

Los argumentos de esta instrucción son:

Loop counter (Identifier). Especifica el nombre del dato que contendrá el valor del contador del bucle.

Start value (num). Especifica el valor inicial deseado para el contador del bucle.

End value (num). Especifica el valor final deseado para el contador del bucle.

Step value (num). Este argumento es opcional. El valor en el que debe incrementarse o reducirse el contador del bucle con cada ciclo. Si no se especifica el valor predeterminado de paso es 1 si el valor inicial es menor que el valor final o -1 si es al contrario.

Por ejemplo:

```
FOR i1 FROM 1 TO Contador-1 DO
    Color:=Secuencia_Colores{i1};
    Principal;
    N_Proc:="main";
    WaitRob\ZeroSpeed;
ENDFOR
```

WHILE

WHILE Condition DO ...

ENDWHILE

Esta instrucción se emplea cuando es necesario repetir una serie de instrucciones siempre y cuando la evaluación de la expresión de una determinada condición dé como resultado el valor TRUE. Se evalúa la expresión de la condición y si el resultado es TRUE, se ejecutan las instrucciones. Se vuelve a evaluar la expresión de la condición y si el resultado continúa siendo TRUE se vuelven a ejecutar. Así hasta que el resultado es FALSE.

Los argumentos de esta instrucción son:

Condition (bool). Especifica la condición cuya evaluación debe dar como resultado TRUE para que se ejecuten las instrucciones.

Por ejemplo:

```
WHILE reg1 < reg2 DO
    ...
    reg1 := reg1 + 1;
ENDWHILE
```

Otra de las formas es trasladarse a una etiqueta dentro de la misma rutina. Para ello se tiene la siguiente instrucción:

GOTO

GOTO Label

Esta instrucción se emplea para transferir la ejecución del programa a otra línea dentro de la misma rutina, señalada mediante una etiqueta. La etiqueta se especifica mediante otra instrucción muy simple denominada Label, mediante la cual simplemente se indica el identificador de la etiqueta en la línea que se desea. La instrucción de la etiqueta (Label) no se cierra con un punto y coma, como la mayoría de instrucciones, si no que termina con dos puntos.

La etiqueta no debe tener el mismo identificador que cualquier otra etiqueta o dato dentro de la misma rutina. El uso de una etiqueta supone la ocultación de los datos globales y las rutinas que tengan el mismo identificador dentro de la rutina en la que se define.

Sólo es posible transferir la ejecución del programa a una etiqueta que se encuentre dentro de una instrucción IF o TEST si la instrucción GOTO se encuentra también dentro de la misma bifurcación de la instrucción.

Sólo es posible transferir la ejecución del programa a una etiqueta que se encuentre dentro de una instrucción FOR o WHILE si la instrucción GOTO se encuentra también dentro de la instrucción.

Los argumentos de esta instrucción son:

Label (Identifier). Especifica la etiqueta que indica dónde debe continuar la ejecución del programa.

Por ejemplo:

```
reg1 := 1;
next:
...
reg1 := reg1 + 1;
IF reg1 <= 5 GOTO next;
```

Y por último se puede detener la ejecución del programa. Para ello se tienen las siguientes instrucciones:

Stop

Stop [\NoRegain] | [\AllMoveTasks]

Esta instrucción se emplea para detener la ejecución del programa. La ejecución del programa se detiene cuando las unidades mecánicas afectadas de la tarea de movimiento actual alcanzan la velocidad cero para el movimiento que se esté realizando. Posteriormente es posible reanudar la ejecución del programa a partir de la instrucción siguiente.

Los argumentos de esta instrucción son los siguientes, siendo ambos opcionales:

[\NoRegain] (switch). Especifica si la unidad mecánica afectada debe regresar a la posición de paro la siguiente vez que se reanude el programa. Si los ejes del robot o los ejes externos han sido alejados de esa posición y se emplea el argumento \NoRegain, estos no regresan a la posición de paro. Si se omite el argumento y los ejes del robot o los ejes externos han sido alejados de la posición de paro, el robot muestra una pregunta en el FlexPendant. A continuación, el usuario debe decidir si el robot debe regresar a la posición de paro.

[\AllMoveTasks] (switch). Especifica que deben detenerse los programas de todas las tareas normales en funcionamiento, excepto la tarea actual. Si se omite el argumento, sólo se detiene el programa de la tarea en la que se ejecuta la instrucción.

Por ejemplo:

```
TPWrite "El programa ha sufrido un fallo";  
Stop;
```

EXIT

EXIT

Esta instrucción se emplea para finalizar la ejecución de un programa, impidiéndose además la posterior reanudación del mismo, es decir, éste sólo puede volverse a ejecutar empezando de nuevo desde la primera instrucción de la rutina main.

Esta instrucción no tiene argumentos.

Por ejemplo:

```
ErrWrite "Estado de error irrecoverable";  
EXIT;
```

Break

Break

Esta instrucción se emplea para provocar la interrupción inmediata de la ejecución de un programa con fines de depuración del código, de forma que continúa siendo posible tras el paro del programa el análisis de variables, valores, etc. Además también se detiene el movimiento del robot, aunque sin esperar a que los ejes del robot o los ejes externos alcancen sus puntos de destino programados para el movimiento que se está realizando. Posteriormente es posible reanudar la ejecución del programa a partir de la instrucción siguiente.

Esta instrucción no tiene argumentos.

Por ejemplo:

```
IF Lectura_Color >= EOF_NUM THEN  
    TPWrite "El archivo que contiene la Secuencia de colores está vacío/incorrecto, revíselo."  
    Break;  
ENDIF
```

SystemStopAction

SystemStopAction [*\Stop*] [*\StopBlock*] [*\Halt*]

Esta instrucción se puede emplear para detener el sistema de robot de distintas formas en función de la gravedad del error o el problema.

Los argumentos de esta instrucción son:

[*\Stop*] (switch). Este argumento implica el paro de la ejecución del programa y los movimientos del robot en todas las tareas de movimiento. No se requiere ninguna acción específica antes del reinicio de la ejecución del programa.

[*\StopBlock*] (switch). Este argumento implica el paro de la ejecución del programa y los movimientos del robot en todas las tareas de movimiento. Todos los punteros de programa deben ser movidos antes de que sea posible reanudar la ejecución del programa.

[*\Halt*] (switch). Este argumento implica el paro de la ejecución del programa y los movimientos del robot en todas las tareas de movimiento y da lugar al estado Motors OFF. Se debe cambiar al estado Motors ON antes de que sea posible reanudar la ejecución del programa.

Por ejemplo:

Por ejemplo:

```
TPWrite "El programa ha sufrido un fallo total";  
SystemStopAction \Halt;
```

3.2.5.2 Movimiento del robot

El lenguaje de RAPID comparte características con otros lenguajes de programación de alto nivel pero su función principal es la controlar robots, por lo que cuenta con un gran número de instrucciones para controlar el movimiento de los mismos. Los movimientos del robot se

programan como movimientos de posición a posición. La trayectoria entre estas dos posiciones es calculada automáticamente por el controlador del robot.

Las características básicas del movimiento del robot, como el tipo de trayectoria, se especifican según la instrucción de posicionamiento seleccionada. Las demás características de movimiento se especifican mediante la definición de los argumentos de la instrucción.

Algunas de las características de movimiento del robot, como las de la siguiente lista, se establecen mediante instrucciones lógicas que se aplican a todos los movimientos.

- 1) Velocidad máxima y ajuste de velocidad.
- 2) Aceleración.
- 3) Gestión de distintas configuraciones de robot.
- 4) Carga útil.
- 5) Comportamiento cerca de puntos singulares.
- 6) Desplazamiento del programa.
- 7) Servo suave.
- 8) Valores de ajuste.
- 9) Activación y desactivación de búfer de eventos.

Instrucciones de posicionamiento

MoveL

MoveL [*\Conc*] *ToPoint* [*\ID*] *Speed* [*\V*] | [*\T*] *Zone* [*\Z*] [*\Inpos*] *Tool* [*\WObj*] [*\Corr*] [*\TLoad*]

Esta instrucción se emplea para trasladar el punto central de la herramienta (TCP) en sentido lineal hacia el punto de destino deseado. Esta instrucción sólo puede usarse en la tarea principal T_ROB1 o, si se cuenta con un sistema MultiMove, en las tareas de movimiento.

Esta instrucción ejecuta el movimiento del robot y de los ejes externos, en el caso de que los haya, de la siguiente forma.

El TCP de la herramienta se mueve linealmente a una velocidad programada constante a lo largo de la trayectoria. A su vez, la herramienta se reorienta en intervalos iguales. Los ejes externos no coordinados, si los hay, se ejecutan a una velocidad constante de forma que alcancen el punto de destino al mismo tiempo que los ejes del robot. Si no es posible alcanzar la velocidad programada para la reorientación o para los ejes externos, se reduce la velocidad del TCP en esa instrucción.

Los argumentos de esta instrucción son:

[*\Conc*] (switch). Este argumento se indica en el caso de que se quieran ejecutar distintas instrucciones consecutivas mientras el robot está en movimiento.

ToPoint (robtargt). Especifica la posición de destino hacia la que deben desplazarse los ejes del robot y los ejes externos. Se define como una posición con identificador o se almacena directamente en la instrucción.

[*\ID*] (identno). La especificación de este argumento es obligatoria en los sistemas MultiMove si el movimiento es sincronizado o sincronizado coordinado. Este argumento no está permitido en ningún otro caso.

Speed (speeddata). Este argumento especifica los datos de velocidad que se aplican a los movimientos. Los datos de velocidad definen la velocidad del TCP, la reorientación de la misma y los ejes externos. Existen un buen número de valores predefinidos para la velocidad.

[\V] (num). Este argumento especifica únicamente la velocidad del TCP, en mm/s, directamente en la instrucción.

[\T] (num). Este argumento especifica el tiempo total, en segundos, que dura el movimiento del robot.

Zone (zonedata). Este argumento especifica los datos de zona del movimiento. Los datos de zona describen el tamaño de la trayectoria de esquina generada, es decir, la exactitud de la posición alcanzada. Existen un buen número de valores predefinidos para la zona.

[\Z] (num). Este argumento especifica únicamente la exactitud de la posición del TCP del robot, directamente en la instrucción.

[\Inpos] (stoppointdata). Este argumento especifica los criterios de convergencia para la posición del TCP del robot en el punto de paro. Los datos de puntos de paro sustituyen a la zona especificada en el parámetro Zone.

Tool (tooldata). Este argumento especifica la herramienta que se está empleando durante el movimiento del robot. tool0 es una herramienta predefinida que representa al robot sin ninguna herramienta montada en él, es decir, el TCP se encuentra en la brida del robot, y no debe ser declarada ni asignada.

[\WObj] (wobjdata). Este argumento especifica el objeto de trabajo, es decir, el sistema de coordenadas con el que está relacionada la posición de robot indicada en la instrucción. Si se omite este parámetro, la posición depende del sistema de coordenadas mundo.

[\Corr] (switch). Mediante este argumento se añaden los datos de corrección escritos en una entrada de corrección mediante una instrucción CorrWrite a la trayectoria y a la posición de destino.

[\TLoad] (loaddata). Este argumento especifica la carga total usada durante el movimiento, es decir, la carga de la herramienta más la carga útil transportada por la herramienta.

Por ejemplo:

```
MoveL Cerrar_Capo1,v2500,z50,tool0\WObj:=wobj_cnv1;
```

```
MoveL Pos_Int4,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
```

```
MoveL Limpieza_1,v1500,fine,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
```

MoveJ

```
MoveJ [\Conc] ToPoint [\ID] Speed [\V] | [\T] Zone [\Z] [\Inpos] Tool [\WObj] [\TLoad]
```

Esta instrucción se emplea para mover el robot rápidamente de un punto a otro cuando no es estrictamente necesario que el movimiento siga una trayectoria lineal. Al igual que la instrucción MoveL, esta instrucción sólo puede usarse en la tarea principal T_ROB1 o, si se cuenta con un sistema MultiMove, en las tareas de movimiento.

Esta instrucción ejecuta el movimiento del robot y de los ejes externos, en el caso de que los haya, de la siguiente forma.

El punto central de la herramienta se mueve hacia el punto de destino declarado mediante la interpolación de los ángulos de los ejes, es decir, cada eje se mueve a una velocidad constante y todos los ejes alcanzan al mismo tiempo el punto de destino, dando lugar a una trayectoria no lineal. La herramienta se reorienta y los ejes externos se mueven al mismo tiempo que se traslada el TCP. Si no es posible alcanzar la velocidad programada para la orientación o para los ejes externos, se reduce la velocidad del TCP en esa instrucción.

Los argumentos de esta instrucción son los mismos que para la instrucción MoveL, salvo el argumento [\Corr] que este caso no existe, por lo que no se volverán a exponer.

Por ejemplo:

```
MoveJ Pos_Int4,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
```

```
MoveJ Limpieza_1,v1500,fine,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
```

MoveC

```
MoveC [\Conc] CirPoint ToPoint [\ID] Speed [\V] | [\T] Zone [\Z] [\Inpos] Tool [\WObj] [\Corr] [\TLoad]
```

Esta instrucción se emplea para trasladar el TCP describiendo un arco de circunferencia hacia un punto de destino programado. Durante el movimiento, la orientación suele permanecer sin cambios respecto del arco. Como en las instrucciones anteriores, esta instrucción sólo puede usarse en la tarea principal T_ROB1 o, si se cuenta con un sistema MultiMove, en las tareas de movimiento.

Esta instrucción ejecuta el movimiento del robot y de los ejes externos, en el caso de que los haya, de la siguiente forma.

El TCP de la herramienta se mueve circularmente a una velocidad programada constante a lo largo de la trayectoria. A su vez, la herramienta se reorienta respecto de la trayectoria circular a una velocidad constante, desde la orientación de la posición inicial hasta la orientación del punto de destino. Por tanto, si la orientación respecto de la trayectoria es la misma en los puntos inicial y final, la orientación relativa permanece sin cambios durante el movimiento. La exactitud de la reorientación a lo largo de la trayectoria sólo depende de la orientación en los puntos inicial y de destino. Si no es posible alcanzar la velocidad programada para la reorientación o para los ejes externos, se reduce la velocidad del TCP en esa instrucción.

Si el punto de inicio, punto de círculo y el punto de destino son colineales, se generará un movimiento lineal, siendo más lógico utilizar la instrucción MoveL.

Los argumentos de esta instrucción son los mismos que para la instrucción MoveL, salvo el argumento CirPoint que es único de esta instrucción, por lo que solo se expondrá este nuevo argumento.

CirPoint (robtargt). Este argumento especifica el punto de círculo del robot, es decir, una posición del arco de circunferencia entre el punto de inicio y el punto de destino. Para conseguir la máxima exactitud, debe estar situado a mitad de camino entre el punto inicial y el de destino. La orientación del punto de círculo no se alcanza, solo se usa para distinguir entre dos sentidos

de orientación posibles. Se define como una posición con identificador o se almacena directamente en la instrucción.

Por ejemplo:

```
MoveLDO Cerrar_Capo3,v300,z1,tool0\WObj:=wobj_cnv1,Cerrar_Capo,0;
```

```
MoveC Cerrar_Capo4,Cerrar_Capo5,v300,z1,tool0\WObj:=wobj_cnv1;
```

MoveAbsJ

```
MoveAbsJ [\Conc] ToJointPos [ID] [\NoEOffs] Speed [V] | [T] Zone [Z] [Inpos] Tool [WObj] [TLoad]
```

Esta instrucción se emplea para mover el robot hacia una posición absoluta definida en posiciones de ejes. Utilizando esta instrucción la posición final del robot no se ve afectada por la herramienta o por el objeto de trabajo que se introduzcan como argumento, ni por el desplazamiento de programa. El robot utiliza estos datos para calcular la carga, la velocidad del TCP y la trayectoria de esquina. Los ejes del robot y los ejes externos se desplazan hasta la posición de destino a lo largo de una trayectoria no lineal, alcanzando todos la posición de destino al mismo tiempo. Como en las instrucciones anteriores, esta instrucción sólo puede usarse en la tarea principal T_ROB1 o, si se cuenta con un sistema MultiMove, en las tareas de movimiento.

Esta instrucción ejecuta el movimiento del robot y de los ejes externos, en el caso de que los haya, de la siguiente forma.

El punto central de la herramienta se mueve hacia la posición absoluta de destino de los ejes declarada mediante la interpolación de los ángulos de los ejes, es decir, cada eje se mueve a una velocidad constante y todos los ejes alcanzan al mismo tiempo el punto de destino de ejes, dando lugar a una trayectoria no lineal. La herramienta se reorienta y los ejes externos se mueven al mismo tiempo que se traslada el TCP. Si no es posible alcanzar la velocidad programada para la orientación o para los ejes externos, se reduce la velocidad del TCP en esa instrucción.

Los argumentos de esta instrucción son los mismos que para la instrucción MoveL, salvo los argumentos ToJointPos y [\NoEOffs] que son únicos de esta instrucción y el argumento [\Corr] que no se encuentra en esta instrucción, por lo que solo de definirán los dos nuevos argumentos.

ToJointPos (jointtarget). Este argumento especifica la posición absoluta de destino de los ejes del robot y de los ejes externos. Se define como una posición con identificador o se almacena directamente en la instrucción.

[\NoEOffs] (switch). Mediante el uso de este argumento el movimiento no se ve afectado por los offsets activos para los ejes externos.

Por ejemplo:

```
MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0 \WObj:=WO_In_Int;
```

Las instrucciones de movimiento MoveL, MoveJ y MoveC presentan tres variantes cada una de ellas, que permiten activar o desactivar los diferentes tipos de señales de salida que existen en RAPID al a vez que ejecutan el movimiento. El movimiento que ejecutará el robot será el mismo que con las instrucciones normales, con la única diferencia que además activarán o desactivarán una determinada señal de salida. Por lo tanto no se van a explicar por completo estas señales, ya que en gran medida son iguales que las normales, si no que se expondrán sus pequeñas diferencias, como por ejemplo algunos argumentos diferentes.

MoveLAO

MoveLAO ToPoint [\ID] Speed [\T] Zone Tool [\WObj] Signal Value [\TLoad]

Esta instrucción se emplea para realizar un movimiento lineal, de la misma forma que mediante la instrucción MoveL, y además, activar la señal de salida analógica que se haya especificado, cuando se alcanza el centro de la trayectoria de esquina.

La mayoría de los argumentos de esta señal son los mismos que los de la señal MoveL, salvo los siguientes:

Signal (signalao). Especifica el nombre de la señal analógica de salida que debe cambiar de valor.

Value (num). Especifica el nuevo valor para la señal.

MoveLDO

MoveLDO ToPoint [\ID] Speed [\T] Zone Tool [\WObj] Signal Value [\TLoad]

Esta instrucción se emplea para realizar un movimiento lineal, de la misma forma que mediante la instrucción MoveL, y además, activar la señal de salida digital que se haya especificado, cuando se alcanza el centro de la trayectoria de esquina.

Los argumentos son los mismos que para la señal anterior pero, al tratarse de una señal digital, el valor de Value solo puede ser 1 o 0.

MoveLGO

MoveLGO ToPoint [\ID] Speed [\T] Zone Tool [\WObj] Signal [\Value] | [\DValue] [\TLoad]

Esta instrucción se emplea para realizar un movimiento lineal, de la misma forma que mediante la instrucción MoveL, y además, activar la señal de salida de grupo que se haya especificado, cuando se alcanza el centro de la trayectoria de esquina.

En esta instrucciones se valor de la señal puede ser un dato de tipo *num* o *dnum*. Se especifica el argumento opcional [\Value] en el caso de que sea de tipo *num* o el argumento opcional [\DValue] en el caso que sea de tipo *dnum*. A pesar de ser opcionales se debe especificar uno de los dos, si no se produce un error.

Por ejemplo:

MoveLAO p2, v1000, z30, tool0\WObj:=wobj_cnv1, Color, 7;

MoveLDO Abrir_Capo3,v50,z1,tool0\WObj:=wobj_cnv1,Abrir_Capo,1;

MoveLGO p2, v1000, z30, tool0, go1 \Value:=5;

MoveJAO

MoveJAO ToPoint [\ID] Speed [\T] Zone Tool [\WObj] Signal Value [\TLoad]

MoveJDO

MoveJDO ToPoint [\ID] Speed [\T] Zone Tool [\WObj] Signal Value [\TLoad]

MoveJGO

MoveJGO ToPoint [\ID] Speed [\T] Zone Tool [\WObj] Signal [\Value] | [\DValue] [\TLoad]

Estas instrucciones se emplean para realizar un movimiento no necesariamente lineal, de la misma forma que mediante la instrucción MoveJ, y además, activar la señal de salida (analógica, digital o de grupo) que se haya especificado, cuando se alcanza el centro de la trayectoria de esquina.

Sus diferencias con MoveJ son las mismas que se han expuesto en las instrucciones previamente explicadas.

MoveCAO

MoveCAO CirPoint ToPoint [\ID] Speed [\T] Zone Tool [\WObj] Signal Value [\TLoad]

MoveCDO

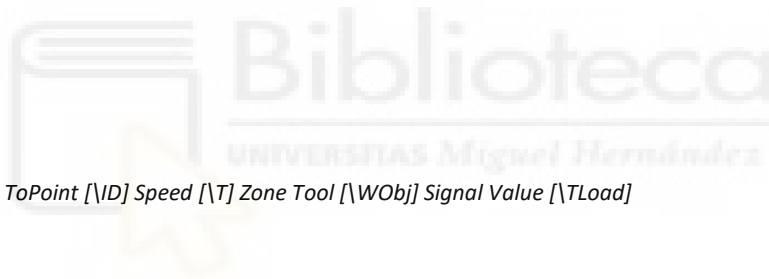
MoveCDO CirPoint ToPoint [\ID] Speed [\T] Zone Tool [\WObj] Signal Value [\TLoad]

MoveCGO

MoveCGO CirPoint ToPoint [\ID] Speed [\T] Zone Tool [\WObj] Signal [\Value] | [\DValue] [\TLoad]

Estas instrucciones se emplean para realizar un movimiento circular, de la misma forma que mediante la instrucción MoveC, y además, activar la señal de salida (analógica, digital o de grupo) que se haya especificado, cuando se alcanza el centro de la trayectoria de esquina.

Sus diferencias con MoveC son las mismas que se han expuesto en las instrucciones previamente explicadas.



Instrucciones de posicionamiento específicas para tareas de pintado

PaintL

PaintL ToPoint [\ID] Speed [\T] Zone Tool [\WObj] [\TLoad]

Esta instrucción tiene exactamente la misma función de movimiento que la instrucción MoveL, es decir, se emplea para trasladar el punto central de la herramienta (TCP) en sentido lineal hacia el punto de destino deseado, con la diferencia que esta es una instrucción específica para trayectorias de pintado.

Si se programa una trayectoria de pintado debe emplearse esta instrucción en lugar de MoveL, ya que la instrucción SetBrush, que define el plano de disparo, debe declararse entre dos instrucciones PaintL.

Los argumentos de esta instrucción se definen de la misma forma que los de la instrucción MoveL, por lo que no volverán a explicarse.

Por ejemplo:

PaintL Paint_10,v2000,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;

También se dispone de las instrucciones de pintado PaintC, que actúa del mismo modo que la instrucción MoveC, y PaintLDO, que actúa del mismo modo que la instrucción MoveLDO.

SetBrush

SetBrush BrushNumber [\X] [\Y] [\Z] [\App] [\Cond]

Esta instrucción se emplea para definir y posicionar el plano de disparo del aplicador de pintura. La instrucción SetBrush se añade en el centro del segmento seleccionado de una trayectoria de pintado, es decir, en el punto medio entre dos instrucciones de movimiento de pintado PaintL. Posteriormente puede ser editado, ya sea gráficamente o mediante la función de edición de eventos. Esta instrucción se representa gráficamente con un plano cuadrado de color blanco, rojo o amarillo que corta la trayectoria de pintado.

La trayectoria de pintado se colorea en función del valor que tiene asignado el plano de disparo o SetBrush. Dependiendo de la posición en la que se encuentra este plano en la trayectoria, su visualización se muestra en blanco, rojo o amarillo.

El color blanco indica que la posición del plano es adecuada.

El color rojo indica que el plano de disparo está posicionado fuera de los segmentos de trayectoria subsiguientes.

El color amarillo indica que el plano de disparo está dentro de la zona.

Los argumentos de esta instrucción son los siguientes:

BrushNumber (num). Este argumento especifica el valor numérico del “cepillo” que se emplea en la instrucción, es decir, hace referencia al color, patrón y distancia de pintado que se emplea en la trayectoria de pintado.

[\X], [\Y] o [\Z] (num). Este argumento especifica la posición en X, Y o Z del plano de disparo que corta a la trayectoria de pintado.

[\App] (num). Este argumento opcional especifica el número de aplicador para la instrucción.

[\Cond] (bool). Este argumento opcional especifica una condición de tipo lógico que debe cumplirse para producirse el disparo del aplicador.

Por ejemplo:

```
SetBrush Color\Z:=2026.49;
```

```
SetBrush 1\X:=4195.16;
```

Instrucciones para la definición de la velocidad

VelSet

VelSet Override Max

Esta instrucción se emplea para aumentar o reducir la velocidad programada de todas las instrucciones de movimiento posteriores. También se emplea para limitar la velocidad máxima de TCP. La velocidad y limitación impuestas por VelSet estarán vigentes hasta que se ejecute una nueva instrucción VelSet.

Los argumentos de esta instrucción son los siguientes:

Override (num). Este argumento especifica la velocidad deseada como porcentaje de la velocidad programada, siendo el 100% la velocidad programada. Este argumento afecta a todos los componentes de velocidad de speeddata (TCP, orientación, ejes externos de rotación y ejes externos lineales), a la redefinición de velocidad programada en la instrucción de posicionamiento, es decir, el argumento \V, y a los movimientos temporizados.

Max (num). Este argumento especifica la velocidad máxima del TCP en mm/s. La velocidad del TCP solo se ve limitada si esta es más lenta que la velocidad programada.

Por ejemplo:

```
VelSet 100,100;
```

```
MoveL Mantenimiento_3,v100,fine,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
```

```
Reset Open_TMant;
```

```
MoveLDO Mantenimiento_2,v100,fine,RB1000_WSC_70_700_200\WObj:=WO_In_Int,Close_TMant,1;
```

```
VelSet 100,MaxRobSpeed();
```

Instrucciones para el control del movimiento si se produce un error o una interrupción

StopMove

StopMove [*\Quick*] [*\AllMotionTasks*]

Esta instrucción se emplea para detener temporalmente los movimientos de los ejes del robot y los ejes externos y cualquier proceso asociado. Para detener los movimientos de los ejes del robot y de los ejes externos no se aplican los frenos. Cualquier proceso asociado con el movimiento en curso se detiene al mismo tiempo que se detiene el movimiento. Una vez detenidos los ejes del robot y los ejes externos se continúa con la ejecución del programa.

Los argumentos de esta instrucción son los siguientes:

[*\Quick*] (switch). Mediante el uso de este argumento opcional el robot se detiene lo antes posible, sin abandonar la trayectoria. Si no se especifica este argumento el robot se detiene en su trayectoria, pero la distancia de frenado es mayor.

[*\AllMotionTasks*] (switch). Mediante el uso de este argumento opcional se detiene el movimiento de todas las unidades mecánicas del sistema. Este argumento sólo puede usarse desde tareas de programa sin movimiento.

Por ejemplo:

TRAP Parada

VAR robtarget Punt_Paro;

VAR jointtarget jointPunt_Paro;

SetAO Señal_Luminosa,1;

StopMove\Quick;

...

ENDTRAP

StartMove

StartMove [*\AllMotionTasks*]

Esta instrucción se emplea tras haberse detenido el movimiento para reanudar el movimiento del robot y los ejes externos y el proceso perteneciente.

El único argumento de esta instrucción es el siguiente:

[*\AllMotionTasks*] (switch). Mediante el uso de este argumento opcional se reinicia el movimiento de todas las unidades mecánicas del sistema. Este argumento sólo puede usarse desde tareas de programa sin movimiento.

Por ejemplo:

TRAP Parada

VAR robtarget Punt_Paro;

VAR jointtarget jointPunt_Paro;

```
StopMove\Quick;  
StorePath;  
...  
RestoPath;  
StartMove;  
ENDTRAP
```

StorePath

StorePath [*\KeepSync*]

Esta instrucción se emplea para almacenar la trayectoria de movimiento que se está ejecutando. Una vez almacenada, se pueden ejecutar otra serie de instrucciones en una rutina TRAP o un gestor de errores por ejemplo. Una vez ejecutado esto, mediante la instrucción RestoPath, que se explicará a continuación, es posible restaurar el movimiento de la trayectoria almacenada.

Sólo es posible tener almacenada una trayectoria de movimiento cada vez.

El único argumento de esta instrucción es el siguiente:

[*\KeepSync*] (switch). Mediante el uso de este argumento opcional se mantiene el modo de movimiento sincronizado tras la instrucción. Este argumento sólo puede usarse si el sistema se encuentra en el modo de movimiento sincronizado antes de la llamada a instrucción.

Por ejemplo:

TRAP Parada

```
VAR robtarget Punt_Paro;  
VAR jointtarget jointPunt_Paro;  
StopMove\Quick;  
StorePath;  
...  
ENDTRAP
```

RestoPath

RestoPath

Esta instrucción se emplea para restablecer una trayectoria de movimiento almacenada anteriormente mediante la instrucción StorePath. La trayectoria de movimiento actual de los ejes del robot y de los ejes externos se elimina y la trayectoria almacenada anteriormente se restablece. Sin embargo, no se produce ningún movimiento de la trayectoria almacenada hasta que se ejecuta la instrucción StartMove.

Si a esta instrucción la precede una instrucción de movimiento, ésta última debe programarse con un punto de paro, es decir con el valor del argumento Zone igual a fine, no un punto de paso.

Esta instrucción no posee argumentos.

Por ejemplo:

TRAP Parada

```
VAR rotarget Punt_Paro;  
VAR jointtarget jointPunt_Paro;  
StopMove\Quick;  
StorePath;  
...  
RestoPath;  
StartMove;
```

ENDTRAP

Instrucciones para el control de ejes adicionales

Estas instrucciones solo afectan a los movimientos de los ejes externos, no a los de los ejes del robot, como si hacían las instrucciones de movimiento.

ActUnit

ActUnit MechUnit

Esta instrucción se emplea para activar una unidad mecánica. Si a esta instrucción la precede una instrucción de movimiento, ésta última debe programarse con un punto de paro, es decir con el valor del argumento Zone igual a fine, no un punto de paso. Es posible usar esta instrucción en el nivel StorePath, pero al ejecutar la instrucción RestoPath deben estar activadas las mismas unidades mecánicas que al ejecutar StorePath.

El único argumento de esta instrucción es el siguiente:

MechUnit (mecunit). Especifica el identificador de la unidad mecánica a activar.

Por ejemplo:

```
PROC main()  
ActUnit CNV1;  
.....  
ENDPROC
```

DeactUnit

DeactUnit MechUnit

Esta instrucción se emplea para desactivar una unidad mecánica. Esta instrucción no se puede utilizar si una de las unidades mecánicas se encuentra en el modo independiente. Si a esta instrucción la precede una instrucción de movimiento, ésta última debe programarse con un punto de paro, es decir con el valor del argumento Zone igual a fine, no un punto de paso. Es posible usar esta instrucción en el nivel StorePath, pero al ejecutar la instrucción RestoPath deben estar activadas las mismas unidades mecánicas que al ejecutar StorePath.

El único argumento de esta instrucción es el siguiente:

MechUnit (mecunit). Especifica el identificador de la unidad mecánica que debe desactivarse.

Por ejemplo:

```
PROC main()  
    ActUnit CNV1;  
    .....  
    DeactUnit CNV1;  
ENDPROC
```

Instrucciones para el seguimiento de transportadores

Las siguientes instrucciones solo son válidas si el controlador del robot está equipado con la opción Conveyor tracking.

WaitWObj

WaitWObj WObj [\RelDist][\MaxTime][\TimeFlag]

Esta instrucción establece una conexión a un objeto de trabajo de la ventana de inicio de la unidad mecánica de transportador. Si no hay ningún objeto en la ventana de inicio, la ejecución del programa espera. Si hay un objeto presente, éste se conecta al transportador y la ejecución prosigue. Si se ejecuta una segunda instrucción WaitWObj mientras otra conexión ya existe, se devuelve un error a no ser que se utilice el argumento opcional \RelDist. Son necesarios 50 ms para conectarse al primer objeto de la ventana de inicio.

Los argumentos de esta instrucción son los siguientes:

WObj (wobjdata). Este argumento especifica el objeto de trabajo móvil, es decir, el sistema de coordenadas móvil, con el que está relacionada la posición de robot indicada en la instrucción.

[\RelDist] (num). Mediante este argumento opcional se especifica la distancia que el objeto debe recorrer dentro de la ventana de inicio para que comience la ejecución. Es decir la ejecución espera hasta que el objeto llega a la distancia especificada dentro de la ventana de inicio.

[\MaxTime] (num). Este argumento opcional especifica el periodo de tiempo máximo permitido para el tiempo de espera, expresado en segundos. Si el tiempo se agota antes de la conexión del

objeto o antes de que se alcance la distancia del argumento `\Reldist`, se llama al gestor de errores si lo hay o se detiene la ejecución.

`[\TimeFlag]` (bool). Si se incluye este argumento opcional, no se considera un error si llega a agotarse el tiempo límite especificado por el argumento `\MaxTime`, sino que simplemente `\TimeFlag` toma el valor de TRUE. Si no se incluye el argumento `\MaxTime` este argumento no se tiene en cuenta.

Por ejemplo:

```
PROC Principal()  
    MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;  
    WaitWObj wobj_cnv1;  
    .....  
ENDPROC
```

DropWObj

DropWObj WObj

Esta instrucción se emplea para desconectarse del objeto actual y así dejar el programa preparado para el siguiente objeto. Al desconectar el objeto de trabajo, la unidad de codificador deja de seguir al objeto, el objeto se elimina de la cola de objetos y no puede ser recuperado. Por lo tanto, si se ejecuta esta instrucción mientras el programa está utilizando activamente el objeto de trabajo coordinado con el transportador, el movimiento se detiene.

La instrucción sólo puede ejecutarse después de emplear un objeto de trabajo fijo en la instrucción de movimiento anterior.

El único argumento de esta instrucción es el siguiente:

WObj (wobjdata). Este argumento especifica el objeto de trabajo móvil, es decir, el sistema de coordenadas móvil, con el que está relacionada la posición de robot indicada en la instrucción.

Por ejemplo:

```
PROC Principal()  
    MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;  
    WaitWObj wobj_cnv1;  
    .....  
    MoveAbsJ Pos_Int8,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;  
    Set Volver;  
    DropWObj wobj_cnv1;  
ENDPROC
```

Espera

WaitTime

WaitTime [*\InPos*] *Time*

Esta instrucción se emplea para pausar la ejecución secuencial de instrucciones y esperar una cantidad de tiempo determinado. Una vez pasa el tiempo especificado se continúa con la ejecución normal del programa.

Los argumentos de esta instrucción son los siguientes:

[*\InPos*] (switch). Este argumento opcional especifica que los ejes del robot y los ejes externos deben estar completamente parados antes de que empiece a contar el tiempo de espera.

Time (num). Este argumento especifica el tiempo, en segundos, que debe estar pausada la ejecución del programa. Su valor mínimo es de 0 s, no tiene límite máximo y tiene una resolución de 0,001 s.

Por ejemplo:

WaitTime 4.5;

WaitRob

WaitRob [*\InPos*] | [*\ZeroSpeed*]

Esta instrucción se emplea para pausar la ejecución secuencial de instrucciones y esperar hasta que el robot y los ejes externos lleguen al punto de paro o tengan una velocidad cero.

Los argumentos de esta instrucción son los siguientes:

[*\InPos*] (switch). Este argumento opcional especifica que para que pueda continuar la ejecución de instrucciones el robot y los ejes externos deben haber llegado al punto de paro.

[*\ZeroSpeed*] (switch). Este argumento opcional especifica que para que pueda continuar la ejecución de instrucciones el robot y los ejes externos deben tener una velocidad cero.

Ambos argumentos son opcionales pero al menos uno de ellos debe ser especificado en la instrucción.

Por ejemplo:

WaitRob \ZeroSpeed;

3.2.5.3 Señales E/S

Modificación del valor de una señal

Set

Set Signal

Esta instrucción se emplea para cambiar a 1 una señal digital de salida especificada.

El único argumento de esta instrucción es el siguiente:

Signal (signaldo). Este argumento especifica el identificador de la señal digital de salida que se desea cambiar a uno.

Por ejemplo:

Set Cambio_Color;

Reset

Reset Signal

Esta instrucción se emplea para poner a cero el valor de una señal digital de salida especificada.

El único argumento de esta instrucción es el siguiente:

Signal (signaldo). Este argumento especifica el identificador de la señal digital de salida que se desea poner a cero.

Por ejemplo:

Reset Finalizado;

SetAO

SetAO Signal Value

Esta instrucción se emplea para cambiar el valor de una señal analógica de salida especificada.

Los argumentos de esta instrucción son los siguientes:

Signal (signaldo). Este argumento especifica el identificador de la señal analógica de salida que se desea cambiar de valor.

Value (num). Este argumento especifica el valor deseado para la señal analógica de salida especificada.

Por ejemplo:

SetAO Señal_Luminosa,3;

Comprobación de señales de entrada o salida

WaitDI

WaitDI Signal Value [\MaxTime] [\TimeFlag] [\Visualize] [\Header] [\Message] | [\MsgArray] [\Wrap] [\Icon] [\Image] [\VisualizeTime] [\UIActiveSignal]

Esta instrucción se emplea para pausar la ejecución secuencial de instrucciones y esperar hasta que se activa o desactiva una señal digital de entrada especificada. Una vez la señal tiene el valor especificado se continúa con la ejecución normal del programa.

Los argumentos de esta instrucción son los siguientes:

Signal (signaldo). Este argumento especifica el identificador de la señal digital de entrada cuyo cambio de valor debe esperarse.

Value (num). Este argumento especifica el valor al que se espera que cambie la señal digital de entrada especificada.

[\MaxTime] (num). Este argumento opcional especifica el periodo máximo de tiempo de espera permitido, expresado en segundos.

[\TimeFlag] (bool). Este argumento opcional especifica si se ha cumplido el tiempo de espera máximo, mediante el cambio de su valor a TRUE. Este argumento no se tiene en cuenta si el argumento MaxTime no se incluye en la instrucción.

[\Visualize] (switch). Este argumento opcional indica la activación visualización de un cuadro de mensaje con la condición que no se cumple, icono, título, líneas de mensaje o imagen, de acuerdo si se han especificado en los siguientes argumentos.

[\Header] (string). Este argumento opcional especifica el texto de cabecera que se escribirá en la parte superior del cuadro de mensaje. Máximo 40 caracteres.

[\Message] (string). Este argumento opcional especifica el texto del mensaje que se escribirá en pantalla. Máximo 50 caracteres.

[\MsgArray] (string). Este argumento opcional especifica varias líneas de texto de una matriz, que se escribirán en la pantalla. Los parámetros \Message o \MsgArray son incompatibles, solo es posible usar uno a la vez. Máximo 5 líneas de 50 caracteres cada una.

[\Wrap] (switch). Este argumento opcional se emplea para concatenar todas las cadenas de caracteres especificadas mediante el argumento \MsgArray, para formar una cadena, con un solo espacio entre las distintas cadenas individuales, distribuida en el número mínimo de líneas posibles.

[\Icon] (icondata). Este argumento opcional indica el icono a mostrar. Sólo puede utilizarse uno de los iconos predefinidos de tipo icondata.

[\Image] (string). Este argumento opcional especifica el nombre de la imagen que debe utilizarse.

[\VisualizeTime] (num). Este argumento opcional especifica el tiempo que aparece el mensaje en pantalla. Este argumento debe tener un valor menor que el del argumento \MaxTime, si se ha especificado.

[\UIActiveSignal] (signaldo). Mediante este argumento opcional se especifica una señal de salida cuyo valor cambiará a 1 cuando se activa el cuadro de mensaje de visualización en la pantalla. Cuando se retira el cuadro de mensaje, la señal se vuelve a establecer en 0.

Por ejemplo:

```
WaitDI Init_Pint1,1;
```

3.2.5.4 Comunicaciones

Lectura o escritura a través de un canal serie o un archivo alfanumérico

Open

```
Open Object [\File]IODevice [\Read] | [\Write] | [\Append] [\Bin]
```

Esta instrucción se emplea para abrir un archivo o un canal serie para su lectura o escritura.

Los argumentos de esta instrucción son los siguientes:

Object (string). Este argumento especifica el directorio donde se encuentra el archivo o canal serie que se desea abrir, por ejemplo "HOME:", "TEMP:", "com1:" o "pc:". El directorio predeterminado es "HOME:".

[\File] (string). Este argumento opcional especifica el nombre del archivo que se desea abrir.

IODevice (iodev). Este argumento especifica una referencia al archivo o canal serie que se desea abrir. Esta referencia se utiliza para las operaciones de lectura y escritura del archivo o canal serie.

[\Read] (switch). Mediante este argumento opcional se indica que se abre un archivo o un canal serie para lectura. La lectura comienza al principio del archivo o canal serie.

[\Write] (switch). Mediante este argumento opcional se indica que se abre un archivo o un canal serie para escritura. La escritura comienza al principio del archivo o canal serie. Si el archivo seleccionado ya existe, se elimina su contenido.

[\Append] (switch). Mediante este argumento opcional se indica que se abre un archivo o un canal serie para escritura. Si el archivo seleccionado ya existe, cualquier información que se escriba a partir de ese momento se escribe al final del archivo.

[\Bin] (switch). Este argumento opcional especifica que el archivo o canal serie se abre en el modo binario.

Por ejemplo:

```
Open date+"_Lista_Colores.txt",infile\Read;
```

Comunicación a través del FlexPendant

TPWrite

TPWrite String [\Num] | [\Bool] | [\Pos] | [\Orient] | [\Dnum]

Esta instrucción se emplea para escribir un mensaje de texto en el FlexPendant. Es posible escribir el valor de determinados tipos de datos al final del mensaje, además de texto.

Los argumentos de esta instrucción son los siguientes:

String (string). Este argumento especifica la cadena de caracteres a escribir, con un máximo de 80 caracteres y 40 caracteres por fila.

[\Num] (num). Este argumento opcional especifica el dato cuyo valor numérico se desea escribir a continuación de la cadena de caracteres.

[\Bool] (bool). Este argumento opcional especifica el dato cuyo valor booleano se desea escribir a continuación de la cadena de caracteres.

[\Pos] (pos). Este argumento opcional especifica el dato cuyo valor de posición se desea escribir a continuación de la cadena de caracteres.

[\Orient] (orient). Este argumento opcional especifica el dato cuyo valor de orientación se desea escribir a continuación de la cadena de caracteres.

[\Dnum] (dnum). Este argumento opcional especifica el dato cuyo valor numérico se desea escribir a continuación de la cadena de caracteres.

Por ejemplo:

TPWrite "El archivo que contiene la Secuencia de colores está vacío/incorrecto, revíselo.";

ErrWrite

ErrWrite [\W] | [\I] Header Reason [\RL2] [\RL3] [\RL4]

Esta instrucción se emplea para mostrar un mensaje de error, de máximo 5 líneas, en el FlexPendant y escribirlo en el registro de eventos.

Los argumentos de esta instrucción son los siguientes:

[\W] (switch). Mediante el uso de este argumento opcional se genera una advertencia que sólo se almacena en el registro de mensajes del robot, no se muestra directamente en la pantalla del FlexPendant.

[\I] (switch). Mediante el uso de este argumento opcional se genera un mensaje de información que sólo se almacena en el registro de eventos, no se muestra directamente en la pantalla del FlexPendant.

Si no se emplea ninguno de los argumentos **\W** o **\I**, se genera un mensaje de error directamente en el FlexPendant y se almacena también en el registro de eventos.

Header (string). Este argumento especifica el título del mensaje de error. Máximo 46 caracteres.

Reason (string). Este argumento especifica el motivo del error.

[\RL2] [\RL3] [\RL4] (string). Cada uno de estos argumentos opcionales sirve para añadir una línea adicional a la especificación del motivo del error.

Por ejemplo:

```
ErrWrite "Color error","El identificador de color "+Num_CE+" es incorrecto, revíselo." \RL2:="Identificador superior a 16 o inferior a 2.";
```

3.2.5.5 Interrupciones

Conexión de interrupciones a rutinas TRAP

CONNECT

CONNECT Interrupt WITH Trap routine

Esta instrucción se emplea para determinar la identidad de una interrupción y conectarla a una rutina TRAP. La interrupción se define mediante la petición de un evento de interrupción, como por ejemplo la activación de una señal digital, y la especificación de su identidad. Por tanto, cuando se produce un evento, la rutina TRAP se ejecuta automáticamente.

Los argumentos de esta instrucción son los siguientes:

Interrupt (intnum). Este argumento especifica la variable a la que se desea asignar la identidad de la interrupción. La declaración de esta variable no debe realizarse dentro de una rutina.

Trap routine (Identifier). Este argumento especifica el nombre de la rutina TRAP a la que se conecta la identidad de la interrupción.

Por ejemplo:

```
CONNECT Paro_Robot WITH Parada;
```

Petición de interrupciones

ISignalDI

ISignalDI [\Single] | [\SingleSafe] Signal TriggValue Interrupt

Esta función se emplea para solicitar y activar interrupciones a partir de una señal digital de entrada. En el momento en el que la señal recibe el valor especificado, se realiza una llamada a la rutina TRAP correspondiente. Una vez ejecutada la rutina, la ejecución del programa continúa a partir del punto en el que se produjo la interrupción.

Los argumentos de esta instrucción son los siguientes:

[\Single] (switch). Este argumento opcional especifica si la interrupción debe producirse una sola vez.

[\SingleSafe] (switch). Este argumento opcional especifica que la interrupción es única y segura, es decir, que no puede ponerse en reposo con la instrucción ISleep. Si se omiten los argumentos \Single y \SingleSafe, se genera una interrupción cada vez que se cumpla la condición.

Signal (signalDI). Este argumento especifica el identificador de la señal a partir de la cual deben producirse las interrupciones.

TriggValue (dionum). Este argumento especifica el valor al que debe cambiar la señal para que se produzca la interrupción.

Interrupt (intnum). Este argumento especifica la identidad de la interrupción. Previamente la interrupción debe estar conectada a una rutina TRAP mediante la instrucción CONNECT.

Por ejemplo:

```
CONNECT Paro_Robot WITH Parada;
```

```
ISignalDI Paro_Propio,1,Paro_Robot;
```

Cancelación de interrupciones

IDelete

IDelete Interrupt

Esta instrucción se emplea para eliminar una interrupción. Esto se hace como medida de prevención, ya que no es posible utilizar más de una vez la identidad de la interrupción sin eliminarla previamente.

El único argumento de esta instrucción es el siguiente:

Interrupt (intnum). Este argumento especifica la identidad de la interrupción.

Por ejemplo:

```
IDelete Paro_Robot;
```

```
CONNECT Paro_Robot WITH Parada;
```

```
ISignalDI Paro_Propio,1,Paro_Robot;
```

3.2.5.6 Otras instrucciones

Eliminación de los avisos de trayectoria de esquina

CornerPathWarning

CornerPathWarning Active

Esta instrucción se emplea para activar o desactivar los avisos de fallo de trayectoria de esquina en todas las instrucciones de movimiento posteriores.

El único argumento de esta instrucción es el siguiente:

Active (bool). Este argumento especifica si los avisos de fallo de trayectoria de esquina deben estar activados (TRUE) o desactivados (FALSE).

Por ejemplo:

CornerPathWarning FALSE;



3.3 Otros softwares utilizados

Para realizar el presente proyecto además del software RobotStudio, ya explicado, se han empleado otros softwares de diseño 3D CAD para poder diseñar y/o modificar los componentes de la estación externos a RobotStudio, es decir, aquellos componentes que no se encuentran inicialmente en las bibliotecas de RobotStudio.

A continuación se realizará una breve descripción de estos softwares.

SolidWorks

SolidWorks es un software CAD (diseño asistido por computadora) para modelado mecánico de piezas y ensamblajes en 3D, a partir de los cuales se pueden obtener planos en 2D. Este software fue desarrollado por SolidWorks Corp., actualmente una filial de Dassault Systèmes, para el sistema operativo Microsoft Windows.



Figura 3.165 - Logotipo de SolidWorks

En el año 1995 se lanzó su primera versión al mercado con el objetivo de hacer la tecnología CAD más accesible a todo el público.

Actualmente este software cuenta con una gran variedad de herramientas que le permiten crear, simular, realiza análisis, administrar datos y gestionar proyectos y procesos maximizando la innovación y la productividad de los recursos de ingeniería. Además este software cuenta con diversas extensiones extra para la fabricación inteligente, diseño y análisis, diseño eléctrico y electrónico, gestión de datos, proyectos y procesos.



Figura 3.166 - Ejemplos de diseños mediante SolidWorks

Este software se ha utilizado para diseñar de los distintos componentes de la estación que por su complejidad no pueden ser diseñados mediante RobotStudio.

Blender

Blender es un software de creación 3D gratuito y de código abierto, destinado al modelado de gráficos tridimensionales, iluminación, animación, simulación, renderizado, composición y seguimiento de movimiento, incluso la edición de vídeo y la creación de juegos. Este un software multiplataforma de forma que es compatible con todas las versiones de Windows, macOS, GNU/Linux , Solaris, FreeBSD e IRIX.



Figura 3.167 - Logotipo del software Blender

Este software se ha empleado principalmente para modificar el modelo 3D CAD de la carrocería del vehículo que se emplea en este proyecto. Este modelo se obtuvo gratuitamente a través de la web GrabCAD, en la cual usuarios comparten de forma gratuita modelos 3D CAD, con lo cual inicialmente el modelo se acercaba al deseado pero requería eliminar, añadir y/o modificar algunos componentes o partes del mismo para obtener el resultado requerido.

Se empleó este software y no otro, ya que el tipo de archivo del modelo y el propio modelo no permitían su correcta modificación mediante los otros softwares expuestos en este apartado.

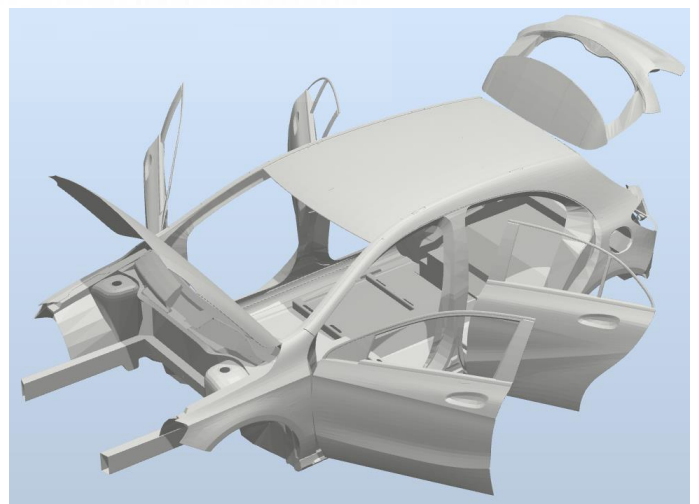
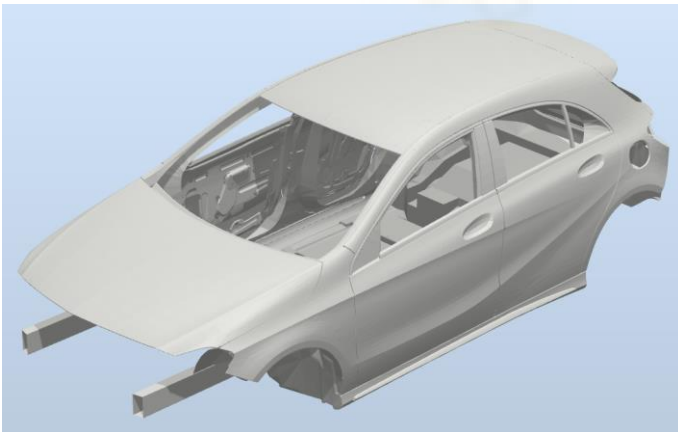


Figura 3.168 - Modelo 3D CAD final de la carrocería Mercedes Benz Clase A A45 AMG

SketchUp

SketchUp es un software de diseño 3D CAD, cuya principal ventaja es permitir realizar diseños en 3D de forma sencilla. Permite conceptualizar y modelar imágenes en 3D de edificios, coches,

personas y cualquier objeto o artículo que se desee, además el programa incluye una galería de objetos, texturas e imágenes listas para descargar.



Figura 3.169 - Logotipo del software SketchUp

Este software se ha empleado para modificar y cambiar de formato algunos de los componentes 3D CAD presentes en la estación.

MeshLab

MeshLab es un sistema de software de código abierto para procesar y editar mallas triangulares 3D. Proporciona un conjunto de herramientas para editar, limpiar, reparar, inspeccionar, renderizar, texturizar y convertir mallas. Ofrece funciones para procesar los datos brutos producidos por las herramientas/dispositivos de digitalización 3D y para preparar los modelos para la impresión 3D.



Figura 3.170 - Logotipo del software MeshLab

Este software se ha utilizado para simplificar algunos de los modelos, en formato .stl, de componentes de la estación, de modo que se reduzca el número de polígonos que los forman y así se reduzca el tamaño del archivo.

4. DISEÑO Y PROGRAMACIÓN DE UNA ESTACIÓN ROBÓTICA DE PINTADO EN SEGUIMIENTO DE LA CARROCERÍA DE UN AUTOMÓVIL

Tras exponer, y por tanto haber adquirido, todos los conocimientos necesarios acerca de las estaciones de recubrimiento de carrocerías de vehículos y sobre el funcionamiento y características del software RobotStudio, en este apartado se procede a exponer como se ha realizado el diseño y programación, mediante este software, de una estación robótica de pintado en seguimiento para la aplicación de la capa base del recubrimiento de la carrocería del modelo de automóvil Mercedes Benz Clase A A45 AMG. Para ello se deberán llevar a cabo un gran número de procesos, desde la correcta selección y/o diseño de los componentes de la estación y su adecuado posicionamiento, hasta la ejecución y simulación de los programas de robot finales.

4.1 Determinación de las necesidades y limitaciones

Aunque este es un proyecto muy abierto en cuanto a la selección de los componentes que lo forman, e incluso en cuanto al tipo y dimensiones de la propia estación, puesto que no es un proyecto que vaya a llevarse a cabo en un empresa real, si no que se trata de un proyecto de libre configuración cuyo propósito principal es el del aprendizaje acerca de la programación offline mediante RobotStudio, sí se han declarado previamente una serie de necesidades y limitaciones autoimpuestas, de forma que la realización del mismo esté enfocada a conseguir los objetivos expuesto en el apartado [1.2 Objetivos](#) y quede limitado por ciertos aspectos.

Necesidades

- 1) Aumentar la productividad disponiendo de una nueva estación robótica de aplicación de la capa base del recubrimiento de la carrocería del vehículo modelo Mercedes Benz Clase A A45 AMG.
- 2) Aplicación de la capa base tanto en la superficie externa como en la interna de la carrocería del vehículo.
- 3) Repetitividad del proceso de aplicación de la capa base sin alteraciones en el resultado.

- 4) Flexibilidad en el diseño de la estación de modo que si en un futuro se quiere reprogramar para otro modelo de vehículo pueda hacerse sin problema.
- 5) La estación diseñada debe realizar la tarea de forma continuada sin sufrir paros o interrupciones.
- 6) El diseño de la estación debe priorizar la economía y la seguridad.
- 7) El diseño de la estación debe ser lo más compacto posible.
- 8) Minimizar al máximo posible el tiempo de ciclo de la estación.

Limitaciones

- 1) Dimensiones del modelo de vehículo Mercedes Benz Clase A A45 AMG, siendo estas aproximadamente las siguientes (en mm):

	Puertas cerradas	Puertas abiertas
Largo	4040	4317
Ancho	1781	3913
Alto	1286	1814

- 2) La longitud máxima de los transportadores o tracks que mueven los robots debe ser de 9 metros.
- 3) La aplicación de la capa base debe realizarse en seguimiento, es decir, el transportador de la carrocería no se detiene para realizar la aplicación.
- 4) La aplicación completa de la capa base, es decir externa e internamente, debe llevarse a cabo en una única estación.
- 5) Debe ser posible realizar el mantenimiento o reparación de las herramientas de aplicación sin acceder al interior de la estación.
- 6) Las capas base a aplicar serán del tipo a base de agua.
- 7) Todos los robots de la estación deben trabajar conjuntamente.
- 8) Los sistemas robóticos deberán detener su movimiento en caso de activación de alguno de los sistemas de seguridad.
- 9) Capacidad de producción de aproximadamente 20 vehículos por hora.
- 10) Debe permitirse el acceso de forma cómoda, sencilla y segura a todos los puntos que sean susceptibles de la realización de inspecciones o mantenimientos.

4.2 Componentes de la estación

4.2.1 Selección y posicionamiento de los equipos de ABB

Puesto que el software RobotStudio se emplea para controlar los sistemas robóticos del fabricante ABB, todos los robots, tracks y herramientas del presente proyecto se seleccionarán del catálogo que ofrece este fabricante.

En este apartado se especificarán los componentes del fabricante ABB presentes en la estación y su posicionamiento a grandes rasgos, ya que sus posiciones exactas se indicarán en el apartado [4.3.2 Importación/creación y posicionamiento de componentes fundamentales](#).

Todos los componentes expuestos en este apartado se importarán a la estación desde la biblioteca de RobotStudio, mediante las herramientas expuestas en los apartados [3.1.5 Importación de robots y otros componentes de la estación](#), [3.1.6 Creación/importación de tracks y mecanismos](#) y [3.1.7 Importación del efector final del robot](#).

4.2.1.1 Selección y posicionamiento de los robots y los tracks

Para cumplir con las necesidades 6, 7 y 8 y con las limitaciones 2, 4 y 7, se ha decidido diseñar una estación con tres manipuladores de robot a cada lado de la misma todos montados sobre tracks, dos de ellos para la aplicación de la capa base, superpuesto uno sobre el otro, y un robot de asistencia, cuya labor será la de abrir y cerrar las puertas del vehículo para permitir así su pintado interior. De esta forma se consigue una estación robótica de pintado compacta, más económica, más rápida, menores longitudes de los tracks y que realiza el pintado completo en una única estación. La disposición de los robots y los tracks en la estación según lo expuesto se muestra en las siguientes Figuras 4.1 y 4.2.

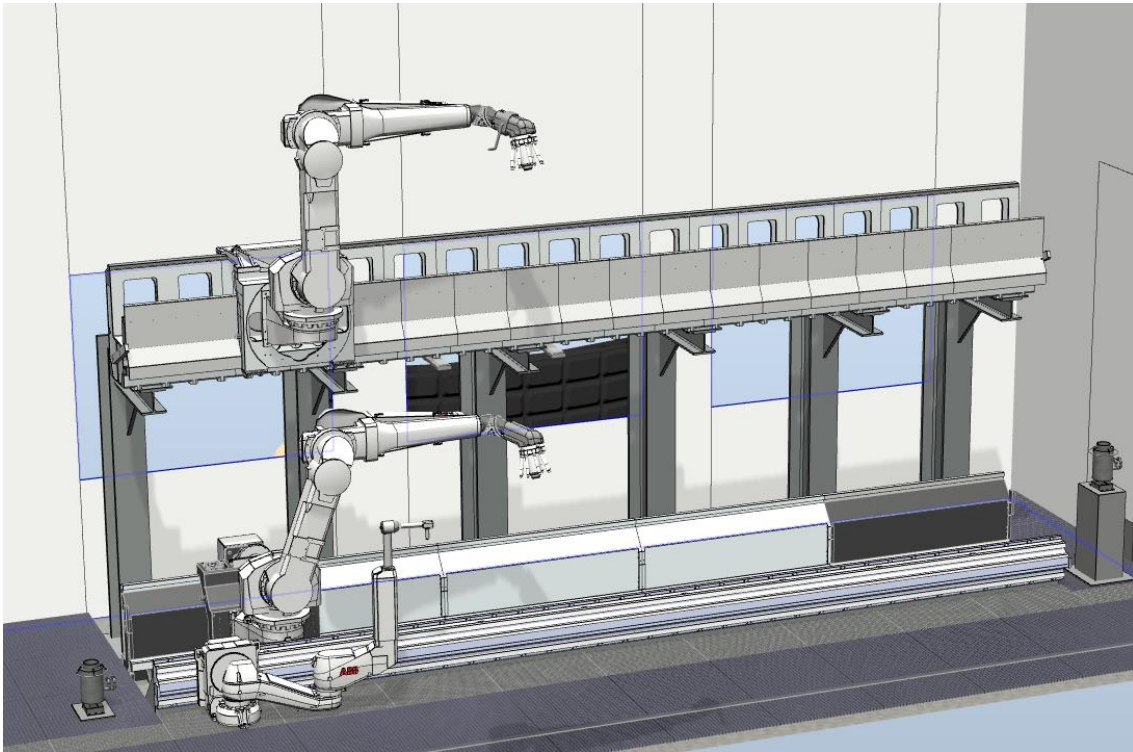


Figura 4.1 - Disposición de los robots y los tracks del lado derecho de la estación

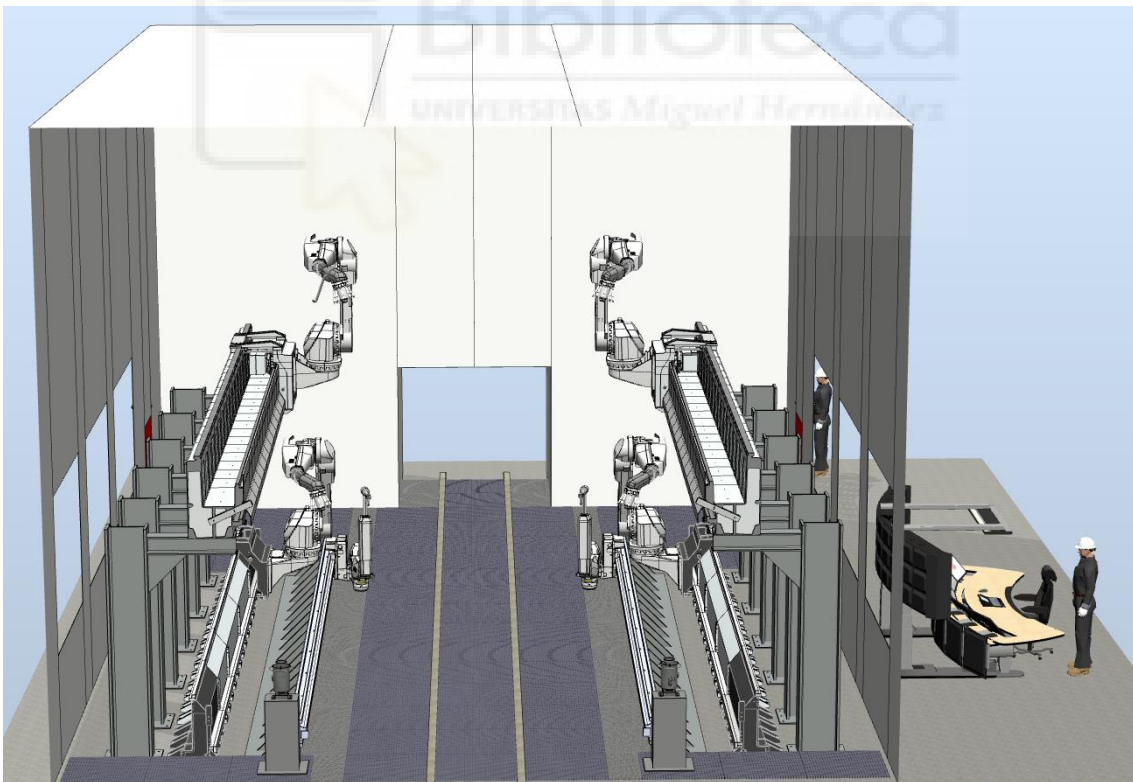


Figura 4.2 - Disposición global de la estación

Una vez se tiene clara la idea de cuantos robots y tracks componen la estación y como estarán posicionados, el siguiente paso es seleccionar sus modelos.

Para la selección de los manipuladores de robot destinados a la aplicación de la capa base los principales aspectos que se han tenido en cuenta son su área de trabajo, que debe tener el tamaño suficiente para poder realizar el pintado de un vehículo de estas características, y su versatilidad, ya que se pretende que esta estación pudiera ser reprogramada para emplearla en el pintado de otros modelos de vehículos.

Así pues de entre los robots de pintura que ofrece ABB se ha seleccionado el robot IRB 5500 para los cuatro robots que realizan el pintado, en sus versiones IRB 5500-23 (Figura 4.3), para los robots inferiores, y IRB 5500-25 (Figura 4.4), para los robots superiores.



Figura 4.3 - Modelo de robot IRB 5500-23



Figura 4.4 - Modelo de robot IRB 5500-25

El robot IRB 5500, en sus dos versiones seleccionadas, gracias entre otras cosas a su gran área de trabajo, su gran aceleración y su velocidad de pintado, es uno de los robots de pintura más eficientes y flexibles que existen actualmente en el mercado.

Este robot permite la instalación sobre el brazo del mismo de componentes de vital importancia para llevar a cabo el pintado, como son el intercambiador de color con hasta 64 válvulas, es decir

32 colores diferentes, dos bombas dosificadoras, el sistema de regulación y control del aire del atomizador en bucle cerrado, el sistema de regulación de la velocidad del plato de campana del atomizador y el sistema de control de alta tensión. Además permite su colocación a tan solo 15 centímetros de distancia de la muñeca del robot, lo cual repercute en un gran ahorro en cuanto a las pérdidas de pintura y disolvente en los cambios de color.

Este robot está diseñado para permitir la utilización tanto de soluciones de pintado a base de agua como a base de disolvente. Además es idóneo para equiparlo con el atomizador RB1000i, expuesto a continuación, de forma que uniendo ambos equipos se consiguen tiempos y acabados de pintado excepcionales.

Las dos versiones del robot seleccionadas, al estar posicionadas sobre tracks, poseen un área de trabajo mucho más grande y flexible, lo cual resulta idóneo para la aplicación de recubrimientos sobre las carrocerías de diferentes modelos de vehículos en la misma estación.

Resumiendo, las principales ventajas de este modelo de robot son las siguientes:

- Mejora la fiabilidad de la estación de pintado en su conjunto.
- Aumenta la flexibilidad de la estación para nuevos procesos y nuevos modelos de vehículos.
- Reduce el impacto ambiental del proceso de pintado.
- Reduce el tamaño de la estación de pintado.
- Reduce el volumen del aire para ventilación.
- Reduce las emisiones del sistema de extracción.

Como ya se ha mencionado y se ha podido observar en las Figuras 4.3 y 4.4, ambas versiones del robot IRB 5500 están posicionadas sobre tracks. Para la versión IRB 5500-23 se ha seleccionado el track IRB 5500 paint rail extended, que estará anclado al suelo de la estación.

Para la versión IRB 5500-25 se ha seleccionado el track IRB 5500 elevated rail, específico para esta versión del robot IRB 5500. Este track estará posicionado sobre un sistema de pilares, de forma que el conjunto formado por el robot y el track queda posicionado sobre los robots inferiores.

Como ya se ha expuesto anteriormente, al estar posicionados sobre tracks, el área de trabajo de los robots se ve ampliamente aumentada, lo cual permite diseñar una estación de pintado más compacta y eficiente, pero manteniendo su versatilidad.

Para los dos robots de asistencia, es decir, para los dos robots encargados de abrir y cerrar las puertas del vehículo, se ha seleccionado el robot IRB 5350, en su versión de cuatro ejes, ideal para la asistencia en piezas de trabajo en movimiento. (Figura 4.5)



Figura 4.5 – Modelo de robot IRB 5350 – versión de 4 ejes

El IRB 5350 es un robot preciso y compacto, diseñado específicamente para la apertura y cierre de las puertas de los vehículos, asistiendo así en las tareas de pintado interior de sus carrocerías.

La versión de cuatro ejes seleccionada, unida al track sobre el que se posiciona, le dota de un área de trabajo mucho mayor, manteniendo su carácter compacto y flexible.

En este caso el modelo de track seleccionado será el IRB 5350 rail, específico para esta versión del robot IRB 5350. El track se sujetará a un sistema de anclaje que a su vez se anclará al suelo de la estación, por delante del rail de pintura de robot IRB 5500-23, de forma que se puedan trabajar conjuntamente sin colisionar.

Todos los tracks seleccionados tendrán una longitud de 9 metros.

Las especificaciones técnicas de todos los modelos de robot se exponen detalladamente en el apartado [8.2 Fichas técnicas de los robots de la estación](#).

4.2.1.2 Selección de las herramientas

La primera herramienta a seleccionar serán los sistemas de aplicación de pintura que controlarán los robots para realizar la aplicación de la capa base sobre la carrocería.

De entre la gran variedad de sistemas de aplicación de pintura a base de agua que ofrece ABB se ha decidido seleccionar, para los cuatro robots de aplicación de la estación, el atomizador RB1000i en su versión exterior SAD-L, un atomizador de plato de campana rotativo de alta velocidad con altas prestaciones, estando esta versión equipada con electrodos externos que hacen posible la aplicación tanto exterior como interior de capas base a base de agua (Figura 4.6). Además es el primer atomizador robótico de pintura digital conectado del mundo.



Figura 4.6 - Modelo de atomizador IRB1000i SAD-L

Este atomizador aumenta las capacidades digitales de la compañía implementando sensores y la plataforma digital ABB Ability en el mismo, para obtener datos inteligentes que permiten mejorar la eficiencia del pintado y reducir en un 75% el desperdicio de pintura debido a los frecuentes cambios de color.

Los sensores detectan problemas de calidad a medida que se lleva a cabo el proceso de pintado, ayudando así a detectar y poder corregir los acabados inadecuados que anteriormente solo podían ser detectados durante las inspecciones tras el proceso completo de pintado. Los sensores de temperatura y vibración y un giroscopio integrados en el atomizador transmiten un conjunto de datos acerca de la estado general del dispositivo y la eficiencia de transferencia de la pintura.

Al controlar el estado de los componentes clave del atomizador, como las campanas rotativas, los motores neumáticos y el cabezal de aire, así como las variables de aceleración, presión, vibración y temperatura, se puede aumentar la eficiencia de la transferencia de pintura hasta en un 10% y se puede reducir el consumo de aire comprimido hasta en un 20%, lo que puede suponer un ahorro significativo a largo plazo.

Por último, los datos extraídos de varios atomizadores conectados se pueden analizar conjuntamente para crear una perspectiva completa del proceso de pintado en tiempo real.

La segunda, y última, herramienta a seleccionar es la agarradera mediante la cual el robot IRB 5350 realiza la apertura y cierre de las puertas del vehículo. Para esta tarea se ha seleccionado el IRB 5350 Gripper, mostrado en la siguiente Figura 4.7.



Figura 4.7 - Modelo de agarradera IRB 5350 Gripper

Se trata de una herramienta de agarre con capacidad de carga de hasta 7 kg que permite agarrar, abrir y cerrar diferentes tipos de puertas de vehículos. Además, esta herramienta incorpora avanzados sensores para la detección de las puertas.

Las herramientas seleccionadas se montarán en el extremo de las muñecas de los robots, por lo que su movimiento será controlado por estos. El TCP del robot pasa de estar situado en el extremo de su muñeca a estar en extremo de la herramienta. En el caso del atomizador IRB1000i, como se explicará posteriormente, se dispondrá de varios TCP para poder variar la distancia de aplicación de la capa base.

4.2.2 Selección y/o diseño de componentes mediante RobotStudio

Como se ha expuesto en los apartados [3.1.8 Modelado de componentes en RobotStudio](#) y [3.1.9 Modificación de componentes en RobotStudio](#), el software RobotStudio dispone de una variedad de herramientas para la creación y modificación de componentes tridimensionales sencillos.

Además de esto, RobotStudio dispone de una biblioteca de componentes tridimensionales de diversos ámbitos, que pueden ser añadidos a la estación mediante las herramientas expuestas en el apartado [3.1.5 Importación de robots y otros componentes de la estación](#).

Los componentes tridimensionales presentes en la estación, diseñados mediante RobotStudio o importados de su biblioteca son los siguientes:

Transportador

Este componente está formado por dos tetraedros, creados mediante la herramienta **Solid - Tetraedro**, que actúan como los raíles del transportador que mueve el conjunto de la carrocería a través de la estación. (Figura 4.8)

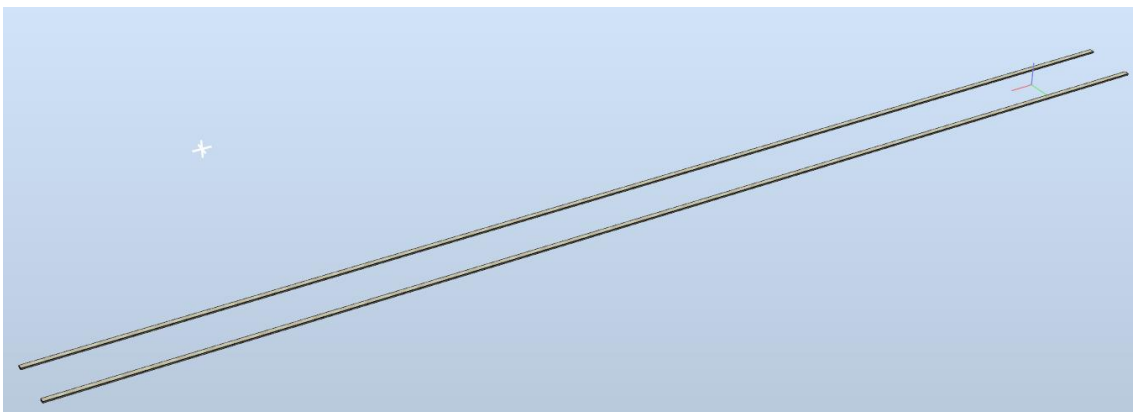


Figura 4.8 - Componente Transportador

Cada uno de los raíles tiene una longitud de 23000 mm, una anchura de 100 mm y una altura de 25 mm. A pesar de estar separados 500 mm, ambos actúan como un único sólido. Para esto, primero se crearon dos sólidos separados y después mediante las herramientas expuestas en los apartados ya mencionados se unen formando un único cuerpo. Para finalizar se le otorga una apariencia de gráficos, es decir un color y textura, de metal esmerilado.

Cabina de control IRC5P

Este componente es un modelo tridimensional de la cabina de control IRC5P, específica para el control y accionamiento de los robots de pintura. En la cabina real, dentro de la misma, se encontrarían el módulo de control y los módulos de accionamiento. (Figura 4.9)



Figura 4.9 – Componente cabina de control IRC5P

Este componente se importa de la biblioteca de equipamiento de RobotStudio. En total se importan 6 cabinas de control IRC5P, situadas en el exterior de la estación, tres a cada lado de la misma, para los tres robots presentes a cada lado de la estación.

Este componente no interviene en la simulación de la estación pero se incluye en la estación para realizar un correcto diseño y dimensionamiento de la misma, ya que lógicamente en la estación real estaría presente.

FlexPendant

Este componente es un modelo tridimensional del sistema FlexPendant de RobotStudio. El FlexPendant es una unidad de operador de mano que se emplea para realizar muchas de las tareas implicadas en el manejo de un sistema de robot, como por ejemplo, ejecutar programas, mover el manipulador, modificar programas del robot, etc. Todos los sistemas de control robóticos de ABB están dotados de un FlexPendant. (Figura 4.10)

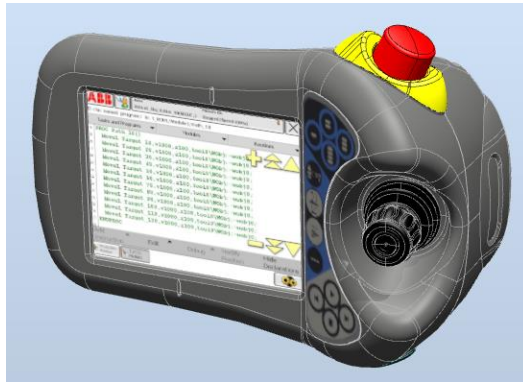


Figura 4.10 - Componente FlexPendant

Este componente se importa de la biblioteca de equipamiento de RobotStudio. En total se importan 6 FlexPendant, situados uno en cada una de las cabinas de control IRC5P.

Pedestal

Este componente está formado por dos tetraedros, creados mediante la herramienta **Solido - Tetraedro**, que actúan como un pedestal sobre el que se fija el componente **Limpiador del atomizador**, expuesto posteriormente. (Figura 4.11)

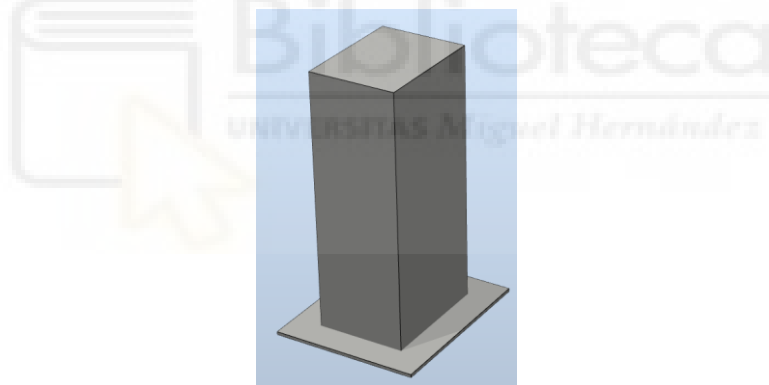


Figura 4.11 - Componente Pedestal

La placa base del componente tiene una longitud de 600mm y una anchura de 500mm. El cuerpo del pedestal tiene una longitud de 400 mm y una anchura de 300 mm. La altura total de este componente es de 895 mm.

Puerta

Este componente es una representación tridimensional simple de las puertas de acceso al interior de la estación y a las zonas de mantenimiento. (Figura 4.12)

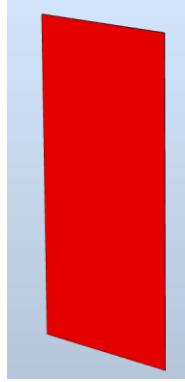


Figura 4.12 - Componente Puerta

El tetraedro que forma la puerta tiene una longitud de 1000 mm, una anchura de 1 mm y una altura de 2000 mm. Se le aplica un color rojo para diferenciarlas claramente de las paredes de la estación.

Suelo

Este componente, formado simplemente por una superficie, actúa como el suelo de la estación, para dotar a la misma de un poco más de realismo. (Figura 4.13)



Figura 4.13 - Componente Suelo

Este componente es una superficie con una longitud de 24000 mm y un ancho de 15000 mm. A este componente se le otorga una apariencia de gráficos, es decir un color y textura, de hormigón.

Mecanismo trampilla de mantenimiento

Este componente tridimensional, formado simplemente por dos superficies, actúa como el mecanismo, formado por una pared con un hueco y una trampilla, que al abrirse permite introducir las herramientas de los robots para realizar la limpieza o mantenimiento de las mismas sin acceder al interior de la estación. (Figura 4.14)

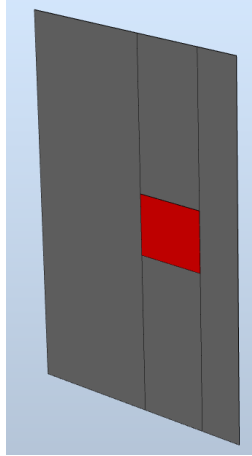


Figura 4.14 - Componente Mecanismo trampilla de mantenimiento

La superficie que actúa de pared tiene una longitud de 2750 mm y una altura de 4000 mm. El hueco está situado a 500 mm a la izquierda y 1700 mm por encima de la esquina derecha inferior. El hueco y la trampilla tienen una longitud de 800 mm y una altura de 650 mm, permitiendo así la entrada de cualquier herramienta de la estación. La trampilla se colorea en rojo para diferenciarla de la pared.

Paredes y techo de la estación

Este componente tridimensional, formado por un gran número de superficies, actúa como las paredes y el techo de la estación al completo. Este componente es simétrico respecto de su eje más largo. Tiene una zona principal donde está situada la estación de pintado propiamente dicha y dos zonas más pequeñas destinadas al mantenimiento y limpieza de las herramientas. (Figura 4.15)

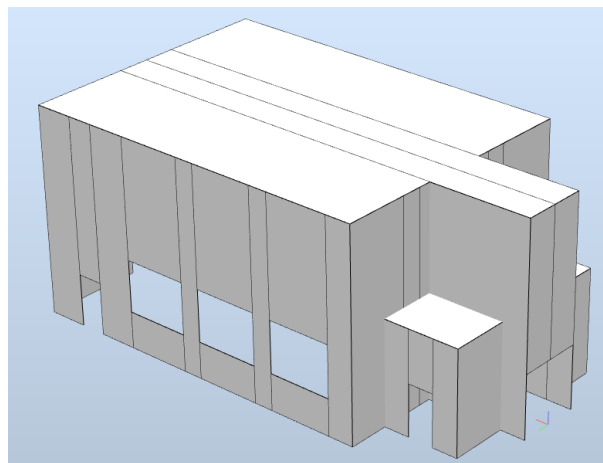


Figura 4.15 - Componente Paredes y techo de la estación

En la zona principal se tienen 3 grandes ventanas a cada lado, de modo que es posible observar el proceso de pintado desde el exterior de la estación, y una puerta a cada lado para permitir el acceso al interior de la estación en caso de necesidad. Lógicamente se tiene una abertura para la entrada y otra para la salida de la carrocería de la estación. Las zonas de mantenimiento tienen una puerta de acceso cada una.

Las dimensiones de este componente se basan en la ubicación del resto de componentes de la estación, de forma que englobe todos los componentes de la estación de pintado, y tenga el suficiente tamaño para no ocasionar colisiones con los robots en movimiento. Estas dimensiones, en mm, se muestran en las siguiente Figura 4.16, donde se muestran las 3 vistas del modelo.

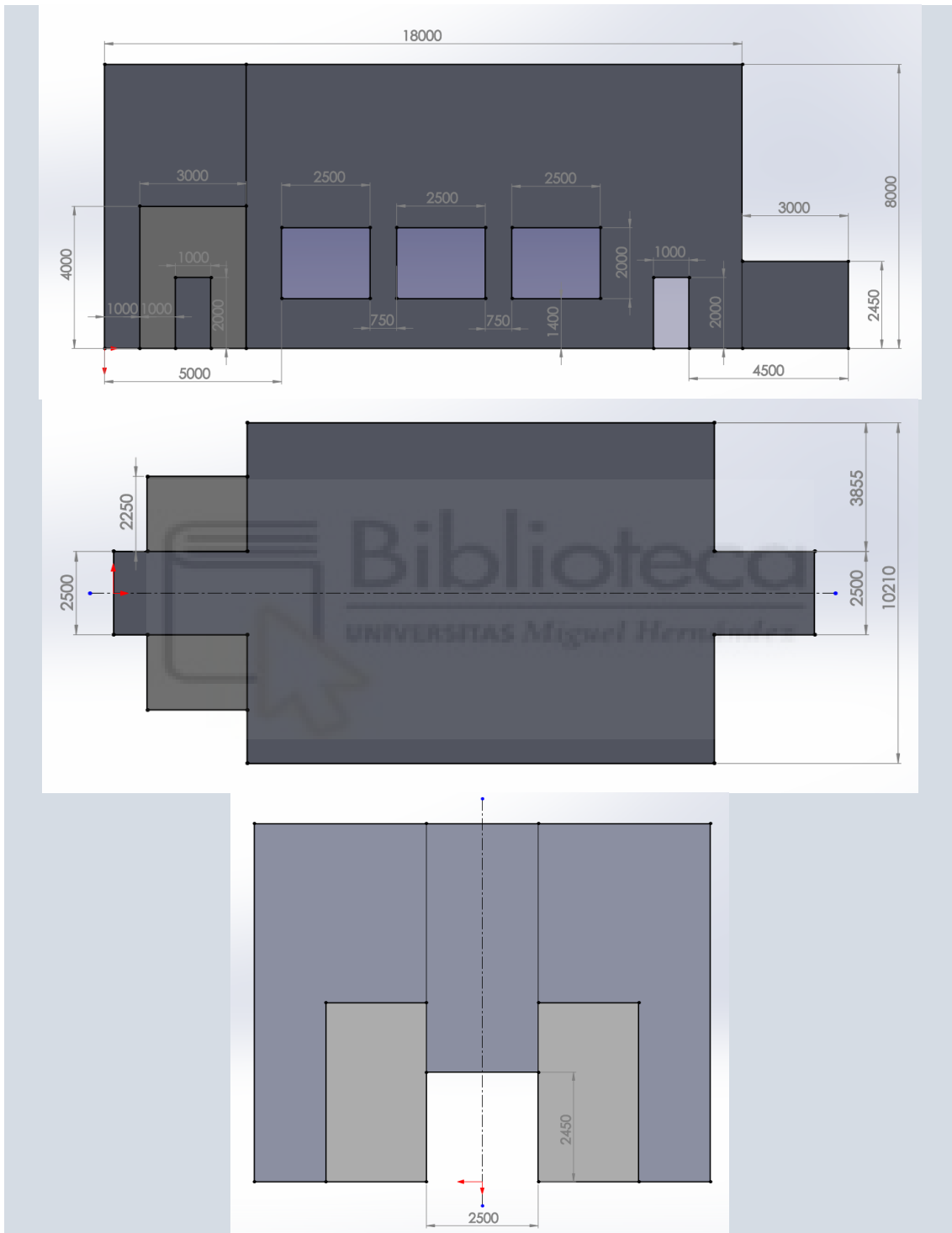


Figura 4.16 – Dimensiones en el alzado, planta y perfil del componente Paredes y techo de la estación

Este componente se diseña mediante la herramienta **Equipment Builder**, expuesta en el apartado [3.1.24 Complementos y PowerPacs](#), mediante la cual es mucho más rápido y simple el diseño de este tipo de componentes.

Valla de protección

Este componente tridimensional actúa como valla de protección para imposibilitar el acceso no autorizado a la zona donde se encuentran posicionadas las cabinas de control IRC5P. Existe un conjunto como el mostrado en la Figura 4.17 a cada lado de la estación, protegiendo los dos conjuntos de cabinas.

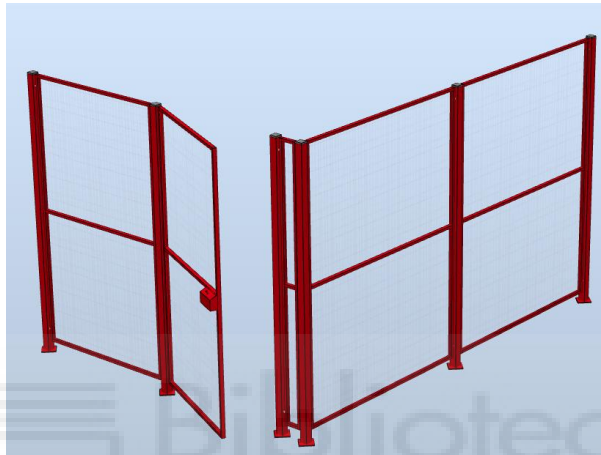


Figura 4.17 - Componente Valla de protección

Este componente está formado por varios módulos de valla y una puerta para acceder al interior del mismo.

Este componente se diseña mediante la herramienta **Equipment Builder**, expuesta en el apartado [3.1.24 Complementos y PowerPacs](#), mediante la cual es mucho más rápido y simple el diseño de este tipo de componentes.

4.2.3 Selección y/o diseño de componentes mediante softwares externos

En este apartado se exponen los componentes de la estación, que por su complejidad, han sido diseñados mediante los softwares expuestos en el apartado [3.3 Otros softwares utilizados](#), e importados a la estación mediante las herramientas expuestas en el apartado [3.1.5 Importación de robots y otros componentes de la estación](#).

Carrocería Mercedes Benz Clase A A45 AMG

Tras los sistemas robóticos, compuestos por robots, tracks y herramientas, este es el componente de mayor importancia en la estación, ya que sobre el mismo se simulará la aplicación de la capa base, lo que implica que todas las trayectorias programadas se basan en la geometría y posición de este componente. Por este motivo se ha intentado que esta carrocería sea lo más fiel posible a la carrocería real del Mercedes Benz Clase A A45 AMG.

Inicialmente se parte del modelo del vehículo Mercedes Benz Clase A A45 AMG completo obtenido gratuitamente a través de la web GrabCAD, en la cual usuarios comparten de forma gratuita modelos 3D CAD, que se muestra en la siguientes Figuras 4.18 y 4.19. Se seleccionó este modelo por su gran diseño y parecido con el modelo real y por la cantidad de componentes que lo formaban, ya que esto permitiría una edición posterior más fácil.



Figura 4.18 - Vista en perspectiva del modelo inicial de carrocería



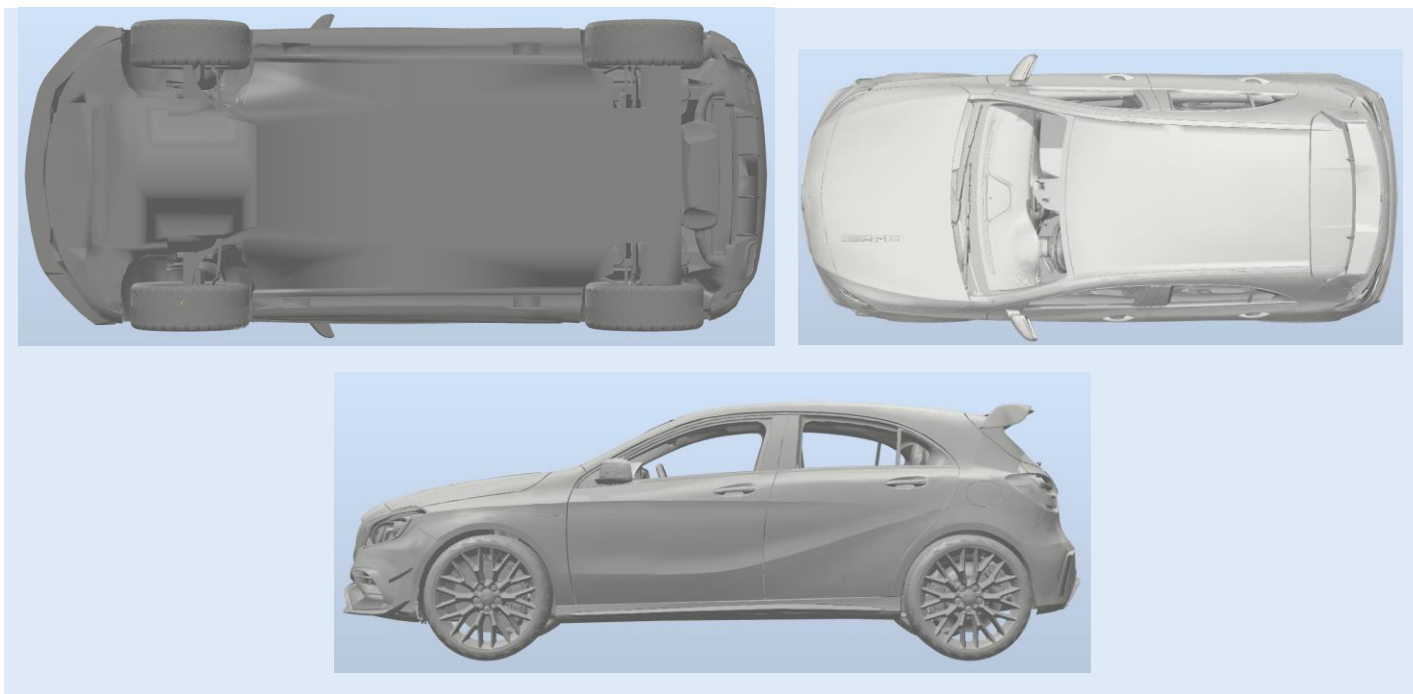


Figura 4.19 - Vistas del modelo inicial de carrocería

Al tratarse del modelo completo del vehículo, se tienen multitud de componentes que no forman parte de la carrocería, como las ruedas, el interior, el motor, etc., y que, o bien no se pintan, o se pintan en otra estación de pintado diferente a la de la carrocería.

Por este motivo, y con el fin de obtener un resultado final lo más fiel a la realidad, se comenzaron a implementar sobre el modelo un gran cantidad de procesos para eliminar y/o modificar partes del mismo y así obtener finalmente la carrocería del vehículo.

El primer paso fue eliminar mediante SolidWorks todos los componentes posibles que no formarán parte de la carrocería, como las ruedas y el interior.

Puesto que el modelo original no permitía hacer muchas más modificaciones mediante este software, se cambió de formato a .stl, obteniéndose así un modelo geométrico mallado compuesto por infinidad de triángulos geométricos, y se comenzó un largo proceso de remodelación mediante el software Blender. Aquí se eliminaron componentes como el alerón, todos los embellecedores, los faros, los parachoques, los retrovisores, las manecillas de las puertas, partes que aún quedaban del interior y muchos más.

Tras esto se separaron los componentes que forman el conjunto de la carrocería, es decir, el chasis, las puertas delanteras, las puertas traseras, el capó y el maletero, y se comenzaron a modificar por separado para asemejarlos a los de la carrocería real. El primer paso fue eliminar, añadir y modificar un gran número de los triángulos geométricos que formaban los modelos, siendo esto un proceso sencillo pero muy costoso en cuanto a tiempo.

Cuando se obtuvo el resultado deseado se mejoraron los componentes que formaban las puertas. Inicialmente las puertas eran una simple superficie, sin ningún armazón interior (Figura 4.20). Para añadirles este armazón se obtuvo un modelo 3D CAD, de la web GrabCAD, de puerta genérica de automóvil que sí contaba con este armazón interior (Figura 4.21).



Figura 4.20 - Modelo de la puerta sin interior



Figura 4.21 - Modelo tridimensional de puerta genérico con armazón interior

Este modelo se cambió a formato .stl y mediante Blender se consiguió aislar el armazón interior de la puerta. Posteriormente, ya que este armazón interior lógicamente no encajaba en dimensiones con la puerta del modelo de Mercedes, se modificó gráficamente mediante Blender para hacer que encajara lo mejor posible con el modelo de puerta, tanto delantera como trasera, del modelo de Mercedes, obteniéndose el resultado mostrado en la Figura 4.22.

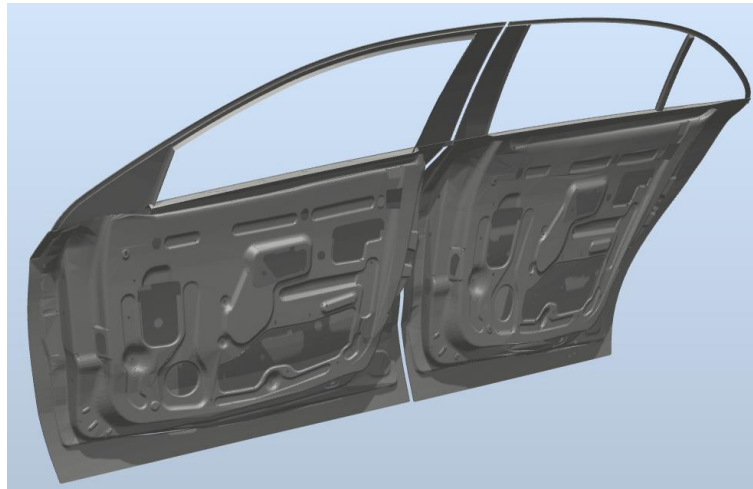


Figura 4.22 - Modelos de puertas terminadas

Una vez echo todo esto, se incorporaron todos los componentes terminados a RobotStudio y se posicionaron correctamente para formar el conjunto de la carrocería completa. Para ello se tuvo que reposicionar el origen local de todos los componentes y después posicionarlos meticulosamente. Finalmente se añadieron todos a un Grupo de componentes y se cambió el color de todos los componentes a gris claro, para que el modelo tenga un color similar al que tendría tras aplicarle la capa de imprimación, obteniéndose el resultado mostrado en las siguientes Figuras 4.23, 4.24, 4.25 y 4.26.

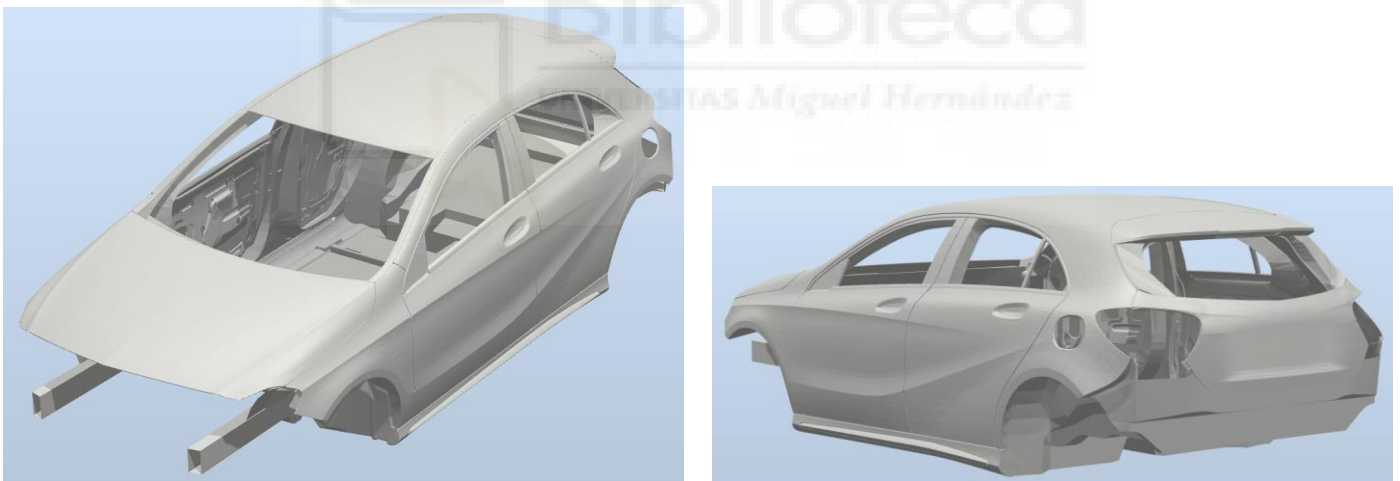


Figura 4.23 - Vista en perspectiva del modelo final de carrocería con las puertas cerradas

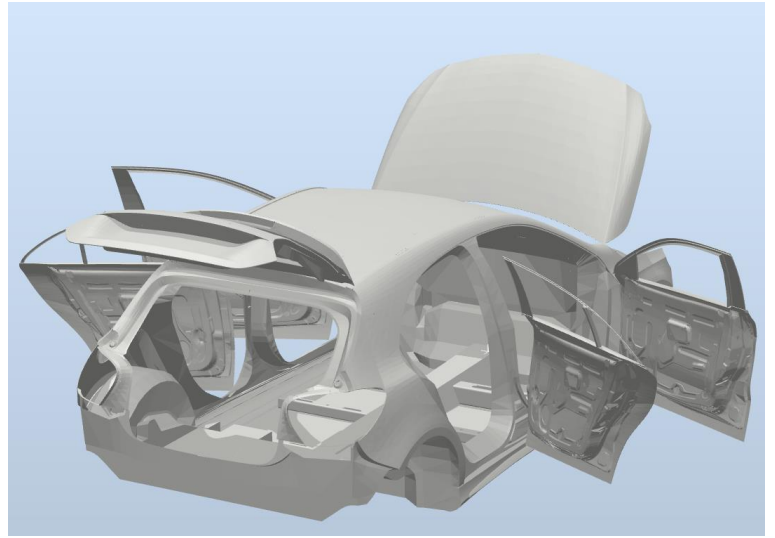
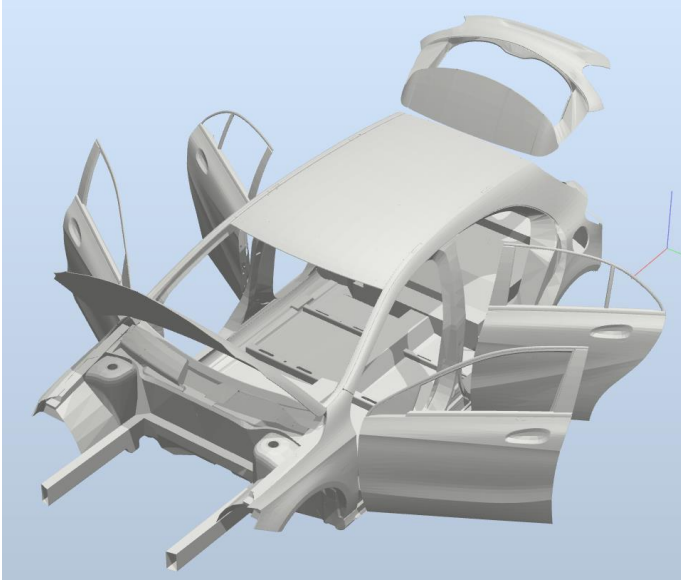
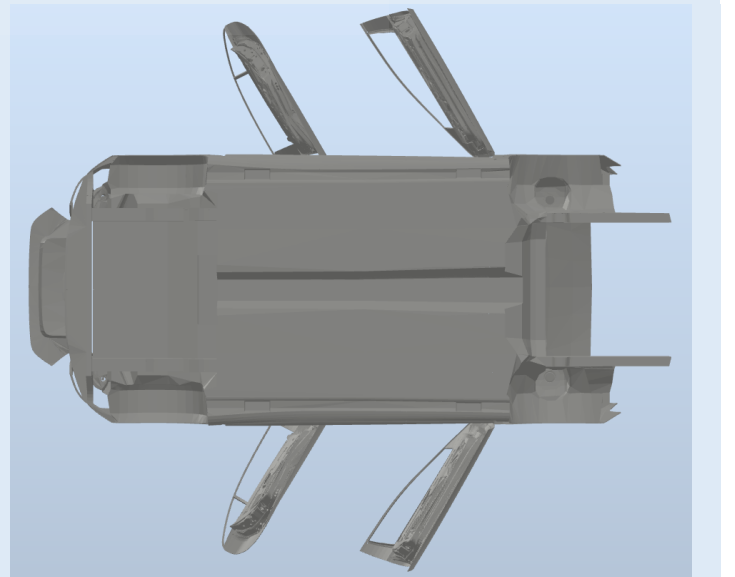
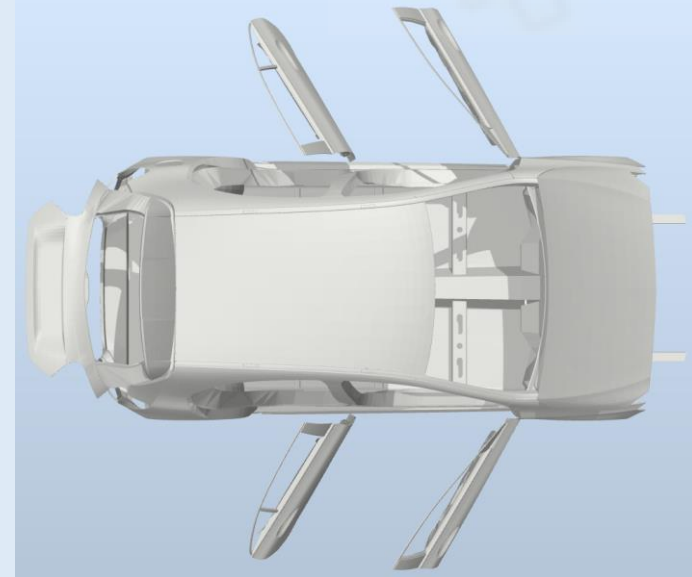
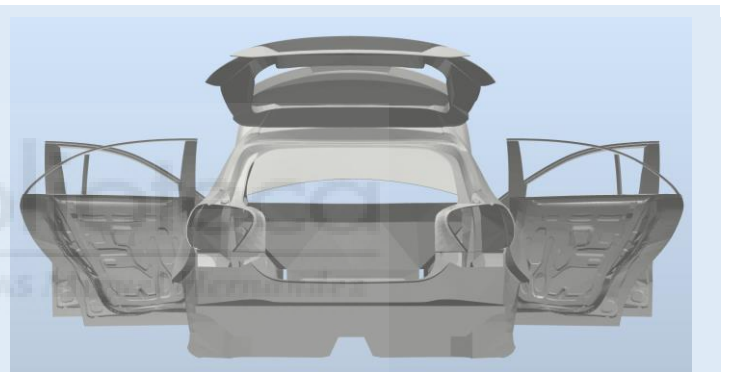
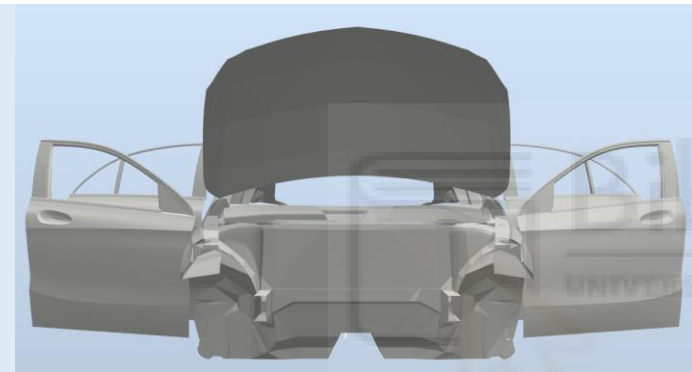


Figura 4.24 - Vista en perspectiva del modelo final de carrocería con las puertas abiertas



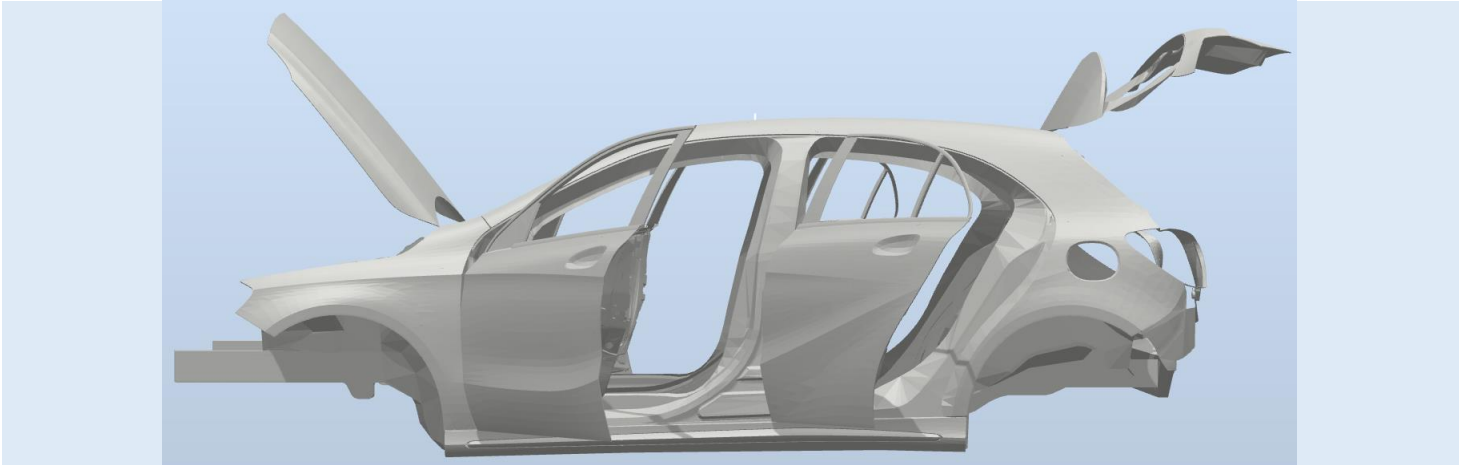


Figura 4.25 - Vistas del modelo final de la carrocería con las puertas abiertas

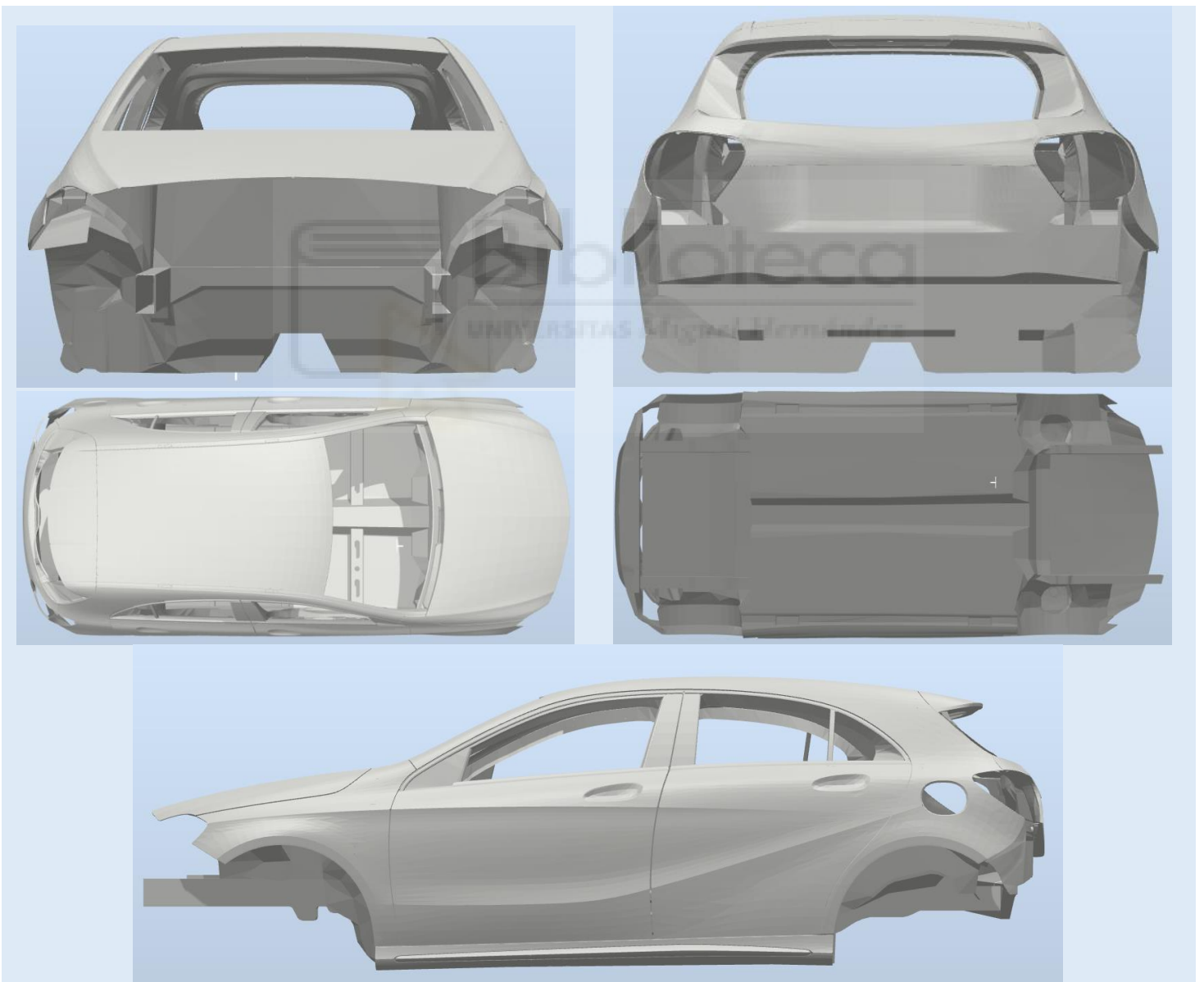


Figura 4.26 - Vistas del modelo final de la carrocería con las puertas cerradas

Este componente se ha explicado con mayor detalle que el resto ya que es muy importante para el desarrollo del proyecto y obtener su aspecto final llevo mucho esfuerzo y tiempo.

Las medidas del componente de la carrocería final son un aspecto muy importante a tener en cuenta para la colocación de la misma y de los robots y tracks en la estación, puesto que debe ser posible la abertura de las puertas, capó y maletero y los robots deben poder acceder al interior de estos componentes para pintarlos, sin que existan colisiones. Las medidas, en mm, se muestran en la siguiente Figura 4.27.

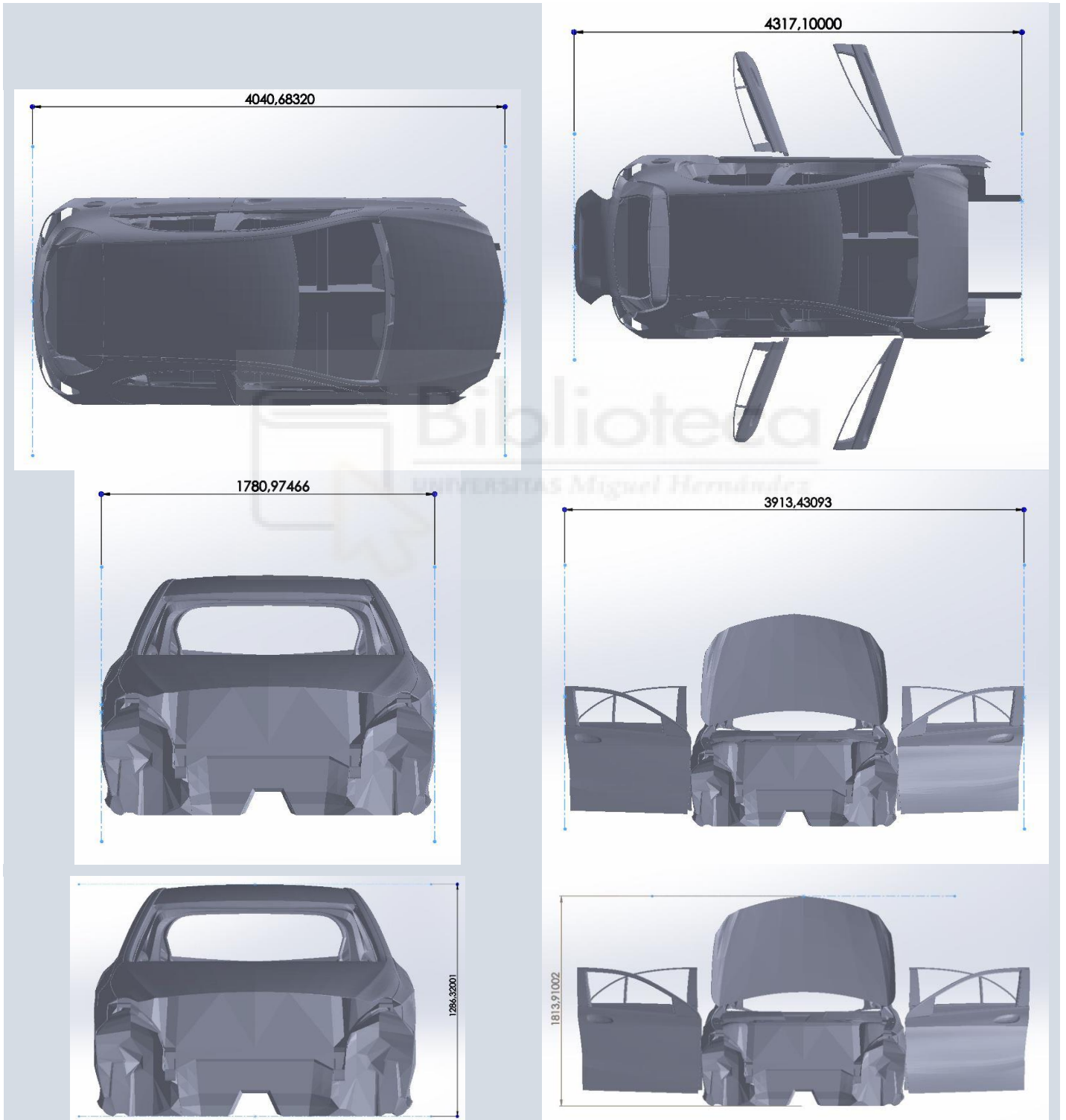


Figura 4.27 - Medidas del modelo final de carrocería (mm)

Pilares

Para sustentar el conjunto formado por track IRB 5500 Elevated Rail, el robot IRB 5500 y la herramienta RB1000i, con un peso total de alrededor de 4400 kg, posicionado sobre el robot IRB 5500 inferior, se diseñan mediante SolidWorks una serie de pilares en los que se anclará el track. (Figura 4.28)

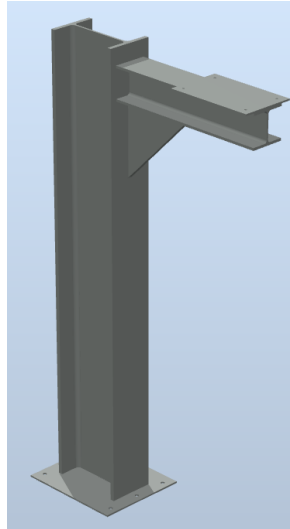


Figura 4.28 - Pilar soporte conjunto robótico superior

Cada uno de los conjuntos robóticos superiores, estará sustentado por seis pilares como el mostrado en la Figura 4.28, situados a la misma distancia unos de otros. Cada uno de estos pilares están compuestos por una placa de anclaje al suelo de 600x700x10 mm, un pilar formado por un perfil metálico HEB 500 de 2950 mm de longitud, una viga formada por un perfil metálico HEB 220 de 1080 mm de longitud, situada a 109,5 mm del extremo superior del pilar, con una placa de anclaje al final de la misma de 572x350x10 mm, sobre la que se anclará el track, y un nervio de refuerzo de 500x500x30 mm.

Este es un diseño rápido, pero necesario, ya que el tamaño del pilar y la viga afectarán al espacio del que dispone el robot IRB 5500 inferior, pero deben dimensionarse correctamente para soportar el peso del conjunto robótico superior.

Para tener un correcto diseño se realizó un análisis estático simple mediante SolidWorks. Para realizarlo se selecciona el acero estructural S275JR para todos los elementos del pilar, se ancla al suelo la placa de anclaje y se aplica una carga igual a los 4400 kg, del conjunto robótico superior, mayorada mediante un coeficiente de 1,2, y dividida por 6, al disponerse de 6 pilares de apoyo, obteniéndose una carga de 880 kg por pilar. Además se aplicará también la fuerza de la gravedad. Los resultados de desplazamientos obtenidos mediante este análisis se muestran en la Figura 4.29.

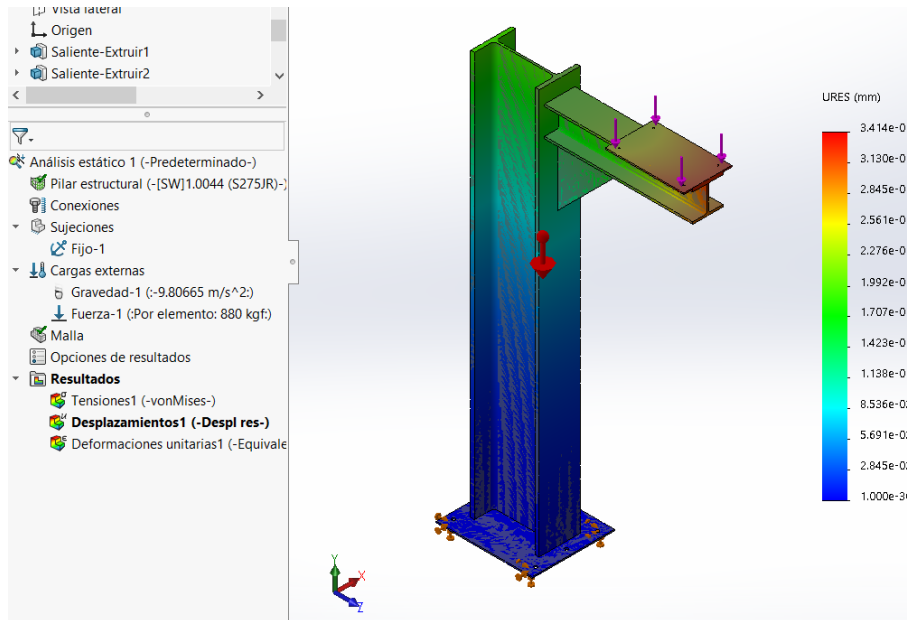


Figura 4.29 - Análisis estático simple del pilar de soporte

Así puede comprobarse que con esta carga se tendría un desplazamiento máximo de 0.3 mm, por lo que el diseño se considera correcto, conociéndose así con certeza el espacio del que dispone el robot IRB 5500 inferior.

Estructura soporte o bancada de la carrocería

Para elevar la carrocería del suelo una distancia suficiente para que todos los robots puedan realizar sin problema la aplicación de la capa base, se diseña mediante SolidWorks una estructura de soporte para la carrocería. (Figura 4.30)

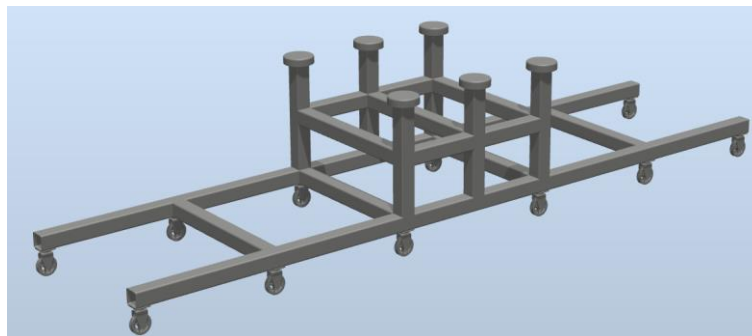


Figura 4.30 - Estructura de soporte de la carrocería

Esta estructura está formada por una serie de perfiles metálicos cuadrados 100.4, cuyas dimensiones se han estimado en función del tamaño de la carrocería. La estructura dispone de una serie de ruedas, ya que esta estructura, junto con la carrocería, es transportada por el transportador de la estación.

Realmente este componente no tendría estas características, pero al desconocerse como podría ser se ha intentado diseñar una aproximación, de forma que la carrocería se posicione sobre el mismo a la distancia apropiada del suelo.

Herramientas de agarre para la abertura del capó y del maletero

Puesto que se debe realizar el pintado interior del capó y del maletero, estos se deberán levantar. Para realizar esta tarea se emplearán los robots superiores IRB 5500, a los atomizadores de los cuales se fijarán las siguientes herramientas diseñadas para esta tarea (Figura 4.31). El robot superior izquierdo será el encargado de levantar y bajar el capó, y el robot superior derecho será el encargado de levantar y bajar el maletero.

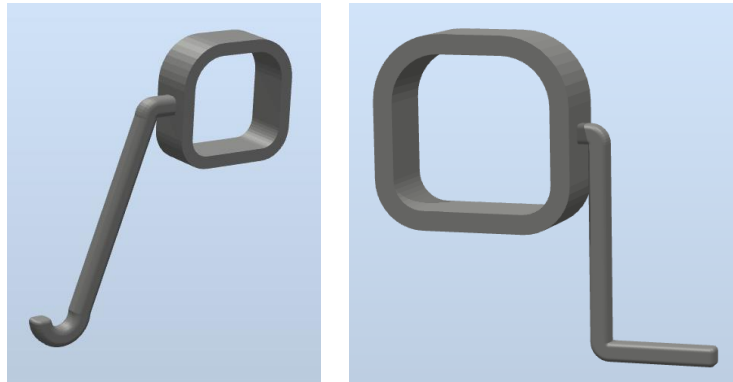


Figura 4.31 - Herramientas de agarre para el robot superior izquierdo (izquierda) y el robot superior derecho (derecha)

Estas herramientas se han diseñado mediante SolidWorks teniendo en cuenta el tamaño del cuerpo del atomizador, la posición del capó y el maletero y el alcance de los robots superiores.

Elementos de agarre para el capó y el maletero

Como se ha expuesto, el capó y el maletero son levantados por los robots IRB 5500 superiores de la estación. Para ello, además de las herramientas de agarre anteriormente expuestas, se necesitan unos elementos de agarre, de forma que las herramientas empujen de estos elementos y no del propio capó o maletero, lo cual dañaría el recubrimiento aplicado. (Figura 4.32)

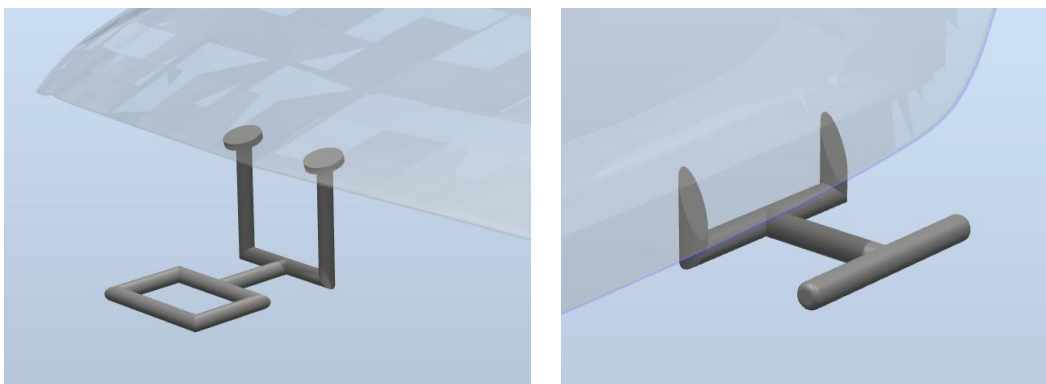


Figura 4.32 - Elementos de agarre para el capó (izquierda) y el maletero (derecha)

Estos elementos se han diseñado mediante SolidWorks basándose en la geometría del capó y del maletero y en el alcance de los robots superiores.

Limpiador automático del atomizador

Para simular la limpieza del plato de campana de los atomizadores, que debe llevarse a cabo en los cambios de color o si se requiere por otra razón, se ha diseñado un modelo de limpiador del plato de campana del atomizador. (Figura 4.33)

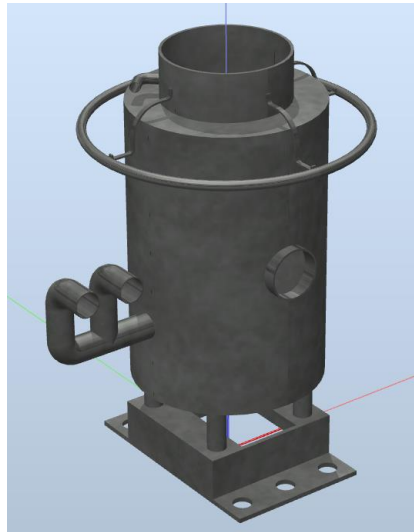


Figura 4.33 - Limpiador automático del atomizador

Este modelo se ha diseñado mediante SolidWorks basándose en el modelo Bell Cleaner External Charge del fabricante CrystalCap, ya que este es un modelo ideal para la limpieza de atomizadores con electrodos externos.

Estos componentes no intervienen en la simulación, pero sirven para programar la trayectoria de limpieza de los atomizadores.

Se añadirá uno por cada robot de aplicación, es decir, 4 en total. Los limpiadores para los robots superiores estarán posicionados sobre el pedestal descrito anteriormente, en el inicio de la estación de pintado. Los limpiadores para los robots inferiores estarán posicionados sobre una placa anclada al suelo de rejilla, al final de la estación de pintado, como se muestra en la siguiente Figura 4.34.

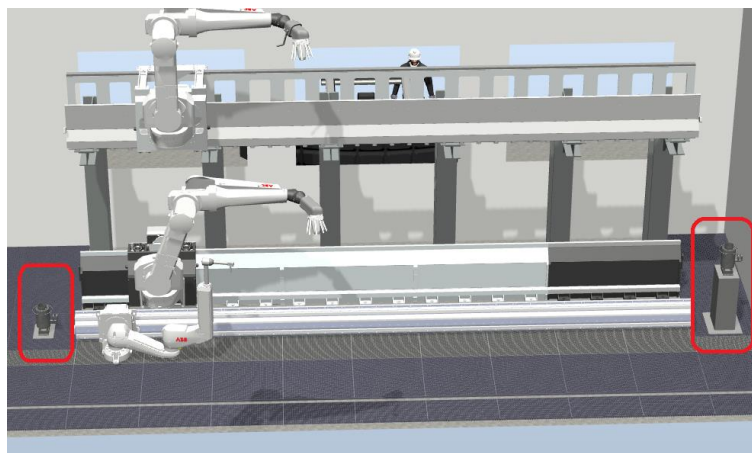


Figura 4.34 - Posicionamiento de los limpiadores de los atomizadores

Suelo de rejilla

El interior de la estación de pintado debe tener un suelo que permita que el material de pintado sobrepulverizado pase a través de él hasta llegar al sistema de limpieza. Para simular esto, se ha diseñado un suelo a medida compuesto por módulos de rejilla de metal de 1000x1000 mm. (Figura 4.35)

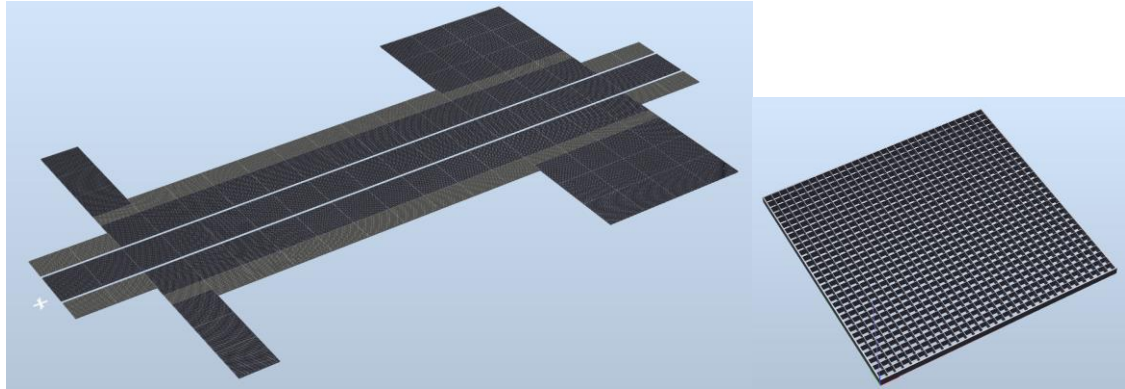


Figura 4.35 - Suelo de rejilla de la estación de pintado

Los modelos de estos módulos se han obtenido gratuitamente a través de la web GrabCAD. Para formar el conjunto se han importado a RobotStudio, se han posicionado correctamente y se han añadido a un grupo de componentes, de forma que todos actúan como un único componente. Para permitir el paso de conjunto formado por el soporte y la carrocería se dejan dos huecos por donde están los raíles del transportador.

Torre de señalización luminosa industrial

Este componente tridimensional actúa como elemento de señalización luminoso y elemento de seguridad, indicando el modo en el que se encuentra la estación. (Figura 4.36)

Si se ilumina elemento de color verde, indica que todos los componentes móviles de la estación están pausados o detenidos, con lo cual sería seguro acceder al interior de la estación. Si se ilumina el elemento de color rojo, indica que los sistemas robóticos están en movimiento y por lo tanto no es posible acceder al interior de la estación. Si se ilumina el elemento de color ámbar, indica que solo está en movimiento uno de los robots para realizar la limpieza de su atomizador o tareas de mantenimiento, por lo que existe un cierto peligro si se accede a la estación.

La iluminación del mismo se consigue combinando el modelo con una serie de fuentes de luz puntuales y un conjunto de componentes inteligentes.



Figura 4.36 - Componente Torre de señalización luminosa industrial

El modelo tridimensional se obtuvo de forma gratuita a través de la web GrabCAD. Posteriormente, se añadió a RobotStudio y se editó para darles el color adecuado a cada uno de los elementos de señal de color del mismo y darles la apariencia gráfica de vidrio transparente.

Se situará una de estas señales luminosas encima de cada una de las puertas de la estación, por lo que en total se tendrán 4 torres de señalización luminosa en la estación.

Operario

Este modelo tridimensional es la representación de un operario de la estación. (Figura 4.37)



Figura 4.37 - Componente Operario

Este componente se obtuvo de la web 3dwarehouse, una web de modelos 3D CAD gratuitos similar a la ya mencionada GrabCAD.

Este componente no interviene en la simulación de la estación pero se incluye en la estación para dotar a la simulación de más realismo.

Puesto de control

Este modelo tridimensional se añade como representación del puesto de control de la estación de pintado. (Figura 4.38)



Figura 4.38 - Componente Puesto de control

Este componente se obtuvo de la web 3dwarehouse, una web de modelos 3D CAD gratuitos similar a la ya mencionada GrabCAD.

Este componente no interviene en la simulación de la estación pero se incluye en la estación para dotar a la simulación de más realismo.

4.3 Creación y programación de la estación en RobotStudio

Una vez se han seleccionado los componentes que intervienen directamente en la aplicación de pintura, es decir, robots, tracks y herramientas, expuestos anteriormente, para la creación y programación de la estación se seguirá el siguiente flujo de trabajo.

El primer paso es la creación de una estación vacía de RobotStudio. Seguidamente se importarán a la estación los robots, tracks y sus herramientas y el conjunto que forman la carrocería y su bancada. También se creará el transportador que mueve el conjunto de la carrocería.

Hecho esto comenzará un proceso iterativo de posicionamiento de estos elementos para conseguir una disposición compacta, que permita que los robots alcancen todas las zonas a pintar de la carrocería, permitiendo un movimiento fluido de los mismos y sin que existan colisiones entre todos los elementos.

Se dice que este proceso es iterativo, ya que pesa a tener definida una disposición conceptual de la estación, al ir posicionando los diferentes elementos se puede observar que no es la mejor posición, que colisionan en algún punto o que su área de trabajo no permite alcanzar alguna zona que sí debería. Además, aun habiendo posicionado todos los elementos en el lugar que se creía idóneo, al programar los objetivos y trayectorias, puede que algunos se tengan que repositionar ligeramente para que todos los objetivos de las trayectorias sean alcanzables.

Una vez posicionados estos elementos se procederá a la creación y configuración de todos los controladores virtuales de robot. Se deberán crear seis controladores virtuales, uno para cada conjunto de robot y track, ya que RobotStudio solo permite un robot de pintura por controlador.

A continuación se crean las conexiones del transportador con los controladores para que estos sean capaces de seguir el movimiento de la carrocería al desplazarse a lo largo de la estación.

Teniendo todos estos aspectos bien configurados, se pasa a la creación de la plantilla de instrucciones, común para todos los controladores de robot, objetos de trabajo, puntos objetivo y trayectorias. Este proceso, junto con la programación en RAPID, es el proceso que más tiempo y esfuerzo ha necesitado. Este también es un proceso iterativo, ya que existen muchas trayectorias con multitud de puntos objetivo en cada una de las mismas, y se deberá comprobar el correcto alcance del robot en cuestión para cada uno de ellos y la correcta configuración del robot en los mismos y para pasar de una trayectoria a otra, lo cual resulta en un continuo reposicionamiento, reorientación y reconfiguración de los puntos objetivo.

Al mismo tiempo que se va desarrollando la creación de las trayectorias, también se desarrolla la lógica de estación, es decir, se van creando componentes inteligentes, conexiones entre los robots de la estación y entre los robots y los componentes inteligentes y enlazamientos de propiedades entre los componentes inteligentes. Esto posibilita que los robots puedan realizar la aplicación de la capa base de forma coordinada y sin colisionar entre ellos. De nuevo este será un proceso iterativo en continuo cambio hasta el final del proceso de creación.

Simultáneamente se crean las señales de E/S de los sistemas de robot, que permiten la comunicación entre los mismos y con los componentes inteligentes. También se sincronizan continuamente las trayectorias creadas para poder comprobar su funcionamiento y modificarlas si fuera necesario.

Todo esto se crea y programa poco a poco, trayectoria a trayectoria, comprobando el alcance, configuración, movimiento del robot, comunicación entre los robots y otros aspectos, realizando continuas simulaciones, hasta conseguir la nube de puntos objetivo y trayectorias que se muestra en la siguiente Figura 4.39.

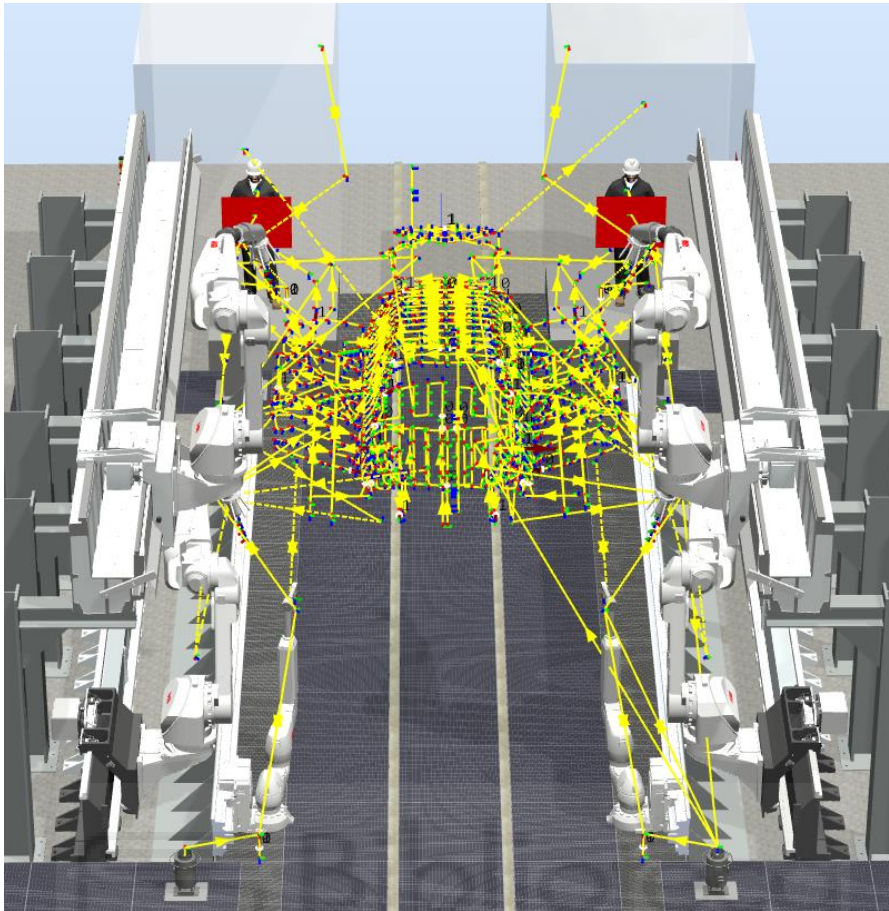


Figura 4.39 - Conjunto de puntos objetivo y trayectorias de la estación finalizada

A medida que aumenta el número de trayectorias, y por lo tanto el número de puntos objetivo, la complejidad de la programación aumenta.

Una vez se tienen programadas las trayectorias, de todos los robots, que realizan el proceso de aplicación de la capa base, se pasa a programar las trayectorias de limpieza, mantenimiento y seguridad de la estación. Al mismo tiempo se programa la lógica de estación para estas tareas. También se crean o importan y se posicionan algunos componentes que intervienen en estas tareas.

Paralelamente a todo este proceso de programación gráfica, se realiza un proceso de programación textual en el editor de lenguaje RAPID de RobotStudio. De la misma forma que la programación de trayectorias, a medida que se desarrolla la estación aumenta la dificultad de programación en RAPID para hacer que todos los elementos funcionen de la forma correcta.

Finalmente se añaden y/o crean y posicionan los componentes de la estación que no desempeñan una función activa en la misma, como por ejemplo, el suelo, las paredes, los operadores, etc.

Cabe destacar, que para la creación de esta estación principal, se han creado paralelamente otras muchas estaciones simples para aprender a emplear y configurar ciertas herramientas de RobotStudio, ya que un error cuando la creación de la estación principal estaba bastante avanzada podía suponer un caos total.

4.3.1 Creación de la estación vacía

El primer paso es crear una estación vacía, para eso se emplea la herramienta **Estación vacía** de la pestaña **Archivo**, como se expone en el apartado [3.1.4 Creación de una estación](#). Así se obtiene un espacio gráfico vacío para comenzar la creación desde cero.

La estación se guarda con el nombre `Pintado_Carroceria_Dinamico.rsstn`, aunque el nombre no es relevante en ningún aspecto.

4.3.2 Importación/creación y posicionamiento de componentes fundamentales

Una vez creada la estación, se van importando y posicionando todos los robots, los tracks, las herramientas y demás componentes fundamentales indicados en el apartado anterior, siguiendo las indicaciones expuestas en los apartados [3.1.5 Importación de robots y otros componentes de la estación](#), [3.1.6 Creación/importación de tracks y mecanismos](#) y [3.1.7 Importación del efector final del robot](#), para su importación, y las del apartado [3.1.11 Posicionamiento de robots y otros componentes de la estación](#), para su posicionamiento.

Se importarán dos robots IRB 5500-23 y dos robots IRB 5500-25, uno para cada lado de la estación. Para esto en el momento de importarlos a la estación se deberán seleccionar estos modelos y sus características, como se muestra en las Figuras 4.40 y 4.41.

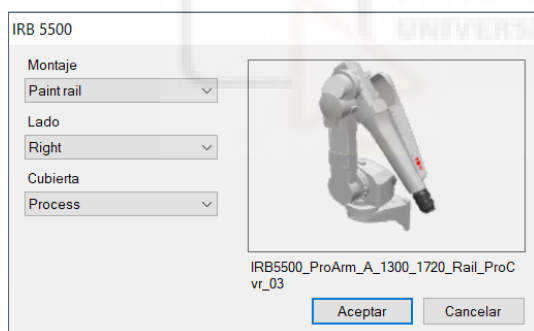


Figura 4.40 - Importación del robot IRB 5500-23

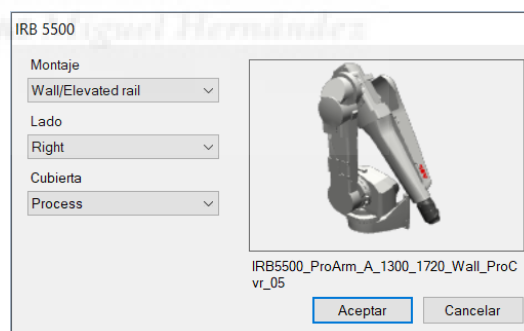


Figura 4.41 - Importación del robot IRB 5500-25

El modelo 5500-23 se especifica seleccionando la opción de montaje **Paint rail**, mientras que para el modelo 5500-25 esta opción se denomina **Wall/Elevated rail**.

Para los dos modelos de robot, se importará uno de los dos con la opción **Lado Right** y el otro con esta opción en versión **Left**. Esta opción es importante ya que de ella depende que el robot disponga de una mayor área de trabajo hacia su lado derecho o hacia su lado izquierdo, es decir, que su eje de rotación 1 permita una mayor rotación en un sentido o en el otro.

Para los cuatro robots IRB 5500 se seleccionará opción de cubierta **Process**, de forma que se añada un componente **Cubierta** al modelo del robot, cuya función sería la de proteger los sistemas de suministro, dosificación, cambio de color, etc. que estarían montados sobre el brazo del robot. Esta opción es importante ya que al seleccionarla el modelo tridimensional del robot está equipado con esta cubierta y por lo tanto es más ancho, aspecto que es importante cuando el robot accede a partes interiores estrechas, como debajo del capó o dentro de las puertas.

También se importarán dos robots IRB 5350 en su versión de cuatro ejes, uno para cada lado de la estación. Como en el caso de los robots anteriores se deberá seleccionar un modelo para cada lado y sus características, como se muestra en las Figuras 4.42 y 4.43.

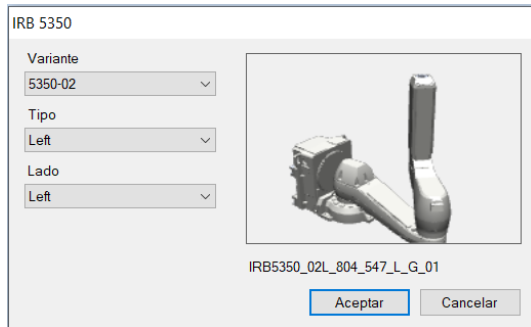


Figura 4.42 -Importación del robot IRB 5350 lado izquierdo

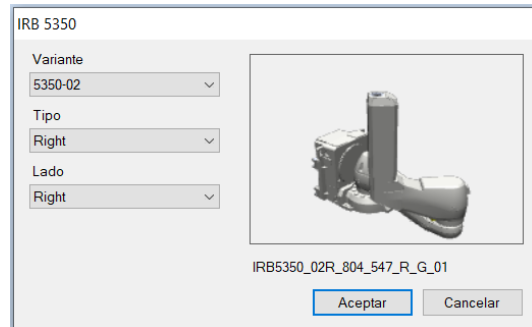


Figura 4.43 -Importación del robot IRB 5350 lado derecho

En ambos robots se selecciona la **Variante** 5350-02, que es la variante de cuatro ejes que pueden montarse sobre track. Para el que está posicionado en el lado izquierdo de la estación se selecciona el **Tipo Left** y **Lado Left**, y para el del lado derecho el **Tipo Right** y **Lado Right**. En este caso es el eje 2 del robot el que varía según el **Tipo** especificado.

Se importan dos tracks IRB 5500 paint rail extended, uno para cada uno de los robots IRB 5500-23, con las opciones que se muestran en la Figura 4.44.

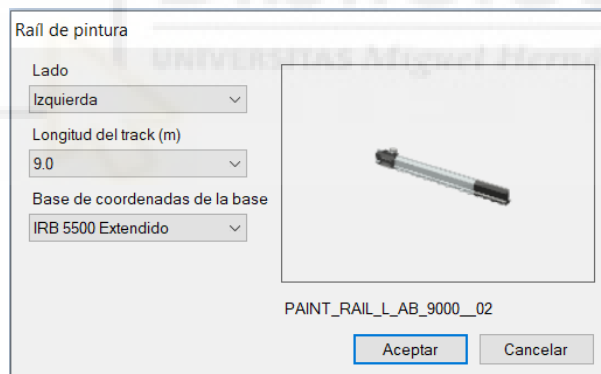


Figura 4.44 -Importación del track IRB 5500 Paint rail extended

Uno de los tracks se importará con la opción **Lado Izquierda** y otro con la opción **Lado Derecha**. La longitud para los dos será de 9 metros. Y en la opción **Base de coordenadas de la base**, se selecciona para los dos la opción **IRB 5500 extendido**, para seleccionar el modelo apropiado para el IRB 5500-23.

Para los dos robots IRB 5500-25, se importan dos tracks IRB 5500 elevated rail, con las opciones que se muestran en la Figura 4.45.

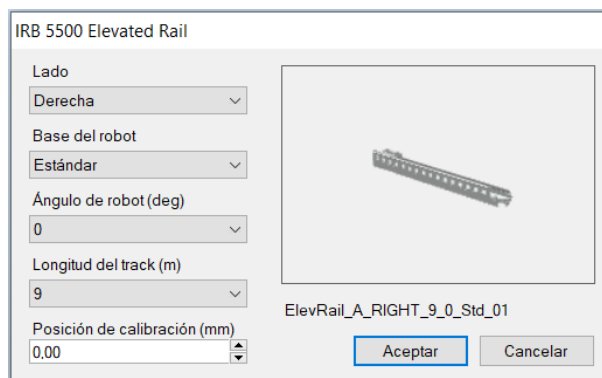


Figura 4.45 -Importación del track IRB 5500 Elevated Rail

Uno de los tracks se importará con la opción **Lado Izquierda** y otro con la opción **Lado Derecha**. Para los dos tracks las siguientes opciones serán iguales. La opción **Base del robot** se selecciona **Estándar**. El ángulo con el que inclinará el robot al posicionarse sobre el track se selecciona 0, de modo que el robot se posicionará en su posición vertical habitual. La longitud será de 9 metros. Y la **Posición de calibración** será 0.

Para los dos robots IRB 5350, se importarán dos tracks IRB 5350 rail, con las opciones que se muestran en la Figura 4.46.

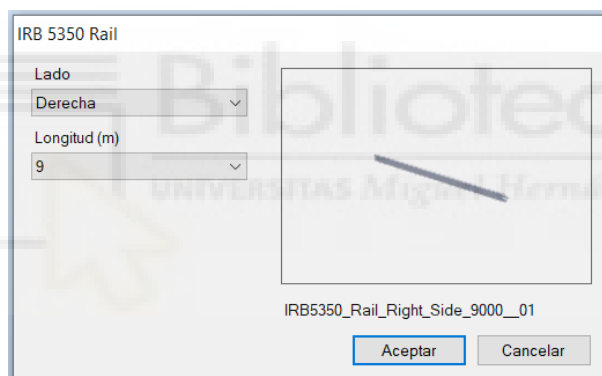


Figura 4.46 -Importación del track IRB 5350 Rail

Uno de los tracks se importará con la opción **Lado Izquierda** y otro con la opción **Lado Derecha**. La longitud para los dos será de 9 metros.

Todos los componentes robots y tracks se importan desde la **Biblioteca ABB** de RobotStudio.

Tras importar los robots y los tracks, el siguiente paso será posicionar los tracks correctamente y conectar los robots a los mismos. El proceso de posicionamiento fue iterativo, realizando multitud de pruebas de configuración hasta alcanzar su posición final. En la siguiente Tabla 4.1 se muestran las posiciones y orientaciones de todos los tracks de la estación respecto del **Sistema de coordenadas mundo de RobotStudio (RS-WCS)**.

	IRB5500 Paint Rail Izq.	IRB5500 Paint Rail Der.	IRB 5500 Elevated Rail Izq.	IRB 5500 Elevated Rail Der.	IRB 5350 Rail Izq.	IRB 5350 Rail Der.
X (mm)	5250	5250	6000	6000	5888	5888
Y (mm)	3800	-3800	2150	-2150	1750	-1750
Z (mm)	0	0	3600	3600	150	150
x (grados)	0	0	0	0	0	0
y (grados)	0	0	0	0	0	0
z (grados)	0	0	0	0	0	0

Tabla 4.1 - Posiciones y orientaciones de los tracks respecto del sistema de coordenadas mundo

Una vez se tienen situados los tracks, se pasa a conectar los robots a los mismos. Conectar el robot al track supone que la posición del robot cambia para situarse en el carro del track, es decir, el robot se une al componente móvil del track. De esta forma la base de coordenadas de la base del robot se sitúa en la misma posición y orientación que la base de coordenadas de la base del track.

Esto implica que el movimiento del track mueve también el robot, de modo que se consigue un área de trabajo y un alcance mucho más grande. Para conectar el robot simplemente se debe seleccionar en el navegador **Diseño** y arrastrarlo hasta el nombre del track al que se quiere conectar.

A continuación se importan las herramientas desde la **Biblioteca de Equipamiento** de RobotStudio. Se importarán 4 atomizadores RB 1000i, en su versión exterior SAD-L. En el cuadro de dialogo que aparece al importar la herramienta se debe seleccionar el modelo RB 1000i_EXT_SAD_L, como se muestra en la siguiente Figura 4.47.

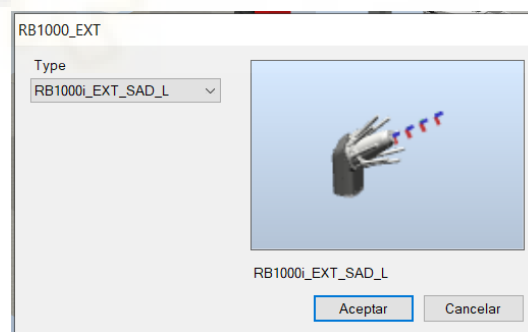


Figura 4.47 - Importación de los atomizadores

Los dos atomizadores de los robots IRB 5500 inferiores se conectan a los mismos de la misma forma que se ha expuesto para los robots y los tracks. Así, el atomizador se conecta a la muñeca del robot, de forma que el movimiento del atomizador pasa a ser controlador por el robot y se pueden seleccionar los TCP de la herramienta.

En el caso de los dos atomizadores de los robots superiores, antes de conectarlos, se deberán importar a la estación los componentes **Herramientas de agarre para la abertura del capo y del maletero**, posicionarlos correctamente en el cuerpo de los atomizadores y conectarlos a estos para que su movimiento dependa de los atomizadores. En este caso no se cambiará la posición de las herramientas de agarre al conectarlas a los atomizadores, puesto que se quiere que se queden fijas donde se han posicionado.

Cuando se tiene posicionadas y conectadas las herramientas de agarre a los atomizadores, estos se conectan a sus respectivos robots.

También se importan dos agarraderas IRB 5350 Gripper. Una vez importadas se conectan a sus respectivos robots.

En la siguiente Figura 4.48 se muestra como quedan posicionados los robots y herramientas del lado derecho al conectarlos.

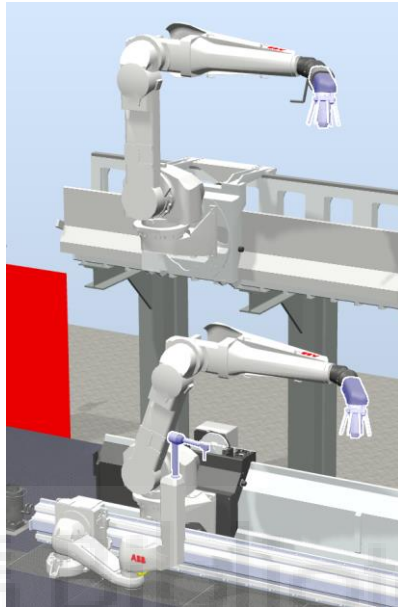


Figura 4.48 - Robots y herramientas del lado derecho conectados

A continuación se crea el componente Transportador, mediante la herramienta **Solido-Tetraedro** expuesta en el apartado [3.1.8 Modelado de componentes en RobotStudio](#).

Para finalizar, se importa el conjunto formado por la bancada de la carrocería y la carrocería, mostrado en la Figura 4.49.

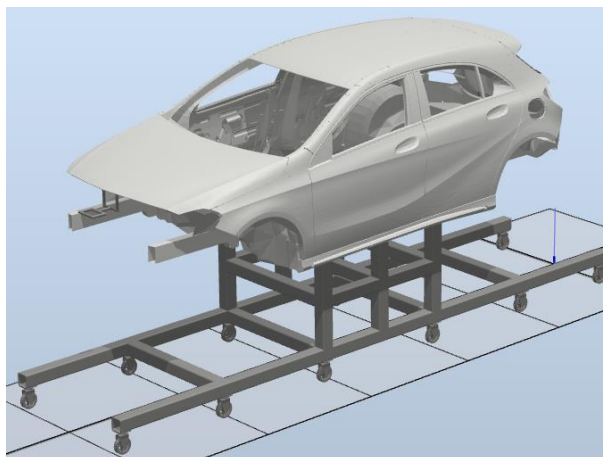


Figura 4.49 - Conjunto carrocería más bancada

Este conjunto se ha creado previamente en otra estación, importando y posicionando correctamente todos los componentes de la carrocería (chasis, puertas, capó y maletero) y la bancada. Tras esto se guarda el conjunto como biblioteca de RobotStudio para así poder importarla y usarla todas las veces que se requiera.

Para posicionar el conjunto primero se modifica la posición del origen local del conjunto, según lo expuesto en el apartado [3.1.9 Modificación de componentes en RobotStudio](#). Este se posiciona en el extremo final del conjunto, es decir, en el lado del maletero, a la altura del punto más bajo de las ruedas de la bancada y en el punto medio del conjunto, como se muestra en la Figura 4.50.

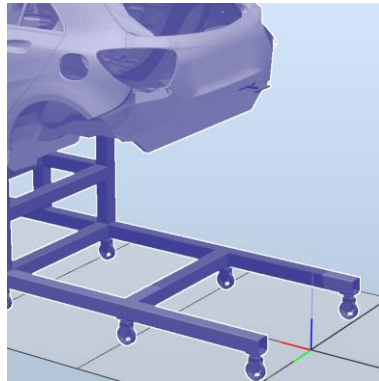


Figura 4.50 - Posición del origen local del conjunto carrocería más bancada

De esta forma se puede colocar el conjunto haciendo coincidir el eje X central de la estación con el eje X central del conjunto, es decir, el conjunto estará centrado en la estación, equidistante a los sistemas robóticos de cada lado de la estación. El conjunto se posicionará haciendo coincidir su origen local con el **Sistema de coordenadas mundo de RobotStudio (RS-WCS)**.

4.3.3 Creación del transportador

El siguiente paso antes de crear los controladores virtuales será la creación del transportador de la estación. La labor de este transportador será la desplazar la carrocería del vehículo a lo largo de la estación de pintado. Además generará carrocerías constantemente cada cierta distancia, hasta que finalice la simulación.

Para crear el transportador se debe seguir el procedimiento descrito en el apartado [3.1.13 Creación de transportadores y conexión con el controlador](#). En el panel **Crear Transportador** se selecciona como geometría del transportador el componente **Transportador**, se le da una longitud de 16000 mm y se selecciona **Repetidor**, como se muestra en la Figura 4.51.

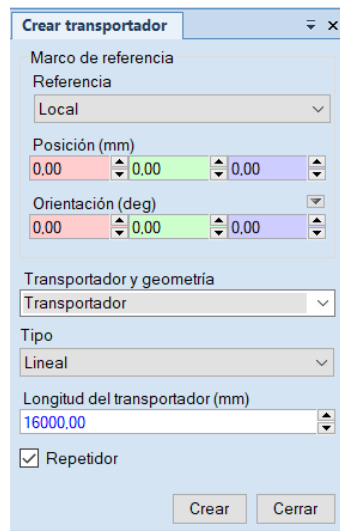


Figura 4.51 - Creación del transportador

De esta forma se crea el transportador, apareciendo ahora en el navegador Diseño. Posteriormente, una vez se hayan creado los controladores, se crearán las conexiones entre el transportador y los controladores.

4.3.4 Creación de los controladores virtuales

Una vez se tienen en la estación correctamente posicionados todos los componentes fundamentales que intervienen en el proceso de pintado, se pasa a crear los controladores virtuales que controlarán los movimientos de los robots y los tracks.

Esto se hace mediante la herramienta **Controlador virtual – Desde diseño**, como se expone en el apartado [3.1.12 Creación de un controlador virtual](#). El proceso de creación del controlador virtual se deberá repetir seis veces, una para conjunto de robot y posicionador. Las opciones de configuración extra, expuesta a continuación, serán para todos los robots de la estación las mismas.

En la primera ventana del cuadro de dialogo que aparece para crear los controladores de la estación (Figura 4.52), estos recibirán los siguientes identificadores:

- Cont_Inf_Izq. Para controlar el conjunto formado por el robot inferior izquierdo IRB 5500-23 y su track IRB 5500 paint rail extended.
- Cont_Inf_Der. Para controlar el conjunto formado por el robot inferior derecho IRB 5500-23 y su track IRB 5500 paint rail extended.
- Cont_Sup_Izq. Para controlar el conjunto formado por el robot superior izquierdo IRB 5500-25 y su track IRB 5500 elevated rail.
- Cont_Sup_Der. Para controlar el conjunto formado por el robot superior derecho IRB 5500-25 y su track IRB 5500 elevated rail.
- Cont_Puertas_Izq. Para controlar el conjunto formado por el robot izquierdo IRB 5350 y su track IRB 5350 rail.

- Cont_Puertas_Der. Para controlar el conjunto formado por el robot derecho IRB 5350 y su track IRB 5350 rail.

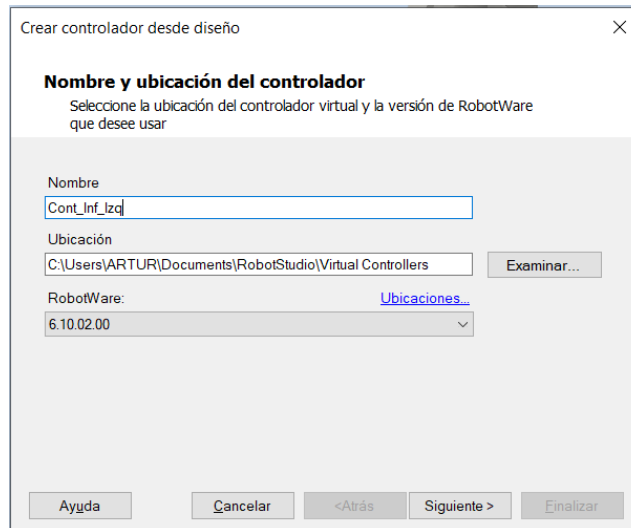


Figura 4.52 - Creación de los controladores - Primera ventana

La ubicación donde se guardarán los controladores se deja la que ofrece por defecto el programa y se selecciona la versión de RobotWare 6.10.02.00.

En la siguiente ventana (Figura 4.53) se seleccionan el robot y el track a añadir al controlador. La razón por la que no se añaden más robots por controlador es que RobotStudio solo permite un robot de pintura por controlador.

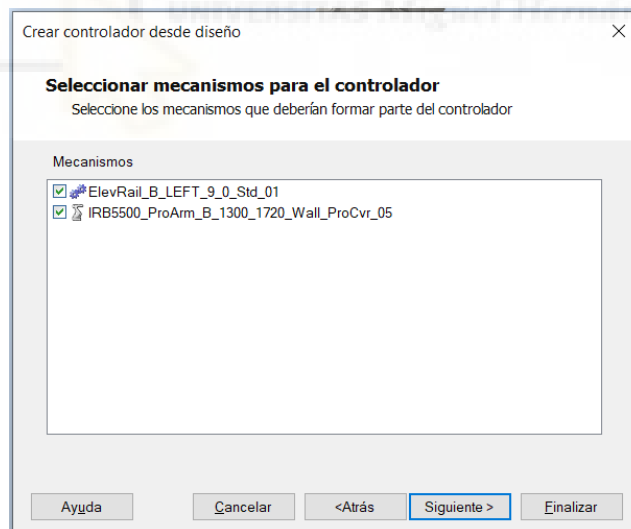


Figura 4.53 - Creación de los controladores - Segunda ventana

En la siguiente ventana (4.54) se deja una única tarea de controlador, ya que solo existe en cada controlador un robot y un track.

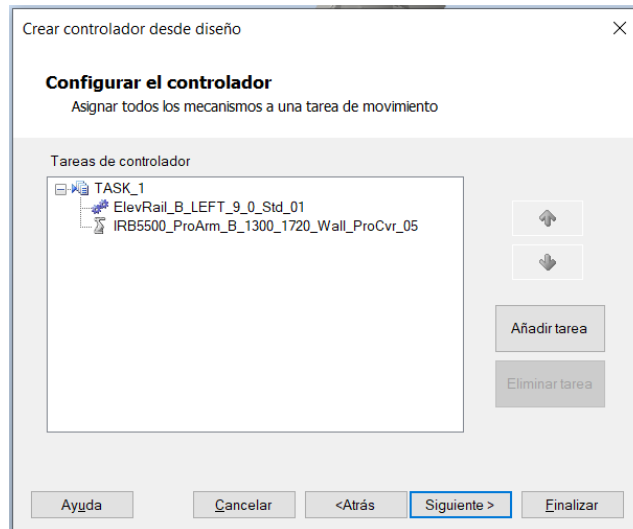


Figura 4.54 - Creación de los controladores - Tercera ventana

Por último, en la última ventana se deberán seleccionar las opciones de configuración del controlador. Además de las que vienen por defecto debido al tipo de robot, es decir, debido a que todos los robots son robots de pintura, se deberán seleccionar las siguientes opciones extra:

- **709-1 DeviceNet Master/Slave.** Esta opción se selecciona automáticamente al seleccionar la opción **606-1 Conveyor Tracking.**
- **888-2 PROFINET Controller/Device.** Esta opción añade a la estación una tarjeta de interfaz virtual ProfiNet. PROFINET es un estándar abierto para Ethernet industrial, empleado en la automatización de procesos. Inicialmente se consideró como opción para programar en ella las señales de E/S pero luego se descartó, por lo que esta opción no tiene mucha importancia en el proyecto.
- **606-1 Conveyor Tracking.** Al seleccionar esta opción el controlador de robot podrá coordinar los movimientos del robot con los del transportador. El robot será capaz ejecutar las trayectorias mientras estas son desplazadas por el transportador.
- **1552-1 Tracking Unit Interface.** Esta opción se selecciona automáticamente al seleccionar la opción **606-1 Conveyor Tracking.**
- **613-1 Collision Detection.** Esta opción se añadió para usar la función de seguridad Evitación de colisión, expuesta en el apartado [3.1.23 Detección de colisiones.](#) Posteriormente se descubrió que esta opción no funciona con los robots de pintura.
- **Paint Options – Conveyor Tracking on PIB.** Esta opción se selecciona automáticamente al seleccionar la opción **606-1 Conveyor Tracking.**

Como ya se ha mencionado, además de estas opciones extra, según el tipo y modelo de robot se tendrán unas opciones por defecto u otras, que no deben cambiarse.

Una vez se han especificado todas las opciones de configuración se pulsa Finalizar para crear el controlador.

4.3.5 Creación de las conexiones del transportador con los controladores

Una vez se tienen los controladores de todos los robots de la estación se pueden crear las conexiones del transportador con ellos. De esta forma se crea una conexión entre el robot y el transportador y un objeto de trabajo asociado al transportador, denominado por defecto wobj_cnv1. Los puntos objetivo programados en este objeto de trabajo serán desplazados por el transportador y el robot será capaz de seguirlo y realizar la trayectoria que componente mientras están en movimiento.

Es imprescindible que exista una conexión entre el transportador y cada uno de los robots para que estos sean capaces de reconocer cuando existe un objeto en el transportador, cuando comenzar a ejecutar las trayectorias sobre la carrocería y ser capaces de seguir el movimiento del mismo. Para crear esta conexión se seguirán los pasos descritos en el apartado [3.1.13 Creación de transportadores y conexión con el controlador](#).

El proceso de creación de la conexión se deberá hacer para cada uno de los robots. Los parámetros expuesto en la siguiente Figura 4.55 son comunes para todos los robots, salvo lógicamente el apartado **Unidad mecánica** en el que se debe especificar el controlador del robot en el que se crea la conexión.

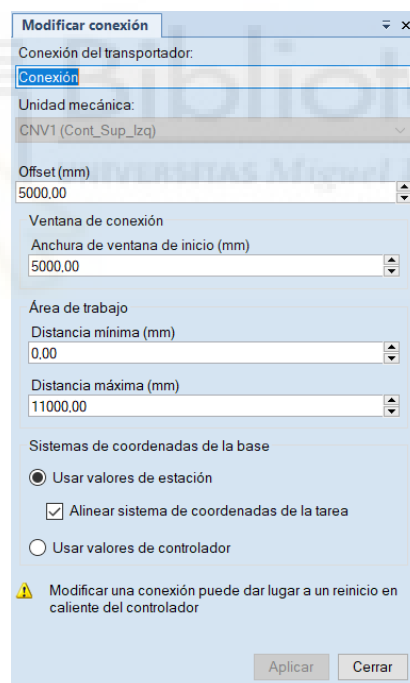


Figura 4.55 - Creación de la conexión transportador - controlador

Se dejará por defecto el nombre **Conexión**, ya que esto no tiene una mayor relevancia. En el apartado **Unidad mecánica** se especifica el identificador del eje externo transportador virtual que se crea al seleccionar la opción de configuración **606-1 Conveyor Tracking**, al crear el transportador. Esta unidad mecánica se denomina por defecto **CNV1 (Identificador del controlador)**. De esta forma se conecta ese eje externo virtual del robot con el controlador del robot indicado y este puede seguirlo.

La distancia de **Offset** se fija en 5000 mm, es decir, la distancia a la que se crea la conexión y se sitúa la base de coordenadas de la base de la unidad mecánica del transportador, respecto de la base de coordenadas de referencia del transportador.

La **Anchura de la ventana de inicio** se fija en 5000 mm, puesto que esa es la longitud total del conjunto carrocería más bancada.

La **Distancia mínima** se fija en 0, ya que el transportador no debe moverse hacia atrás en ningún momento.

La **Distancia máxima** se fija en 11000 mm, ya que más allá de esa distancia los robots no tienen alcance para poder continuar trabajando sobre la carrocería.

En el apartado **Sistema de coordenadas de la base** se indica la opción **Usar valores de estación**, con la opción añadida de **Alinear sistema de coordenadas de la tarea**.

De esta forma se obtendrá una conexión como la mostrada en la Figura 4.56 para cada uno de los robots de la estación.

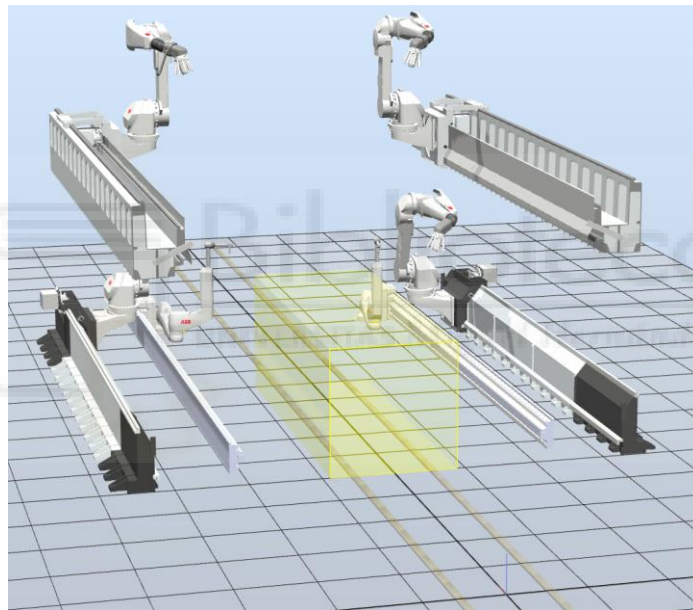


Figura 4.56 - Conexión creada para cada robot

A continuación se procede a especificar algunos elementos de la configuración del transportador y a añadir la carrocería completa al mismo.

En el panel Movimiento del transportador, al que se accede seleccionando el transportador con el botón derecho del ratón, se especifica una velocidad de avance del transportador de 40 mm/s y una aceleración de 0 mm/s², como se muestra en la siguiente Figura 4.57.

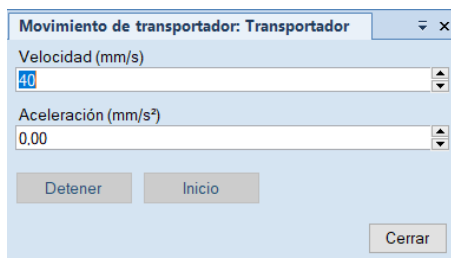


Figura 4.57 - Parámetros de movimiento del transportador

El valor para la velocidad se fija en 40 mm/s a partir de una serie de pruebas que se han realizado en otra estación secundaria.

En esta estación secundaria se programa el pintado completo de uno de los lados de la carrocería estando esta fija. Solo se programa uno de los lados ya que la carrocería es simétrica, y aunque las trayectorias no lo son totalmente, en su mayoría si son iguales. De esta forma se puede comprobar que las trayectorias cubren por completo el área de pintado que se requiere y se obtiene un tiempo aproximado de pintado de la carrocería.

Para obtener la velocidad del transportador se divide la longitud de los tracks, ya que los robots no pueden desplazarse más allá de esa longitud para seguir con el pintado, entre el tiempo aproximado que se tardó en realizar el pintado en parado. De esta forma se obtiene un valor cercano a 40 mm/s, que posteriormente se comprueba que efectivamente es el valor idóneo.

Seguidamente, se añade el conjunto formado por la carrocería y la bancada, denominado **Carroceria_Completa**, como objeto del transportador, es decir, como objeto que moverá el transportador y que creará de forma automática continuamente.

En el panel **Objeto del transportador** se especifican los parámetros mostrados en la siguiente Figura 4.58. Como pieza el componente **Carroceria_Completa**. Como **Separación** se fija 7900 mm, siendo esta la distancia entre las sucesivas carrocerías creadas en el transportador.

Al haberse posicionado el componente **Carroceria_Completa** centrado en la estación y se quiere que aparezca en el transportador desde la posición (0,0,0), en los apartados **Compensar posición** y **Compensar orientación** se deja todo a 0.

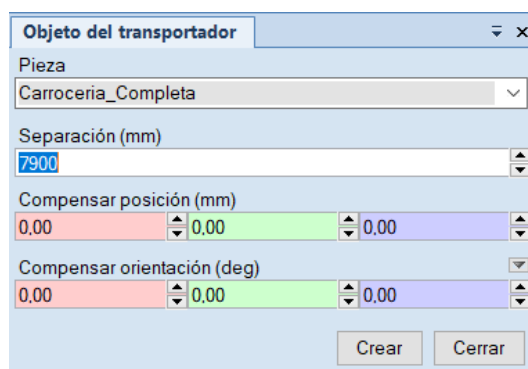


Figura 4.58 - Parámetros del objeto del transportador

4.3.6 Creación de la plantilla de instrucciones y objetos de trabajo

Antes de comenzar a crear los puntos objetivo de los que estarán compuestas las trayectorias se debe crear la plantilla de instrucciones de la estación y los objetos de trabajo que se requieran para programar algunos de los puntos objetivo en los mismos.

La plantilla de instrucciones de la estación se crea en uno de los controladores según lo expuesto en el apartado [3.1.17 Plantillas de instrucciones](#), añadiendo todas las instrucciones necesarias, obteniéndose la plantilla con las instrucciones mostrada en la siguiente Figura 4.59.

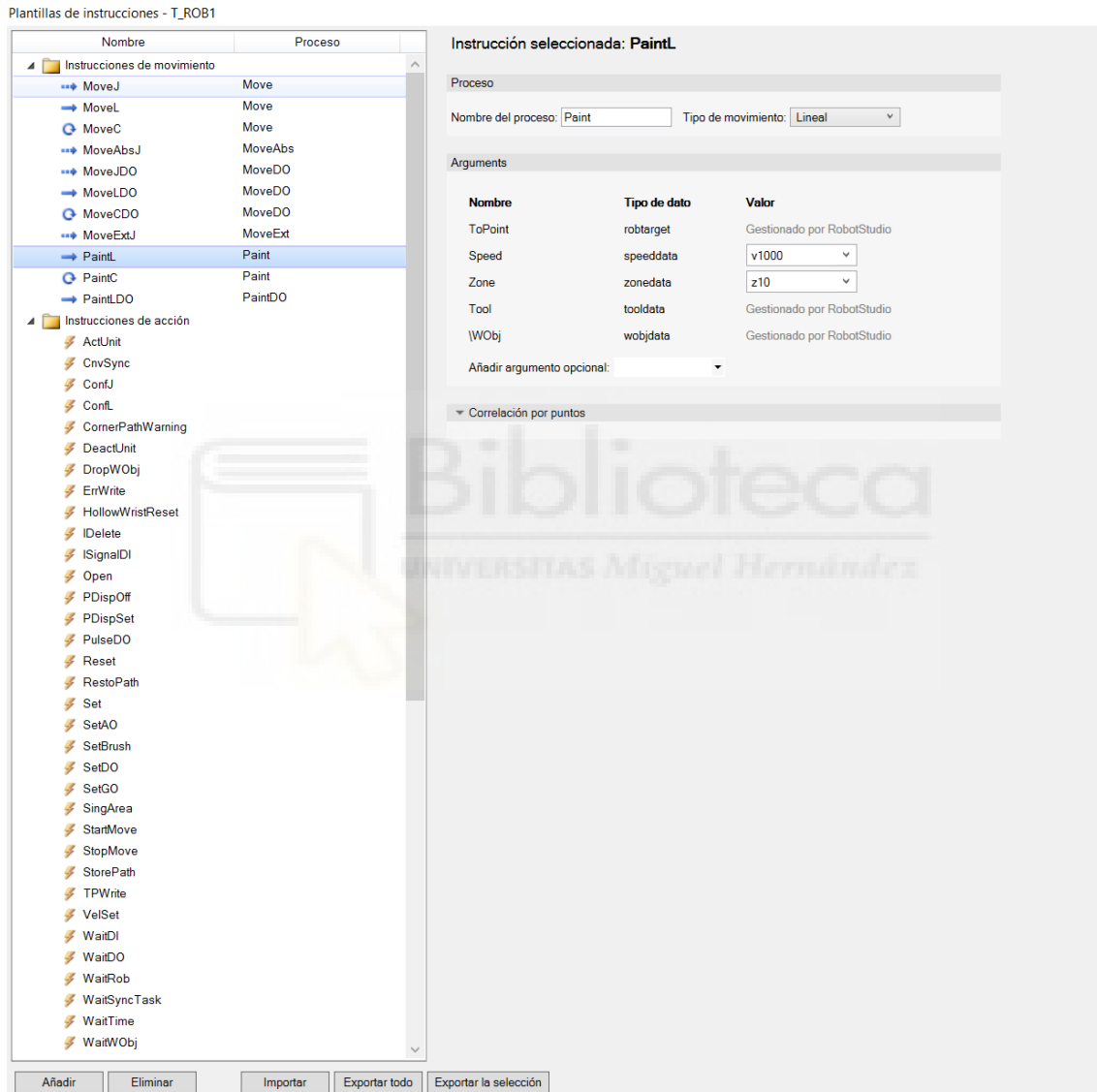


Figura 4.59 - Plantilla de instrucciones final de la estación

Una vez se tiene la plantilla creada en uno de los controladores, se exporta la plantilla completa mediante la opción **Exportar todo** y se importa en el resto de controladores mediante la opción **Importar**.

En cada uno de los controladores de robot de la estación se tiene por defecto el objeto de trabajo predeterminado **wobj0** y el objeto de trabajo asociado al transportador **wobj_cnv1**, situados ambos en el mismo punto que la base de coordenadas de la base de la unidad mecánica del transportador.

Además de estos dos, para cada controlador, se creará un tercer objeto de trabajo para referenciar en él puntos objetivo de posiciones intermedias, posiciones de parada, etc., en definitiva, para referenciar en él todos los puntos objetivo estáticos, ya que la gran mayoría de objetivos se referenciarán al objeto de trabajo **wobj_cnv1**, de modo que sean desplazados por el transportador y el robot pueda ejecutar la aplicación de la capa base en seguimiento.

Para crear un nuevo objeto de trabajo se deben seguir los pasos expuestos en el apartado [3.1.10 Descripción de los sistemas de coordenadas](#).

Este nuevo objeto de trabajo se denominará **WO_In_Int**, y sus parámetros se especificarán según se muestra en la Figura 4.60.

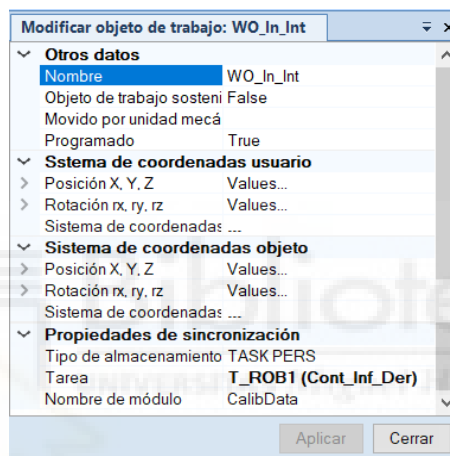


Figura 4.60 - Parámetros del objeto de trabajo WO_In_Int

Este objeto de trabajo estará posicionado en la misma posición que la base de coordenadas de la base del robot y será estático.

4.3.7 Creación de puntos objetivo y programación de trayectorias, procedimientos y rutinas

Una vez se han realizado correctamente todos los procesos expuestos en los apartados anteriores, se comienza una de las partes del proyecto de mayor duración, más iterativas y de mayor esfuerzo, y por lo tanto, una de las más importantes, la creación de los objetivos y programación de las trayectorias para realizar la aplicación de la capa base sobre la carrocería en movimiento.

Los contenidos expuestos en este apartado y los de los dos siguientes apartados [4.3.8 Desarrollo de la lógica de estación](#) y [4.3.9 Creación de señales de E/S](#), se desarrollan paralelamente durante el proceso de creación de la estación, ya que están estrechamente relacionados.

En este apartado se describirá tanto la parte de programación gráfica de objetivos y trayectorias como la programación en el editor de lenguaje RAPID de RobotStudio de todas las trayectorias de todos los robots de la estación.

4.3.7.1 Objetivos

La exposición individual de la posición, orientación y configuración de cada uno de los puntos objetivo creados en la estación es totalmente inviable, puesto que en la estación del presente proyecto se han creado 1863 puntos objetivo. Se puede ver por la cantidad de puntos objetivo que este es un claro ejemplo de programación fuera de línea de una estación robótica, ya que programar todos estos objetivos manualmente sería inconcebible. En las siguientes Figuras 4.61 y 4.62 se muestra la “nube” que forman todos los objetivos creados en la estación, en la cual se puede apreciar la forma de la carrocería.

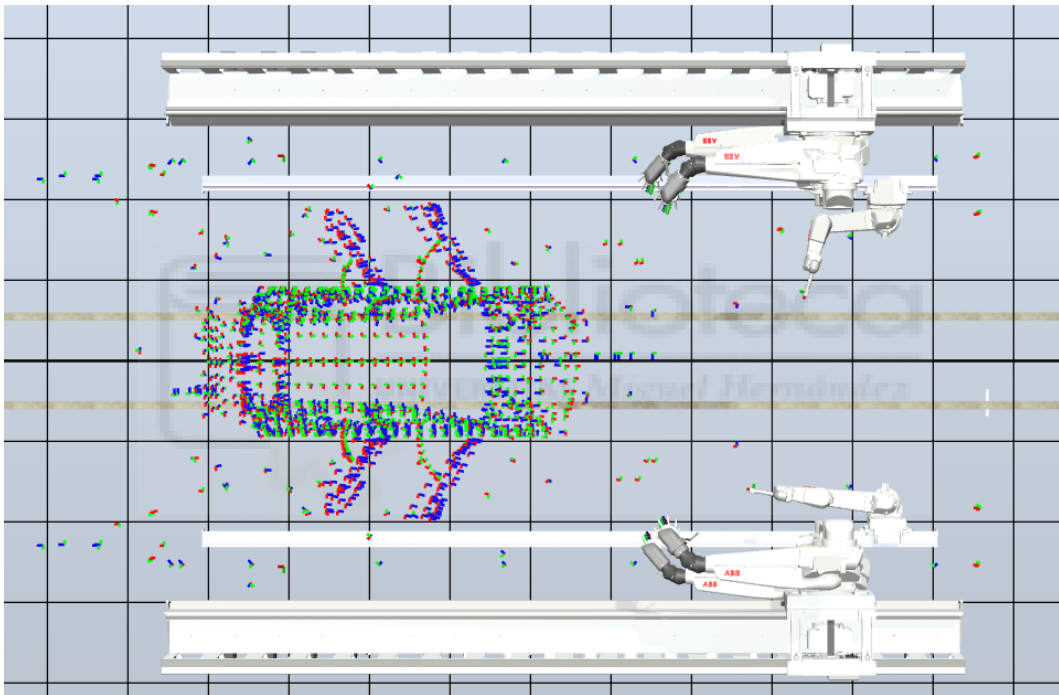


Figura 4.61 - Vista superior de la nube de puntos objetivo

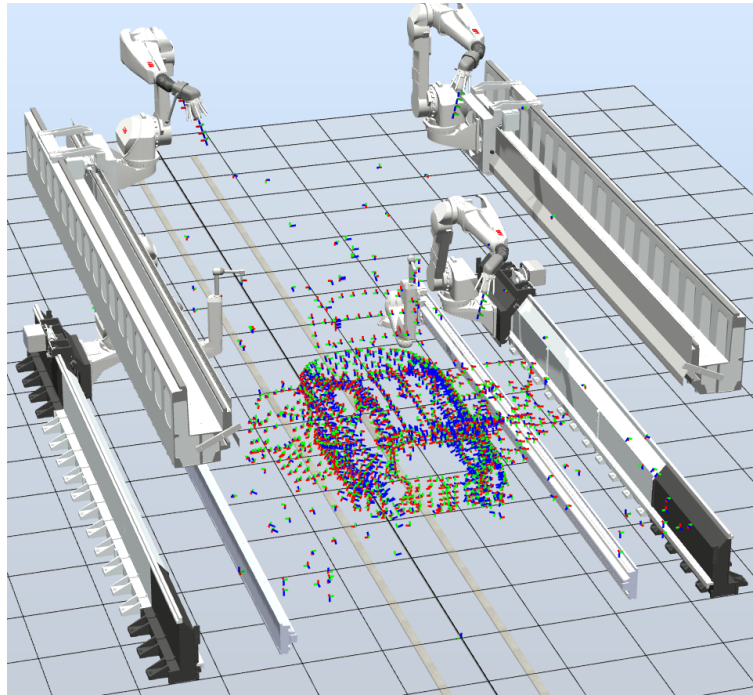


Figura 4.62 - Vista trasera en perspectiva de la nube de puntos objetivo

En cada uno de los controladores de robot se tiene la siguiente cantidad de objetivos:

- **Cont_Inf_Izq:** 703 objetivos.
- **Cont_Inf_Der:** 697 objetivos.
- **Cont_Sup_Izq:** 183 objetivos.
- **Cont_Sup_Der:** 188 objetivos.
- **Cont_Puertas_Izq:** 45 objetivos.
- **Cont_Puertas_Der:** 47 objetivos.

Cada uno de estos puntos o bien se han programado de forma individual o se han creado al programar una trayectoria automática, especificando la posición de cada uno, mediante las herramientas expuestas en el apartado [3.1.15 Creación de objetivos y trayectorias](#).

Después de esto, mediante las herramientas expuestas en el apartado [3.1.16 Orientaciones y configuraciones de los objetivos](#), siendo la herramienta más importante **Ver herramienta en posición**, se ha tenido que comprobar uno por uno su orientación, de modo que la herramienta del robot quede correctamente orientada en cada uno de ellos.

Una vez orientados correctamente se debe comprobar que el robot en cuestión es capaz de alcanzar cada uno de los objetivos con la orientación programada, como queda posicionado el robot y si colisionaría con algún otro componente en esa posición específica, principalmente mediante la herramienta **Ver robot en posición**. En el caso de que exista algún problema se deberá repositonar o reorientar el objetivo que lo causa. Si no se puede solucionar de ninguna forma se deberá replantear la trayectoria que los emplea y volver a crearlos.

El mayor número de problemas se encontraron al crear los objetivos que se emplean para las trayectorias de pintado interior, como debajo del capo y del maletero y dentro de las puertas,

puesto que el robot, e incluso el atomizador, son elementos de un tamaño considerable y el espacio para maniobrar en esas zonas es reducido.

Para finalizar se debe comprobar que la configuración en cada uno de los objetivos es la adecuada para poder ejecutar la trayectoria y para que el paso de una trayectoria a la siguiente también sea posible.

Para la creación de todos los puntos objetivo que dependen de la posición de la carrocería, esta se colocará en el transportador, mediante la opción **Colocar objeto en el transportador**, y se posicionará mediante la opción **Jog**, en el punto inicial de la conexión del transportador, es decir en el punto X = 5000, respecto del sistema de coordenadas mundo de RobotStudio, o la posición cero del transportador (Figura 4.63). Cuando se ejecute la simulación, al situarse completamente la carrocería dentro de la **Ventana de inicio** del transportador, es decir, al pasar por completo el inicio de la venta se iniciará el movimiento de los robots y por tanto la ejecución de las trayectorias.

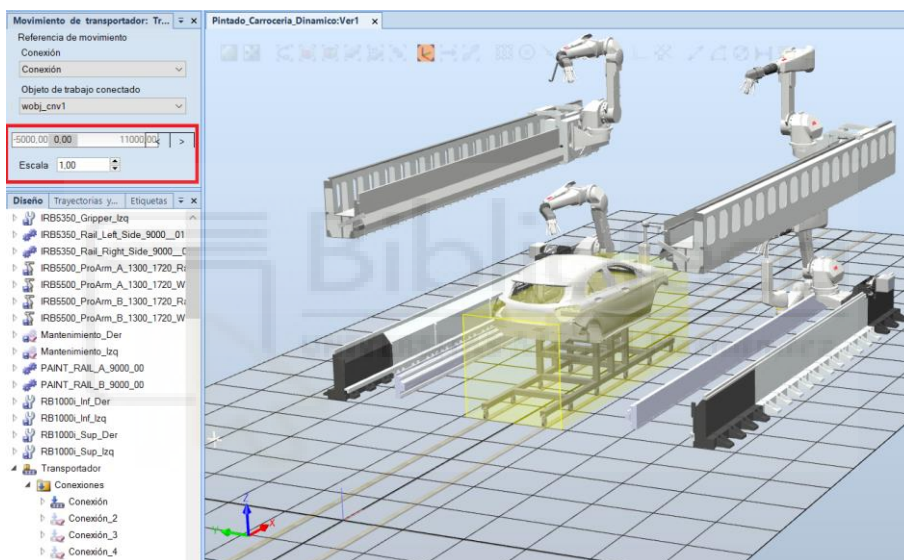


Figura 4.63 - Carrocería situada en el inicio de la Ventana de inicio del transportador

Es muy importante que estando la carrocería en esa posición, el área de trabajo de cada uno de los robots permita alcanzar todos los objetivos programados para el mismo, ya que de lo contrario se producirán errores al ejecutar el programa de robot. Como puede observarse en la Figura 4.61 todos los objetivos programados sobre la trayectoria están creados con la carrocería en esa posición.

Un aspecto muy importante al crear una serie de objetivos es que el track, sobre el que está posicionado el robot, debe estar posicionado, en el momento de programarlos, en una posición adecuada para que el robot tenga alcance a todos ellos. Es decir, previa a la creación de los objetivos se debe posicionar el track en una posición desde la que el robot sea capaz de ejecutar la trayectoria que estos formarán.

Existen dos tipo principales de puntos objetivo en este proyecto, los que se programan sobre la carrocería y formarán las trayectorias para la aplicación de la capa base, y que por lo tanto serán

desplazados por el transportador, y los puntos objetivo de posiciones intermedias, posiciones de espera, posiciones de limpieza y mantenimiento, etc., que serán estáticos, en ningún momento varían su ubicación. Muchos de estos puntos objetivo serán objetivos de ejes.

Los puntos objetivo estáticos se crearán referenciados al objeto de trabajo **WO_In_Int**. Estos objetivos se crearán de forma individual, es decir, uno por uno, mediante las herramientas **Posición** y **Programar posición**, expuestas en el apartado [3.1.15 Creación de objetivos y trayectorias](#).

Los puntos que se emplean para formar las trayectorias para la aplicación de la capa base, se crearán referenciados al objeto de trabajo **wobj_cnv1**, el objeto de trabajo del transportador. De esta forma al entrar el conjunto de la carrocería en la **Ventana de inicio** del transportador, todos estos objetivos, y por lo tanto las trayectorias, se desplazarán conjuntamente con la carrocería y el robot podrá realizar la aplicación de la capa base en seguimiento.

Estos objetivos se crean mediante dos herramientas diferentes. Por un lado, los objetivos que se emplean para las trayectorias que realizan la abertura y cierre del capó, maletero y las puertas, y los objetivos empleados como puntos intermedios entre algunas trayectorias, en definitiva, todos los objetivos que tienen que desplazarse con la carrocería pero no son objetivos de pintado, se programan mediante las herramientas expuestas en el apartado [3.1.15 Creación de objetivos y trayectorias](#).

Por otro lado, los objetivos de pintado, es decir, todos los objetivos que se emplean en trayectorias que realizan la aplicación de la capa base se programan mediante la herramienta **Inset Paint Stroke**, proporcionada por el **Painting PowerPac**, expuesto en el apartado [3.1.24 Complementos y PowerPacs](#).

Esta herramienta se emplea para crear de forma manual, dentro de una trayectoria ya existente, instrucciones de movimiento lineal de pintado (**PaintL**), y por lo tanto objetivos, e instrucciones **SetBrush**, empleadas para determinar el punto de accionamiento del atomizador, creando así una trayectoria de pintado o una de las partes de una trayectoria de pintado.

Para acceder a esta herramienta se selecciona la pestaña **Painting**, de forma que se accede a todas las opciones del **Painting PowerPac**. En el navegador **Programming**, se selecciona el subnavegador **Intructions**. Dentro de este se selecciona el controlador en el que se encuentra la trayectoria y la trayectoria en la que quiere programar las instrucciones de pintado. Se selecciona la trayectoria mediante el botón derecho del ratón y se elige la herramienta **Insert Paint Stroke**, como se muestra en la Figura 4.64.

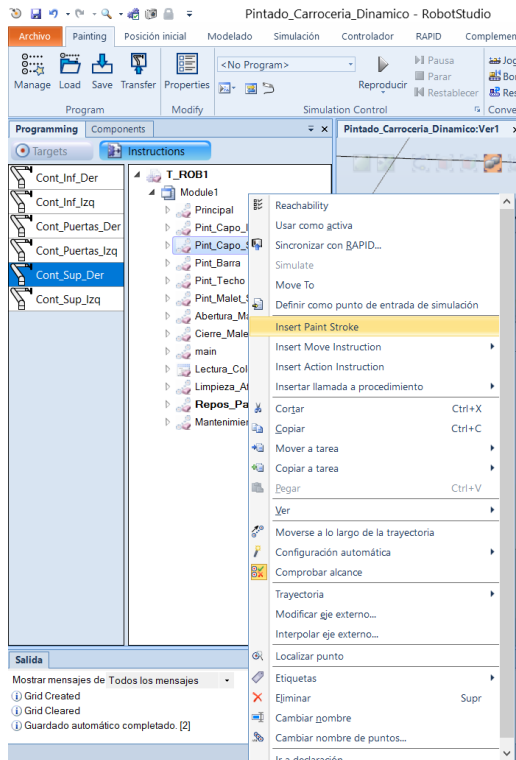


Figura 4.64 - Herramienta Insert Paint Stroke

Al seleccionar la herramienta se abre el panel **Manual Path**, mostrado en la siguiente Figura 4.65.

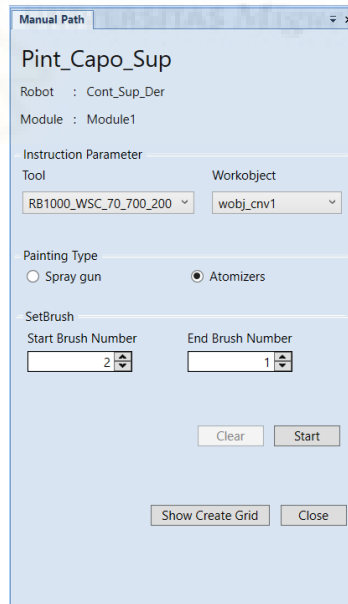


Figura 4.65 - Panel Manual Path

En este panel se muestra el nombre de la trayectoria en el que se insertarán las instrucciones de pintado, el nombre del controlador y el nombre del módulo de RAPID donde se programarán. A continuación se debe indicar el TCP de la herramienta que se empleará, ya que las herramientas de pintado disponen de más de un TCP para variar la distancia de pintado, y el objeto de trabajo al que estarán referidos los objetivos que se crean al crear las instrucciones.

El atomizador RB1000i_EXT_SAD_L dispone de 4 TCP. Uno en el punto final de la herramienta, otro a 100 mm de distancia de la misma, otro a 200 mm de distancia y el último a 300 mm de distancia de la herramienta, como se muestra en la Figura 4.66.

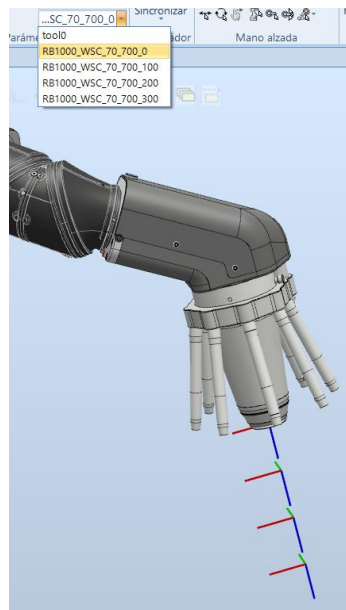


Figura 4.66 - TCPs del atomizador RB1000i_EXT_SAD_L

Estos TCP hacen más sencilla la programación de las trayectorias de pintado con diferentes distancias de pintado. La distancia de pintado es la distancia que existe entre la herramienta de pintado y la pieza de trabajo a pintar.

En el presente proyecto la distancia de pintado se ha establecido en 200 mm, por lo que en la mayoría de los objetivos de pintado la herramienta y TCP empleados será **RB1000_WSC_70_700_200**.

Tras esto se debe especificar el tipo de pintado a realizar, **Spray gun** o **Atomizers**. En este proyecto se seleccionará la opción **Atomizers** para todas las trayectorias de pintado, ya que se emplea un atomizador. Después se debe indicar el número de “cepillo”, es decir, el número de color, con el que se inicia la trayectoria de pintado y con el que se finaliza. De esta forma ya se tendrían especificados todos los parámetros del **Paint Stroke**.

Para facilitar el posicionamiento de los objetivos, de los que estará formada la trayectoria, esta herramienta ofrece la posibilidad de crear una cuadrícula, de dimensiones especificadas, sobre el componente a pintar. Esto se hace seleccionando la opción **Show Create Grid**, de forma que el panel **Manual Path** cambia al mostrado en la Figura 4.67.

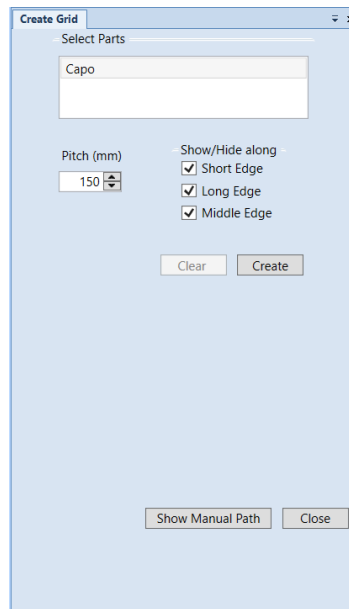


Figura 4.67 - Creación de una cuadrícula sobre el elemento a pintar

En este panel se especifica el componente de la carrocería sobre el que se quiere crear la cuadrícula, el tamaño de cada uno de los cuadrados de la cuadrícula, **Pitch**, y los ejes a lo largo de los cuales se crea. En este proyecto se especifica un **Pitch** de 150 mm y que se cree a lo largo de todos los ejes, para todos los componentes en los que se ha necesitado crear una cuadrícula.

Una vez se han especificado estos parámetros se pulsa **Create**, obteniéndose un resultado similar al mostrado en la Figura 4.68 para el caso del capó.

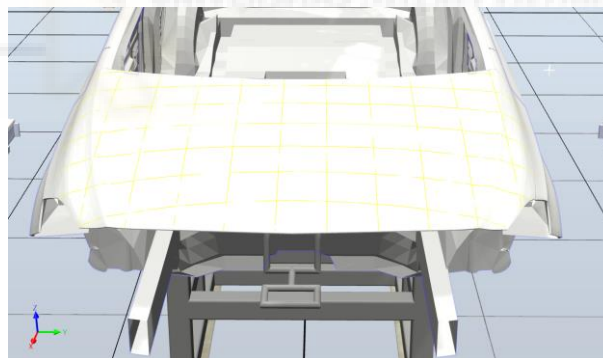


Figura 4.68 - Cuadrícula creada para el componente capó

Esta cuadrícula permite seleccionar objetivos sobre el componente manteniendo una distancia fija de 150 mm entre ellos.

A continuación se pulsa sobre la opción **Show Manual Path**, para volver al anterior panel, y se pulsa **Start**. En este momento se comienzan a seleccionar de forma ordenada sobre el componente los objetivos que formarán la trayectoria. Al seleccionar un punto sobre el componente se crea en ese punto un objetivo, visualizándose mediante un pequeño sistema de coordenadas. Al seleccionar el siguiente punto se comienza a visualizar la trayectoria, visualizándose una línea que une los puntos. Cuando se han creado todos los objetivos de la trayectoria se pulsa **Finish**.

Esta herramienta creará automáticamente un punto objetivo de aproximación y otro de salida de la trayectoria. El punto de aproximación se crea a una cierta distancia del primer objetivo creado, siguiendo la dirección de la trayectoria. El punto de salida se crea a una cierta distancia del último objetivo creado, siguiendo la dirección de la trayectoria.

De forma automática, también creará una instrucción **SetBrush** situada en el punto medio entre el punto de aproximación y el primer objetivo y otra situada en el punto medio entre el último objetivo y el punto de salida, obteniéndose así una trayectoria de pintado completa. Como se ha mencionado al principio de la explicación de la herramienta, mediante la misma se crean los objetivos especificados y todas las instrucciones de movimiento para desplazarse entre los mismos.

En este proyecto se han establecido los siguientes parámetros de pintado para todas las trayectorias de aplicación de la capa base de todos los robots:

- **Distancia de pintado:** 200 mm.
- **Distancia entre trazos de pintado paralelos o Paso de pintado:** 150 mm.
Esta distancia, como ya se ha indicado en el parámetro **Pitch** al crear las cuadrículas, indica la distancia entre objetivos situados en tramos paralelos de la trayectoria de pintado.
- **Ancho del patrón de pintado:** 300 mm.
Al ser el ancho del patrón de pintado el doble que el paso de pintado, en todas las trayectorias se tendrá un solapamiento de pintado del 50%.
- **Largo del patrón de pintado:** 150 mm.

Con estos parámetros se obtendrá un patrón de pintado similar al de la Figura 4.69.

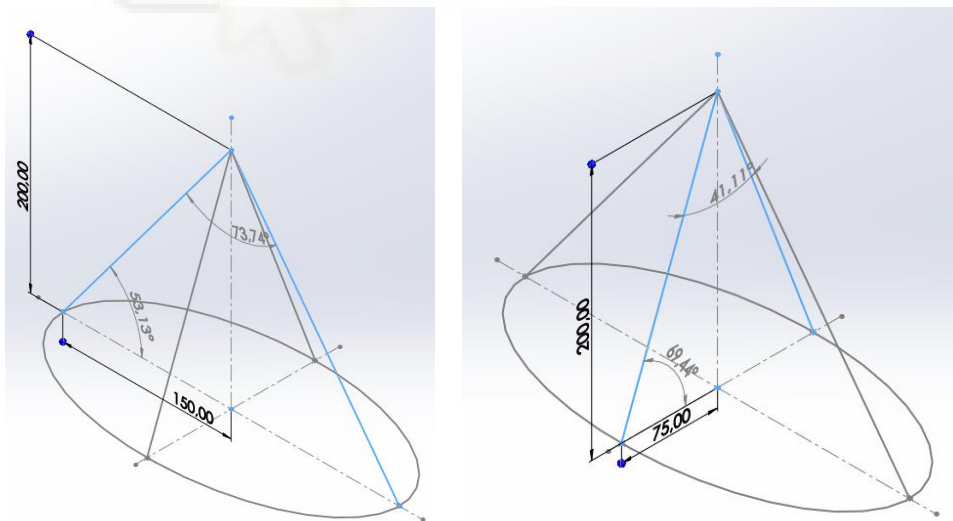


Figura 4.69 - Dimensiones del patrón de pintado

4.3.7.2 Trayectorias, procedimientos y rutinas

Una vez se han creado los objetivos que forman las trayectorias, estas se crean mediante las herramientas y métodos expuestos en el apartado [3.1.15 Creación de objetivos y trayectorias](#).

Una de las principales limitaciones en la programación de trayectorias con robots de pintura, además del espacio en algunas zonas interiores, es que giro relativo entre los ejes 4 y 6 de la muñeca no debe superar nunca ± 2 revoluciones, ya que si esto ocurre se produce un error. Para solucionar esto se han creado una serie de objetivos de ejes intermedios que permiten posicionar los ejes de los robots de forma adecuada para poder pasar a la siguiente trayectoria sin que se de este error.

Cuando se tienen las trayectorias creadas y configuradas, se sincronizan con RAPID para generar el código RAPID de las mismas, según lo expuesto en el apartado [3.1.18 Sincronización del controlador virtual con RAPID](#). De esta forma se obtiene el conjunto de objetivos y trayectorias mostrado en la Figura 4.39.

Las velocidades de pintado de todos los robots de aplicación IRB 5500 se han establecido en 1500 mm/s para las zonas interiores, debajo del capo y maletero y las zonas interiores de las puertas, y 800 mm/s para todo el exterior de la carrocería. En las zonas interiores la velocidad es mayor puesto que no se requiere un acabado de tanta calidad como en el exterior de la carrocería, aun así, con esa velocidad se conseguiría un acabado óptimo para esas zonas interiores.

A continuación se procede a realizar la explicación de cada una de las trayectorias de cada uno de los robots. Al sincronizarse con RAPID las trayectorias programadas gráficamente pasan a denominarse **Procedimientos**. Además de procedimientos de movimiento, se han programado otros procedimientos que realizan otras tareas para el correcto funcionamiento de la estación y rutinas **TRAP** que se emplean para las tareas de seguridad, limpieza y mantenimiento.

Se expondrán todos estos procedimientos y rutinas de forma que se puedan comprender claramente los programas de todos los robots, y por tanto se entienda como funciona la estación, y en definitiva una parte importante del proyecto. Esta explicación se hará exponiendo paralelamente el código RAPID y la parte gráfica, ya que se crean y trabajan conjuntamente.

En la siguiente Tabla 4.2 se muestran los procedimientos presentes en cada uno de los controladores.

Cont_Inf_Izq	Cont_Inf_Der	Cont_Sup_Izq	Cont_Sup_Der
Lectura_Colores	Lectura_Colores	Abertura_Capo	Abertura_Maletero
Limpieza_Atom_Color	Limpieza_Atom_Color	Cierre_Capo	Cierre_Maletero
main (Punto de entrada)	main (Punto de entrada)	Lectura_Colores	Lectura_Colores
Mantenimiento_Atom	Mantenimiento_Atom	Limpieza_Atom_Color	Limpieza_Atom_Color
Pint_Bajo_Capo	Pint_Bajo_Capo	main (Punto de entrada)	main (Punto de entrada)
Pint_Bajos_1	Pint_Bajos_1_1	Mantenimiento_Atom	Mantenimiento_Atom
Pint_Bajos_2	Pint_Bajos_2_2	Pint_Barra	Pint_Barra
Pint_Lateral	Pint_Lat	Pint_Capo_Inf	Pint_Capo_Inf
Pint_Malet_Inf	Pint_Maletero_Inf	Pint_Capo_Sup	Pint_Capo_Sup
Pint_PI_Del	Pint_PD_Del	Pint_Malet_Sup	Pint_Malet_Sup
Pint_PI_Tras	Pint_PD_Tras	Pint_Techo	Pint_Techo
Principal	Principal	Principal	Principal
Cont_Puertas_Izq	Cont_Puertas_Der		
Abertura_Cierre_PI_Del	Abertura_Cierre_PD_Del		
Abertura_Cierre_PI_Tras	Abertura_Cierre_PD_Tras		
main (Punto de entrada)	main (Punto de entrada)		
Mantenimiento_Gripper	Mantenimiento_Gripper		
Principal	Principal		

Tabla 4.2 - Procedimientos en cada uno de los controladores de robot

Además de estos procedimientos, se tienen una serie de rutinas TRAP, que se emplean para ejecutar las medidas de seguridad, limpieza y mantenimiento, que también incluyen trayectorias. Las rutinas TRAP presentes en cada uno de los controladores de robot se muestran en la siguiente Tabla 4.3.

Cont_Inf_Izq	Cont_Inf_Der	Cont_Sup_Izq
Limpieza_Atomizador	Limpieza_Atomizador	Paro_Puerta
Parada	Parada	Limpieza_Atomizador
Mantenimiento_Atomizador	Mantenimiento_Atomizador	Parada
		Mantenimiento_Atomizador
Cont_Sup_Der	Cont_Puertas_Izq	Cont_Puertas_Der
Limpieza_Atomizador	Parada	Parada
Parada		
Mantenimiento_Atomizador		

Tabla 4.3 - Rutinas TRAP en cada uno de los controladores de robot

Para realizar la programación en RAPID de todos estos procedimientos y rutinas se han empleado todas las herramientas y conocimientos expuestos en el apartado [3.2 Programación básica en RAPID](#).

En cada uno de los controladores de robot se crea un módulo de programa denominado **Module1**, donde se programarán todos los procedimientos y rutinas, como se muestra en la siguiente Figura 4.70.

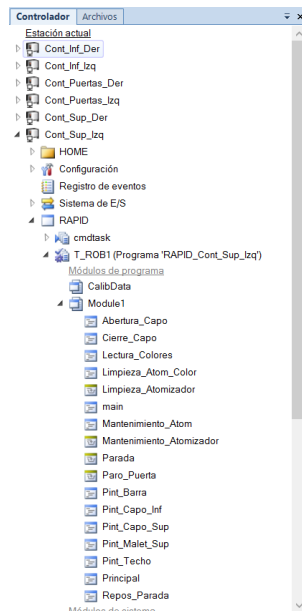


Figura 4.70 – Desplegable del Module1 del controlador Cont_Sup_Izq

Para facilitar la explicación de los procedimientos, estos se dividirán en tres grupos, procedimientos de pintado, procedimientos de ejecución y procedimientos de asistencia. Estos procedimientos se explicarán según el orden en el que se ejecutan en el programa.

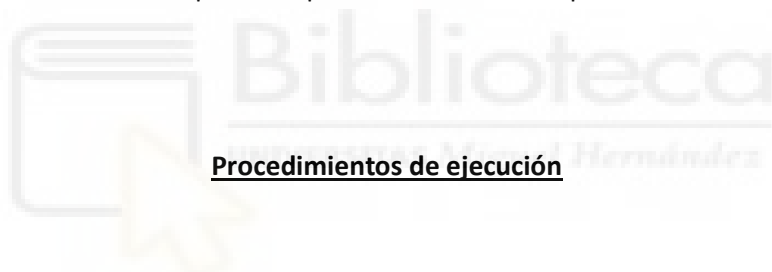
Controlador virtual Cont Inf Izq

El código completo del programa de robot de este controlador se encuentra en el apartado [8.1.1 Controlador virtual Cont Inf Izq](#). Debido a su tamaño no se volverá a mostrar en este apartado.

Antes de exponer los procedimientos y rutinas del programa, se exponen brevemente las variables globales definidas para el mismo.

- VAR num **Color**;
El número de esta variable se emplea como código del color de pintado.
- VAR intnum **Limp_Atom**;
Esta variable se emplea para identificar la interrupción asociada la rutina TRAP **Limpieza_Atomizador**.
- VAR intnum **Paro_Robot**;
Esta variable se emplea para identificar la interrupción asociada la rutina TRAP **Parada**.
- VAR intnum **Mant_Atom**;
Esta variable se emplea para identificar la interrupción asociada la rutina TRAP **Mantenimiento_Atomizador**.
- VAR num **Lectura_Color**;
El número de esta variable indica el número leído del archivo **date+ "_Lista_Colores.txt"**.

- VAR iodev **infile**;
Esta variable se emplea como referencia al archivo **date+"_Lista_Colores.txt"**.
- VAR num **Contador**;
Esta variable se emplea como un contador.
- VAR num **Secuencia_Colores{1000}**;
En esta variable se guarda la secuencia de colores presentes en el archivo **date+"_Lista_Colores.txt"**. Esta variable puede contener hasta un máximo de 1000 números de código de color.
- VAR string **date**;
En esta variable se guarda la fecha.
- VAR string **N_Proc**;
Esta variable indica el procedimiento que se está ejecutando en cada momento. Se empleará dentro de las rutinas TRAP para conocer como debe ser la trayectoria de salida según el procedimiento que se estuviera ejecutando.
- VAR string **Limpieza**;
Esta variable indica el tipo de limpieza del atomizador que debe realizarse.



main (Punto de entrada de la simulación)

Este es el procedimiento en el que se inicia la ejecución del programa de robot. Desde él se debe llamar a la ejecución del resto de procedimientos que forman el programa.

El procedimiento realiza las siguientes tareas de forma secuencial.

1. Se resetean todas las señales de E/S creadas para el proyecto y que intervienen en el mismo.
2. Se borran las conexiones a las identidades de interrupción asociadas a las rutinas TRAP, como medida de seguridad para que no existan posibles errores en cuanto a las conexiones con estas rutinas.
3. Se realizan las conexiones entre las identidades de interrupción y las rutinas TRAP.
4. Se especifica que señal digital de entrada debe cambiar su valor al indicado para que la ejecución normal del programa se detenga y se pase a ejecutar la rutina TRAP asociada a la identidad indicada.
5. Se cambia a 1 el valor de la señal digital de salida **Get_Contador**. Mediante este cambio de valor se comunica a los controladores **Cont_Puertas_Izq** y **Cont_Puertas_Der** el número de ciclos a realizar.
6. Se fija el valor de la variable **Color** igual al primer valor de variable **Secuencia_Colores**, es decir, se establece el color del primer ciclo de pintado. Los valores de la variable

- Secuencia_Colores** se obtienen mediante el procedimiento **Lectura_Colores**, expuesto a continuación.
7. Se cambia a 1 el valor de la señal digital de salida **Cambio_Color**. Mediante este cambio se hace que el componente inteligente **Aplicadores_Pintura** lea la variable **Color** actual y la pase a los aplicadores.
 8. Se activa la unidad mecánica **CNV1**, es decir, el transportador.
 9. En este punto se comienza un bucle FOR, para la ejecución repetitiva de todos los procedimientos de pintado y asistencia. Este bucle se ejecutará un número de veces igual al número de colores indicados en el archivo **date+_Lista_Colores.txt**. Es decir, se ejecuta hasta que se pinten tantos vehículos como colores se han especificado.
 10. Dentro de este bucle se ejecuta el procedimiento **Principal**, que contiene las llamadas a todos los procedimientos de pintado y asistencia. Con la ejecución completa de este procedimiento se ejecutan todas las trayectorias que realizan la aplicación completa de la capa base.
 11. Una vez ejecutado se resetea la señal digital de salida **Cambio_Color**.
 12. Se fija la variable **N_Proc:="main"**.
 13. A continuación se espera a que el movimiento del robot sea nulo.
 14. Aquí comienza una expresión condicional IF, empleada para determinar si existe un cambio de color, y por lo tanto se debe limpiar el atomizador y cambiar de color, si es el último de los ciclos de pintado, y por lo tanto se debe limpiar el atomizar para dejarlo limpio, o si no se debe cambiar de color para la siguiente carrocería.
 15. En el caso de que la siguiente carrocería se deba pintar de otro color, el valor de la variable **Limpieza** pasa a ser **"Limp1"**. Se ejecuta el procedimiento **Limpieza_Atom_Color**, que limpia el atomizador en función del valor de la variable **Limpieza**. Se cambia el valor de la variable **Color** al del siguiente número de la variable **Secuencia_Colores**. Se cambia a 1 el valor de la señal digital de salida **Cambio_Color**. Se ejecutan una secuencia de instrucciones de movimiento que simulan el purgado y cambio de color del atomizador. Por último se mueve el robot hasta su posición inicial.
 16. En el caso de que sea el último ciclo de pintado, el valor de la variable **Limpieza** pasa a ser **"Limp1"**. Se ejecuta el procedimiento **Limpieza_Atom_Color**, que limpia el atomizador en función del valor de la variable **Limpieza**. Por último se mueve el robot hasta su posición inicial.
 17. Si no se cambia de color para la siguiente carrocería, se cambia el valor de la variable **Color** al del siguiente número de la variable **Secuencia_Colores**. Se cambia a 1 el valor de la señal digital de salida **Cambio_Color**. Por último se mueve el robot hasta su posición inicial.
 18. Cuando se termina de ejecutar el bucle FOR, es decir, cuando se terminan de pintar todas las carrocerías especificadas, se resetea la señal digital de salida **Get_Contador** y se desactiva la unidad mecánica **CNV1**. De esta forma terminaría la ejecución del programa de robot.

Principal

Dentro de este procedimiento se realizan las llamadas secuenciales a todos los procedimientos que se emplean en este controlador para realizar la aplicación de la capa base sobre la carrocería. Entre las distintas llamadas a los procedimientos se ejecutan instrucciones de movimiento y de acción, sobre todo de espera y de activación/desactivación de señales, para coordinar todos los movimientos de los robots y que el paso de una trayectoria a otra se realice de forma correcta sin causar errores.

La secuencia de ejecución, explicada de forma abreviada, es la siguiente:

1. Se resetean algunas señales, se mueve el robot a la posición inicial y se espera que la carrocería entre en la **Ventana de inicio** del transportador, mediante la instrucción **WaitWObj**.
2. Cuando la carrocería llega a esa posición, se espera a que el controlador **Cont_Sup_Izq** le indique que debe continuar, a través de la señal digital de entrada **Init_Pint1**.
3. Se llama a la ejecución del procedimiento **Pint_Bajo_Capo**, que pinta la parte izquierda de la zona del chasis situada debajo del capó.
4. Se traslada el robot a una posición intermedia mientras se cambia el valor de la señal digital de salida **Abrir_P** a 1, lo cual le indica al controlador **Cont_Puertas_Izq** que debe continuar su movimiento.
5. Se espera a que el controlador **Cont_Puertas_Izq** le indique que debe continuar, a través de la señal digital de entrada **Init_Pint2**.
6. Se resetea la señal digital de salida **Abrir_P**.
7. Se ejecuta el procedimiento **Pint_PI_Del**, que pinta el interior de puerta izquierda delantera.
8. Se traslada el robot a una posición intermedia mientras se cambia el valor de la señal digital de salida **Abrir_P** a 1, lo cual le indica al controlador **Cont_Puertas_Izq** que debe continuar su movimiento.
9. Se espera a que el controlador **Cont_Puertas_Izq** le indique que debe continuar, a través de la señal digital de entrada **Init_Pint2**.
10. Se ejecuta el procedimiento **Pint_PI_Tras**, que pinta el interior de puerta izquierda trasera.
11. Se resetea la señal digital de salida **Abrir_P**.
12. Se traslada el robot a una posición intermedia y se espera a que el controlador **Cont_Sup_Der** le indique que debe continuar, a través de la señal digital de entrada **Init_Pint2**.
13. Se ejecuta el procedimiento **Pint_Malet_Inf**, que pinta la parte izquierda del interior del maletero.
14. Se ejecutan una serie de instrucciones de movimiento para reposicionar el robot y se resetea la señal digital de salida **Avance_Inf_P**, mediante la cual se le había indicado, dentro del procedimiento **Pint_Malet_Inf**, al controlador **Cont_Puertas_Izq** que debía continuar su ejecución.
15. Se ejecuta el procedimiento **Pint_Bajos_1**, que pinta la zona de los bajos de la rueda delantera izquierda.
16. Se mueve el robot a una posición intermedia y se ejecuta el procedimiento **Pint_Lateral**, que pinta todo el lateral izquierdo de la carrocería.
17. Se ejecuta el procedimiento **Pint_Bajos_2**, que pinta la zona de los bajos de la rueda trasera izquierda.

18. Se cambia a 1 el valor de la señal digital de salida **Finalizado**, que indica que se han finalizado los procesos de pintado de la carrocería para este controlador.
19. Se traslada el robot a una posición intermedia.
20. Se cambia a 1 el valor de la señal digital de salida **Volver**, lo cual le indica al controlador **Cont_Puertas_Izq** que debe continuar su movimiento.
21. Se deja de seguir a la carrocería del transportador, es decir, se suelta el objeto de trabajo **wobj_cnv1** mediante la instrucción **DropWObj**.

Lectura Colores

Este procedimiento se emplea para leer los números de código de color que se pasan diariamente en un fichero de texto a la carpeta **HOME** del controlador de cada robot de pintura.

En el fichero, que se nombra de la siguiente forma **FECHA_Lista_Colores.txt**, por ejemplo, **2021-01-05_Lista_Colores.txt**, se escriben los números de color separados por espacios, como por ejemplo, 2 5 5 8 6 6 13 13 10 11. Este procedimiento abre ese fichero, lee cada uno de los números de código de color y los va guardando en la variable **Secuencia_Colores** para así saber qué color emplear en cada ciclo de pintado. Los números de los códigos de color tendrán que ser mayores que 2 y menores que 16.

Esto se ha hecho de esta forma ya que se cree lógico que en cada periodo de trabajo se tengan códigos de color diferentes, es decir, se tenga una secuencia de pintado diferente, y es más simple y cómodo editar un archivo de texto o insertar uno nuevo cada día, que tener que editar el programa de robot y volver a cargarlo en el robot.

Este procedimiento será común en todos los robots de aplicación de la capa base, programado totalmente igual.

El procedimiento realiza las siguientes tareas de forma secuencial.

1. Se crea una variable local para el procedimiento, VAR string **Num_CE**;
2. Se lee la fecha del día actual y se guarda en la variable **date**.
3. Se "abre" para lectura el fichero con los códigos de color, **date+"_Lista_Colores.txt"**.
4. Se guarda en la variable **Lectura_Color** el primer código de color del archivo.
5. Se cambia el valor de la variable Contador a 1.
6. La variable **Num_CE** pasa a tomar el valor de la variable **Contador** pero en formato textual, es decir, se cambia el valor de la variable **Contador** de número a cadena de caracteres.
7. Se realiza una comprobación de que el primer número de color leído del archivo es correcto, es decir, es mayor que 2 y menor que 16. En caso contrario se mostraría un mensaje de error en el **FlexPendant** y en la **Ventana de salida** de RobotStudio y se detendría la ejecución del programa de robot.
8. Si es correcto, el primero de los números de la variable **Secuencia_Colores** pasa a tomar el valor de la variable **Lectura_Color**.
9. A continuación comienza un bucle **WHILE**, en el que mientras el número de código de color leído del archivo, es decir, la variable **Lectura_Color**, sea menor que **EOF_NUM**, número predefinido en RobotStudio con un valor de 9.998E+36, se continúan leyendo los números de código de color del archivo.
10. El primer paso dentro del bucle es sumar 1 a la variable **Contador**.

11. Tras esto la variable **Num_CE** pasa a tomar de nuevo el valor de la variable **Contador** pero en formato textual.
12. Se lee el siguiente número del archivo de códigos de color y se cambia el valor de la variable **Lectura_Color** a este nuevo número leído.
13. Se realiza la misma comprobación de seguridad que antes.
14. Si es correcto, el número en la posición **Contador** de la variable **Secuencia_Colores** pasa a tomar el valor de este código de color leído. A continuación se vuelve a comenzar el bucle.
15. Finalmente, como el último número de la variable **Secuencia_Colores** siempre tendrá el valor de **EOF_NUM**, se hace que este último número tenga el valor 0, dejando así esta variable únicamente formada por la secuencia de números de código de color.

Limpeza Atom Color

Este procedimiento se emplea para simular la limpieza del plato de campana del atomizador. Dentro de este procedimiento se tiene una expresión condicional **IF**, que ejecuta unas instrucciones u otras en función del valor de la variable **Limpeza**.

Si su valor es igual a "**Limp1**", simplemente se cambia el valor de la señal digital de salida **RUN_Limp** a 1 y se ejecutan las instrucciones de movimiento para realizar la limpieza.

Si su valor es igual a "**Limp2**", se hace lo mismo que en el caso anterior pero además se añaden una serie de instrucciones de movimiento de pintado para simular el proceso de purgado del atomizador.

Una vez ejecutado el **IF**, se resetea la señal **RUN_Limp**.

Este proceso será igual para todos los controladores de robots de aplicación de la capa base, variando únicamente las posiciones de los objetivos para realizar la trayectoria de limpieza, ya que los limpiadores de cada robot están posicionados en un lugar diferente.

Mantenimiento Atom

Este procedimiento se emplea para realizar el mantenimiento o la limpieza manual de los atomizadores desde el área de mantenimiento, lo que permite no tener que acceder al interior de la estación para llevar a cabo estas tareas.

El robot ejecutará una serie de instrucciones de movimiento para posicionarse delante de la trampilla de mantenimiento. En ese momento activa la señal digital de salida **Open_TMant**, es decir cambia su valor a 1, que está conectada al componente inteligente **PoseMover [Open]**, que al activarse mueve la trampilla de mantenimiento hasta dejar el hueco completamente abierto.

Una vez abierta, se fija la velocidad máxima del TCP del robot en 100 mm/s de forma que si, en caso de fallo, chocase con el operario no le causase un problema grave. El robot introduce la herramienta a través del hueco de mantenimiento a esa velocidad y espera hasta que la señal **Mantenimiento** vuelva a 0. Al mismo tiempo el valor de la señal analógica de salida **Señal_Luminosa** cambia para indicar si el robot está en movimiento o parado.

Cuando se termina el mantenimiento y se resetea la señal **Mantenimiento**, el robot saca la herramienta a través del hueco, se resetea la señal **Open_TMant** y se activa la señal **Close_TMant**, conectada al componente inteligente **Pose_Mover[HomePose]**, que cierra la trampilla.

Una vez tiene la herramienta fuera de la trampilla, la velocidad del robot vuelve a fijarse en la máxima velocidad que puede alcanzar el modelo de robot, para ya no estar limitado, y se ejecutan una serie de instrucciones de movimiento para alejarse.

Este proceso será igual para todos los controladores de robots de aplicación de la capa base, variando simplemente los objetivos de las instrucciones de acercamiento o alejamiento a la zona de mantenimiento.

Procedimientos de pintado

Pint Bajo Capó

Este procedimiento de pintado está compuesto en su totalidad por instrucciones de movimiento, tanto normales (**MoveL**) como de pintado (**PaintL**) y dos instrucciones **SetBrush**, que indican los puntos de accionamiento del atomizador, es decir, los puntos de inicio de pintado. Para la primera de estas instrucciones **SetBrush** su parámetro **BrushNumber** será igual a la variable **Color**, la cual indica el color a aplicar en cada ciclo de pintado. Para la segunda instrucción **SetBrush** este parámetro tendrá el valor 1, puesto que el valor 1 se emplea para detener el atomizador.

Este procedimiento, mostrado en la Figura 4.71, se emplea para realizar la aplicación de la capa base en la parte izquierda del chasis que se encuentra bajo el capó. El atomizador se aproxima a la zona de pintado, realiza la trayectoria de pintado y se aleja de la zona, para así facilitar la transición a la siguiente trayectoria.

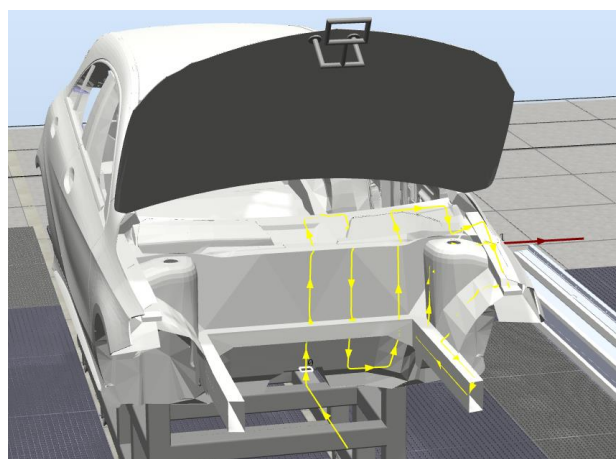


Figura 4.71 - Procedimiento Pint_Bajo_Capó - Cont_Inf_Izq

Pint PI Del

Este procedimiento de pintado está formado a su vez por 3 PaintStrokes o trayectorias de pintado, es decir, se tienen tres activaciones y desactivaciones del atomizador. Este procedimiento está compuesto en su totalidad por instrucciones de movimiento de pintado (**PaintL**) y seis instrucciones **SetBrush**, 3 de accionamiento con el parámetro **BrushNumber** igual a la variable **Color** y tres de desactivación con este parámetro igual a 1.

Este procedimiento, mostrado en la Figura 4.72, se emplea para realizar la aplicación de la capa base en la parte interior de la puerta delantera izquierda y en la zona interior del chasis que cierra esta puerta.

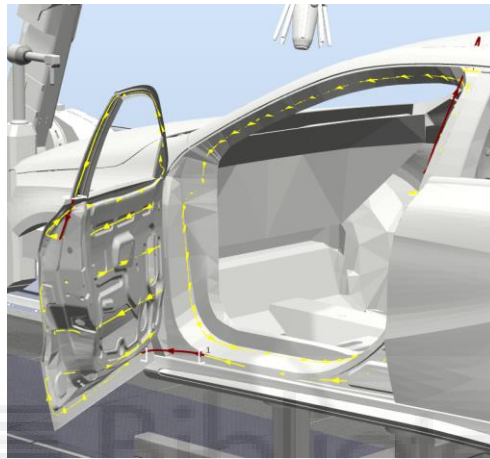


Figura 4.72 - Procedimiento Pint_PI_Del - Cont_Inf_Izq

Pint PI Tras

Este procedimiento de pintado está formado a su vez por 3 PaintStrokes o trayectorias de pintado, es decir, se tienen tres activaciones y desactivaciones del atomizador. Este procedimiento está compuesto en su totalidad por instrucciones de movimiento, tanto normales (**MoveL**) como de pintado (**PaintL**) y seis instrucciones **SetBrush**, 3 de accionamiento con el parámetro **BrushNumber** igual a la variable **Color** y tres de desactivación con este parámetro igual a 1.

Este procedimiento, mostrado en la Figura 4.73, se emplea para realizar la aplicación de la capa base en la parte interior de la puerta trasera izquierda y en la zona interior del chasis que cierra esta puerta.

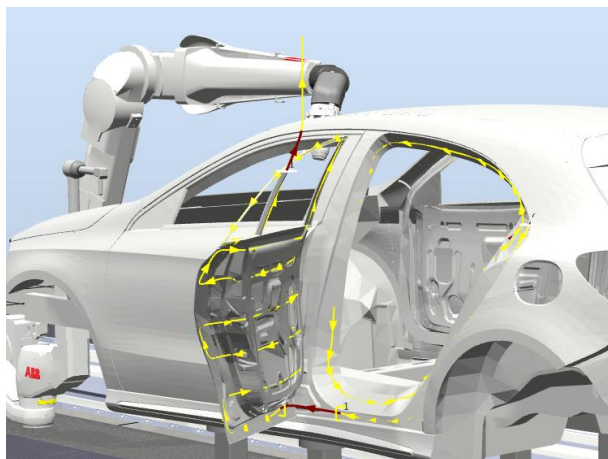


Figura 4.73 - Procedimiento Pint_PI_Tras - Cont_Inf_Izq

Pint Malet Inf

Este procedimiento de pintado está formado a su vez por 2 PaintStrokes o trayectorias de pintado, es decir, se tienen dos activaciones y desactivaciones del atomizador. Este procedimiento está compuesto en su totalidad por instrucciones de movimiento, tanto normales (**MoveL**) como de pintado (**PaintL**) y cuatro instrucciones **SetBrush**, 2 de accionamiento con el parámetro **BrushNumber** igual a la variable **Color** y 2 de desactivación con este parámetro igual a 1.

Este procedimiento, mostrado en la Figura 4.74, se emplea para realizar la aplicación de la capa base en la parte interior izquierda del maletero y en la zona interior izquierda del chasis que cierra el maletero.

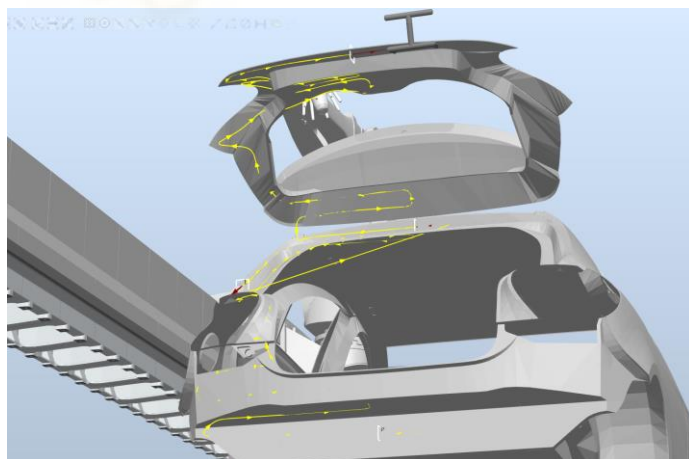


Figura 4.74 - Procedimiento Pint_Malet_Inf - Cont_Inf_Izq

Pint Bajos 1

Este procedimiento de pintado está compuesto en su totalidad por instrucciones de movimiento, tanto normales (**MoveL**) como de pintado (**PaintL**) y dos instrucciones **SetBrush**, 1 de accionamiento con el parámetro BrushNumber igual a la variable Color y 1 de desactivación con este parámetro igual a 1.

Este procedimiento, mostrado en la Figura 4.75, se emplea para realizar la aplicación de la capa base en la parte interior del hueco de la rueda delantera izquierda.

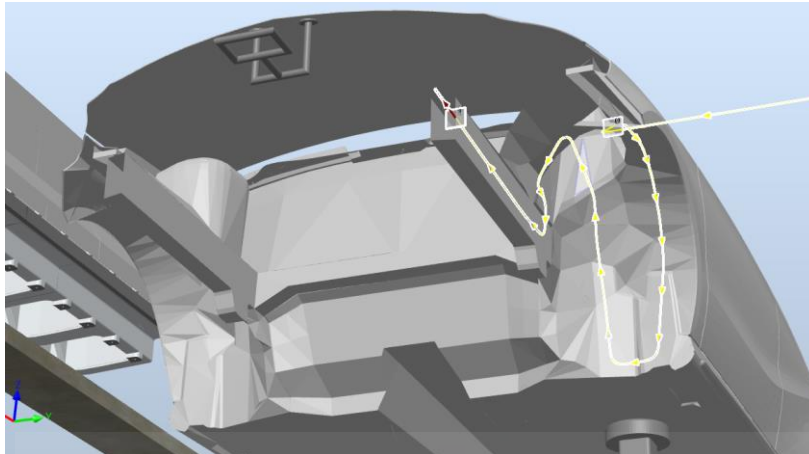


Figura 4.75 - Procedimiento Pint_Bajos_1 - Cont_Inf_Izq

Pint Lateral

Este procedimiento de pintado está formado a su vez por 2 PaintStrokes o trayectorias de pintado, es decir, se tienen dos activaciones y desactivaciones del atomizador. Este procedimiento está compuesto en su totalidad por instrucciones de movimiento de pintado (**PaintL**) y cuatro instrucciones **SetBrush**, 2 de accionamiento con el parámetro **BrushNumber** igual a la variable **Color** y tres de desactivación con este parámetro igual a 2.

Este procedimiento, mostrado en la Figura 4.76, se emplea para realizar la aplicación de la capa base sobre todo el lateral izquierdo de la carrocería, incluyendo chasis y exterior de las puertas.

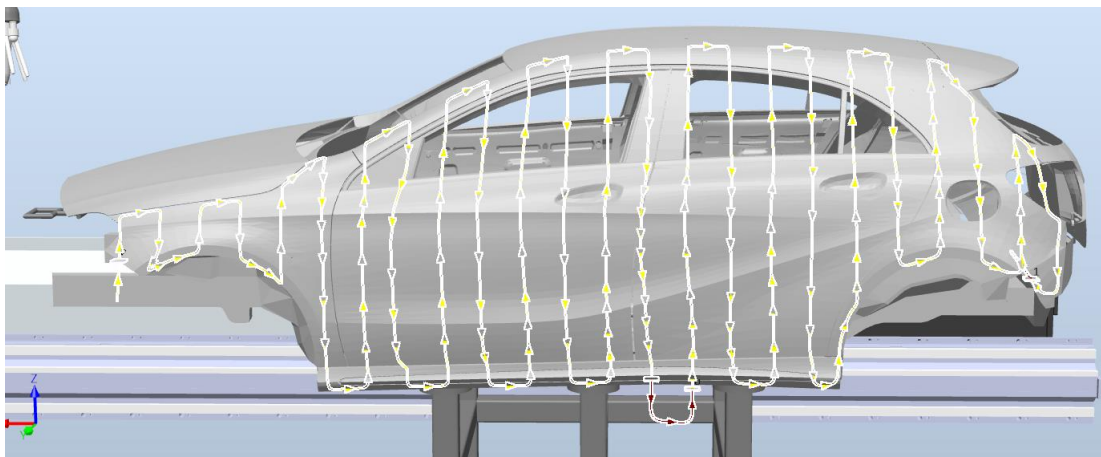


Figura 4.76 - Procedimiento Pint_Lateral - Cont_Inf_Izq

Pint Bajos 2

Este procedimiento de pintado está compuesto en su totalidad por instrucciones de movimiento, tanto normales (**MoveL**) como de pintado (**PaintL**) y dos instrucciones **SetBrush**, 1 de accionamiento con el parámetro BrushNumber igual a la variable Color y 1 de desactivación con este parámetro igual a 1.

Este procedimiento, mostrado en la Figura 4.77, se emplea para realizar la aplicación de la capa base en la parte interior del hueco de la rueda trasera izquierda.

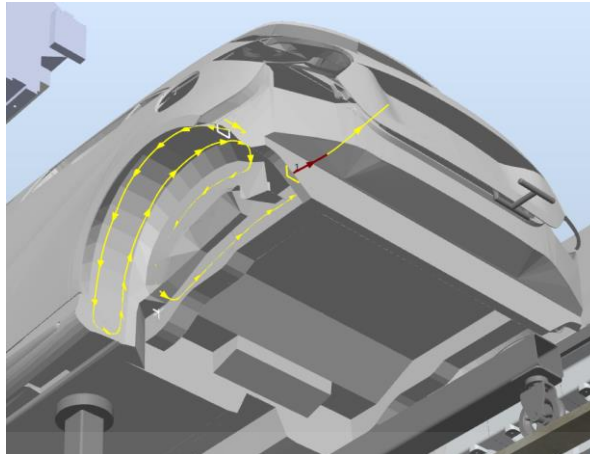


Figura 4.77 - Procedimiento Pint_Bajos_2 - Cont_Inf_Izq

Controlador virtual Cont Inf Der

El código completo del programa de robot de este controlador se encuentra en el apartado [8.1.2 Controlador virtual Cont Inf Der](#). Debido a su tamaño no se volverá a mostrar en este apartado.

Las variables globales definidas para este controlador son las mismas y juegan el mismo papel que las del controlador anterior **Cont_Inf_Izq**, por lo que no se volverán a exponer.

Procedimientos de ejecución

main (Punto de entrada de la simulación)

Este es el procedimiento en el que se inicia la ejecución del programa de robot. Desde él se debe llamar a la ejecución del resto de procedimientos que forman el programa.

Este procedimiento realiza las mismas tareas en el mismo orden que el procedimiento **main** del controlador **Cont_Inf_Izq**, expuesto anteriormente. Simplemente varía, como es lógico, la posición de los objetivos de las instrucciones de movimiento y algunas señales de entrada y

salida, pero las tareas principales y su secuencia es la misma. En este controlador no se tiene la señal digital de salida **Get_Contador**, que se tenía en el controlador **Cont_Inf_Izq**, y se tiene la señal digital de salida **Finalizado2**.

Principal

Dentro de este procedimiento se realizan las llamadas secuenciales a todos los procedimientos que se emplean en este controlador para realizar la aplicación de la capa base sobre la carrocería. Entre las distintas llamadas a los procedimientos se ejecutan instrucciones de movimiento y de acción, sobre todo de espera y de activación/desactivación de señales, para coordinar todos los movimientos de los robots y que el paso de una trayectoria a otra se realice de forma correcta sin causar errores.

La secuencia de ejecución, explicada de forma abreviada, es la siguiente:

1. Se resetean algunas señales, se mueve el robot a la posición inicial y se espera que la carrocería entre en la **Ventana de inicio** del transportador, mediante la instrucción **WaitWObj**.
2. Cuando la carrocería llega a esa posición, se espera a que el controlador **Cont_Sup_Izq** le indique que debe continuar, a través de la señal digital de entrada **Init_Pint1**.
3. Después de esto se esperan 4.5 segundos, para evitar colisiones entre el robot de este controlador y el robot del controlador **Cont_Inf_Izq**.
4. Se llama a la ejecución del procedimiento **Pint_Bajo_Capo**, que pinta la parte derecha de la zona del chasis situada debajo del capó.
5. Se traslada el robot a una posición intermedia mientras se cambia el valor de la señal digital de salida **Abrir_P** a 1, lo cual le indica al controlador **Cont_Puertas_Der** que debe continuar su movimiento.
6. Se resetea la señal digital de salida **Continuar**. Esta se había activado dentro del procedimiento **Pint_Bajo_Capo** para indicar al controlador **Cont_Sup_Izq**, que se había terminado el pintado de la zona debajo del capó y por lo tanto podía continuar.
7. Se espera a que el controlador **Cont_Puertas_Der** le indique que debe continuar, a través de la señal digital de entrada **Init_Pint2**.
8. Se resetea la señal digital de salida **Abrir_P**.
9. Se ejecuta el procedimiento **Pint_PD_Del**, que pinta el interior de puerta derecha delantera.
10. Se traslada el robot a una posición intermedia mientras se cambia el valor de la señal digital de salida **Abrir_P** a 1, lo cual le indica al controlador **Cont_Puertas_Der** que debe continuar su movimiento.
11. Se espera a que el controlador **Cont_Puertas_Der** le indique que debe continuar, a través de la señal digital de entrada **Init_Pint2**.
12. Se ejecuta el procedimiento **Pint_PD_Tras**, que pinta el interior de puerta derecha trasera.
13. Se resetea la señal digital de salida **Abrir_P**.
14. Se traslada el robot a una posición intermedia y se espera a que el controlador **Cont_Sup_Der** le indique que debe continuar, a través de la señal digital de entrada **Init_Pint2**.
15. Después de esto se espera 1 segundo, para evitar colisiones entre el robot de este controlador y el robot del controlador **Cont_Inf_Izq**.

16. Se ejecuta el procedimiento **Pint_Maletero_Inf**, que pinta la parte derecha del interior del maletero.
17. Se ejecutan una serie de instrucciones de movimiento para reposicionar el robot, mientras se cambia el valor de la señal digital de salida **Continuar** a 1, que le indica al controlador **Cont_Sup_Der** que el pintado del interior del maletero ha finalizado y puede continuar.
18. Se resetea la señal digital de salida **Avance_Inf_P**, mediante la cual se le había indicado, dentro del procedimiento **Pint_Maletero_Inf**, al controlador **Cont_Puertas_Der** que debía continuar su ejecución.
19. Se ejecuta el procedimiento **Pint_Bajos_1_1**, que pinta la zona de los bajos de la rueda delantera derecha.
20. Se mueve el robot a una posición intermedia y se ejecuta el procedimiento **Pint_Lat**, que pinta todo el lateral derecho de la carrocería.
21. Se ejecuta el procedimiento **Pint_Bajos_2_2**, que pinta la zona de los bajos de la rueda trasera derecha.
22. Se cambia a 1 el valor de la señal digital de salida **Finalizado**, que indica que se han finalizado los procesos de pintado de la carrocería para este controlador.
23. Se traslada el robot a una posición intermedia.
24. Se cambia a 1 el valor de la señal digital de salida **Volver**, lo cual le indica al controlador **Cont_Puertas_Der** que debe continuar su movimiento.
25. Se resetea la señal digital de salida **Continuar**.
26. Se deja de seguir a la carrocería del transportador, es decir, se suelta el objeto de trabajo **wobj_cnv1** mediante la instrucción **DropWObj**.

Lectura Colores

Como ya se ha mencionado en el controlador anterior, el código de RAPID de este procedimiento es el mismo para todos los robots de aplicación de pintura, y puesto que ya se ha expuesto para el controlado **Cont_Inf_Izq**, no se volverá a exponer.

Limpieza Atom Color

Este procedimiento realiza las mismas tareas en el mismo orden que el procedimiento **Limpieza_Atom_Color** del controlador **Cont_Inf_Izq**, expuesto anteriormente. Simplemente varía, como es lógico, la posición de los objetivos de las instrucciones de movimiento, pero las tareas principales y su secuencia es la misma.

Mantenimiento Atom

Este procedimiento realiza las mismas tareas en el mismo orden que el procedimiento **Mantenimiento_Atom** del controlador **Cont_Inf_Izq**, expuesto anteriormente. Simplemente varía, como es lógico, la posición de los objetivos de las instrucciones de movimiento, pero las tareas principales y su secuencia es la misma.

Procedimientos de pintado

Pint Bajo Capó

Este procedimiento de pintado está compuesto en su totalidad por instrucciones de movimiento, tanto normales (**MoveL**) como de pintado (**PaintL**) y dos instrucciones **SetBrush**, que indican los puntos de accionamiento del atomizador, es decir, los puntos de inicio de pintado. Para la primera de estas instrucciones **SetBrush** su parámetro **BrushNumber** será igual a la variable **Color**, la cual indica el color a aplicar en cada ciclo de pintado. Para la segunda instrucción **SetBrush** este parámetro tendrá el valor 1, puesto que el valor 1 se emplea para detener el atomizador.

Este procedimiento, mostrado en la Figura 4.78, se emplea para realizar la aplicación de la capa base en la parte derecha del chasis que se encuentra bajo el capó. El atomizador se aproxima a la zona de pintado, realiza la trayectoria de pintado y se aleja de la zona, para así facilitar la transición a la siguiente trayectoria.

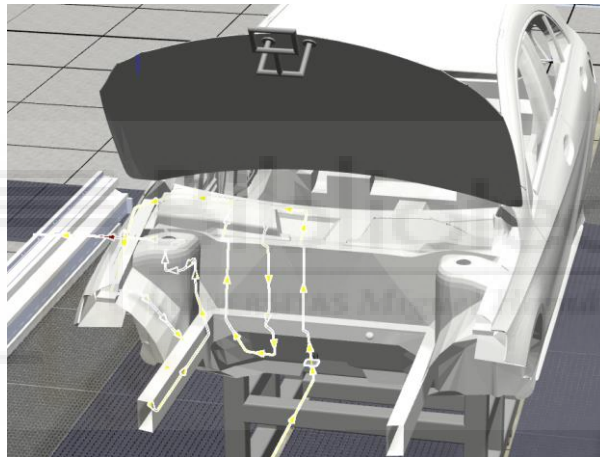


Figura 4.78 - Procedimiento Pint_Bajo_Capó - Cont_Inf_Der

Pint PD Del

Este procedimiento de pintado está formado a su vez por 3 PaintStrokes o trayectorias de pintado, es decir, se tienen tres activaciones y desactivaciones del atomizador. Este procedimiento está compuesto en su totalidad por instrucciones de movimiento de pintado (**PaintL**) y seis instrucciones **SetBrush**, 3 de accionamiento con el parámetro **BrushNumber** igual a la variable **Color** y tres de desactivación con este parámetro igual a 1.

Este procedimiento, mostrado en la Figura 4.79, se emplea para realizar la aplicación de la capa base en la parte interior de la puerta delantera derecha y en la zona interior del chasis que cierra esta puerta.

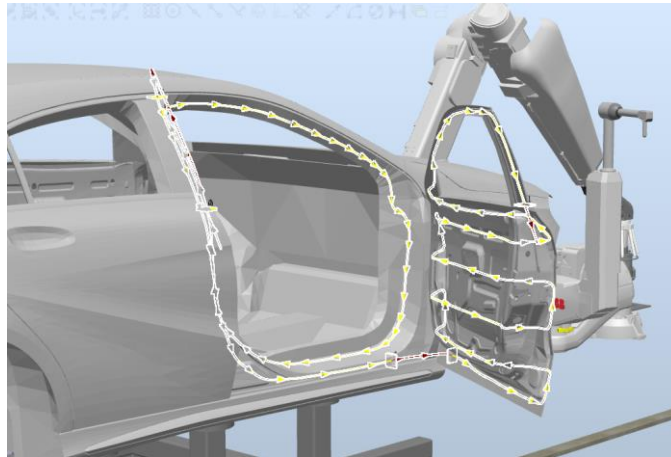


Figura 4.79 - Procedimiento Pint_PD_Del - Cont_Inf_Der

Pint PD Tras

Este procedimiento de pintado está formado a su vez por 3 PaintStrokes o trayectorias de pintado, es decir, se tienen tres activaciones y desactivaciones del atomizador. Este procedimiento está compuesto en su totalidad por instrucciones de movimiento, tanto normales (**MoveL**) como de pintado (**PaintL**) y seis instrucciones **SetBrush**, 3 de accionamiento con el parámetro **BrushNumber** igual a la variable **Color** y tres de desactivación con este parámetro igual a 1.

Este procedimiento, mostrado en la Figura 4.80, se emplea para realizar la aplicación de la capa base en la parte interior de la puerta trasera derecha y en la zona interior del chasis que cierra esta puerta.

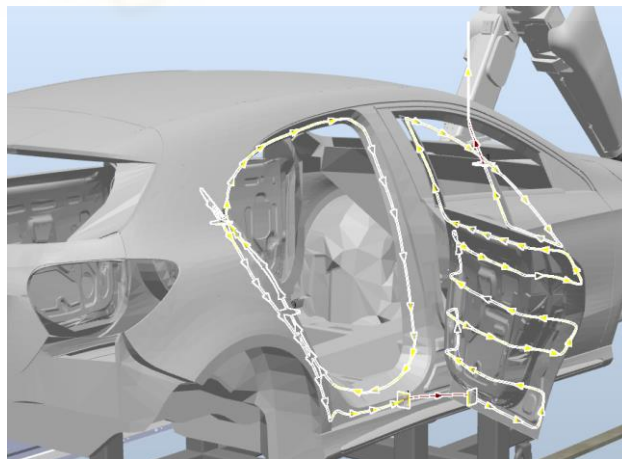


Figura 4.80 - Procedimiento Pint_PD_Tras - Cont_Inf_Der

Pint Maletero Inf

Este procedimiento de pintado está formado a su vez por 2 PaintStrokes o trayectorias de pintado, es decir, se tienen dos activaciones y desactivaciones del atomizador. Este

procedimiento está compuesto en su totalidad por instrucciones de movimiento, tanto normales (**MoveL**) como de pintado (**PaintL**) y cuatro instrucciones **SetBrush**, 2 de accionamiento con el parámetro **BrushNumber** igual a la variable **Color** y 2 de desactivación con este parámetro igual a 1. En el momento de pasar del primer PaintStroke al segundo, se incluye una instrucción **WaitTime** para esperar 2.5 segundos y así evitar que el robot de este controlador colisione con el robot del controlador **Cont_Inf_Izq**.

Este procedimiento, mostrado en la Figura 4.81, se emplea para realizar la aplicación de la capa base en la parte interior derecha del maletero y en la zona interior derecha del chasis que cierra el maletero.

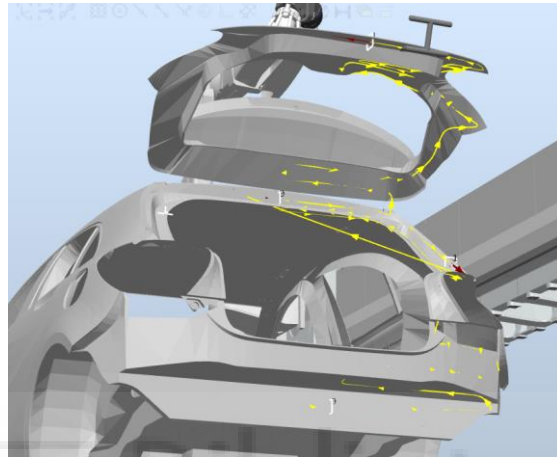


Figura 4.81 - Procedimiento Pint_Maletero_Inf - Cont_Inf_Der

Pint Bajos 1 1

Este procedimiento de pintado está compuesto en su totalidad por instrucciones de movimiento, tanto normales (**MoveL**) como de pintado (**PaintL**) y dos instrucciones **SetBrush**, 1 de accionamiento con el parámetro **BrushNumber** igual a la variable **Color** y 1 de desactivación con este parámetro igual a 1.

Este procedimiento, mostrado en la Figura 4.82, se emplea para realizar la aplicación de la capa base en la parte interior del hueco de la rueda delantera derecha.

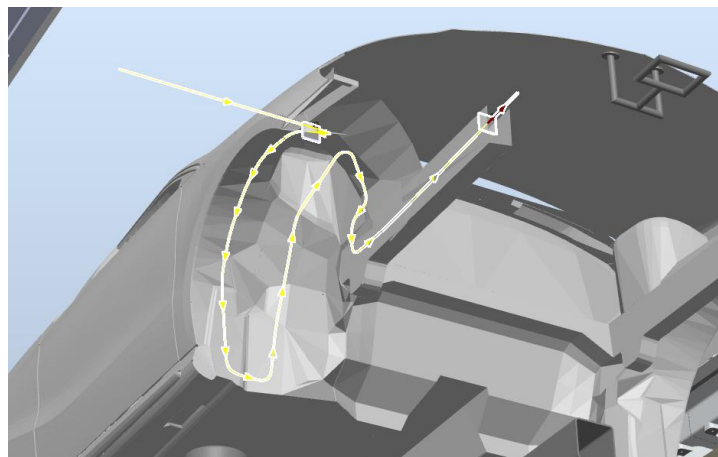


Figura 4.82 - Procedimiento Pint_Bajos_1_1 - Cont_Inf_Der

Pint Lat

Este procedimiento de pintado está formado a su vez por 2 PaintStrokes o trayectorias de pintado, es decir, se tienen dos activaciones y desactivaciones del atomizador. Este procedimiento está compuesto en su totalidad por instrucciones de movimiento de pintado (**PaintL**) y cuatro instrucciones **SetBrush**, 2 de accionamiento con el parámetro **BrushNumber** igual a la variable **Color** y tres de desactivación con este parámetro igual a 2.

Este procedimiento, mostrado en la Figura 4.83, se emplea para realizar la aplicación de la capa base sobre todo el lateral derecho de la carrocería, incluyendo chasis y exterior de las puertas.

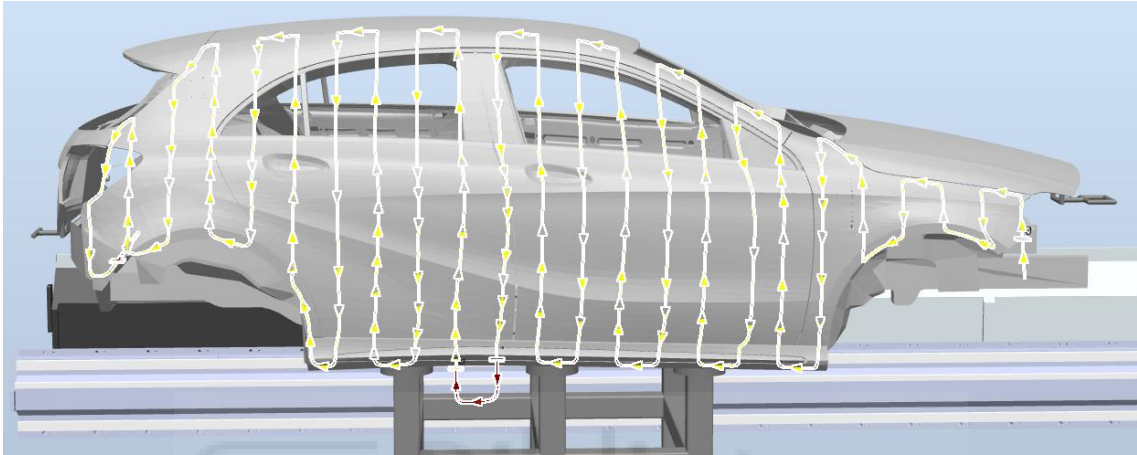


Figura 4.83 - Procedimiento Pint_Lat - Cont_Inf_Der

Pint Bajos 2 2

Este procedimiento de pintado está compuesto en su totalidad por instrucciones de movimiento, tanto normales (**MoveL**) como de pintado (**PaintL**) y dos instrucciones **SetBrush**, 1 de accionamiento con el parámetro **BrushNumber** igual a la variable **Color** y 1 de desactivación con este parámetro igual a 1.

Este procedimiento, mostrado en la Figura 4.84, se emplea para realizar la aplicación de la capa base en la parte interior del hueco de la rueda trasera derecha.

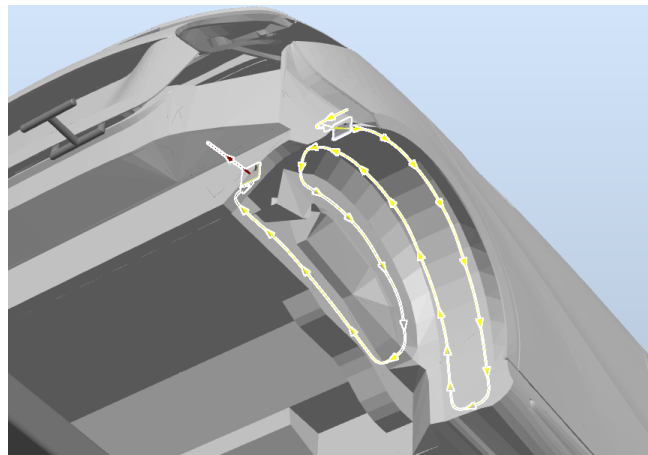


Figura 4.84 - Procedimiento Pint_Bajos_2_2 - Cont_Inf_Der

Controlador virtual Cont_Sup_Izq

El código completo del programa de robot de este controlador se encuentra en el apartado [8.1.3 Controlador virtual Cont_Sup_Izq](#). Debido a su tamaño no se volverá a mostrar en este apartado.

Las variables globales definidas para este controlador son las mismas y juegan el mismo papel que las del controlador anterior **Cont_Inf_Izq**, por lo que no se volverán a exponer. Solo existe una variable más, ya que se tiene una rutina TRAP más en este controlador, siendo esta variable la siguiente:

- VAR intnum **Puerta_Abierta**;
Esta variable se emplea para identificar la interrupción asociada a la rutina TRAP **Paro_Puerta**.

Procedimientos de ejecución

main (Punto de entrada de la simulación)

Este es el procedimiento en el que se inicia la ejecución del programa de robot. Desde él se debe llamar a la ejecución del resto de procedimientos que forman el programa.

Este procedimiento realiza las mismas tareas en el mismo orden que el procedimiento **main** del controlador **Cont_Inf_Izq**, expuesto anteriormente. Simplemente varía, como es lógico, la posición de los objetivos de las instrucciones de movimiento y algunas señales de entrada y salida, pero las tareas principales y su secuencia es la misma. En este controlador no se tiene la señal digital de salida **Get_Contador**, que se tenía en el controlador **Cont_Inf_Izq**.

Principal

Dentro de este procedimiento se realizan las llamadas secuenciales a todos los procedimientos que se emplean en este controlador para realizar la aplicación de la capa base sobre la carrocería. Entre las distintas llamadas a los procedimientos se ejecutan instrucciones de movimiento y de acción, sobre todo de espera y de activación/desactivación de señales, para coordinar todos los movimientos de los robots y que el paso de una trayectoria a otra se realice de forma correcta sin causar errores.

La secuencia de ejecución, explicada de forma abreviada, es la siguiente:

1. Se resetean algunas señales, se mueve el robot a la posición inicial y se espera que la carrocería entre en la **Ventana de inicio** del transportador, mediante la instrucción **WaitWObj**.
2. Se activa la señal digital de salida **Clock_ON**, empleada como punto de activación de uno de los cronómetros de la simulación.

3. Se resetea la señal digital de salida **Finalizado**.
4. Se ejecuta el procedimiento **Abertura_Capo**, mediante el cual se realiza la trayectoria para abrir el capó. Una vez abierto este se mantendrá en la posición abierta gracias a la ayuda de un elemento hidráulico, similar al que mantiene el capó o el maletero abiertos en un vehículo finalizado.
5. Se resetea la señal digital de salida **Clock_ON**.
6. Se traslada el robot a una posición intermedia y se espera hasta que la señal digital de entrada **Continuar** se active. Esta señal se activará cuando la señal digital de salida **Continuar**, del controlador **Cont_Inf_Der** se active, es decir, al finalizar el proceso **Pint_Bajo_Capo** de este controlador.
7. Se ejecuta el procedimiento **Pint_Capo_Inf**, mediante el cual se pinta la parte izquierda de la superficie interior del capó.
8. Se traslada el robot a una posición intermedia y se espera 1.5 segundos, para evitar que en el siguiente procedimiento el robot colisione con el robot del controlador **Cont_Sup_Der**.
9. Se traslada el robot a una posición de aproximación para el cierre del capó y se espera a que la señal digital de entrada **Continuar2** se active. Esta señal se activará cuando la señal digital de salida **Continuar2** del controlador **Cont_Sup_Der** se active, es decir, cuando el robot del controlador **Cont_Sup_Der** finalice el pintado de la parte derecha de la superficie interior del capó.
10. Se ejecuta el procedimiento **Cierre_Capo**, mediante el cual se realiza la trayectoria para cerrar el capó.
11. Se traslada el robot a una posición intermedia y se espera 3.5 segundos, para evitar que en el siguiente procedimiento el robot colisione con el robot del controlador **Cont_Sup_Der**.
12. Se ejecuta el procedimiento **Pint_Capo_Sup**, mediante el cual se realiza la trayectoria para pintar la parte izquierda de la superficie exterior del capó.
13. Se ejecuta el procedimiento **Pint_Barra**, mediante el cual se realiza la trayectoria para pintar la barra izquierda que une el pilar delantero izquierdo del chasis con el techo.
14. Cuando se termina este procedimiento se espera 2 segundos, para evitar que en el siguiente procedimiento el robot colisione con el robot del controlador **Cont_Sup_Der**.
15. Se ejecuta el procedimiento **Pint_Techo**, mediante el cual se realiza la trayectoria para pintar la parte izquierda del techo.
16. Se ejecuta el procedimiento **Pint_Malet_Sup**, mediante el cual se realiza la trayectoria para pintar la parte izquierda de la superficie exterior del maletero.
17. Se traslada el robot a una posición intermedia y al mismo tiempo se activa la señal digital de salida **Continuar3**. Esta señal le indica al controlador **Cont_Sup_Der** que puede continuar y realizar la trayectoria que abre el maletero.
18. Se activa la señal digital de salida **Finalizado**.
19. Se traslada el robot a una posición intermedia y al mismo tiempo se desactiva la señal digital de salida **Continuar3**.
20. Se deja de seguir a la carrocería del transportador, es decir, se suelta el objeto de trabajo **wobj_cnv1** mediante la instrucción **DropWObj**.

Lectura Colores

Como ya se ha mencionado en el controlador anterior, el código de RAPID de este procedimiento es el mismo para todos los robots de aplicación de pintura, y puesto que ya se ha expuesto para el controlado **Cont_Inf_Izq**, no se volverá a exponer.

Limpieza Atom Color

Este procedimiento realiza las mismas tareas en el mismo orden que el procedimiento **Limpieza_Atom_Color** del controlador **Cont_Inf_Izq**, expuesto anteriormente. Simplemente varía, como es lógico, la posición de los objetivos de las instrucciones de movimiento, pero las tareas principales y su secuencia es la misma.

Mantenimiento Atom

Este procedimiento realiza las mismas tareas en el mismo orden que el procedimiento **Mantenimiento_Atom** del controlador **Cont_Inf_Izq**, expuesto anteriormente. Simplemente varía, como es lógico, la posición de los objetivos de las instrucciones de movimiento y existen algunas instrucciones de movimiento más para realizar el acercamiento a la trampilla de mantenimiento, pero las tareas principales y su secuencia es la misma.



Procedimientos de asistencia

Abertura Capó

Este procedimiento de asistencia está compuesto en su totalidad por instrucciones de movimiento lineal y circular (**MoveL** y **MoveC**). Dentro de este procedimiento se activará la señal digital de salida **Abrir_Capó**, para indicarle al componente inteligente **Abertura_Capó** que puede rotar el capó 60° para abrirlo. Al final del procedimiento también se activa la señal digital de salida **Inic_Pint_Inf**, que les indica a los robots de los controladores **Cont_Inf_Izq** y **Cont_Inf_Der** que pueden iniciar el pintado de la parte del chasis que se encuentra bajo el capó.

Este procedimiento, mostrado en la Figura 4.85, se emplea para realizar la trayectoria mediante la cual se realiza la apertura del capó, para posibilitar el pintado de su superficie interior y de la zona de chasis que se encuentra bajo el mismo.

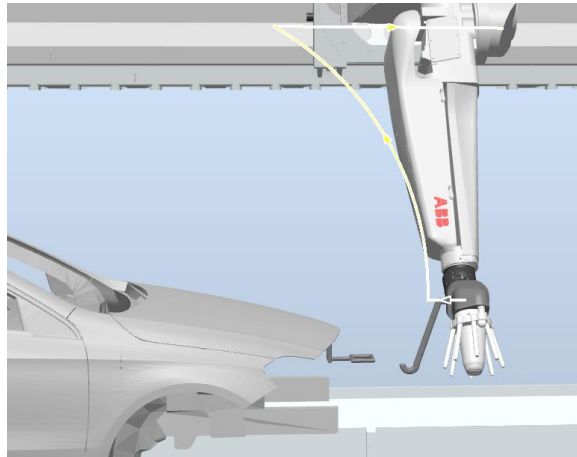


Figura 4.85 - Procedimiento Abertura_Capo - Cont_Sup_Izq

En esta trayectoria el robot emplea el TCP del centro de la muñeca, puesto que el añadido de la pistola para el agarre del capó no posee TCP. Se deberá diseñar la trayectoria para que coincida el extremo de la herramienta de agarre con el elemento de agarre del capó, pero programando los objetivos para la muñeca del robot.

Para programar esta trayectoria se emplea la herramienta **Trayectoria automática**. Para emplear esta herramienta previamente se diseña una curva que sirva de guía para la trayectoria, es decir, la trayectoria sigue un trozo de la curva auxiliar, mostrada en la Figura 4.86.

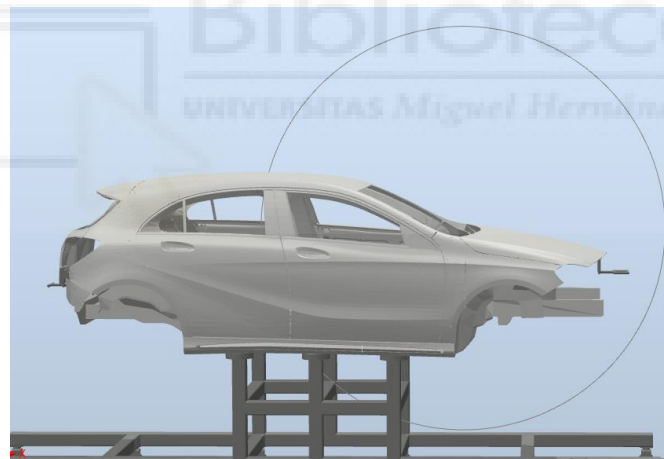


Figura 4.86 - Curva auxiliar para la programación de la trayectoria de apertura del capó

Como se ha mencionado anteriormente, una vez abierto el capó, este se quedará en la posición abierta gracias a la ayuda de un elemento hidráulico o neumático.

Cierre Capo

Este procedimiento de asistencia está compuesto en su totalidad por instrucciones de movimiento lineal y circular (**MoveL** y **MoveC**). Dentro de este procedimiento se activará la señal digital de salida **Cerrar_Capo**, para indicarle al componente inteligente **Abertura_Capo** que puede rotar el capó 60° para cerrarlo. Al final del procedimiento también se activa la señal digital

de salida **Continuar3**, que le indica al robot del controlador **Cont_Sup_Der** que puede iniciar el pintado de la superficie superior del capó.

Este procedimiento, mostrado en la Figura 4.87, se emplea para realizar la trayectoria mediante la cual se realiza el cierre del capó.

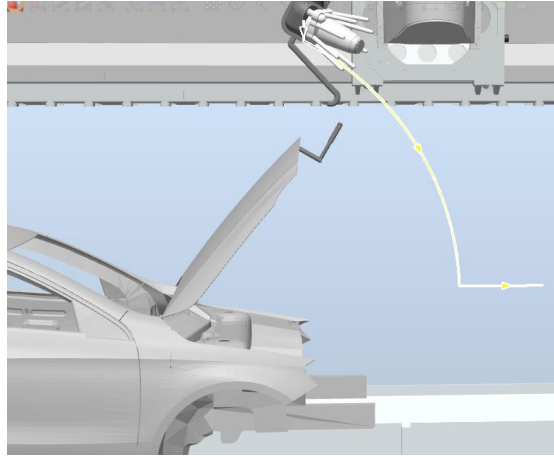


Figura 4.87 - Procedimiento Cierre_Capó - Cont_Sup_Izq

Esta trayectoria de cierre se programa de la misma forma que la trayectoria de apertura explicada anteriormente.



Pint_Capó_Inf

Este procedimiento de pintado está compuesto en su totalidad por instrucciones de movimiento, tanto normales (**MoveL**) como de pintado (**PaintL**) y dos instrucciones **SetBrush**, que indican los puntos de accionamiento del atomizador, es decir, los puntos de inicio de pintado. Para la primera de estas instrucciones **SetBrush** su parámetro **BrushNumber** será igual a la variable **Color**, la cual indica el color a aplicar en cada ciclo de pintado. Para la segunda instrucción **SetBrush** este parámetro tendrá el valor 1, puesto que el valor 1 se emplea para detener el atomizador.

Este procedimiento, mostrado en la Figura 4.88, se emplea para realizar la aplicación de la capa base en la parte izquierda de la superficie interior del capó. El atomizador se aproxima a la zona de pintado, realiza la trayectoria de pintado y se aleja de la zona, para así facilitar la transición a la siguiente trayectoria.

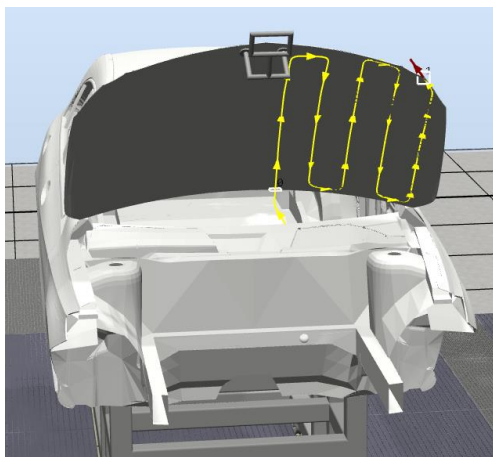


Figura 4.88 - Procedimiento Pint_Capo_Inf - Cont_Sup_Izq

Pint Capo Sup

Este procedimiento de pintado está compuesto en su totalidad por instrucciones de movimiento de pintado (**PaintL**) y dos instrucciones **SetBrush**, 1 de accionamiento con el parámetro **BrushNumber** igual a la variable **Color** y 1 de desactivación con este parámetro igual a 1.

Este procedimiento, mostrado en la Figura 4.89, se emplea para realizar la aplicación de la capa base en la parte izquierda de la superficie superior del capó. El atomizador se aproxima a la zona de pintado, realiza la trayectoria de pintado y se aleja de la zona, para así facilitar la transición a la siguiente trayectoria.

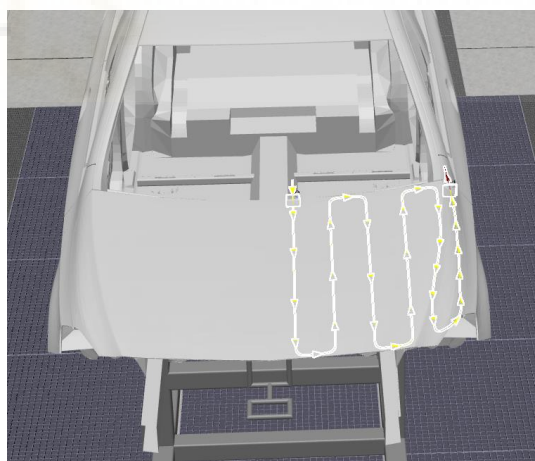


Figura 4.89 - Procedimiento Pint_Capo_Sup - Cont_Sup_Izq

Pint Barra

Este procedimiento de pintado está compuesto en su totalidad por instrucciones de movimiento de pintado (**PaintL**) y dos instrucciones **SetBrush**, 1 de accionamiento con el parámetro **BrushNumber** igual a la variable **Color** y 1 de desactivación con este parámetro igual a 1.

Este procedimiento, mostrado en la Figura 4.90, se emplea para realizar la aplicación de la capa base sobre la barra izquierda que une el pilar delantero izquierdo del chasis con el techo.

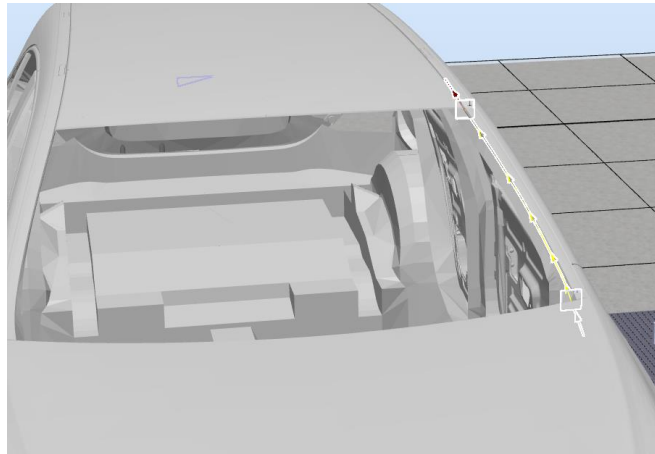


Figura 4.90 - Procedimiento Pint_Barra - Cont_Sup_Izq

Pint_Techo

Este procedimiento de pintado está compuesto en su totalidad por instrucciones de movimiento de pintado (**PaintL**) y dos instrucciones **SetBrush**, 1 de accionamiento con el parámetro **BrushNumber** igual a la variable **Color** y 1 de desactivación con este parámetro igual a 1.

Este procedimiento, mostrado en la Figura 4.91, se emplea para realizar la aplicación de la capa base sobre la parte izquierda del techo de la carrocería.

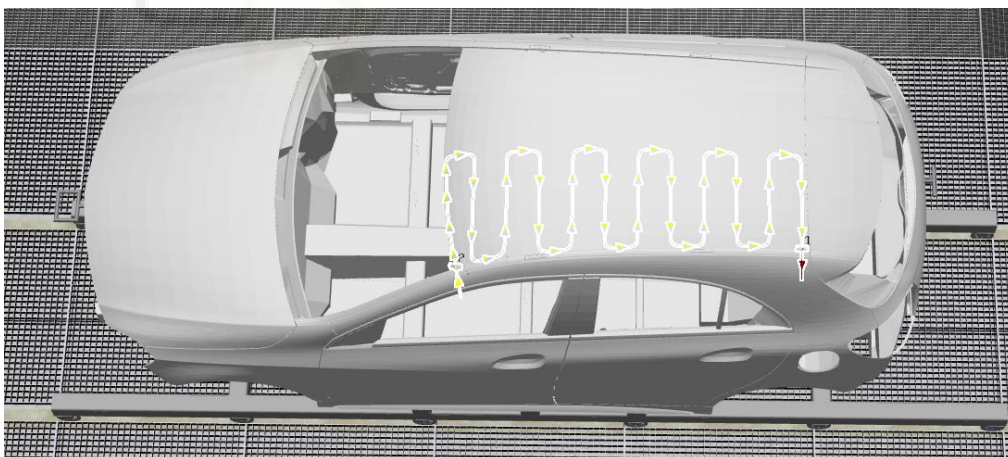


Figura 4.91 - Procedimiento Pint_Techo - Cont_Sup_Izq

Pint_Malet_Sup

Este procedimiento de pintado está compuesto en su totalidad por instrucciones de movimiento de pintado (**PaintL**) y dos instrucciones **SetBrush**, 1 de accionamiento con el parámetro **BrushNumber** igual a la variable **Color** y 1 de desactivación con este parámetro igual a 1.

Este procedimiento, mostrado en la Figura 4.92, se emplea para realizar la aplicación de la capa base sobre la parte izquierda de la superficie exterior del maletero.

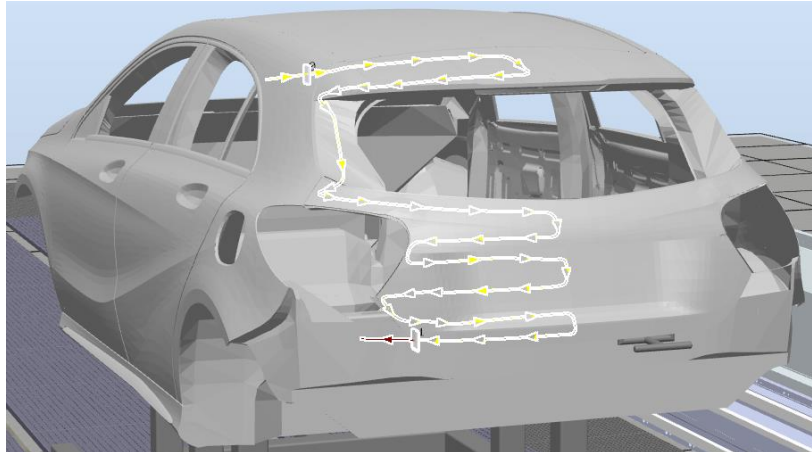


Figura 4.92 - Procedimiento Pint_Malet_Sup - Cont_Sup_Izq

Controlador virtual Cont Sup Der

El código completo del programa de robot de este controlador se encuentra en el apartado [8.1.4 Controlador virtual Cont Sup Der](#). Debido a su tamaño no se volverá a mostrar en este apartado.

Las variables globales definidas para este controlador son las mismas y juegan el mismo papel que las del controlador anterior **Cont_Inf_Izq**, por lo que no se volverán a exponer.

Procedimientos de ejecución

main (Punto de entrada de la simulación)

Este es el procedimiento en el que se inicia la ejecución del programa de robot. Desde él se debe llamar a la ejecución del resto de procedimientos que forman el programa.

Este procedimiento realiza las mismas tareas en el mismo orden que el procedimiento **main** del controlador **Cont_Inf_Izq**, expuesto anteriormente. Simplemente varía, como es lógico, la posición de los objetivos de las instrucciones de movimiento y algunas señales de entrada y salida, pero las tareas principales y su secuencia es la misma. En este controlador no se tiene la señal digital de salida **Get_Contador**, que se tenía en el controlador **Cont_Inf_Izq**.

Principal

Dentro de este procedimiento se realizan las llamadas secuenciales a todos los procedimientos que se emplean en este controlador para realizar la aplicación de la capa base sobre la carrocería. Entre las distintas llamadas a los procedimientos se ejecutan instrucciones de movimiento y de acción, sobre todo de espera y de activación/desactivación de señales, para coordinar todos los movimientos de los robots y que el paso de una trayectoria a otra se realice de forma correcta sin causar errores.

La secuencia de ejecución, explicada de forma abreviada, es la siguiente:

1. Se resetean algunas señales, se mueve el robot a la posición inicial y se espera que la carrocería entre en la **Ventana de inicio** del transportador, mediante la instrucción **WaitWObj**.
2. Se resetea la señal digital de salida **Finalizado**.
3. Se traslada el robot a una posición intermedia y se espera hasta que la señal digital de entrada **Continuar** se active. Esta señal se activará cuando la señal digital de salida **Continuar**, del controlador **Cont_Inf_Der** se active, es decir, al finalizar el proceso **Pint_Bajo_Capó** de este controlador.
4. Después de esto se esperan 3.5 segundos, para evitar colisiones entre el robot de este controlador y el robot del controlador **Cont_Sup_Izq**.
5. Se ejecuta el procedimiento **Pint_Capó_Inf**, mediante el cual se pinta la parte derecha de la superficie interior del capó.
6. Se traslada el robot a una posición intermedia y al mismo tiempo se activa la señal digital de salida **Continuar2**. Esta señal le indica al controlador **Cont_Sup_Izq** que puede continuar y realizar la trayectoria que cierra el capó.
7. Se espera a que la señal digital de entrada **Continuar3** se active. Esta señal se activará cuando la señal digital de salida **Continuar3**, del controlador **Cont_Sup_Izq** se active, es decir, al finalizar el proceso **Cierre_Capó** de este controlador, indicándole así al robot del presente controlador que puede iniciar el pintado de la superficie superior del capó.
8. Se ejecuta el procedimiento **Pint_Capó_Sup**, mediante el cual se realiza la trayectoria para pintar la parte derecha de la superficie exterior del capó.
9. Se ejecuta el procedimiento **Pint_Barra**, mediante el cual se realiza la trayectoria para pintar la barra derecha que une el pilar delantero derecho del chasis con el techo.
10. Se ejecuta el procedimiento **Pint_Techo**, mediante el cual se realiza la trayectoria para pintar la parte derecha del techo.
11. Se ejecuta el procedimiento **Pint_Malet_Sup**, mediante el cual se realiza la trayectoria para pintar la parte derecha de la superficie exterior del maletero.
12. Se traslada el robot a una posición intermedia y se espera hasta que la señal digital de entrada **Continuar3** se active. Esta señal se activará cuando la señal digital de salida **Continuar3**, del controlador **Cont_Sup_Izq** se active, es decir, al finalizar el proceso **Pint_Malet_Sup** de este controlador, indicándole así al robot del presente controlador que puede iniciar la trayectoria de abertura del maletero.
13. Se ejecuta el procedimiento **Abertura_Maletero**, mediante el cual se realiza la trayectoria para abrir el maletero. Una vez abierto este se mantendrá en la posición abierta gracias a la ayuda de un elemento hidráulico, similar al que mantiene el capó o el maletero abiertos en un vehículo finalizado.
14. Se traslada el robot a una posición intermedia y se espera hasta que la señal digital de entrada **Continuar** se active. Esta señal se activará cuando la señal digital de salida

Continuar, del controlador **Cont_Inf_Der** se active, es decir, al finalizar el proceso **Pint_Maletero_Inf** de este controlador, indicándole así al robot del presente controlador que puede iniciar la trayectoria de cierre del maletero.

15. Se resetea la señal digital de salida **Inic_Pint_Inf**, la cual se había activado dentro del procedimiento **Abertura_Maletero**, para indicar a los robots de los controladores **Cont_Inf_Izq** y **Cont_Inf_Der** que podían iniciar el pintado de la superficie interior del maletero y la zona del chasis que cierra el maletero.
16. Se ejecuta el procedimiento **Cierre_Maletero**, mediante el cual se realiza la trayectoria para cerrar el maletero.
17. Se activa la señal digital de salida **Finalizado**.
18. Se traslada el robot a una posición intermedia.
19. Se deja de seguir a la carrocería del transportador, es decir, se suelta el objeto de trabajo **wobj_cnv1** mediante la instrucción **DropWObj**.

Lectura Colores

Como ya se ha mencionado en el controlador anterior, el código de RAPID de este procedimiento es el mismo para todos los robots de aplicación de pintura, y puesto que ya se ha expuesto para el controlado **Cont_Inf_Izq**, no se volverá a exponer.

Limpieza Atom Color

Este procedimiento realiza las mismas tareas en el mismo orden que el procedimiento **Limpieza_Atom_Color** del controlador **Cont_Inf_Izq**, expuesto anteriormente. Simplemente varía, como es lógico, la posición de los objetivos de las instrucciones de movimiento, pero las tareas principales y su secuencia es la misma.

Mantenimiento Atom

Este procedimiento realiza las mismas tareas en el mismo orden que el procedimiento **Mantenimiento_Atom** del controlador **Cont_Inf_Izq**, expuesto anteriormente. Simplemente varía, como es lógico, la posición de los objetivos de las instrucciones de movimiento y existen algunas instrucciones de movimiento más para realizar el acercamiento a la trampilla de mantenimiento, pero las tareas principales y su secuencia es la misma.

Procedimientos de asistencia

Abertura Maletero

Este procedimiento de asistencia está compuesto en su totalidad por instrucciones de movimiento lineal y circular (**MoveL** y **MoveC**). Dentro de este procedimiento se activará la señal

digital de salida **Abrir_Maletero**, para indicarle al componente inteligente **Abertura_Maletero** que puede rotar el maletero 70° para abrirlo. Al final del procedimiento también se activa la señal digital de salida **Inic_Pint_Inf**, que les indica a los robots de los controladores **Cont_Inf_Izq** y **Cont_Inf_Der** que pueden iniciar el pintado la parte interior del maletero y en la zona interior del chasis que cierra el maletero.

Este procedimiento, mostrado en la Figura 4.93, se emplea para realizar la trayectoria mediante la cual se realiza la abertura del maletero, para posibilitar el pintado de su superficie interior y de la zona interior del chasis que cierra el mismo.

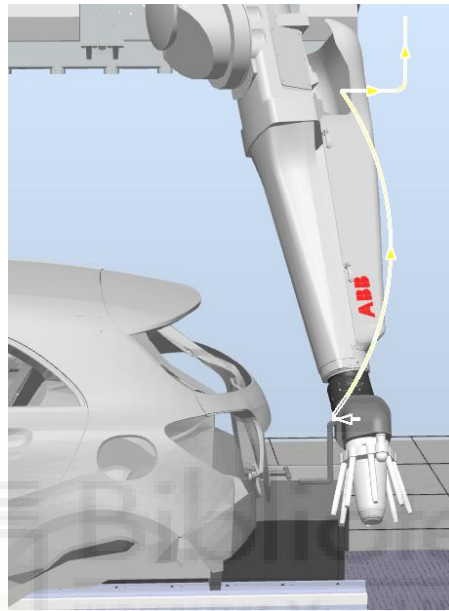


Figura 4.93 - Procedimiento Abertura_Maletero - Cont_Sup_Der

En esta trayectoria el robot emplea el TCP del centro de la muñeca, puesto que el añadido de la pistola para el agarre del maletero no posee TCP. Se deberá diseñar la trayectoria para que coincida el extremo de la herramienta de agarre con el elemento de agarre del maletero, pero programando los objetivos para la muñeca del robot.

Para programar esta trayectoria se emplea la herramienta **Trayectoria automática**. Para emplear esta herramienta previamente se diseña una curva que sirva de guía para la trayectoria, es decir, la trayectoria sigue un trozo de la curva auxiliar, mostrada en la Figura 4.94.

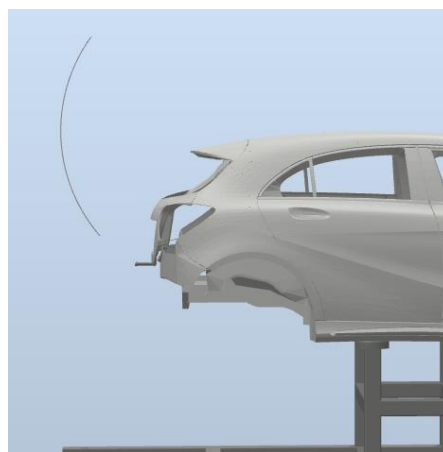


Figura 4.94 - Curva auxiliar para la programación de la trayectoria de abertura del maletero

Como se ha mencionado anteriormente, una vez abierto el maletero este se quedará en la posición abierta gracias a la ayuda de un elemento hidráulico o neumático auxiliar.

Cierre Maletero

Este procedimiento de asistencia está compuesto en su totalidad por instrucciones de movimiento lineal y circular (**MoveL** y **MoveC**). Dentro de este procedimiento se activará la señal digital de salida **Cerrar_Maletero**, para indicarle al componente inteligente **Abertura_Maletero** que puede rotar el maletero 70° para cerrarlo.

Este procedimiento, mostrado en la Figura 4.95, se emplea para realizar la trayectoria mediante la cual se realiza el cierre del maletero.

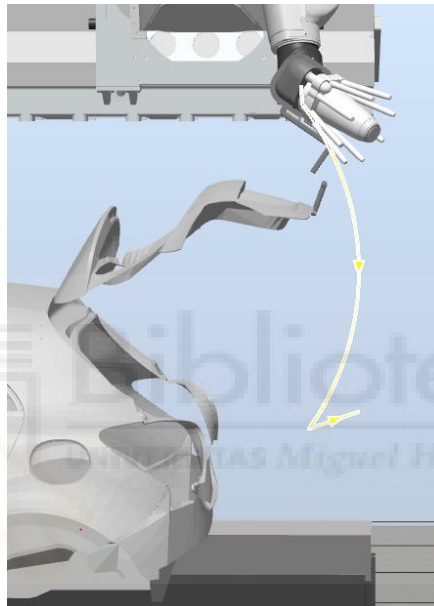


Figura 4.95 - Procedimiento Cierre_Maletero - Cont_Sup_Der

Esta trayectoria de cierre se programa de la misma forma que la trayectoria de abertura explicada anteriormente.

Procedimientos de pintado

Pint Capo Inf

Este procedimiento de pintado está compuesto en su totalidad por instrucciones de movimiento, tanto normales (**MoveL**) como de pintado (**PaintL**) y dos instrucciones **SetBrush**, que indican los puntos de accionamiento del atomizador, es decir, los puntos de inicio de pintado. Para la primera de estas instrucciones **SetBrush** su parámetro **BrushNumber** será igual a la variable **Color**, la cual indica el color a aplicar en cada ciclo de pintado. Para la segunda

instrucción **SetBrush** este parámetro tendrá el valor 1, puesto que el valor 1 se emplea para detener el atomizador.

Este procedimiento, mostrado en la Figura 4.96, se emplea para realizar la aplicación de la capa base en la parte derecha de la superficie interior del capó. El atomizador se aproxima a la zona de pintado, realiza la trayectoria de pintado y se aleja de la zona, para así facilitar la transición a la siguiente trayectoria.

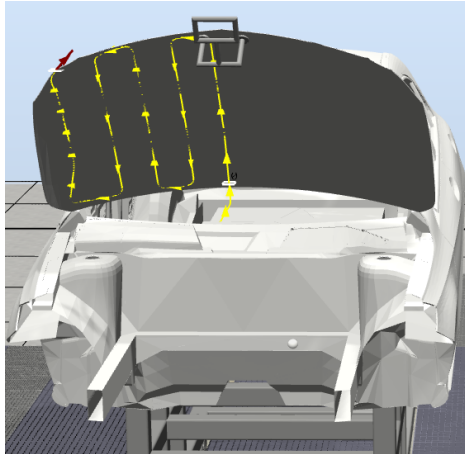


Figura 4.96 - Procedimiento Pint_Capo_Inf - Cont_Sup_Der

Pint Capo Sup

Este procedimiento de pintado está compuesto en su totalidad por instrucciones de movimiento de pintado (**PaintL**) y dos instrucciones **SetBrush**, 1 de accionamiento con el parámetro **BrushNumber** igual a la variable **Color** y 1 de desactivación con este parámetro igual a 1.

Este procedimiento, mostrado en la Figura 4.97, se emplea para realizar la aplicación de la capa base en la parte derecha de la superficie superior del capó. El atomizador se aproxima a la zona de pintado, realiza la trayectoria de pintado y se aleja de la zona, para así facilitar la transición a la siguiente trayectoria.

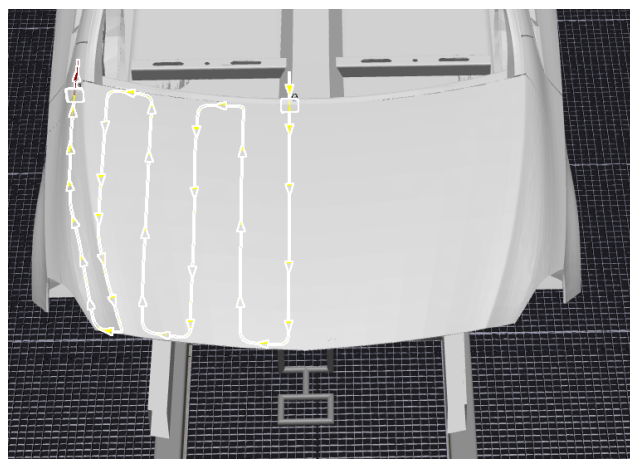


Figura 4.97 - Procedimiento Pint_Capo_Sup - Cont_Sup_Der

Pint Barra

Este procedimiento de pintado está compuesto en su totalidad por instrucciones de movimiento de pintado (**PaintL**) y dos instrucciones **SetBrush**, 1 de accionamiento con el parámetro **BrushNumber** igual a la variable **Color** y 1 de desactivación con este parámetro igual a 1.

Este procedimiento, mostrado en la Figura 4.98, se emplea para realizar la aplicación de la capa base sobre la barra derecha que une el pilar delantero derecho del chasis con el techo.

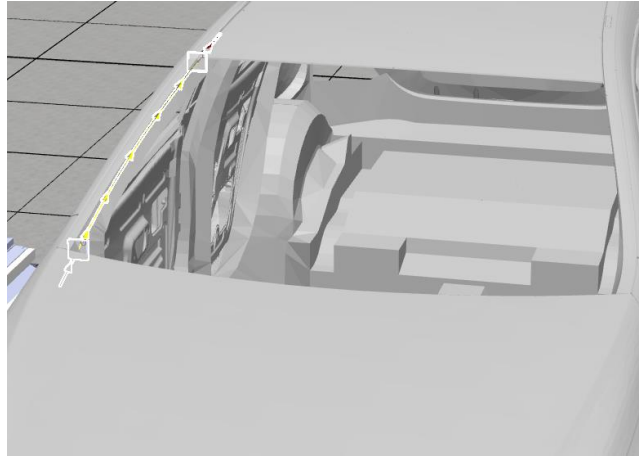


Figura 4.98 - Procedimiento Pint_Barra - Cont_Sup_Der

Pint Techo

Este procedimiento de pintado está compuesto en su totalidad por instrucciones de movimiento de pintado (**PaintL**) y dos instrucciones **SetBrush**, 1 de accionamiento con el parámetro **BrushNumber** igual a la variable **Color** y 1 de desactivación con este parámetro igual a 1.

Este procedimiento, mostrado en la Figura 4.99, se emplea para realizar la aplicación de la capa base sobre la parte derecha del techo de la carrocería.

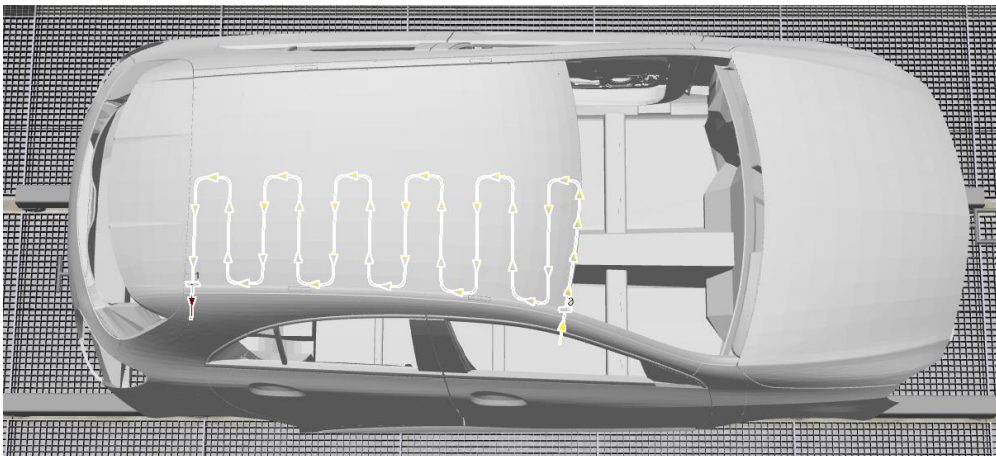


Figura 4.99 - Procedimiento Pint_Techo - Cont_Sup_Der

Pint Malet Sup

Este procedimiento de pintado está compuesto en su totalidad por instrucciones de movimiento de pintado (**PaintL**) y dos instrucciones **SetBrush**, 1 de accionamiento con el parámetro **BrushNumber** igual a la variable **Color** y 1 de desactivación con este parámetro igual a 1.

Este procedimiento, mostrado en la Figura 4.100, se emplea para realizar la aplicación de la capa base sobre la parte derecha de la superficie exterior del maletero.

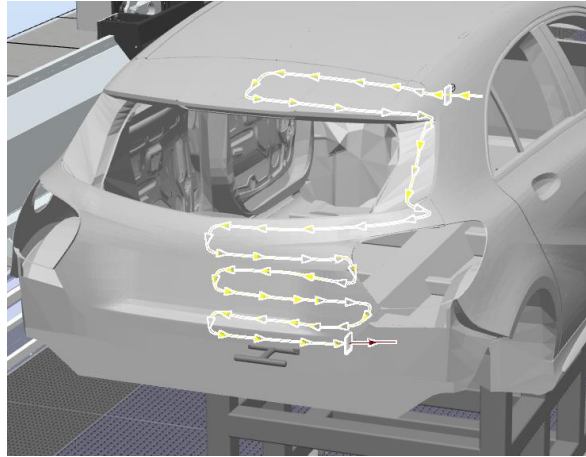


Figura 4.100 - Procedimiento Pint_Malet_Sup - Cont_Sup_Der



Controlador virtual Cont Puertas Izq

El código completo del programa de robot de este controlador se encuentra en el apartado [8.1.5 Controlador virtual Cont Puertas Izq](#). Debido a su tamaño no se volverá a mostrar en este apartado.

En este controlador solo se tienen las variables globales **Paro_Robot**, **N_Proc** y **Contador**, ya expuestas en el controlador **Cont_Inf_Izq**, por lo que no se volverán a exponer.

Procedimientos de ejecución

main (Punto de entrada de la simulación)

Este es el procedimiento en el que se inicia la ejecución del programa de robot. Desde él se debe llamar a la ejecución del resto de procedimientos que forman el programa.

El procedimiento realiza las siguientes tareas de forma secuencial.

1. Se resetean todas las señales de E/S creadas para el proyecto y que intervienen en el mismo.
2. Se borra la conexión a la identidad de interrupción **Paro_Robot** asociada a la rutina TRAP **Parada**, como medida de seguridad para que no existan posibles errores en cuanto a la conexión con esta rutina.
3. Se realiza la conexión entre esta identidad de interrupción y la rutina TRAP **Parada**.
4. Se especifica que señal digital de entrada debe cambiar su valor al indicado para que la ejecución normal del programa se detenga y se pase a ejecutar la rutina TRAP **Parada**.
5. Se activa la unidad mecánica CNV1, es decir, el transportador.
6. En este punto se comienza un bucle FOR, para la ejecución repetitiva de todos los procedimientos de asistencia y, si se requiere, de mantenimiento. Este bucle se ejecutará un número de veces igual al valor de la variable **Contador**. El valor de esta variable se fija a través de un componente inteligente **RapidVariable**. Mediante este componente inteligente se obtiene el valor de la variable **Contador** del controlador **Cont_Inf_Izq** y se pasa ese valor a la variable **Contador** del presente controlador. Esta variable representa el número de ciclos de pintado que deben llevarse a cabo.
7. Dentro de este bucle se ejecuta el procedimiento **Principal**, que contiene las llamadas a todos los procedimientos de asistencia que forman el programa. Con la ejecución completa de este procedimiento se ejecutan todas las trayectorias que realizan la abertura y cierre de las puertas del lado izquierdo de la carrocería.
8. Una vez ejecutado este procedimiento se espera a que la señal digital de entrada **Finalizado** se active. Esta señal se activará cuando la señal digital de salida **Finalizado2**, del controlador **Cont_Inf_Der** se active, es decir, al finalizar un ciclo completo de pintado.
9. En este punto comienza una expresión condicional IF, de modo que si la señal digital de entrada **Mantenimiento** se ha activado, y se mantiene activa, en algún momento de la ejecución del programa, se realizará el mantenimiento de la herramienta del robot, es decir, el mantenimiento del IRB5350 Gripper.

Para realizar dicho mantenimiento se llevan a cabo las siguientes instrucciones:

1. Primero se activa la señal digital de salida **Paro_Resto**, que activará en el resto de controladores la rutina TRAP **Parada**, de modo que el resto de robots detengan su movimiento.
2. A continuación se activa la señal digital de salida **Stop_Conveyor**, que detiene el avance del transportador.
3. Se cambia de valor la señal analógica de salida **Señal_Luminosa**.
4. Se ejecuta el procedimiento **Mantenimiento_Gripper**, que ejecuta las instrucciones para realizar el mantenimiento.
5. Se traslada el robot a la posición inicial.
6. Se resetea la señal digital de salida **Paro_Resto**, de modo que en el resto de controladores se continúe la ejecución dentro de la rutina TRAP **Parada**, que hasta ahora estaba en espera.
7. Se resetea la señal digital de salida **Close_TMant**, que se había activado en el procedimiento **Mantenimiento_Gripper**, para cerrar la trampilla de mantenimiento.
8. Se activa la señal digital de salida **Pos_Parada**, que indica que en ese controlador ya se ha llegado a los pasos finales de la rutina TRAP en cuestión o en este caso del mantenimiento.

9. Se espera a que la señal digital de entrada **OK_PosP** se active, lo cual indica que todos los controladores se están listos para retomar su movimiento.
 10. Se espera 0.25 segundos, para asegurar que todos los controladores comienzan de nuevo su movimiento a la vez.
 11. Se resetea la señal digital de salida **Stop_Conveyor**, de modo que el transportador vuelve a moverse.
 12. Se resetea la señal digital de salida **Pos_Parada**
 13. Se cambia de valor la señal analógica de salida **Señal_Luminosa**.
10. Cuando se termina de ejecutar el bucle FOR, es decir, cuando se termina el proceso de pintado de todas las carrocerías especificadas, se desactiva la unidad mecánica CNV1. De esta forma terminaría la ejecución del programa de robot.

Principal

Dentro de este procedimiento se realizan las llamadas secuenciales a todos los procedimientos que se emplean en este controlador para realizar la apertura y cierre de las puertas del lado izquierdo de las carrocerías. Entre las distintas llamadas a los procedimientos se ejecutan instrucciones de movimiento y de acción, sobre todo de espera y de activación/desactivación de señales, para coordinar todos los movimientos de los robots y que el paso de una trayectoria a otra se realice de forma correcta sin causar errores.

La secuencia de ejecución, explicada de forma abreviada, es la siguiente:

1. Se resetean algunas señales, se mueve el robot a la posición inicial y se espera que la carrocería entre en la **Ventana de inicio** del transportador, mediante la instrucción **WaitWObj**.
2. Se espera a que la señal digital de entrada **Abrir** se active. Esta señal se activará cuando la señal digital de salida **Abrir_P**, del controlador **Cont_Inf_Izq** se active, es decir, cuando el robot de este controlador se posicione en la posición de espera para pintar el interior de la puerta izquierda delantera, indicándole así al robot del presente controlador que puede iniciar la trayectoria para abrir la puerta delantera izquierda de la carrocería.
3. Se ejecuta el procedimiento **Abertura_Cierre_PI_Del**, mediante el cual se realiza la apertura de la puerta izquierda delantera, se mantiene la puerta abierta mientras el robot del controlador **Cont_Inf_Izq** pinta su interior, y se cierra la misma.
4. Tras esto se espera a que la señal digital de entrada **Abrir** se active de nuevo. Esta señal se activará de nuevo cuando la señal digital de salida **Abrir_P**, del controlador **Cont_Inf_Izq** se active otra vez, es decir, cuando el robot de este controlador se posicione en la posición de espera para pintar el interior de la puerta izquierda trasera, indicándole así al robot del presente controlador que puede iniciar la trayectoria para abrir la puerta trasera izquierda de la carrocería.
5. Se ejecuta el procedimiento **Abertura_Cierre_PI_Tras**, mediante el cual se realiza la apertura de la puerta izquierda trasera, se mantiene la puerta abierta mientras el robot del controlador **Cont_Inf_Izq** pinta su interior, y se cierra la misma.
6. Tras esto se espera que la señal digital de entrada **Avance** se active. Esta señal se activará cuando la señal digital de salida **Avance_Inf_P**, del controlador **Cont_Inf_Izq** se active, es decir, cuando el robot de ese controlador está en una posición que permite que el robot del presente controlador avance hasta posicionarse en una posición donde

no dificulta el resto de procedimientos de pintado del robot del controlador **Cont_Inf_Izq**.

7. Se traslada el robot a la posición que se ha mencionado.
8. Se espera a que se active la señal digital de entrada **Volver**. Esta señal se activará cuando la señal digital de salida **Volver**, del controlador **Cont_Inf_Izq** se active, es decir, cuando se ha finalizado el procedimiento **Principal** de ese controlador.
9. Se traslada el robot a su posición inicial.
10. Se deja de seguir a la carrocería del transportador, es decir, se suelta el objeto de trabajo **wobj_cnv1** mediante la instrucción **DropWObj**.

Mantenimiento Gripper

Este procedimiento se emplea para realizar el mantenimiento o la limpieza manual de la herramienta de agarre de los robots de asistencia IRB 5350 desde el área de mantenimiento, lo que permite no tener que acceder al interior de la estación para llevar a cabo estas tareas.

El robot ejecutará una serie de instrucciones de movimiento para posicionarse delante de la trampilla de mantenimiento. En ese momento activa la señal digital de salida **Open_TMant**, es decir cambia su valor a 1, que está conectada al componente inteligente **PoseMover [Open]**, que al activarse mueve la trampilla de mantenimiento hasta dejar el hueco completamente abierto.

Una vez abierta, se fija la velocidad máxima del TCP del robot en 100 mm/s de forma que si, en caso de fallo, chocase con el operario no le causase un problema grave. El robot introduce la herramienta a través del hueco de mantenimiento a esa velocidad y espera hasta que la señal digital de entrada **Mantenimiento** se desactive. Al mismo tiempo el valor de la señal analógica de salida **Señal_Luminosa** cambia para indicar si el robot está en movimiento o parado.

Cuando se termina el mantenimiento y se resetea la señal **Mantenimiento**, el robot saca la herramienta a través del hueco, se resetea la señal **Open_TMant** y se activa la señal **Close_TMant**, conectada al componente inteligente **Pose_Mover[HomePose]**, que cierra la trampilla.

Una vez tiene la herramienta fuera de la trampilla, la velocidad del robot vuelve a fijarse en la máxima velocidad que puede alcanzar el modelo de robot, para ya no estar limitado, y se ejecutan una serie de instrucciones de movimiento para alejarse.

Este proceso será igual para todos los controladores de robots de asistencia, variando simplemente los objetivos de las instrucciones de acercamiento o alejamiento a la zona de mantenimiento.

Procedimientos de asistencia

Abertura Cierre PI Del

Este procedimiento de asistencia está compuesto en su totalidad por instrucciones de movimiento lineal y circular (**MoveL** y **MoveC**). Dentro de este procedimiento se activará la señal digital de salida **Abrir_Puerta_Del**, para indicarle al componente inteligente **Abertura_PIs** que puede rotar la puerta delantera izquierda 60° para abrirla.

Una vez se ha realizado la trayectoria para abrir la puerta se activa la señal digital de salida **Int_Pint_P**, mediante la cual se le indica al robot del controlador **Cont_Inf_Izq** que puede iniciar el pintado del interior de la puerta.

A continuación el robot avanza en el eje X, pero manteniendo la posición en el eje Y y Z, es decir, los ejes del robot se quedan fijos y el robot como conjunto se mueve mediante el track a la misma velocidad que el transportador, de modo que mantiene la puerta abierta para que el robot del controlador **Cont_Inf_Izq** realce el pintado interior de la misma.

Cuando ese robot finaliza el pintado se activa la señal digital de salida **Cerrar_Puerta_Del**, para indicarle al componente inteligente **Abertura_PIs** que puede rotar la puerta delantera izquierda 60° para cerrarla.

Este procedimiento, mostrado en la Figura 4.101, se emplea para realizar la trayectoria mediante la cual se realiza la abertura y cierre de la puerta delantera izquierda, para posibilitar el pintado de su superficie interior y de la zona de chasis que se encuentra bajo la misma.

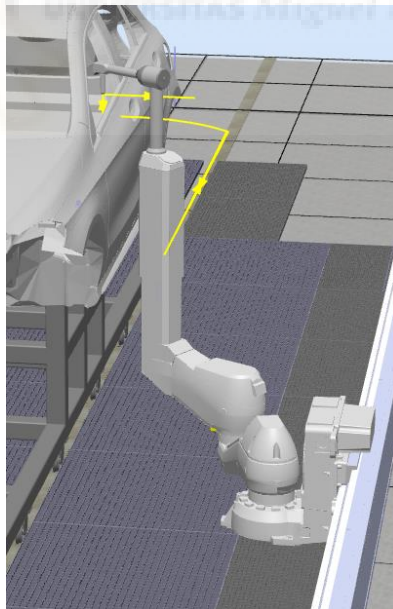


Figura 4.101 - Procedimiento Abertura_Cierre_PI_Del - Cont_Puertas_Izq

Para programar esta trayectoria se emplea como guía una curva previamente diseñada y posicionada a la altura adecuada. Al programar los objetivos sobre esta curva se consigue un movimiento de abertura circular cuyo centro se encuentra en las bisagras de la puerta. Esta curva se muestra en la siguiente Figura 102.

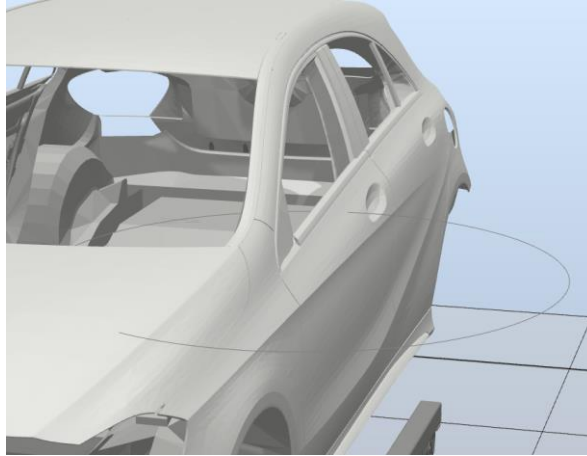


Figura 4.102 - Curva auxiliar para la programación de la trayectoria de abertura y cierre de la puerta delantera izquierda

Abertura Cierre PI Tras

Este procedimiento de asistencia está compuesto en su totalidad por instrucciones de movimiento lineal y circular (**MoveL** y **MoveC**). Dentro de este procedimiento se activará la señal digital de salida **Abrir_Puerta_Tras**, para indicarle al componente inteligente **Abertura_PIs** que puede rotar la puerta trasera izquierda 60° para abrirla.

Una vez se ha realizado la trayectoria para abrir la puerta se activa la señal digital de salida **Int_Pint_P**, mediante la cual se le indica al robot del controlador **Cont_Inf_Izq** que puede iniciar el pintado del interior de la puerta.

A continuación el robot avanza en el eje X, pero manteniendo la posición en el eje Y y Z, es decir, los ejes del robot se quedan fijos y el robot como conjunto se mueve mediante el track a la misma velocidad que el transportador, de modo que mantiene la puerta abierta para que el robot del controlador **Cont_Inf_Izq** realce el pintado interior de la misma.

Cuando ese robot finaliza el pintado se activa la señal digital de salida **Cerrar_Puerta_Tras**, para indicarle al componente inteligente **Abertura_PIs** que puede rotar la puerta trasera izquierda 60° para cerrarla.

Este procedimiento, mostrado en la Figura 4.103, se emplea para realizar la trayectoria mediante la cual se realiza la abertura y cierre de la puerta trasera izquierda, para posibilitar el pintado de su superficie interior y de la zona de chasis que se encuentra bajo la misma.

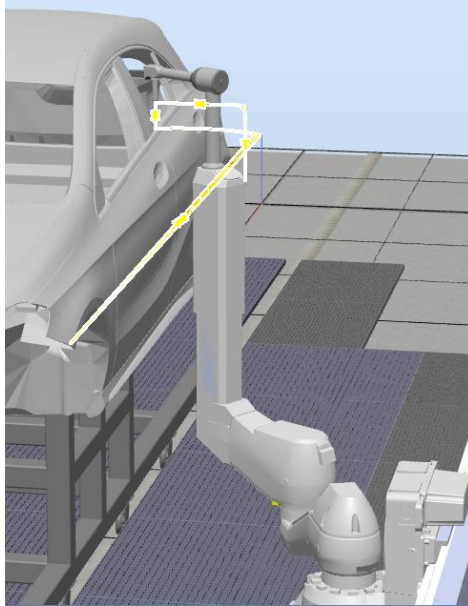


Figura 4.103 - Procedimiento Abertura_Cierre_PI_Tras - Cont_Puertas_Izq

Para programar esta trayectoria se emplea como guía una curva previamente diseñada y posicionada a la altura adecuada. Al programar los objetivos sobre esta curva se consigue un movimiento de abertura circular cuyo centro se encuentra en las bisagras de la puerta. Esta curva se muestra en la siguiente Figura 104.



Figura 4.104 - Curva auxiliar para la programación de la trayectoria de abertura y cierre de la puerta trasera izquierda

Controlador virtual Cont Puertas Der

El código completo del programa de robot de este controlador se encuentra en el apartado [8.1.6 Controlador virtual Cont Puertas Der](#). Debido a su tamaño no se volverá a mostrar en este apartado.

En este controlador solo se tienen las variables globales **Paro_Robot**, **N_Proc** y **Contador**, ya expuestas en el controlador **Cont_Inf_Izq**, por lo que no se volverán a exponer.

Procedimientos de ejecución

main (Punto de entrada de la simulación)

Este es el procedimiento en el que se inicia la ejecución del programa de robot. Desde él se debe llamar a la ejecución del resto de procedimientos que forman el programa.

Este procedimiento realiza las mismas tareas en el mismo orden que el procedimiento **main** del controlador **Cont_Puertas_Izq**, expuesto anteriormente. Simplemente varía, como es lógico, la posición de los objetivos de las instrucciones de movimiento y algunas señales de entrada y salida, pero las tareas principales y su secuencia es la misma.

Principal

Dentro de este procedimiento se realizan las llamadas secuenciales a todos los procedimientos que se emplean en este controlador para realizar la abertura y cierre de las puertas del lado derecho de las carrocerías. Entre las distintas llamadas a los procedimientos se ejecutan instrucciones de movimiento y de acción, sobre todo de espera y de activación/desactivación de señales, para coordinar todos los movimientos de los robots y que el paso de una trayectoria a otra se realice de forma correcta sin causar errores.

La secuencia de ejecución, explicada de forma abreviada, es la siguiente:

1. Se resetean algunas señales, se mueve el robot a la posición inicial y se espera que la carrocería entre en la **Ventana de inicio** del transportador, mediante la instrucción **WaitWObj**.
2. Se espera a que la señal digital de entrada **Abrir** se active. Esta señal se activará cuando la señal digital de salida **Abrir_P**, del controlador **Cont_Inf_Der** se active, es decir, cuando el robot de este controlador se posicione en la posición de espera para pintar el interior de la puerta derecha delantera, indicándole así al robot del presente controlador que puede iniciar la trayectoria para abrir la puerta delantera derecha de la carrocería.
3. Se ejecuta el procedimiento **Abertura_Cierre_PD_Del**, mediante el cual se realiza la abertura de la puerta derecha delantera, se mantiene la puerta abierta mientras el robot del controlador **Cont_Inf_Der** pinta su interior, y se cierra la misma.

4. Tras esto se espera a que la señal digital de entrada **Abrir** se active de nuevo. Esta señal se activará de nuevo cuando la señal digital de salida **Abrir_P**, del controlador **Cont_Inf_Der** se active otra vez, es decir, cuando el robot de este controlador se posicione en la posición de espera para pintar el interior de la puerta derecha trasera, indicándole así al robot del presente controlador que puede iniciar la trayectoria para abrir la puerta trasera derecha de la carrocería.
5. Se ejecuta el procedimiento **Abertura_Cierre_PD_Tras**, mediante el cual se realiza la abertura de la puerta derecha trasera, se mantiene la puerta abierta mientras el robot del controlador **Cont_Inf_Der** pinta su interior, y se cierra la misma.
6. Tras esto se espera que la señal digital de entrada **Avance** se active. Esta señal se activará cuando la señal digital de salida **Avance_Inf_P**, del controlador **Cont_Inf_Der** se active, es decir, cuando el robot de ese controlador está en una posición que permite que el robot del presente controlador avance hasta posicionarse en una posición donde no dificulta el resto de procedimientos de pintado del robot del controlador **Cont_Inf_Der**.
7. Se traslada el robot a la posición que se ha mencionado.
8. Se espera a que se active la señal digital de entrada **Volver**. Esta señal se activará cuando la señal digital de salida **Volver**, del controlador **Cont_Inf_Der** se active, es decir, cuando se ha finalizado el procedimiento **Principal** de ese controlador.
9. Se traslada el robot a su posición inicial.
10. Se deja de seguir a la carrocería del transportador, es decir, se suelta el objeto de trabajo **wobj_cnv1** mediante la instrucción **DropWObj**.

Mantenimiento Gripper

Este procedimiento realiza las mismas tareas en el mismo orden que el procedimiento **Mantenimiento_Gripper** del controlador **Cont_Puertas_Izq**, expuesto anteriormente. Simplemente varía, como es lógico, la posición de los objetivos de las instrucciones de movimiento para realizar el acercamiento a la trampilla de mantenimiento, pero las tareas principales y su secuencia es la misma.

Procedimientos de asistencia

Abertura Cierre PD Del

Este procedimiento de asistencia está compuesto en su totalidad por instrucciones de movimiento lineal y circular (**MoveL** y **MoveC**). Dentro de este procedimiento se activará la señal digital de salida **Abrir_Puerta_Del**, para indicarle al componente inteligente **Abertura_PD** que puede rotar la puerta delantera derecha 60° para abrirla.

Una vez se ha realizado la trayectoria para abrir la puerta se activa la señal digital de salida **Int_Pint_P**, mediante la cual se le indica al robot del controlador **Cont_Inf_Der** que puede iniciar el pintado del interior de la puerta.

A continuación el robot avanza en el eje X, pero manteniendo la posición en el eje Y y Z, es decir, los ejes del robot se quedan fijos y el robot como conjunto se mueve mediante el track a la misma velocidad que el transportador, de modo que mantiene la puerta abierta para que el robot del controlador **Cont_Inf_Der** realce el pintado interior de la misma.

Cuando ese robot finaliza el pintado se activa la señal digital de salida **Cerrar_Puerta_Del**, para indicarle al componente inteligente **Abertura_PDs** que puede rotar la puerta delantera derecha 60° para cerrarla.

Este procedimiento, mostrado en la Figura 4.105, se emplea para realizar la trayectoria mediante la cual se realiza la abertura y cierre de la puerta delantera derecha, para posibilitar el pintado de su superficie interior y de la zona de chasis que se encuentra bajo la misma.

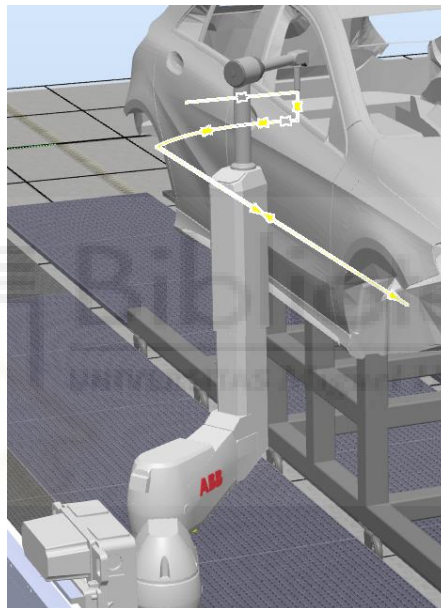


Figura 4.105 - Procedimiento Abertura_Cierre_PD_Del - Cont_Puertas_Der

Para programar esta trayectoria se emplea como guía una curva previamente diseñada y posicionada a la altura adecuada. Al programar los objetivos sobre esta curva se consigue un movimiento de abertura circular cuyo centro se encuentra en las bisagras de la puerta. Esta curva se muestra en la siguiente Figura 106.

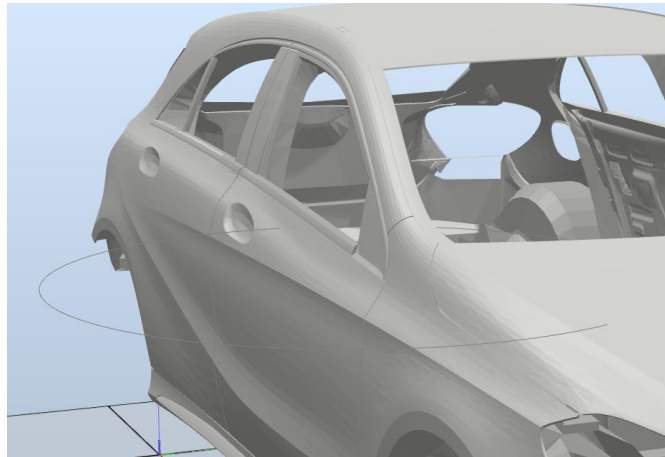


Figura 4.106 - Curva auxiliar para la programación de la trayectoria de abertura y cierre de la puerta delantera derecha

Abertura Cierre PD Tras

Este procedimiento de asistencia está compuesto en su totalidad por instrucciones de movimiento lineal y circular (**MoveL** y **MoveC**). Dentro de este procedimiento se activará la señal digital de salida **Abrir_Puerta_Tras**, para indicarle al componente inteligente **Abertura_PD**s que puede rotar la puerta trasera derecha 60° para abrirla.

Una vez se ha realizado la trayectoria para abrir la puerta se activa la señal digital de salida **Int_Pint_P**, mediante la cual se le indica al robot del controlador **Cont_Inf_Der** que puede iniciar el pintado del interior de la puerta.

A continuación el robot avanza en el eje X, pero manteniendo la posición en el eje Y y Z, es decir, los ejes del robot se quedan fijos y el robot como conjunto se mueve mediante el track a la misma velocidad que el transportador, de modo que mantiene la puerta abierta para que el robot del controlador **Cont_Inf_Der** realce el pintado interior de la misma.

Cuando ese robot finaliza el pintado se activa la señal digital de salida **Cerrar_Puerta_Tras**, para indicarle al componente inteligente **Abertura_PD**s que puede rotar la puerta trasera derecha 60° para cerrarla.

Este procedimiento, mostrado en la Figura 4.107, se emplea para realizar la trayectoria mediante la cual se realiza la abertura y cierre de la puerta trasera derecha, para posibilitar el pintado de su superficie interior y de la zona de chasis que se encuentra bajo la misma.

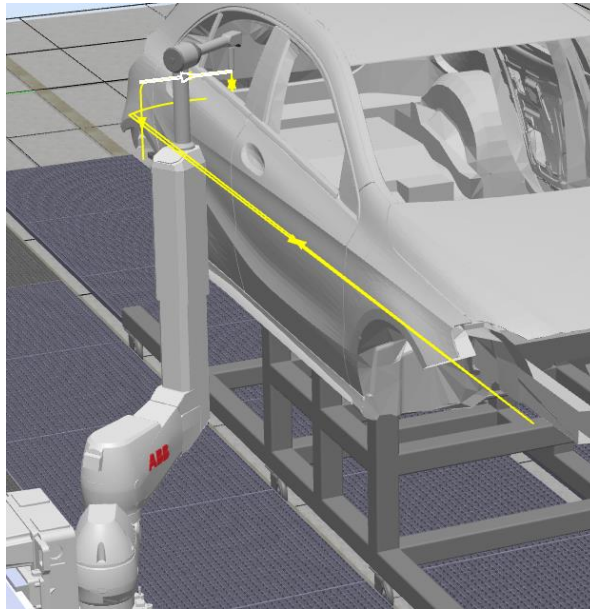


Figura 4.107 - Procedimiento Abertura_Cierre_PD_Tras - Cont_Puertas_Der

Para programar esta trayectoria se emplea como guía una curva previamente diseñada y posicionada a la altura adecuada. Al programar los objetivos sobre esta curva se consigue un movimiento de abertura circular cuyo centro se encuentra en las bisagras de la puerta. Esta curva se muestra en la siguiente Figura 108.

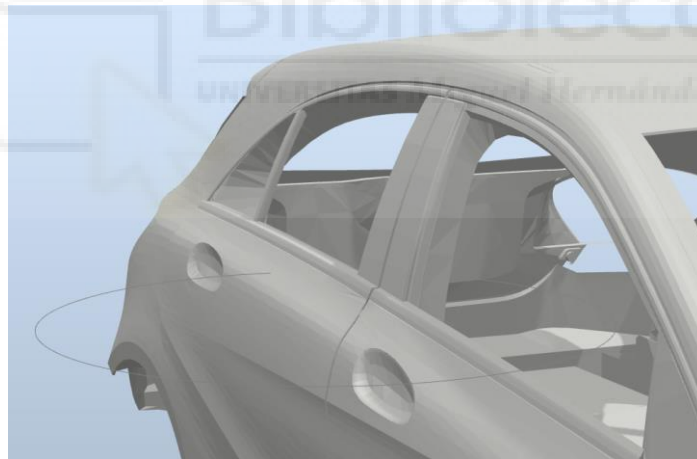


Figura 4.108 - Curva auxiliar para la programación de la trayectoria de abertura y cierre de la puerta trasera derecha

4.3.7.3 Rutinas TRAP

Como puede observarse en la Tabla 4.3, expuesta anteriormente, las rutinas TRAP **Parada**, **Limpieza_Atomizador** y **Mantenimiento_Atomizador** son comunes a los cuatro controladores de robots de aplicación de pintura.

Estas rutinas tienen la misma estructura de programación y realizan las mismas tareas en los cuatro controladores, variando, como es lógico, la posición de los objetivos de las instrucciones de movimiento en cada controlador y variando ligeramente su longitud en función de los procedimientos que contenga ese controlador. Por estos motivos se expondrá una única vez cada una de estas tres rutinas TRAP.

El controlador **Cont_Sup_Izq** también tiene la rutina TRAP **Paro_Puerta**, que también se expondrá.

En los dos controladores de los robots de asistencia IRB5350 también se tiene la rutina TRAP **Parada**, pero difiere en algunos aspectos con la programada para los robots de aplicación, por lo que expondrá a parte.

Rutina TRAP Parada

La ejecución de esta rutina TRAP se activa cuando la señal digital de entrada **Paro_Propio** se activa. Esta rutina TRAP se emplea para que cuando otro controlador lo solicite, es decir, cuando una señal de otro controlador cambie a 1 el valor de la señal digital de entrada **Paro_Propio**, el robot se detenga en el punto donde se encuentre, almacene esa posición, realice una trayectoria de salida que le permita moverse desde el punto de paro hasta una posición de reposo, sin colisionar con ningún elemento, espere a que la señal **Paro_Propio** se desactive, vuelva al punto donde se paró, sin colisionar, y retome la trayectoria que estaba ejecutando cuando se detuvo.

Esta rutina realiza las siguientes tareas de forma secuencial.

1. Se crean las variables locales de la rutina VAR robtarg **Punt_Paro** y VAR jointtarg **jointPunt_Paro**. Posteriormente en la variable **Punt_Paro** se almacenará la ubicación del punto de paro como un dato robtarg, es decir, con la posición y orientación del TCP, la configuración de ejes del robot y la posición del track. En la variable **jointPunt_Paro** se almacena también la ubicación del punto de paro pero como dato jointtarg, es decir, la posición de los ejes del robot y la posición del track.
2. Se cambia el valor de la señal analógica de salida **Señal_Luminosa** a 1, para cambiar el color de las señales luminosas a ámbar.
3. Se detiene el movimiento del robot lo más rápido posible.
4. Se almacena la trayectoria de movimiento que se estaba ejecutando. De esta forma se puede iniciar un nuevo movimiento temporal y reiniciar después el movimiento original que se ha almacenado.
5. Se almacena en la variable **Punt_Paro** la ubicación del punto de paro, es decir, el punto donde se ha detenido el robot como un dato robtarg.
6. Se almacena también en la variable **jointPunt_Paro** la ubicación del punto de paro pero en este caso como dato jointtarg.
7. Se analiza el valor de la variable **N_Proc** mediante una instrucción **TEST**, para según su valor ejecutar una trayectoria de salida u otra. Es decir, en función del procedimiento, y

el tramo dentro del procedimiento en algunos casos, donde se detuvo el robot se debe realizar una trayectoria de salida u otra para así evitar que el robot colisiones con la carrocería.

Dentro de cada uno de los procedimientos del programa de robot se fija un valor diferente para la variable **N_Proc**, de modo que puede saberse en cada instante en que procedimiento se encuentra el programa de robot. En algunos procedimientos dentro un único procedimiento se tienen varios valores para la variable **N_Proc**, en función de la posición del robot.

Dentro de cada uno de los casos de TEST, se programa una trayectoria de salida desde el punto donde se ha detenido el robot hasta una posición intermedia de reposo, en la que el robot se posicione de forma que no entorpezca el paso de un operario si se requiriera ni pueda colisionar con un robot de la estación en movimiento. Lógicamente el robot debe realizar la trayectoria de salida sin colisionar con la carrocería u otro elemento.

Una vez se alcanza la posición de reposo se cambia el valor de la señal analógica de salida **Señal_Luminosa** a 2, para cambiar el color de las señales luminosas a verde, y se espera hasta que la señal digital de entrada **Paro_Propio** se desactive.

Cuando se desactiva la señal **Paro_Propio**, el robot realiza la trayectoria de salida en sentido contrario hasta posicionarse de nuevo en el punto de paro donde se detuvo.

8. Se activa la señal digital de salida **Pos_Parada**, que indica que el robot está situado de nuevo en el punto de paro y listo para retomar la trayectoria donde se detuvo.
9. Se espera hasta que la señal digital de entrada **OK_PosP** se active. Esta señal se activa cuando todos los robots de la estación están situados de nuevo en sus respectivos puntos de paro, es decir, cuando las señales digitales de salida **Pos_Parada** de todos los controladores de robot se han activado. Esto permite que el movimiento de todos los robots y del transportador se reanude al mismo tiempo, posibilitando así que puedan continuar con las trayectorias donde se detuvieron.
10. Se restablece la trayectoria en la que se paró el robot y que previamente se había almacenado.
11. Se reanuda el movimiento del robot en esa trayectoria almacenada, que al inicio de la rutina se había detenido.
12. Se espera un tiempo de 0.25 segundos para dar un pequeño margen a la coordinación de la activación de todos los robots. Pasado ese tiempo el robot continuará con la ejecución normal del programa de robot desde la instrucción de la trayectoria donde se detuvo.
13. Se resetea la señal **Pos_Parada**.
14. Se cambia el valor de la señal analógica de salida **Señal_Luminosa** a 3, para cambiar el color de las señales luminosas a rojo.

Rutina TRAP Limpieza Atomizador

La ejecución de esta rutina TRAP se activa cuando la señal digital de entrada **Limpieza_Atom** se activa. Esta rutina TRAP se emplea en el caso de que sea necesaria la limpieza automática del atomizador durante la ejecución del programa de robot. Es decir, si durante el proceso de pintado se detecta que el atomizador no está pintando de forma adecuada, debido por ejemplo a un atasco en el suministro de pintura en el atomizador, mediante esta rutina el robot del controlador en el que se ejecuta detiene su movimiento en el punto donde se encuentra, almacena esa posición, realiza una trayectoria de salida que le permita moverse desde el punto de paro hasta una posición intermedia cercana al limpiador del atomizador, ejecuta el procedimiento de limpieza **Limpieza_Atom_Color**, vuelve a la posición intermedia, espera a que la señal **Limpieza_Atom** se desactive, vuelva al punto donde se detuvo, sin colisionar, y retoma la trayectoria que estaba ejecutando cuando se detuvo.

Al mismo tiempo en el resto de robots de la estación se ejecuta la rutina TRAP **Parada**, expuesta anteriormente, para pausar la ejecución de sus programas de robot y moverlos a una posición de reposo.

Esta rutina realiza las siguientes tareas de forma secuencial.

1. Se crean las variables locales de la rutina VAR robtargt **Punt_Paro**, VAR jointtargt **jointPunt_Paro** y VAR string **N_PROC2**.
Posteriormente en la variable **Punt_Paro** se almacenará la ubicación del punto de paro como un dato robtargt, es decir, con la posición y orientación del TCP, la configuración de ejes del robot y la posición del track.
En la variable **jointPunt_Paro** se almacena también la ubicación del punto de paro pero como dato jointtargt, es decir, la posición de los ejes del robot y la posición del track.
En la variable **N_PROC2** se almacena el valor de la variable **N_Proc** en el momento del paro, ya que dentro de la rutina TRAP se ejecutará el procedimiento **Limpieza_Atom_Color**, lo cual hará que el valor de **N_Proc** cambie. Al terminar la rutina TRAP se quiere que el valor de la variable **N_Proc** sea el que tenía en el momento de paro, ya que al retomar el movimiento en la trayectoria donde se detuvo debe tener el valor de esa trayectoria.
2. Se detiene el movimiento del robot lo más rápido posible.
3. Se activa la señal digital de salida **Paro_Resto**, mediante la cual se activará en el resto de robots la rutina TRAP **Parada**, expuesta anteriormente.
4. Se activa la señal digital de salida **Stop_Conveyor**, mediante la cual se detiene el movimiento del transportador.
5. Se cambia el valor de la señal analógica de salida **Señal_Luminosa** a 1, para cambiar el color de las señales luminosas a ámbar.
6. Se almacena la trayectoria de movimiento que se estaba ejecutando. De esta forma se puede iniciar un nuevo movimiento temporal y reiniciar después el movimiento original que se ha almacenado.
7. El valor de la variable global **Limpieza** se fija en "**Limp2**".
8. Se almacena en la variable **Punt_Paro** la ubicación del punto de paro, es decir, el punto donde se ha detenido el robot como un dato robtargt.
9. Se almacena también en la variable **jointPunt_Paro** la ubicación del punto de paro pero en este caso como dato jointtargt.

10. Se analiza el valor de la variable **N_Proc** mediante una instrucción **TEST**, para según su valor ejecutar una trayectoria de salida u otra. Es decir, en función del procedimiento, y el tramo dentro del procedimiento en algunos casos, donde se detuvo el robot se debe realizar una trayectoria de salida u otra para así evitar que el robot colisiones con la carrocería.

Dentro de cada uno de los procedimientos del programa de robot se fija un valor diferente para la variable **N_Proc**, de modo que puede saberse en cada instante en que procedimiento se encuentra el programa de robot. En algunos procedimientos dentro un único procedimiento se tienen varios valores para la variable **N_Proc**, en función de la posición del robot.

Dentro de cada uno de los casos de **TEST**, se programa una trayectoria de salida desde el punto donde se ha detenido el robot hasta una posición intermedia cercana al limpiador del atomizador. Lógicamente el robot debe realizar la trayectoria de salida sin colisionar con la carrocería u otro elemento.

Una vez se alcanza la posición intermedia se ejecuta el procedimiento **Limpieza_Atom_Color**, que ejecuta la limpieza del atomizador en función del valor de la variable **Limpieza**.

Tras esto el robot vuelve a la posición intermedia y espera a que la señal digital de entrada **Limpieza_Atom** se desactive. Al mismo tiempo se cambia el valor de la señal analógica de salida **Señal_Luminosa** a 2, para cambiar el color de las señales luminosas a verde.

Cuando se desactiva la señal **Limpieza_Atom**, el robot realiza la trayectoria de salida en sentido contrario hasta posicionarse de nuevo en el punto de paro donde se detuvo. Al mismo tiempo se desactiva la señal de salida digital **Paro_Resto**, para que el resto de robots de la estación retomen el movimiento.

11. Se fija el valor de la variable **N_Proc** igual al de la variable **N_PROC2**, es decir, se devuelve a la variable **N_Proc** el valor que tenía cuando se detuvo el robot.
12. Se activa la señal digital de salida **Pos_Parada**, que indica que el robot está situado de nuevo en el punto de paro y listo para retomar la trayectoria donde se detuvo.
13. Se espera hasta que la señal digital de entrada **OK_PosP** se active. Esta señal se activa cuando todos los robots de la estación están situados de nuevo en sus respectivos puntos de paro, es decir, cuando las señales digitales de salida **Pos_Parada** de todos los controladores de robot se han activado. Esto permite que el movimiento de todos los robots y del transportador se reanude al mismo tiempo, posibilitando así que puedan continuar con las trayectorias donde se detuvieron.
14. Se restablece la trayectoria en la que se paró el robot y que previamente se había almacenado.
15. Se reanuda el movimiento del robot en esa trayectoria almacenada, que al inicio de la rutina se había detenido.
16. Se espera un tiempo de 0.25 segundos para dar un pequeño margen a la coordinación de la activación de todos los robots. Pasado ese tiempo el robot continuará con la ejecución normal del programa de robot desde la instrucción de la trayectoria donde se detuvo.
17. Se resetea la señal digital de salida **Stop_Conveyor**, lo cual hace que el movimiento del transportador se reanude.
18. Se resetea la señal **Pos_Parada**.
19. Se cambia el valor de la señal analógica de salida **Señal_Luminosa** a 3, para cambiar el color de las señales luminosas a rojo.

Rutina TRAP Mantenimiento Atomizador

La ejecución de esta rutina TRAP se activa cuando la señal digital de entrada **Mantenimiento** se activa. Esta rutina TRAP se emplea en el caso de que sea necesario realizar un proceso de mantenimiento manual o una limpieza manual del atomizador del robot del controlador en el que se activa. Es decir, si durante el proceso de pintado se detecta que el atomizador no está pintando o actuando de forma adecuada, debido a algún problema técnico, mediante esta rutina el robot del controlador en el que se ejecuta detiene su movimiento en el punto donde se encuentra, almacena esa posición, realiza una trayectoria de salida que le permita moverse desde el punto de paro hasta una posición intermedia cercana al limpiador del atomizador, ejecuta el procedimiento de limpieza **Limpieza_Atom_Color**, para dejar el atomizador limpio para su mantenimiento, se traslada a una posición intermedia orientada para dirigirse hacia la zona de mantenimiento, ejecuta el procedimiento de mantenimiento **Mantenimiento_Atom**, dentro del cual espera a que la señal **Mantenimiento** se desactive, vuelve a la posición intermedia cercana al limpiador del atomizador, vuelve al punto donde se detuvo, sin colisionar, y retoma la trayectoria que estaba ejecutando cuando se detuvo.

Al mismo tiempo en el resto de robots de la estación se ejecuta la rutina TRAP **Parada**, expuesta anteriormente, para pausar la ejecución de sus programas de robot y moverlos a una posición de reposo.

Esta rutina realiza las siguientes tareas de forma secuencial.

1. Se crean las variables locales de la rutina VAR robtargget **Punt_Paro**, VAR jointtargget **jointPunt_Paro** y VAR string **N_PROC2**.
Posteriormente en la variable **Punt_Paro** se almacenará la ubicación del punto de paro como un dato robtargget, es decir, con la posición y orientación del TCP, la configuración de ejes del robot y la posición del track.
En la variable **jointPunt_Paro** se almacena también la ubicación del punto de paro pero como dato jointtargget, es decir, la posición de los ejes del robot y la posición del track.
En la variable **N_PROC2** se almacena el valor de la variable **N_Proc** en el momento del paro, ya que dentro de la rutina TRAP se ejecutarán los procedimientos **Mantenimiento_Atom** y **Limpieza_Atom_Color**, lo cual hará que el valor de **N_Proc** cambie. Al terminar la rutina TRAP se quiere que el valor de la variable **N_Proc** sea el que tenía en el momento de paro, ya que al retomar el movimiento en la trayectoria donde se detuvo debe tener el valor de esa trayectoria.
2. Se detiene el movimiento del robot lo más rápido posible.
3. Se activa la señal digital de salida **Paro_Resto**, mediante la cual se activará en el resto de robots la rutina TRAP **Parada**, expuesta anteriormente.
4. Se activa la señal digital de salida **Stop_Conveyor**, mediante la cual se detiene el movimiento del transportador.
5. Se cambia el valor de la señal analógica de salida **Señal_Luminosa** a 1, para cambiar el color de las señales luminosas a ámbar.
6. Se almacena la trayectoria de movimiento que se estaba ejecutando. De esta forma se puede iniciar un nuevo movimiento temporal y reiniciar después el movimiento original que se ha almacenado.
7. El valor de la variable global **Limpieza** se fija en **"Limp1"**.
8. Se almacena en la variable **Punt_Paro** la ubicación del punto de paro, es decir, el punto donde se ha detenido el robot como un dato robtargget.

9. Se almacena también en la variable **jointPunt_Paro** la ubicación del punto de paro pero en este caso como dato **jointtarget**.

10. Se analiza el valor de la variable **N_Proc** mediante una instrucción **TEST**, para según su valor ejecutar una trayectoria de salida u otra. Es decir, en función del procedimiento, y el tramo dentro del procedimiento en algunos casos, donde se detuvo el robot se debe realizar una trayectoria de salida u otra para así evitar que el robot colisiones con la carrocería.

Dentro de cada uno de los procedimientos del programa de robot se fija un valor diferente para la variable **N_Proc**, de modo que puede saberse en cada instante en que procedimiento se encuentra el programa de robot. En algunos procedimientos dentro un único procedimiento se tienen varios valores para la variable **N_Proc**, en función de la posición del robot.

Dentro de cada uno de los casos de **TEST**, se programa una trayectoria de salida desde el punto donde se ha detenido el robot hasta una posición intermedia cercana al limpiador del atomizador. Lógicamente el robot debe realizar la trayectoria de salida sin colisionar con la carrocería u otro elemento.

Una vez se alcanza la posición intermedia se ejecuta el procedimiento **Limpieza_Atom_Color**, que ejecuta la limpieza del atomizador en función del valor de la variable **Limpieza**.

Tras esto el robot se traslada a una posición intermedia orientada para dirigirse hacia la zona de mantenimiento.

Se ejecuta el procedimiento **Mantenimiento_Atom**, mediante el cual se permite realizar el mantenimiento o limpieza manual del atomizador desde la zona de mantenimiento, sin acceder al interior de la estación.

Una vez finalizado el mantenimiento, el robot vuelve a la posición intermedia cercana al limpiador del atomizador. Al mismo tiempo se desactiva la señal de salida digital **Paro_Resto**, para que el resto de robots de la estación retomen el movimiento.

El robot realiza la trayectoria de salida en sentido contrario hasta posicionarse de nuevo en el punto de paro donde se detuvo.

11. Se fija el valor de la variable **N_Proc** igual al de la variable **N_PROC2**, es decir, se devuelve a la variable **N_Proc** el valor que tenía cuando se detuvo el robot.

12. Se activa la señal digital de salida **Pos_Parada**, que indica que el robot está situado de nuevo en el punto de paro y listo para retomar la trayectoria donde se detuvo.

13. Se espera hasta que la señal digital de entrada **OK_PosP** se active. Esta señal se activa cuando todos los robots de la estación están situados de nuevo en sus respectivos puntos de paro, es decir, cuando las señales digitales de salida **Pos_Parada** de todos los controladores de robot se han activado. Esto permite que el movimiento de todos los robots y del transportador se reanude al mismo tiempo, posibilitando así que puedan continuar con las trayectorias donde se detuvieron.

14. Se restablece la trayectoria en la que se paró el robot y que previamente se había almacenado.

15. Se reanuda el movimiento del robot en esa trayectoria almacenada, que al inicio de la rutina se había detenido.

16. Se espera un tiempo de 0.25 segundos para dar un pequeño margen a la coordinación de la activación de todos los robots. Pasado ese tiempo el robot continuará con la ejecución normal del programa de robot desde la instrucción de la trayectoria donde se detuvo.

17. Se resetea la señal digital de salida **Stop_Conveyor**, lo cual hace que el movimiento del transportador se reanude.
18. Se resetea la señal **Pos_Parada**.
19. Se cambia el valor de la señal analógica de salida **Señal_Luminosa** a 3, para cambiar el color de las señales luminosas a rojo.

Rutina TRAP Paro Puerta

Esta rutina TRAP solo está programada en el controlador **Cont_Sup_Izq**. La ejecución de esta rutina TRAP se activa cuando la señal digital de entrada **P_Puerta** se activa. Esta señal se activa cuando una de las puertas de acceso a la estación se abre y corta uno de los sensores lineales que hay situados en la parte superior de las mismas. Estas puertas deberán estar cerradas para imposibilitar su abertura si no es con una llave u otro mecanismo, pero si por un error no se cierra y alguien intenta acceder al interior de la estación esta rutina sirve como medida de seguridad.

Si durante el proceso de pintado se detecta que una de las puertas de acceso a la estación se ha abierto, mediante esta rutina el robot del controlador **Cont_Sup_Izq** detiene su movimiento en el punto donde se encuentra, almacena esa posición, realiza una trayectoria de salida que le permita moverse desde el punto de paro hasta una posición intermedia de reposo, espera a que la señal **P_Puerta** se desactive, es decir, espera a que se cierre la puerta, vuelve al punto donde se detuvo, sin colisionar, y retoma la trayectoria que estaba ejecutando cuando se detuvo.

Al mismo tiempo en el resto de robots de la estación se ejecuta la rutina TRAP **Parada**, expuesta anteriormente, para pausar la ejecución de sus programas de robot y moverlos a una posición de reposo.

Esta rutina realiza las siguientes tareas de forma secuencial.

1. Se crean las variables locales de la rutina VAR robtargat **Punt_Paro** y VAR jointtargat **jointPunt_Paro**. Posteriormente en la variable **Punt_Paro** se almacenará la ubicación del punto de paro como un dato robtargat, es decir, con la posición y orientación del TCP, la configuración de ejes del robot y la posición del track. En la variable **jointPunt_Paro** se almacena también la ubicación del punto de paro pero como dato jointtargat, es decir, la posición de los ejes del robot y la posición del track.
2. Se detiene el movimiento del robot lo más rápido posible.
3. Se activa la señal digital de salida **Paro_Resto**, mediante la cual se activará en el resto de robots la rutina TRAP **Parada**, expuesta anteriormente.
4. Se activa la señal digital de salida **Stop_Conveyor**, mediante la cual se detiene el movimiento del transportador.
5. Se cambia el valor de la señal analógica de salida **Señal_Luminosa** a 1, para cambiar el color de las señales luminosas a ámbar.
6. Se almacena la trayectoria de movimiento que se estaba ejecutando. De esta forma se puede iniciar un nuevo movimiento temporal y reiniciar después el movimiento original que se ha almacenado.
7. Se almacena en la variable **Punt_Paro** la ubicación del punto de paro, es decir, el punto donde se ha detenido el robot como un dato robtargat.

8. Se almacena también en la variable **jointPunt_Paro** la ubicación del punto de paro pero en este caso como dato jointtarget.
9. Se analiza el valor de la variable **N_Proc** mediante una instrucción **TEST**, para según su valor ejecutar una trayectoria de salida u otra. Es decir, en función del procedimiento, y el tramo dentro del procedimiento en algunos casos, donde se detuvo el robot se debe realizar una trayectoria de salida u otra para así evitar que el robot colisiones con la carrocería.

Dentro de cada uno de los procedimientos del programa de robot se fija un valor diferente para la variable **N_Proc**, de modo que puede saberse en cada instante en que procedimiento se encuentra el programa de robot. En algunos procedimientos dentro un único procedimiento se tienen varios valores para la variable **N_Proc**, en función de la posición del robot.

Dentro de cada uno de los casos de **TEST**, se programa una trayectoria de salida desde el punto donde se ha detenido el robot hasta una posición intermedia de reposo, en la que el robot se posicione de forma que no entorpezca el paso de un operario si se requiriera. Lógicamente el robot debe realizar la trayectoria de salida sin colisionar con la carrocería u otro elemento.

Una vez se alcanza la posición de reposo se cambia el valor de la señal analógica de salida **Señal_Luminosa** a 2, para cambiar el color de las señales luminosas a verde, y se espera hasta que la señal digital de entrada **P_Puerta** se desactive.

Cuando se desactiva la señal **P_Puerta**, el robot realiza la trayectoria de salida en sentido contrario hasta posicionarse de nuevo en el punto de paro donde se detuvo.

Al mismo tiempo se desactiva la señal de salida digital **Paro_Resto**, para que el resto de robots de la estación retomen el movimiento.

10. Se activa la señal digital de salida **Pos_Parada**, que indica que el robot está situado de nuevo en el punto de paro y listo para retomar la trayectoria donde se detuvo.
11. Se espera hasta que la señal digital de entrada **OK_PosP** se active. Esta señal se activa cuando todos los robots de la estación están situados de nuevo en sus respectivos puntos de paro, es decir, cuando las señales digitales de salida **Pos_Parada** de todos los controladores de robot se han activado. Esto permite que el movimiento de todos los robots y del transportador se reanude al mismo tiempo, posibilitando así que puedan continuar con las trayectorias donde se detuvieron.
12. Se restablece la trayectoria en la que se paró el robot y que previamente se había almacenado.
13. Se reanuda el movimiento del robot en esa trayectoria almacenada, que al inicio de la rutina se había detenido.
14. Se espera un tiempo de 0.25 segundos para dar un pequeño margen a la coordinación de la activación de todos los robots. Pasado ese tiempo el robot continuará con la ejecución normal del programa de robot desde la instrucción de la trayectoria donde se detuvo.
15. Se resetea la señal digital de salida **Stop_Conveyor**, lo cual hace que el movimiento del transportador se reanude.
16. Se resetea la señal **Pos_Parada**.
17. Se cambia el valor de la señal analógica de salida **Señal_Luminosa** a 3, para cambiar el color de las señales luminosas a rojo.

Rutina TRAP Parada (Robots IRB5350)

Esta versión de la rutina TRAP **Parada** solo está programada en los controladores **Cont_Puertas_Izq** y **Cont_Puertas_Der**, ya que en estos controladores se requieren unas modificaciones en la programación de la rutina para que funcione correctamente.

La ejecución de esta rutina TRAP se activa cuando la señal digital de entrada **Paro_Propio** se activa. Esta rutina TRAP se emplea para que cuando otro controlador lo solicite, es decir, cuando una señal de otro controlador cambie a 1 el valor de la señal digital de entrada **Paro_Propio**, el robot se detenga en el punto donde se encuentre, almacene esa posición, realice una trayectoria de salida que le permita moverse desde el punto de paro hasta una posición de reposo, sin colisionar con ningún elemento, espere a que la señal **Paro_Propio** se desactive, vuelva al punto donde se paró, sin colisionar, y retome la trayectoria que estaba ejecutando cuando se detuvo.

Esta rutina realiza las siguientes tareas de forma secuencial.

1. Se crean las variables locales de la rutina VAR jointtarget **jointPunt_Paro**, VAR num **ZOffset** y VAR jointtarget **jointPunt_Paro_ZOffset**.
En la variable **jointPunt_Paro** se almacena la ubicación del punto de paro como dato jointtarget, es decir, la posición de los ejes del robot y la posición del track.
La variable **ZOffset** se calculará más tarde como la diferencia entre el rango máximo del tercer eje del robot, el eje lineal, y la posición de ese eje en el punto de paro.
La variable **jointPunt_Paro_ZOffset** se calculará posteriormente como una combinación de las variables **jointPunt_Paro** y **ZOffset**.
2. Se cambia el valor de la señal analógica de salida **Señal_Luminosa** a 1, para cambiar el color de las señales luminosas a ámbar.
3. Se detiene el movimiento del robot lo más rápido posible.
4. Se almacena la trayectoria de movimiento que se estaba ejecutando. De esta forma se puede iniciar un nuevo movimiento temporal y reiniciar después el movimiento original que se ha almacenado.
5. Se almacena también en la variable **jointPunt_Paro** la ubicación del punto de paro pero en este caso como dato jointtarget.
6. Se calcula el valor de la variable **ZOffset** como la diferencia entre el rango máximo del tercer eje del robot, el eje lineal, y la posición de ese eje en el punto de paro.
7. Se establece la variable **jointPunt_Paro_ZOffset** igual a la variable **jointPunt_Paro**.
8. Se establece el valor del tercer eje de la variable **jointPunt_Paro_ZOffset** igual a la variable **ZOffset**. De esta forma en algunos puntos de las trayectorias de apertura y cierre de las puertas se puede desplazar, mediante una instrucción de movimiento de ejes, el TCP de la herramienta una distancia de offset para que al salir el robot no choque con la puerta.
9. Se analiza el valor de la variable **N_Proc** mediante una instrucción **TEST**, para según su valor ejecutar una trayectoria de salida u otra. Es decir, en función del procedimiento, y el tramo dentro del procedimiento en algunos casos, donde se detuvo el robot se debe realizar una trayectoria de salida u otra para así evitar que el robot colisiones con la carrocería.
Dentro de cada uno de los procedimientos del programa de robot se fija un valor diferente para la variable **N_Proc**, de modo que puede saberse en cada instante en que procedimiento se encuentra el programa de robot. En algunos procedimientos dentro

un único procedimiento se tienen varios valores para la variable **N_Proc**, en función de la posición del robot.

Dentro de cada uno de los casos de TEST, se programa una trayectoria de salida desde el punto donde se ha detenido el robot hasta una posición intermedia de reposo, en la que el robot se posicione de forma que no entorpezca el paso de un operario si se requiriera ni pueda colisionar con un robot de la estación en movimiento. Lógicamente el robot debe realizar la trayectoria de salida sin colisionar con la carrocería u otro elemento.

Una vez se alcanza la posición de reposo se cambia el valor de la señal analógica de salida **Señal_Luminosa** a 2, para cambiar el color de las señales luminosas a verde, y se espera hasta que la señal digital de entrada **Paro_Propio** se desactive.

Cuando se desactiva la señal **Paro_Propio**, el robot realiza la trayectoria de salida en sentido contrario hasta posicionarse de nuevo en el punto de paro donde se detuvo.

10. Se activa la señal digital de salida **Pos_Parada**, que indica que el robot está situado de nuevo en el punto de paro y listo para retomar la trayectoria donde se detuvo.
11. Se espera hasta que la señal digital de entrada **OK_PosP** se active. Esta señal se activa cuando todos los robots de la estación están situados de nuevo en sus respectivos puntos de paro, es decir, cuando las señales digitales de salida **Pos_Parada** de todos los controladores de robot se han activado. Esto permite que el movimiento de todos los robots y del transportador se reanude al mismo tiempo, posibilitando así que puedan continuar con las trayectorias donde se detuvieron.
12. Se restablece la trayectoria en la que se paró el robot y que previamente se había almacenado.
13. Se reanuda el movimiento del robot en esa trayectoria almacenada, que al inicio de la rutina se había detenido.
14. Se espera un tiempo de 0.25 segundos para dar un pequeño margen a la coordinación de la activación de todos los robots. Pasado ese tiempo el robot continuará con la ejecución normal del programa de robot desde la instrucción de la trayectoria donde se detuvo.
15. Se resetea la señal **Pos_Parada**.
16. Se cambia el valor de la señal analógica de salida **Señal_Luminosa** a 3, para cambiar el color de las señales luminosas a rojo.

4.3.8 Desarrollo de la lógica de estación

De manera paralela a la creación de objetivos, procedimientos y rutinas, expuestas en el apartado anterior, se debe ir desarrollando la lógica de estación, de tal forma que se creen las conexiones y enlazamientos para hacer posible la correcta ejecución y simulación de los programas de robot, de modo que pueda comprobarse el correcto funcionamiento de lo programado en cada momento. Todos estos procesos se llevan a cabo según lo expuesto en los apartados [3.1.19 Lógica de estación](#) y [3.1.20 Componentes inteligentes](#).

Mediante la lógica de estación se consigue la comunicación entre los robots de la estación para llevar a cabo de forma coordinada la aplicación de la capa base sobre la carrocería. Además de entre los robots, también será necesaria la comunicación entre los robots y algunos componentes inteligentes creados para simular ciertos procesos y entre los propios componentes inteligentes. En la siguiente Figura 4.109 se muestra la lógica de estación completa desarrollada para el presente proyecto.

En este apartado se exponen todos los componentes inteligentes que forman la lógica de estación, agrupandolos según la tarea que desempeñan.

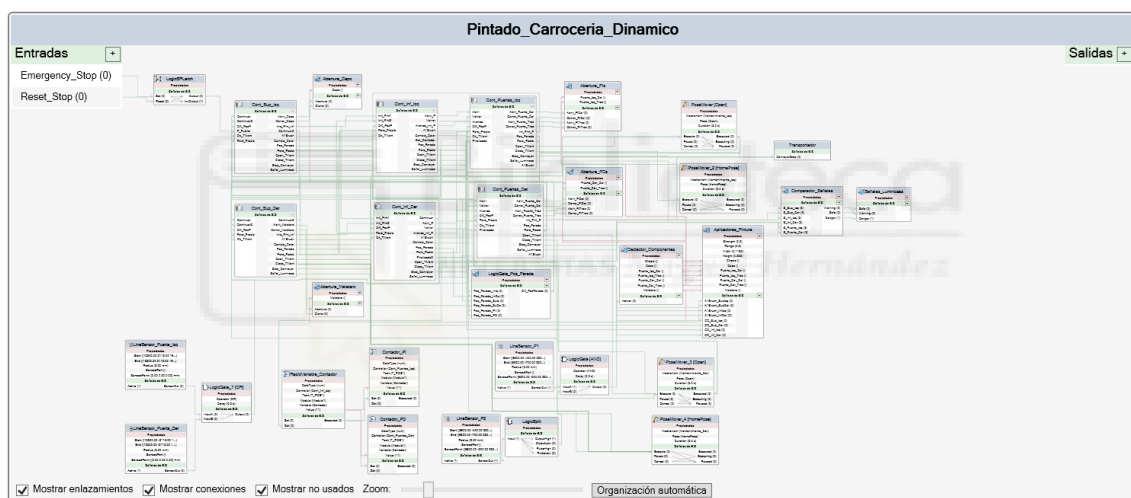


Figura 4.109 - Lógica de estación completa

Detector Componentes y auxiliares

Este conjunto de componentes inteligentes, mostrado en la Figura 4.110, que tiene como componente principal el componente inteligente creado **Detector_Componentes**, se emplea para detectar cada uno de los elementos que componen la carrocería (chasis, capo, puertas y maletero) que está en ese momento en el transportador y sobre la que se va a aplicar la capa base.

Estos componentes se detectan, se quedan almacenados en el componente **Detector_Componentes** y se envían a otros componentes inteligentes de la estación mediante enlazamiento de propiedad. En una estación real esto no sería necesario, ya que los

componentes inteligentes a los que se pasan los componentes detectados se emplean para simular procesos que en la estación real no serían simulados, como la abertura de las puertas o la simulación del patrón de pintado.

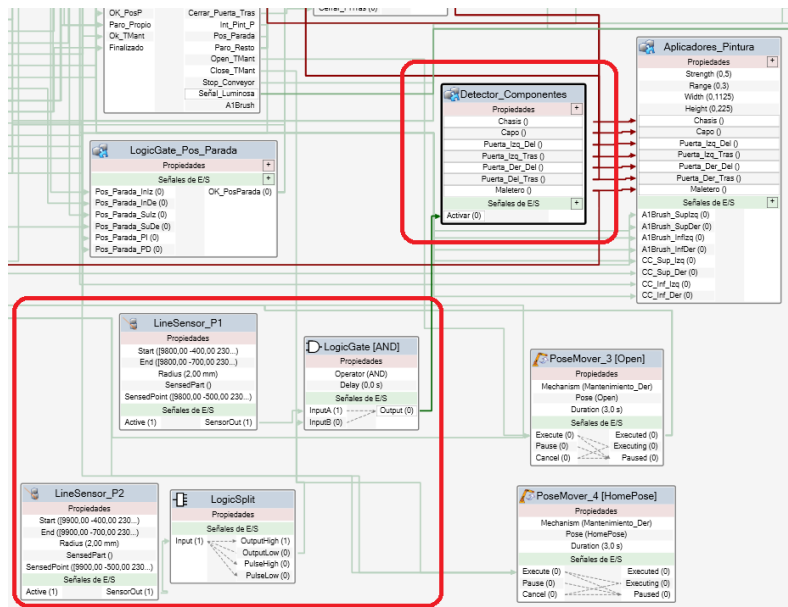


Figura 4.110 - Conjunto de componentes inteligentes para detectar los componentes de la carrocería

Todo el conjunto de detección de componentes está formado por los siguientes componentes:

- Detector_Componentes.
- LineSensor_P1.
- LineSensor_P2.
- LogicSplit.
- LogicGate [AND].

El funcionamiento de este conjunto de componentes es el siguiente. Se colocan los sensores lineales **LineSensor_P1** y **LineSensor_P2** en las posiciones [(9800,-400,300),(9800,-700,300)] y [(9900,-400,300),(9900,-700,300)] respectivamente, respecto del sistema de coordenadas mundo de RobotStudio, como se muestra en la Figura 4.111.

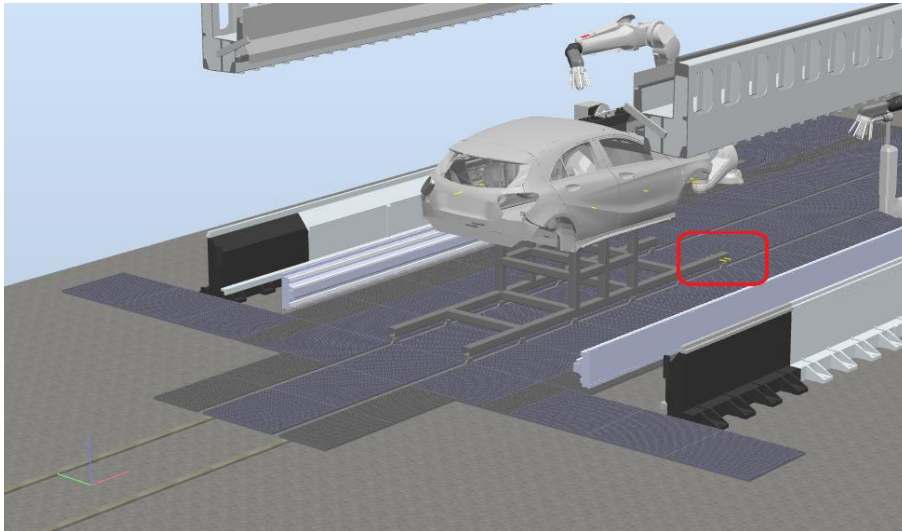


Figura 4.111 – Posición de los sensores lineales LineSensor_P1 y LineSensor_P2

Al avanzar la carrocería por el transportador el primero de los sensores activados será el **LineSensor_P1**. La salida de este sensor está conectada a la entrada **InputA** de la puerta lógica **LogicGate [AND]**. La salida del segundo sensor lineal, **LineSensor_P2**, está conectada a un **LogicSplit**. La salida **OutputLow** de este último, que invierte el valor de la salida del sensor, está conectada a la entrada **InputB** de la puerta lógica. De esta forma cuando el conjunto de la carrocería avanza e intersecta el primer sensor lineal, la salida del primer sensor se activa, mientras que la del segundo continúa a 0, haciendo que la salida **OutputLow** del **LogicSplit** sea 1 y activando así la puerta lógica AND, cuya salida está conectada a la entrada **Activar** del componente inteligente **Detector_Componentes**, mediante la cual se activa este componente.

El componente inteligente **Detector_Componentes** está formado a su vez por 7 sensores lineales, cuya señal de entrada de activación, **Active**, está conectada a la entrada **Activar**, es decir, cuando el componente **Detector_Componentes** se activa, se activan todos estos sensores lineales. Los componentes detectados por cada uno de estos sensores se pasan a una serie de propiedades creadas de tipo **Part**, donde se almacenan los identificadores de estos componentes. (Figura 4.112)

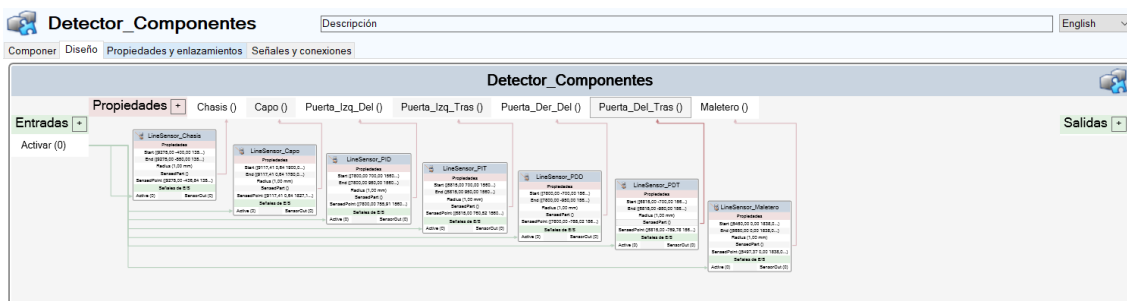


Figura 4.112 - Lógica del componente inteligente Detector_Componentes

Los 7 sensores se encuentran posicionados en la estación para que en el momento que se active el componente **Detector_Componentes**, es decir, cuando la carrocería intersecte el sensor lineal **LineSensor_P1**, se encuentren posicionados intersectando cada uno de los componentes de la carrocería, de modo que los detecten, como se muestra en la Figura 4.113.

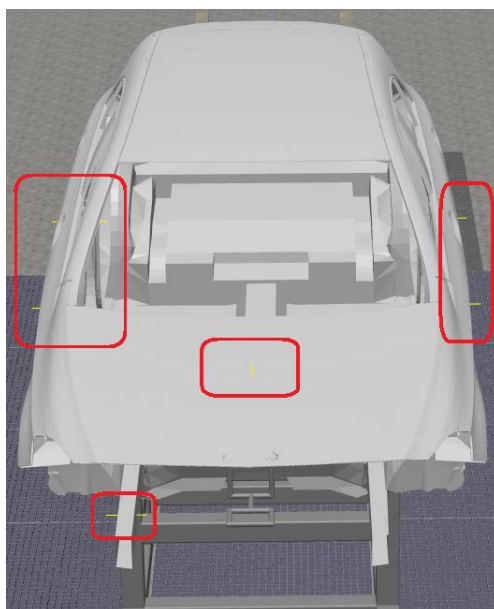


Figura 4.113 - Posicionamiento de los sensores que detectan los componentes de la carrocería

Cuando la carrocería intersecta el segundo de los sensores lineales, **LineSensor_P2**, su salida se activa, haciendo que el valor de la salida **OutputLow** del LogicSplit cambie a 0, desactivando la puerta lógica AND y por lo tanto el componente **Detector_Componentes**. Esto se hace para dejar fijados los componentes de la carrocería detectados hasta que la siguiente carrocería active de nuevo el componente **Detector_Componentes**. Si no se fijan en ese momento, como la carrocería sigue avanzado, los componentes detectados por cada sensor cambiarían, haciendo que los identificadores de componentes que se pasan a otros componentes inteligentes no sean los adecuados.

Todos los identificadores de los componentes detectados se pasan mediante varios enlazamientos de propiedades al componente inteligente creado **Aplicadores_Pintura**.

El identificador Capo se pasa mediante un enlazamiento de propiedad al componente inteligente creado **Abertura_Capo**.

El identificador Maletero se pasa mediante un enlazamiento de propiedad al componente inteligente creado **Abertura_Maletero**.

Los identificadores **Puerta_Izq_Del** y **Puerta_Izq_Tras** se pasan mediante dos enlazamientos de propiedades al componente inteligente creado **Abertura_PIs**.

Los identificadores **Puerta_Der_Del** y **Puerta_Der_Tras** se pasan mediante dos enlazamientos de propiedades al componente inteligente creado **Abertura_PDs**.

Abertura Capó

Este componente inteligente creado, mostrado en la Figura 4.114, se emplea para simular la abertura y cierre del capó de la carrocería mientras está en movimiento en el transportador.

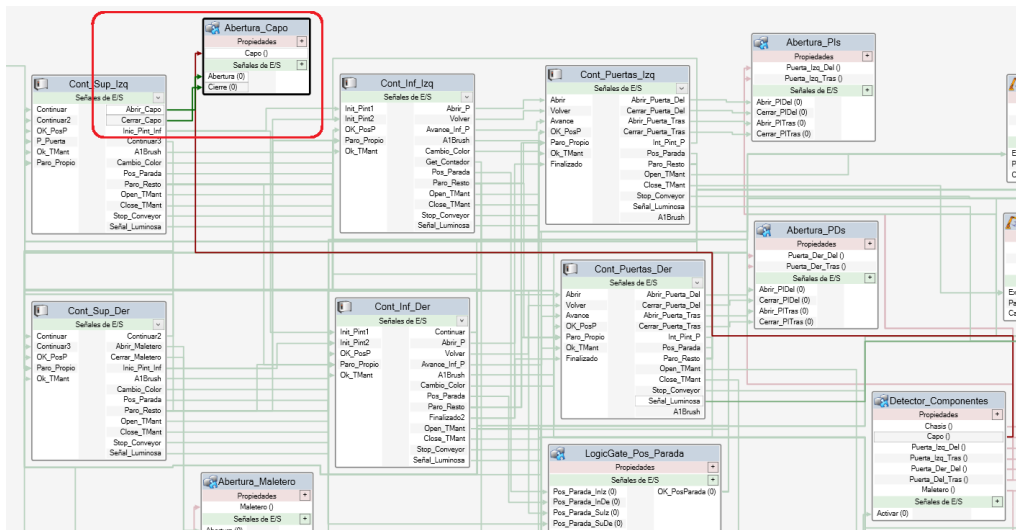


Figura 4.114 - Componente inteligente *Abertura_Capó*

Este componente tiene las entradas **Abertura** y **Cierre**. La primera de ellas se activa mediante la activación de la señal digital **Abrir_Capó** del controlador **Cont_Sup_Izq**. La segunda de ellas se activa mediante la activación de la señal digital **Cerrar_Capó** del controlador **Cont_Sup_Izq**.

Este componente también tiene la propiedad **Capó** de tipo **Part**. Esta propiedad está enlazada a la propiedad **Sup_Capó** del componente inteligente **Detector_Componentes**, en la cual se indica el identificador del capó de la carrocería sobre la que se trabaja.

El componente inteligente **Abertura_Capó** a su vez está formado, como se muestra en la Figura 4.115, por dos componentes inteligentes **Rotator2**.

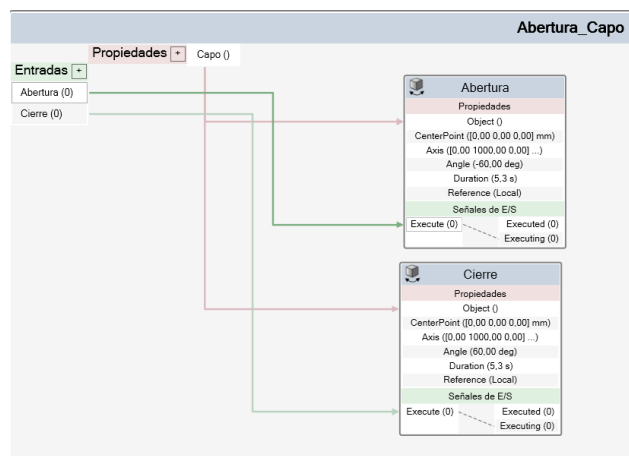


Figura 4.115 - Lógica del componente inteligente *Abertura_Capó*

El primero de ellos se ha denominado **Abertura** y es el encargado de simular la abertura del capó. Su señal de entrada de activación, **Execute**, está conectada a la señal de entrada del componente global **Abertura**. La propiedad del componente global **Capó** se conecta con la

propiedad **Object** del componente inteligente **Abertura**, de este modo sabe que componente de la estación debe rotar. Cuando se activa, rota el capó de la carrocería -60 grados respecto del eje Y de su origen local, simulando así su abertura. La duración de este giro será de 5.3 segundos. Este tiempo se ha obtenido mediante pruebas, de forma que se coordine el movimiento del robot al realizar la trayectoria de abertura del capó con el movimiento del propio capó.

El segundo de ellos se ha denominado **Cierre** y es el encargado de simular el cierre del capó. Su señal de entrada de activación, **Execute**, está conectada a la señal de entrada del componente global **Cierre**. La propiedad del componente global **Capo** se conecta con la propiedad **Object** del componente inteligente **Cierre**, de este modo sabe que componente de la estación debe rotar. Cuando se activa, rota el capó de la carrocería 60 grados respecto del eje Y de su origen local, simulando así su cierre. La duración de este giro será de 5.3 segundos. Este tiempo se ha obtenido mediante pruebas, de forma que se coordine el movimiento del robot al realizar la trayectoria de cierre del capó con el movimiento del propio capó.

Abertura Maletero

Este componente inteligente creado, mostrado en la Figura 4.116, se emplea para simular la abertura y cierre del maletero de la carrocería mientras está en movimiento en el transportador.

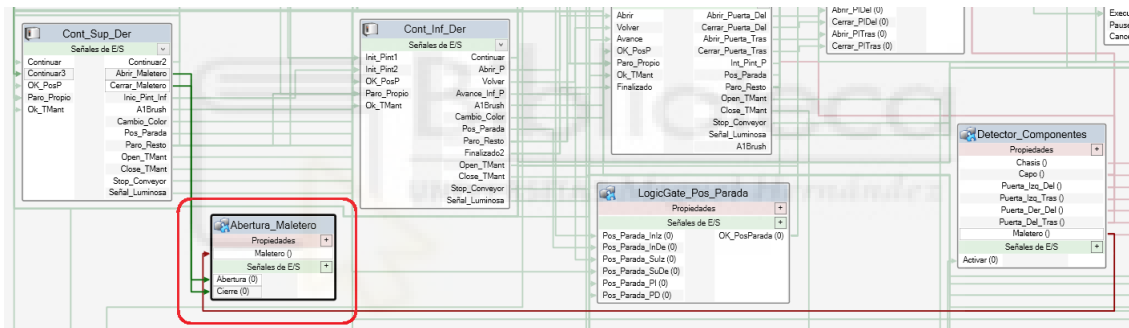


Figura 4.116 - Componente inteligente Abertura_Maletero

Este componente tiene las entradas **Abertura** y **Cierre**. La primera de ellas se activa mediante la activación de la señal digital **Abrir_Maletero** del controlador **Cont_Sup_Der**. La segunda de ellas se activa mediante la activación de la señal digital **Cerrar_Maletero** del controlador **Cont_Sup_Der**.

Este componente también tiene la propiedad **Maletero** de tipo **Part**. Esta propiedad está enlazada a la propiedad **Maletero** del componente inteligente **Detector_Componentes**, en la cual se indica el identificador del maletero de la carrocería sobre la que se trabaja.

El componente inteligente **Abertura_Maletero** a su vez está formado, como se muestra en la Figura 4.117, por dos componentes inteligentes **Rotator2**.

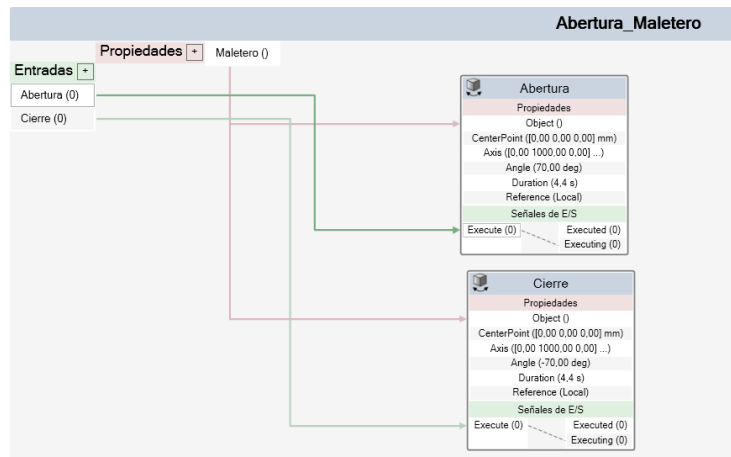


Figura 4.117 - Lógica del componente inteligente Abertura_Maletero

El primero de ellos se ha denominado **Abertura** y es el encargado de simular la apertura del maletero. Su señal de entrada de activación, **Execute**, está conectada a la señal de entrada del componente global **Abertura**. La propiedad del componente global **Mailetero** se conecta con la propiedad **Object** del componente inteligente **Abertura**, de este modo sabe que componente de la estación debe rotar. Cuando se activa, rota el maletero de la carrocería 70 grados respecto del eje Y de su origen local, simulando así su apertura. La duración de este giro será de 4.4 segundos. Este tiempo se ha obtenido mediante pruebas, de forma que se coordine el movimiento del robot al realizar la trayectoria de apertura del maletero con el movimiento del propio maletero.

El segundo de ellos se ha denominado **Cierre** y es el encargado de simular el cierre del maletero. Su señal de entrada de activación, **Execute**, está conectada a la señal de entrada del componente global **Cierre**. La propiedad del componente global **Mailetero** se conecta con la propiedad **Object** del componente inteligente **Cierre**, de este modo sabe que componente de la estación debe rotar. Cuando se activa, rota el maletero de la carrocería -70 grados respecto del eje Y de su origen local, simulando así su cierre. La duración de este giro será de 4.4 segundos. Este tiempo se ha obtenido mediante pruebas, de forma que se coordine el movimiento del robot al realizar la trayectoria de cierre del maletero con el movimiento del propio maletero.

Abertura Pls

Este componente inteligente creado, mostrado en la Figura 4.118, se emplea para simular la apertura y cierre de las puertas del lado izquierdo de la carrocería, tanto la delantera como la trasera, mientras está en movimiento en el transportador.

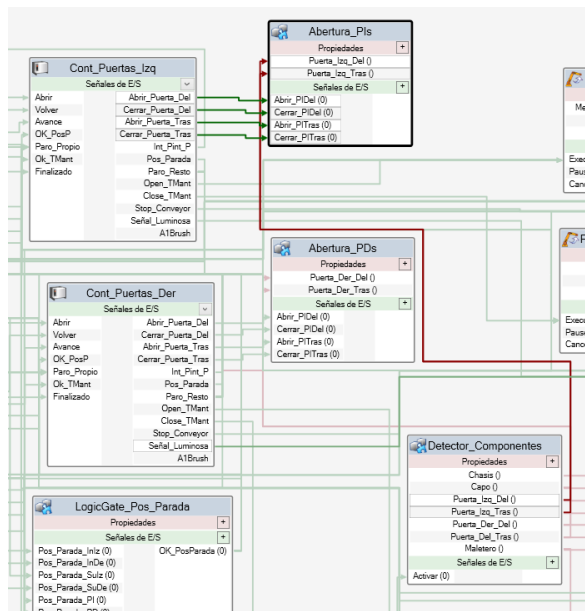


Figura 4.118 - Componente inteligente *Abertura_PIs*

Este componente tiene las entradas **Abrir_PIDel**, **Cerrar_PIDel**, **Abrir_PITras** y **Cerrar_PITras**. La señal **Abrir_PIDel** se activa mediante la activación de la señal digital **Abrir_Puerta_Del** del controlador **Cont_Puertas_Izq**. La señal **Cerrar_PIDel** se activa mediante la activación de la señal digital **Cerrar_Puerta_Del** del controlador **Cont_Puertas_Izq**. La señal **Abrir_PITras** se activa mediante la activación de la señal digital **Abrir_Puerta_Tras** del controlador **Cont_Puertas_Izq**. La señal **Cerrar_PITras** se activa mediante la activación de la señal digital **Cerrar_Puerta_Tras** del controlador **Cont_Puertas_Izq**.

Este componente también tiene las propiedades de tipo **Part** denominadas **Puerta_Izq_Del** y **Puerta_Izq_Tras**. La propiedad **Puerta_Izq_Del** está enlazada a la propiedad **Puerta_Izq_Del** del componente inteligente **Detector_Componentes**, en la cual se indica el identificador de la puerta izquierda delantera de la carrocería sobre la que se trabaja. La propiedad **Puerta_Izq_Tras** está enlazada a la propiedad **Puerta_Izq_Tras** del componente inteligente **Detector_Componentes**, en la cual se indica el identificador de la puerta izquierda trasera de la carrocería sobre la que se trabaja.

El componente inteligente **Abertura_PIs** a su vez está formado, como se muestra en la Figura 4.119, por cuatro componentes inteligentes **Rotator2**.

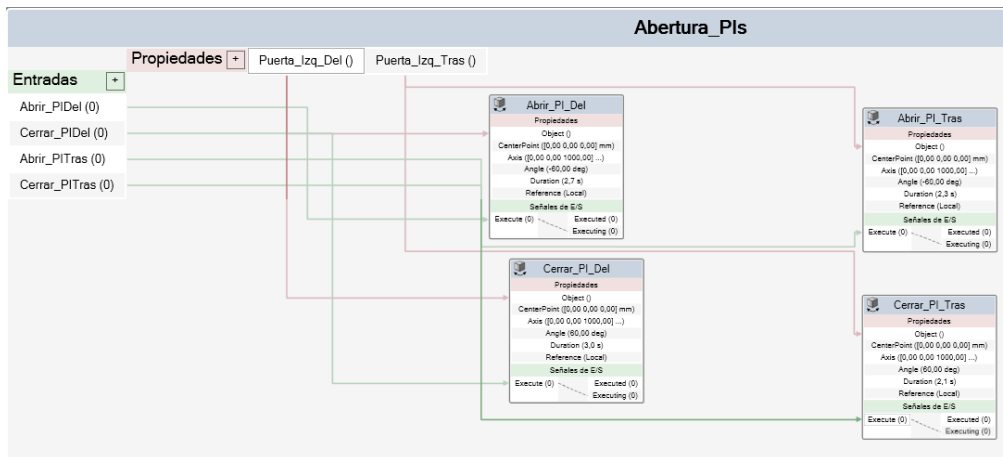


Figura 4.119 - Lógica del componente inteligente Abertura_PIs

El primero de ellos se ha denominado **Abrir_PI_Del** y es el encargado de simular la abertura de la puerta izquierda delantera. Su señal de entrada de activación, **Execute**, está conectada a la señal de entrada del componente global **Abrir_PIDel**. La propiedad del componente global **Puerta_Izq_Del** se conecta con la propiedad **Object** del componente inteligente **Abrir_PI_Del**, de este modo sabe que componente de la estación debe rotar. Cuando se activa, rota la puerta delantera izquierda de la carrocería -60 grados respecto del eje Z de su origen local, simulando así su abertura. La duración de este giro será de 2.7 segundos. Este tiempo se ha obtenido mediante pruebas, de forma que se coordine el movimiento del robot al realizar la trayectoria de abertura de la puerta izquierda delantera con el movimiento de la propia puerta izquierda delantera.

El segundo de ellos se ha denominado **Cerrar_PI_Del** y es el encargado de simular el cierre de la puerta izquierda delantera. Su señal de entrada de activación, **Execute**, está conectada a la señal de entrada del componente global **Cerrar_PIDel**. La propiedad del componente global **Puerta_Izq_Del** se conecta con la propiedad **Object** del componente inteligente **Cerrar_PI_Del**, de este modo sabe que componente de la estación debe rotar. Cuando se activa, rota la puerta izquierda delantera de la carrocería 60 grados respecto del eje Z de su origen local, simulando así su cierre. La duración de este giro será de 3.0 segundos. Este tiempo se ha obtenido mediante pruebas, de forma que se coordine el movimiento del robot al realizar la trayectoria de cierre de la puerta izquierda delantera con el movimiento de la propia puerta izquierda delantera.

El tercero de ellos se ha denominado **Abrir_PI_Tras** y es el encargado de simular la abertura de la puerta izquierda trasera. Su señal de entrada de activación, **Execute**, está conectada a la señal de entrada del componente global **Abrir_PITras**. La propiedad del componente global **Puerta_Izq_Tras** se conecta con la propiedad **Object** del componente inteligente **Abrir_PI_Tras**, de este modo sabe que componente de la estación debe rotar. Cuando se activa, rota la puerta trasera izquierda de la carrocería -60 grados respecto del eje Z de su origen local, simulando así su abertura. La duración de este giro será de 2.3 segundos. Este tiempo se ha obtenido mediante pruebas, de forma que se coordine el movimiento del robot al realizar la trayectoria de abertura de la puerta izquierda trasera con el movimiento de la propia puerta izquierda trasera.

El cuarto de ellos se ha denominado **Cerrar_PI_Tras** y es el encargado de simular el cierre de la puerta izquierda trasera. Su señal de entrada de activación, **Execute**, está conectada a la señal de entrada del componente global **Cerrar_PITras**. La propiedad del componente global **Puerta_Izq_Tras** se conecta con la propiedad **Object** del componente inteligente

Cerrar_PI_Tras, de este modo sabe que componente de la estación debe rotar. Cuando se activa, rota la puerta izquierda trasera de la carrocería 60 grados respecto del eje Z de su origen local, simulando así su cierre. La duración de este giro será de 2.1 segundos. Este tiempo se ha obtenido mediante pruebas, de forma que se coordine el movimiento del robot al realizar la trayectoria de cierre de la puerta izquierda trasera con el movimiento de la propia puerta izquierda trasera.

Abertura PDs

Este componente inteligente creado, mostrado en la Figura 4.120, se emplea para simular la apertura y cierre de las puertas del lado derecho de la carrocería, tanto la delantera como la trasera, mientras está en movimiento en el transportador.

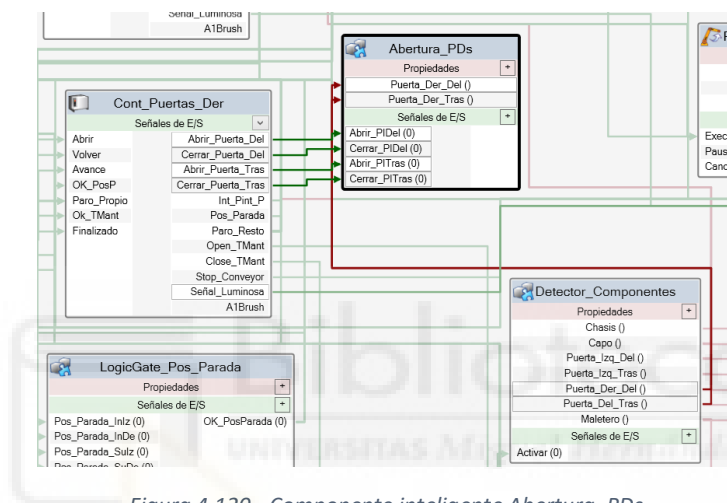


Figura 4.120 - Componente inteligente Abertura_PDs

Este componente tiene las entradas **Abrir_PIDel**, **Cerrar_PIDel**, **Abrir_PITras** y **Cerrar_PITras**. Estas señales deberían de haber sido renombradas, pero al reutilizarse el componente inteligente **Abertura_PIs** para crear este, no es posible cambiarles el nombre.

La señal **Abrir_PIDel** se activa mediante la activación de la señal digital **Abrir_Puerta_Del** del controlador **Cont_Puertas_Der**. La señal **Cerrar_PIDel** se activa mediante la activación de la señal digital **Cerrar_Puerta_Del** del controlador **Cont_Puertas_Der**. La señal **Abrir_PITras** se activa mediante la activación de la señal digital **Abrir_Puerta_Tras** del controlador **Cont_Puertas_Der**. La señal **Cerrar_PITras** se activa mediante la activación de la señal digital **Cerrar_Puerta_Tras** del controlador **Cont_Puertas_Der**.

Este componente también tiene las propiedades de tipo **Part** denominadas **Puerta_Der_Del** y **Puerta_Der_Tras**. La propiedad **Puerta_Der_Del** está enlazada a la propiedad **Puerta_Der_Del** del componente inteligente **Detector_Componentes**, en la cual se indica el identificador de la puerta derecha delantera de la carrocería sobre la que se trabaja. La propiedad **Puerta_Der_Tras** está enlazada a la propiedad **Puerta_Der_Tras** del componente inteligente **Detector_Componentes**, en la cual se indica el identificador de la puerta derecha trasera de la carrocería sobre la que se trabaja.

El componente inteligente **Abertura_PDs** a su vez está formado, como se muestra en la Figura 4.121, por cuatro componentes inteligentes **Rotator2**.

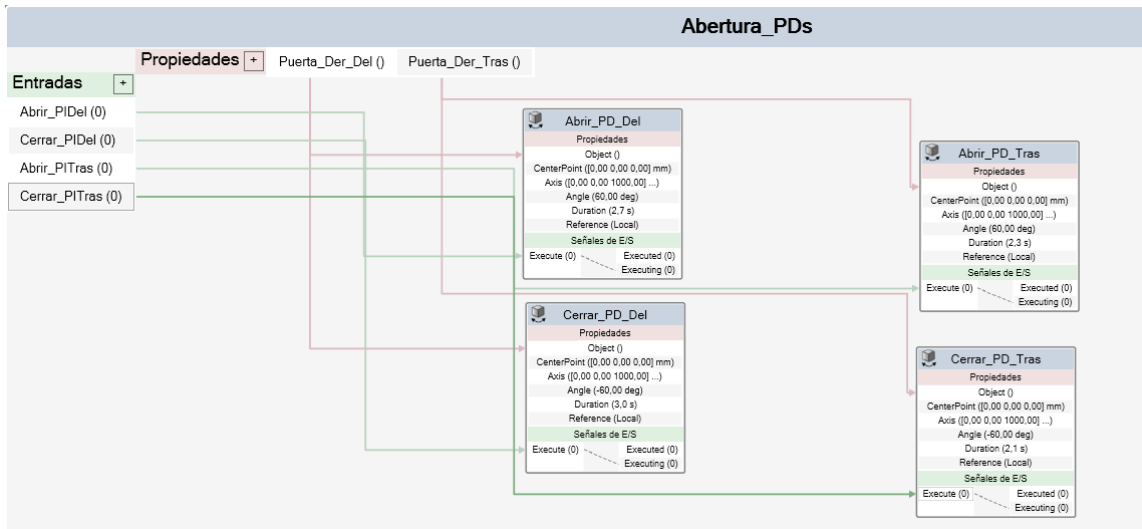


Figura 4.121 - Lógica del componente inteligente Abertura_PDs

El primero de ellos se ha denominado **Abrir_PD_Del** y es el encargado de simular la apertura de la puerta derecha delantera. Su señal de entrada de activación, **Execute**, está conectada a la señal de entrada del componente global **Abrir_PIDel**. La propiedad del componente global **Puerta_Der_Del** se conecta con la propiedad **Object** del componente inteligente **Abrir_PD_Del**, de este modo sabe que componente de la estación debe rotar. Cuando se activa, rota la puerta delantera derecha de la carrocería 60 grados respecto del eje Z de su origen local, simulando así su apertura. La duración de este giro será de 2.7 segundos. Este tiempo se ha obtenido mediante pruebas, de forma que se coordine el movimiento del robot al realizar la trayectoria de apertura de la puerta derecha delantera con el movimiento de la propia puerta derecha delantera.

El segundo de ellos se ha denominado **Cerrar_PD_Del** y es el encargado de simular el cierre de la puerta derecha delantera. Su señal de entrada de activación, **Execute**, está conectada a la señal de entrada del componente global **Cerrar_PIDel**. La propiedad del componente global **Puerta_Der_Del** se conecta con la propiedad **Object** del componente inteligente **Cerrar_PD_Del**, de este modo sabe que componente de la estación debe rotar. Cuando se activa, rota la puerta derecha delantera de la carrocería -60 grados respecto del eje Z de su origen local, simulando así su cierre. La duración de este giro será de 3.0 segundos. Este tiempo se ha obtenido mediante pruebas, de forma que se coordine el movimiento del robot al realizar la trayectoria de cierre de la puerta derecha delantera con el movimiento de la propia puerta derecha delantera.

El tercero de ellos se ha denominado **Abrir_PD_Tras** y es el encargado de simular la apertura de la puerta derecha trasera. Su señal de entrada de activación, **Execute**, está conectada a la señal de entrada del componente global **Abrir_PITras**. La propiedad del componente global **Puerta_Der_Tras** se conecta con la propiedad **Object** del componente inteligente **Abrir_PD_Tras**, de este modo sabe que componente de la estación debe rotar. Cuando se activa, rota la puerta trasera derecha de la carrocería 60 grados respecto del eje Z de su origen local, simulando así su apertura. La duración de este giro será de 2.3 segundos. Este tiempo se ha obtenido mediante pruebas, de forma que se coordine el movimiento del robot al realizar la trayectoria de apertura de la puerta derecha trasera con el movimiento de la propia puerta derecha trasera.

El cuarto de ellos se ha denominado **Cerrar_PD_Tras** y es el encargado de simular el cierre de la puerta derecha trasera. Su señal de entrada de activación, **Execute**, está conectada a la señal de entrada del componente global **Cerrar_PITras**. La propiedad del componente global **Puerta_Der_Tras** se conecta con la propiedad **Object** del componente inteligente **Cerrar_PD_Tras**, de este modo sabe que componente de la estación debe rotar. Cuando se activa, rota la puerta derecha trasera de la carrocería -60 grados respecto del eje Z de su origen local, simulando así su cierre. La duración de este giro será de 2.1 segundos. Este tiempo se ha obtenido mediante pruebas, de forma que se coordine el movimiento del robot al realizar la trayectoria de cierre de la puerta derecha trasera con el movimiento de la propia puerta derecha trasera.

Conjunto Paro Puerta

Este conjunto de componentes inteligentes, mostrado en la Figura 4.122, se emplea como sistema de seguridad para detectar la apertura de alguna de las dos puertas de acceso al interior de la estación, activándose en ese caso la señal digital de entrada **P_Puerta** del controlador **Cont_Sup_Izq**, que activa en ese controlador la rutina TRAP **Paro_Puerta**, expuesta en el apartado anterior [4.3.7.3 Rutinas TRAP](#), mediante la cual se detiene el movimiento de los robots y el transportador y se reposicionan los robots en posiciones intermedias de reposo hasta que se cierre la puerta.

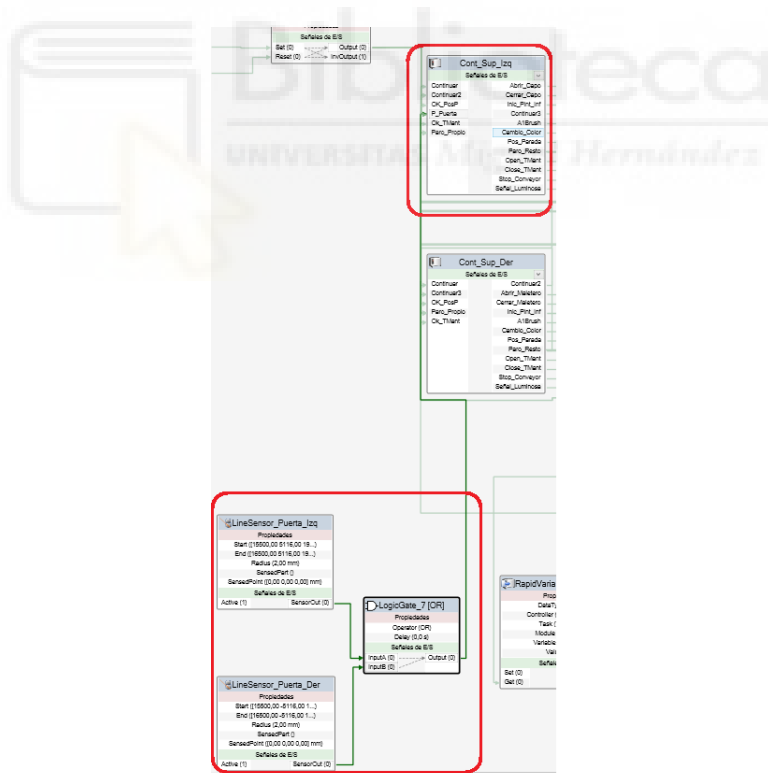


Figura 4.122 - Conjunto de componentes inteligentes Paro_Puerta

Este conjunto **Paro_Puerta** está formado por los siguientes componentes:

- Controlador Cont_Sup_Izq.
- LineSensor_Puerta_Izq.
- LineSensor_Puerta_Der.

- LogicGate [OR].

El funcionamiento de este conjunto de componentes es el siguiente. Se colocan los sensores lineales **LineSensor_Puerta_Izq** y **LineSensor_Puerta_Der** en las posiciones $[(15500, 5116, 1990), (16500, 5116, 1990)]$ y $[(15500, -5116, 1990), (16500, -5116, 1990)]$ respectivamente, respecto del sistema de coordenadas mundo de RobotStudio, como se muestra en la Figura 4.123.

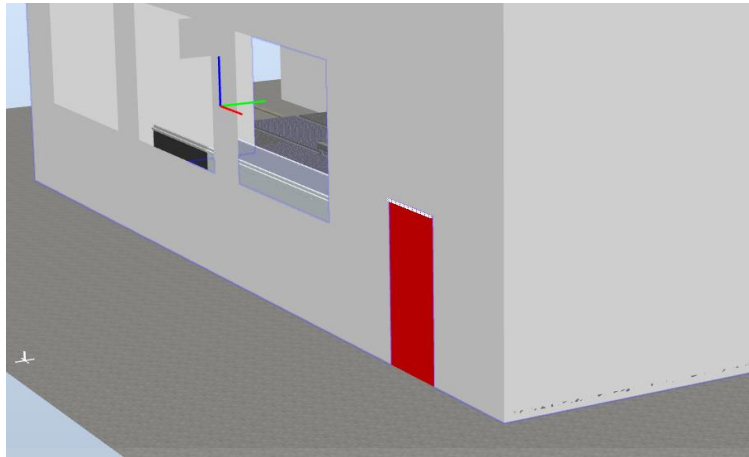


Figura 4.123 - Posición del sensor lineal *LineSensor_Puerta_Der*

De esta forma los sensores quedan posicionados en la parte superior de las puertas de acceso al interior de la estación, de forma que si una de ellas se abre, el sensor de esa puerta la detecta y se activa su salida. Mediante la puerta lógica OR se especifica que tanto si se activa la señal de salida del primer sensor o la del segundo o las dos, se activa la señal de salida de la puerta lógica, la cual está conectada con la señal digital de entrada **P_Puerta** del controlador **Cont_Sup_Izq**. Esta señal de entrada activa en ese controlador la rutina TRAP **Paro_Puerta**, mediante la cual se detiene el movimiento de los robots y el transportador y se repositionan los robots en posiciones intermedias de reposo hasta que se cierre la puerta.

Conjunto Obtener Contador

Este conjunto de componentes inteligentes, mostrado en la Figura 4.124, se emplea para obtener el valor de la variable **Contador** del controlador **Cont_Inf_Izq** y pasar ese valor a las variables **Contador** de los controladores **Cont_Puertas_Izq** y **Cont_Puertas_Der**.

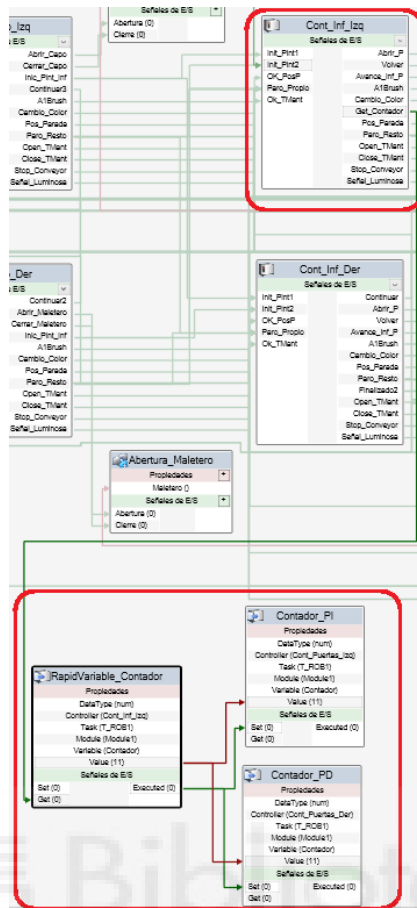


Figura 4.124 - Conjunto de componentes inteligentes *Obtener_Contador*

Este conjunto **Obtener_Contador** está formado por los siguientes componentes:

- Controlador **Cont_Inf_Izq**.
- **RapidVariable_Contador**.
- **Contador_PI** (**RapidVariable**).
- **Contador_PD** (**RapidVariable**).

El funcionamiento de este conjunto de componentes es el siguiente. Cuando en el controlador **Cont_Inf_Izq** se activa la señal digital de salida **Get_Contador** se activa la señal de entrada **Get** del componente inteligente **RapidVariable_Contador**. Al activarse esta señal de entrada de este componente inteligente se obtiene el valor de la variable **Contador** del controlador **Cont_Inf_Izq**. Cuando se ha obtenido, se activa la señal de salida **Executed**, que está conectada a las señales de entrada **Set** de los componentes inteligentes **Contador_PI** y **Contador_PD**. Al activarse la señal de entrada **Set** de estos componentes inteligentes se establece el valor obtenido anteriormente de la variable **Contador** del **Cont_Inf_Izq**, como el valor de las variables **Contador** de los controladores **Cont_Puertas_Izq** y **Cont_Puertas_Der**.

LogicGate Pos Parada

Este componente inteligente, mostrado en la Figura 4.125, se emplea como una puerta lógica AND para todas las señales digitales de salida **Pos_Parada** de todos los controladores de robot.

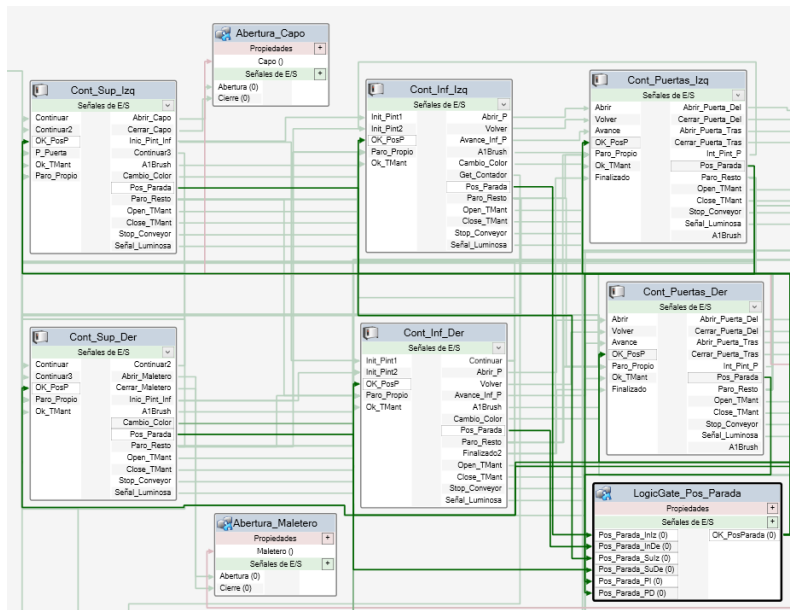


Figura 4.125 - Componente inteligente LogicGate_Pos_Parada

El funcionamiento de este componente es el siguiente. Cuando en alguno de los controladores de robot se ejecuta una rutina TRAP, ya sea de mantenimiento, limpieza o parada, en el resto se ejecuta la rutina TRAP **Parada**. Tras ejecutar parte del código de la rutina TRAP en la que se encuentre cada uno de los robots, estos vuelven a situarse en el punto espacial donde se detuvieron. En ese instante activan la señal digital de salida **Pos_Parada**, mediante la cual indican que están situados en el punto de paro, a partir del cual pueden retomar el procedimiento donde se detuvieron. Como para cada uno de los controladores de robot las rutinas TRAP tienen una cantidad de instrucciones diferente, mediante este componente inteligente se asegura que todos estén posicionados en el punto de paro donde se detuvieron y así se pueda reanudar el movimiento de la estación completa a la vez.

Este componente inteligente tiene las entradas **Pos_Parada_InIz**, **Pos_Parada_InDe**, **Pos_Parada_Sulz**, **Pos_Parada_SuDe**, **Pos_Parada_PI** y **Pos_Parada_PD**. Cada una de estas señales de entrada está conectada con la señal digital de salida Pos_Parada de los controladores **Cont_Inf_Izq**, **Cont_Inf_Der**, **Cont_Sup_Izq**, **Cont_Sup_Der**, **Cont_Puertas_Izq** y **Cont_Puertas_Der**, respectivamente.

Así pues, este componente actúa como un puerta lógica AND con 6 entradas diferentes, de modo que hasta que no se activen y se mantengan activas todas ellas, no se activará la señal de salida **OK_PosParada** del componente inteligente. Esta señal de salida está conectada a las señales digitales de entrada **OK_PosP** de todos los controladores de robot, indicándoles así que todos están listos para continuar el movimiento.

El componente inteligente **LogicGate_Pos_Parada** a su vez está formado, como se muestra en la Figura 4.126, por cinco componentes inteligentes **LogicGate [AND]**.

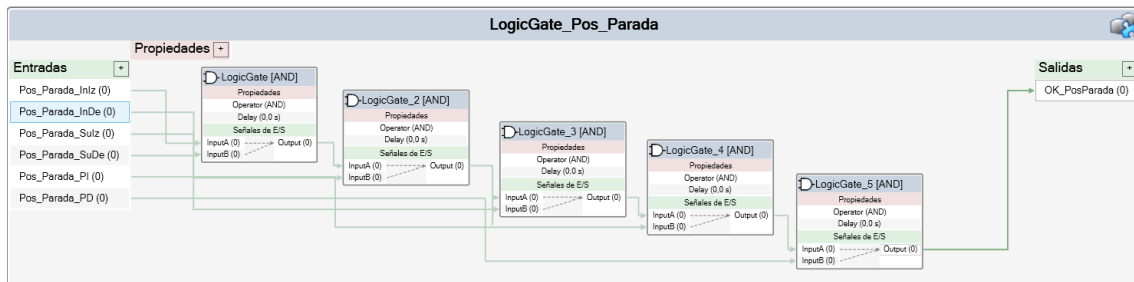


Figura 4.126 - Lógica del componente inteligente LogicGate_Pos_Parada

Conjunto Paro Emergencia

Este conjunto inteligente, mostrado en la Figura 4.127, se emplea para realizar una parada de emergencia de la estación, deteniendo el movimiento todos los robots y posicionándolos en una posición intermedia de reposo hasta que se active la señal de **Reset_Stop**.

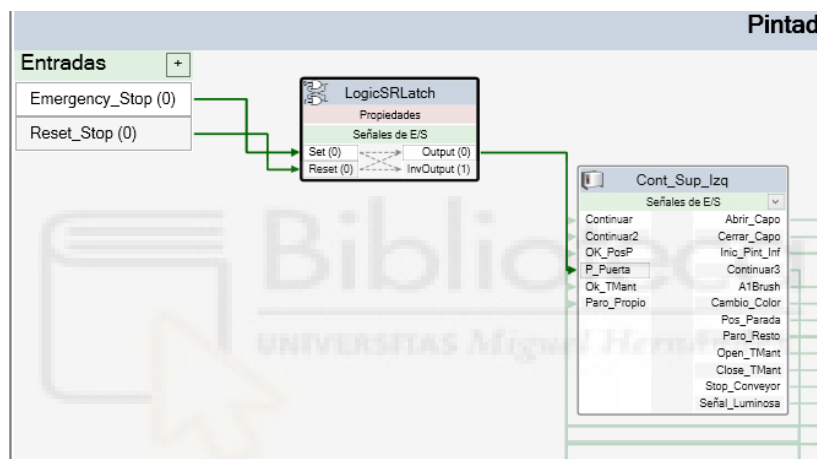


Figura 4.127 - Conjunto inteligente Paro_Emergencia

Este conjunto **Paro_Emergencia** está formado por los siguientes componentes:

- Controlador Cont_Sup_Izq.
- LogicSRLatch.
- Señal de entrada de la estación Emergency_Stop.
- Señal de entrada de la estación Reset_Stop.

El funcionamiento de este conjunto de componentes es el siguiente. Cuando se pulsa en el **Simulador de E/S**, de la pestaña **Simulación**, la señal de entrada de la estación **Emergency_Stop**, esta se activa momentáneamente para después desactivarse automáticamente. Al activarse momentáneamente se activa, momentáneamente también, la señal de entrada **Set** del componente inteligente **LogicSRLatch**, lo cual produce la activación permanente de su señal de salida **Output**. Esta señal de salida está conectada con la señal digital de entrada **P_Puerta**, del controlador **Cont_Sup_Izq**. Esta señal activa en ese controlador la rutina TRAP **Paro_Puerta**, expuesta en el apartado anterior [4.3.7.3 Rutinas TRAP](#), mediante la cual se detiene el movimiento de los robots y el transportador y se reposicionan los robots en posiciones intermedias de reposo hasta que se desactive dicha señal **P_Puerta**.

Cuando se quiere reiniciar el movimiento de la estación, se pulsa en el **Simulador de E/S**, de la pestaña **Simulación**, la señal de entrada de la estación **Reset_Stop**, la cual se activa momentáneamente para activar momentáneamente la señal de entrada **Reset** del **LogicSRLatch**, que desactiva permanentemente su señal de salida **Output**, y por lo tanto desactiva la señal **P_Puerta** del controlador **Cont_Sup_Izq**, reanudándose así el movimiento del transportador y todos los robots de la estación en el punto y procedimiento donde se detuvieron.

Transportador

Este componente inteligente, mostrado en la Figura 4.128, se crea automáticamente al crear el transportador de la estación y permite, mediante la activación de su señal de entrada **ConveyorStop**, detener el movimiento del transportador.

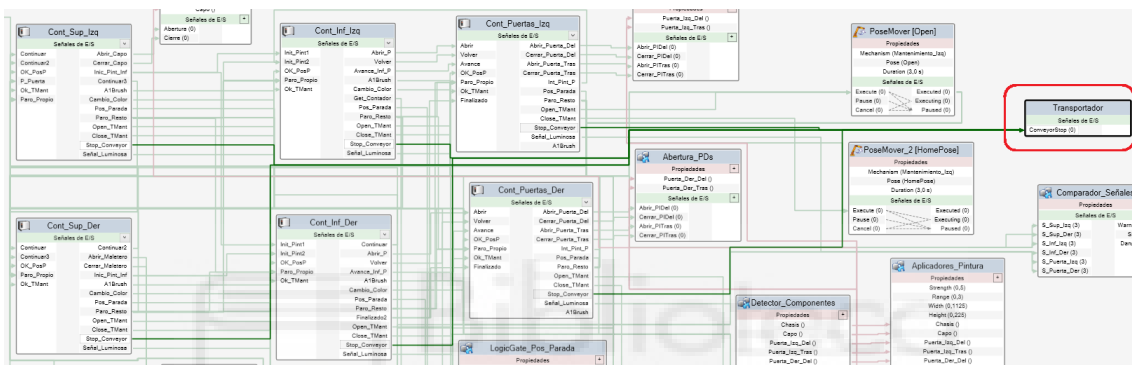


Figura 4.128 - Componente inteligente Transportador

La señal digital de salida **Stop_Conveyor**, presente en todos los controladores de robot, se activa para el controlador en cuestión en el caso de que a través de ese controlador se quiera detener el movimiento del transportador. Las señales **Stop_Conveyor** de todos los robots están conectadas con la entrada **ConveyorStop** del componente inteligente **Transportador**, de forma que si alguna de ellas se activa, el movimiento de transportador se detiene.

Conjunto Abertura Mantenimiento Izq

Este conjunto de componentes inteligentes, mostrado en la Figura 4.129, se emplea para simular la abertura y cierre de la trampilla del mecanismo de mantenimiento de la zona de mantenimiento izquierda, permitiendo así que el robot del lado izquierdo de la estación que haya solicitado una operación de mantenimiento de la herramienta, pueda introducirla a través del hueco de la trampilla para que un operario realice su mantenimiento, sin necesidad de acceder al interior de la estación.

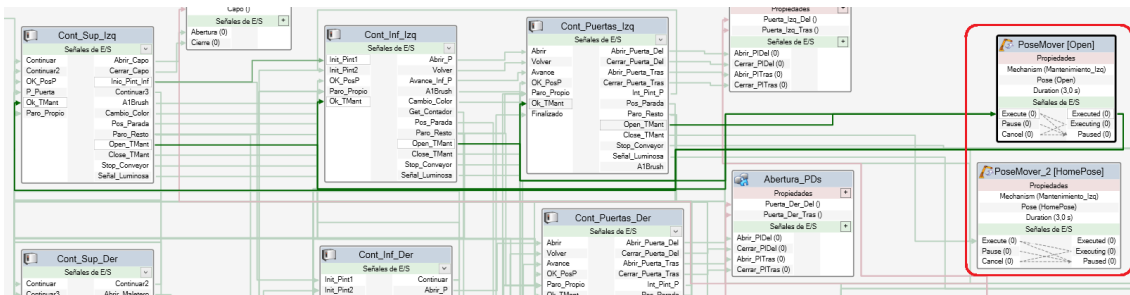


Figura 4.129 - Conjunto de componentes inteligentes Abertura_Mantenimiento_Izq

Este conjunto **Abertura_Mantenimiento_Izq** está formado por los siguientes componentes:

- Controlador Cont_Inf_Izq.
- Controlador Cont_Sup_Izq.
- Controlador Cont_Puertas_Izq.
- PoseMover [Open].
- PoseMover_2 [HomePose]

El funcionamiento de este conjunto de componentes es el siguiente. Cada uno de los controladores de los robots del lado izquierdo de la estación, **Cont_Inf_Izq**, **Cont_Sup_Izq** y **Cont_Puertas_Izq**, tienen programadas una señal digital de salida denominada **Open_TMant** y otra denominada **Close_TMant**, y una señal digital de entrada denominada **Ok_TMant**.

Las señales digitales de salida **Open_TMant** están conectadas a la entrada **Execute** del componente inteligente **PoseMover [Open]**. Cuando en uno de estos controladores se activa la señal **Open_TMant**, debido a que se requiere el mantenimiento de su herramienta, se activa la entrada del componente inteligente, el cual hace que el componente **Trampilla** del mecanismo **Mantenimiento_Izq** se desplace en 3 segundos hasta su posición **Open**, que previamente se ha creado en el mismo. Esta posición **Open** es la posición del mecanismo en la que la trampilla está totalmente abierta. Cuando se alcanza esta posición del mecanismo, se activa la señal digital de salida **Executed** del componente inteligente **PoseMover [Open]**, que está conectada a las señales digitales de entrada **Ok_TMant** de los controladores nombrados anteriormente. De esta forma el componente inteligente le comunica al controlador de robot que la trampilla está abierta y por lo tanto puede introducir la herramienta a través del hueco sin riesgo de colisión.

Por otro lado, las señales digitales de salida **Close_TMant** están conectadas a la entrada **Execute** del componente inteligente **PoseMover_2 [HomePose]**. Cuando el mantenimiento de la herramienta termina y esta se mueve fuera del hueco de mantenimiento, en el controlador de robot se activa la señal digital de salida **Close_TMant**, que activa el componente inteligente **PoseMover_2 [HomePose]**, el cual hace que el componente **Trampilla** del mecanismo **Mantenimiento_Izq** se desplace en 3 segundos hasta su posición **HomePose**, que previamente se ha creado en el mismo. Esta posición **HomePose** es la posición del mecanismo en la que la trampilla está totalmente cerrada.

Conjunto Abertura Mantenimiento Der

Este conjunto de componentes inteligentes, mostrado en la Figura 4.130, se emplea para simular la apertura y cierre de la trampilla del mecanismo de mantenimiento de la zona de mantenimiento derecha, permitiendo así que el robot del lado derecho de la estación que haya solicitado una operación de mantenimiento de la herramienta, pueda introducirla a través del hueco de la trampilla para que un operario realice su mantenimiento, sin necesidad de acceder al interior de la estación.

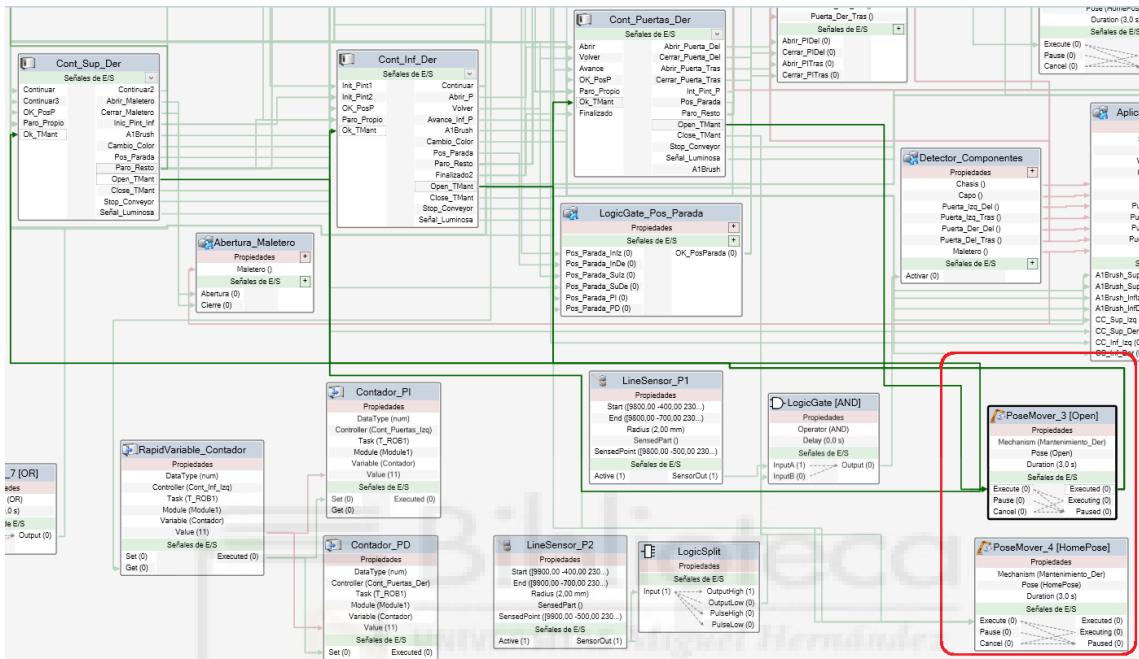


Figura 4.130 - Conjunto de componentes inteligentes Abertura_Mantenimiento_Der

Este conjunto **Abertura_Mantenimiento_Der** está formado por los siguientes componentes:

- Controlador Cont_Inf_Der.
- Controlador Cont_Sup_Der.
- Controlador Cont_Puertas_Der.
- PoseMover_3 [Open].
- PoseMover_4 [HomePose]

El funcionamiento de este conjunto de componentes es el siguiente. Cada uno de los controladores de los robots del lado derecho de la estación, **Cont_Inf_Der**, **Cont_Sup_Der** y **Cont_Puertas_Der**, tienen programadas una señal digital de salida denominada **Open_TMant** y otra denominada **Close_TMant**, y una señal digital de entrada denominada **Ok_TMant**.

Las señales digitales de salida **Open_TMant** están conectadas a la entrada **Execute** del componente inteligente **PoseMover_3 [Open]**. Cuando en uno de estos controladores se activa la señal **Open_TMant**, debido a que se requiere el mantenimiento de su herramienta, se activa la entrada del componente inteligente, el cual hace que el componente **Trampilla** del mecanismo **Mantenimiento_Der** se desplace en 3 segundos hasta su posición **Open**, que previamente se ha creado en el mismo. Esta posición **Open** es la posición del mecanismo en la que la trampilla está totalmente abierta. Cuando se alcanza esta posición del mecanismo, se

activa la señal digital de salida **Executed** del componente inteligente **PoseMover_3 [Open]**, que está conectada a las señales digitales de entrada **Ok_TMant** de los controladores nombrados anteriormente. De esta forma el componente inteligente le comunica al controlador de robot que la trampilla está abierta y por lo tanto puede introducir la herramienta a través del hueco sin riesgo de colisión.

Por otro lado, las señales digitales de salida **Close_TMant** están conectadas a la entrada **Execute** del componente inteligente **PoseMover_4 [HomePose]**. Cuando el mantenimiento de la herramienta termina y esta se mueve fuera del hueco de mantenimiento, en el controlador de robot se activa la señal digital de salida **Close_TMant**, que activa el componente inteligente **PoseMover_4 [HomePose]**, el cual hace que el componente **Trampilla** del mecanismo **Mantenimiento_Der** se desplace en 3 segundos hasta su posición **HomePose**, que previamente se ha creado en el mismo. Esta posición **HomePose** es la posición del mecanismo en la que la trampilla está totalmente cerrada.

Aplicadores Pintura

Este componente inteligente, mostrado en la Figura 4.131, es uno de los más complejos de la estación y se emplea para simular la aplicación de la pintura sobre la carrocería en la que se está trabajando. Este componente simula los patrones de pintado en forma de cono que producirían los atomizadores durante el proceso de pintado, activándose y desactivándose en los momentos adecuados.

Este componente cambiaría el color de la carrocería si esta estuviera fija en la estación. Desafortunadamente, los componentes inteligentes **PaintApplicator**, que se encuentran dentro de este y son los encargados de cambiar el color, no funcionan, es decir, no simulan el cambio de color, sobre componentes que se mueven sobre un transportador.

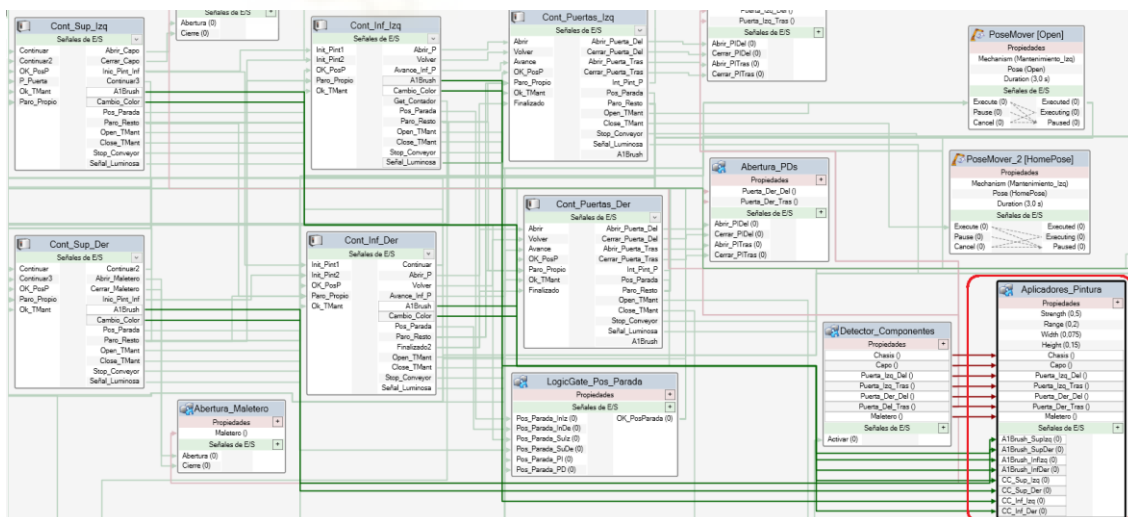


Figura 4.131 - Componente inteligente Aplicadores_Pintura

El componente inteligente tiene las siguientes entradas:

- **A1Brush_SupIzq**. Esta señal está conectada a la señal digital de grupo de salida **A1Brush** de controlador **Cont_Sup_Izq**.

- **A1Brush_SupDer**. Esta señal está conectada a la señal digital de grupo de salida **A1Brush** de controlador **Cont_Sup_Der**.
- **A1Brush_InfIzq**. Esta señal está conectada a la señal digital de grupo de salida **A1Brush** de controlador **Cont_Inf_Izq**.
- **A1Brush_InfDer**. Esta señal está conectada a la señal digital de grupo de salida **A1Brush** de controlador **Cont_Inf_Der**.
- **CC_Sup_Izq**. Esta señal está conectada a la señal digital de salida **Cambio_Color** de controlador **Cont_Sup_Izq**.
- **CC_Sup_Der**. Esta señal está conectada a la señal digital de salida **Cambio_Color** de controlador **Cont_Sup_Der**.
- **CC_Inf_Izq**. Esta señal está conectada a la señal digital de salida **Cambio_Color** de controlador **Cont_Inf_Izq**.
- **CC_Inf_Der**. Esta señal está conectada a la señal digital de salida **Cambio_Color** de controlador **Cont_Inf_Der**.

Las señales **A1Brush** de los controladores de robot se activan cuando en una trayectoria de pintado se alcanza una instrucción **SetBrush**, cambiando su valor al valor especificado en esta instrucción. Estos valores se trasladan al componente inteligente Aplicadores_Pintura mediante las conexiones expuestas.

Las señales digitales de salida **Cambio_Color** de los controladores se activan cuando se finaliza cada uno de los ciclos de pintado, es decir, cuando se termina de ejecutar el procedimiento **Principal** de los controladores. Mediante estas señales se indica al componente inteligente **Aplicadores_Pintura** que se ha terminado el ciclo y que es posible que se deba cambiar el color.

Por otro lado, el componente inteligente tiene las siguientes propiedades:

- **Strength**. Esta propiedad indica la opacidad con la que se mostraría la pintura aplicada sobre el componente, en el caso de que los componentes inteligentes **PaintApplicator** funcionarían. El valor de esta propiedad se expresa en tanto por uno, es decir de 0 a 1. Esta propiedad puede entenderse como la fuerza con la que se depositará la pintura sobre el componente, de forma que a mayor fuerza mayor será la opacidad del color aplicado y menos se verá el color base del componente. El valor de esta propiedad se fija en 0.5.
- **Range**. El valor de esta propiedad indica la altura, en metros, del cono de pintura que simula el patrón de pintado mediante los componentes inteligentes **PaintApplicator**, expuestos a continuación. El valor de esta propiedad se fija en 0.2, puesto que es la distancia que existe en todas las trayectorias de pintado entre los atomizadores y la carrocería.
- **Width**. El valor de esta propiedad indica la mitad de la longitud, en metros, de la diagonal menor de la base del cono de pintura que simula el patrón de pintado mediante los componentes inteligentes **PaintApplicator**, expuestos a continuación. El valor de esta propiedad se fija en 0.075, puesto que es la mitad del valor **Largo del patrón de pintado**, fijado en el apartado [4.3.7.1 Objetivos](#).

- **Height.** El valor de esta propiedad indica la mitad de la longitud, en metros, de la diagonal mayor de la base del cono de pintura que simula el patrón de pintado mediante los componentes inteligentes **PaintApplicator**, expuestos a continuación. El valor de esta propiedad se fija en 0.15, puesto que es la mitad del valor **Ancho del patrón de pintado**, fijado en el apartado [4.3.7.1 Objetivos](#).
- **Chasis.** Esta propiedad está conectada a la propiedad **Chasis** del componente inteligente **Detector_Componentes**, recibándose así el identificador del chasis de la carrocería sobre la que se está trabajando.
- **Capo.** Esta propiedad está conectada a la propiedad **Capo** del componente inteligente **Detector_Componentes**, recibándose así el identificador del capó de la carrocería sobre la que se está trabajando.
- **Puerta_Izq_Del.** Esta propiedad está conectada a la propiedad **Puerta_Izq_Del** del componente inteligente **Detector_Componentes**, recibándose así el identificador de la puerta izquierda delantera de la carrocería sobre la que se está trabajando.
- **Puerta_Izq_Tras.** Esta propiedad está conectada a la propiedad **Puerta_Izq_Tras** del componente inteligente **Detector_Componentes**, recibándose así el identificador de la puerta izquierda trasera de la carrocería sobre la que se está trabajando.
- **Puerta_Der_Del.** Esta propiedad está conectada a la propiedad **Puerta_Der_Del** del componente inteligente **Detector_Componentes**, recibándose así el identificador de la puerta derecha delantera de la carrocería sobre la que se está trabajando.
- **Puerta_Der_Tras.** Esta propiedad está conectada a la propiedad **Puerta_Der_Tras** del componente inteligente **Detector_Componentes**, recibándose así el identificador de la puerta derecha trasera de la carrocería sobre la que se está trabajando.
- **Maletero.** Esta propiedad está conectada a la propiedad **Maletero** del componente inteligente **Detector_Componentes**, recibándose así el identificador del maletero de la carrocería sobre la que se está trabajando.

Este componente inteligente está formado por los siguientes componentes inteligentes, mostrados en la Figura 4.132:

- Obtener_Color.
- Aplicador_Sup_Izq.
- Aplicador_Sup_Der.
- Aplicador_Inf_Izq.
- Aplicador_Inf_Der.

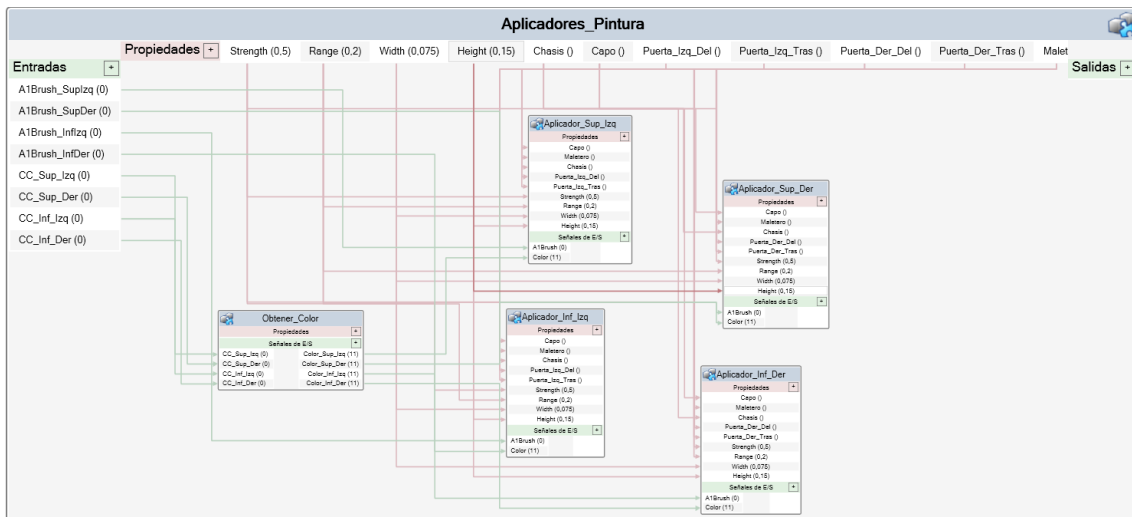


Figura 4.132 - Lógica del componente inteligente Aplicadores_Pintura

Hasta este punto el funcionamiento del componente inteligente **Aplicadores_Pintura** es el siguiente. Cuando se inician los programas de robot o cuando se termina cada ciclo de pintado se activan las señales **Cambio_Color** de los controladores de robots de pintado. Esto hace que se activen las señales de entrada del componente inteligente que comienzan por **CC_**. Estas señales activan las señales de entrada, con su mismo nombre, del componente inteligente **Obtener_Color**.

El componente inteligente **Obtener_Color** se emplea para obtener el valor de las variables **Color** de los controladores de robots de pintado. El valor de estas variables se pasa a través de las salidas de este componente inteligente a los componentes inteligentes **Aplicador_**, según el controlador de robot al que se refieren. Los valores de la variable **Color** se emplearán en los componentes inteligentes **Aplicador_** para dotar de ese color a los conos de pintura que simulan el patrón de pintado.

Por otro lado, cuando en una trayectoria de pintado se alcanza una instrucción **SetBrush**, el valor de la señal **A1Brush** del controlador pasa a tener el valor especificado en la instrucción **SetBrush**. Mediante esta instrucción se indica el color de pintado de la trayectoria, cuyo valor al inicio de la trayectoria es igual al de la variable **Color** y al final es igual a 1, que indica que se detiene el pintado.

Los valores de las señales **A1Brush** de los controladores de los robots de pintado se pasan a las señales de entrada del componente inteligente **Aplicadores_Pintura** cuyo nombre comienza por **A1Brush_**, según el controlador de robot al que se refieren. Estas entradas a su vez están conectadas a las entradas **A1Brush** de los componentes inteligentes **Aplicador_**, según el controlador de robot al que se refieren.

Por otro lado, las propiedades del componente inteligente **Aplicadores_Pintura** se conectan a las propiedades con su mismo nombre de los componentes inteligentes **Aplicador_**.

Los componentes inteligentes **Aplicador_** se emplearán para simular el patrón de pintado de los atomizadores, activándose y desactivándose en el momento adecuado. Si la carrocería estuviera estática también modificarían el color de la misma al pasar por encima de ella.

El componente inteligente **Obtener_Color** está formado por los siguientes componentes inteligentes, mostrados en la Figura 4.133:

- Color_Sup_Izq (RapidVariable)
- Color_Sup_Der (RapidVariable)
- Color_Inf_Izq (RapidVariable)
- Color_Inf_Der (RapidVariable)

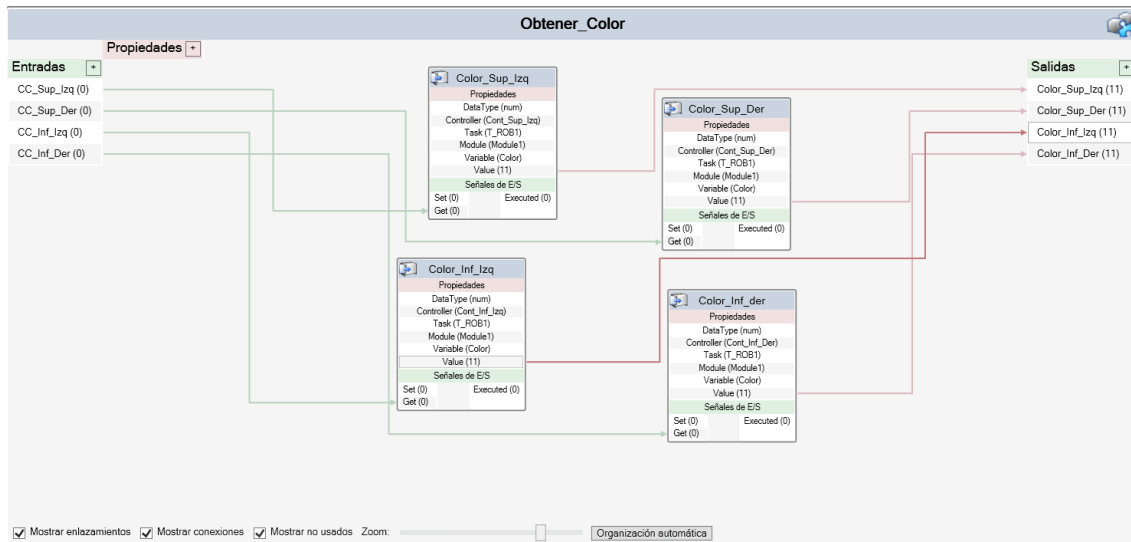


Figura 4.133 - Lógica del componente inteligente *Obtener_Color*

Al activarse las señales de entrada de este componente, se activan las señales de entrada **Get** de los componentes inteligentes de tipo **RapidVariable** que hacen referencia al mismo controlador. De esta forma se obtienen los valores de la variable **Color** de cada controlador. Estos valores se pasan a las salidas del componente inteligente, en función del controlador al que hacen referencia.

Cada uno de los componentes inteligentes **Aplicador_(Referencia a controlador)** está formado por los siguientes componentes inteligentes, mostrados en la Figura 4.134:

- ColorTable
- Comparar
- 5 componentes inteligentes PaintApplicator. Para los robots de lado derecho se tendrá un PaintApplicator para el componente **Capo**, otro para el **Chasis**, otro para el **Maletero**, otro para la **Puerta delantera derecha** y otro para la **Puerta trasera derecha**. Esto se hace para poder simular el cono de pintado sobre cualquiera de los componentes que pueden pintar estos robots. Para los robots de lado izquierdo se tendrá un PaintApplicator para el componente **Capo**, otro para el **Chasis**, otro para el **Maletero**, otro para la **Puerta delantera izquierda** y otro para la **Puerta trasera izquierda**. Esto se hace para poder simular el cono de pintado sobre cualquiera de los componentes que pueden pintar estos robots.

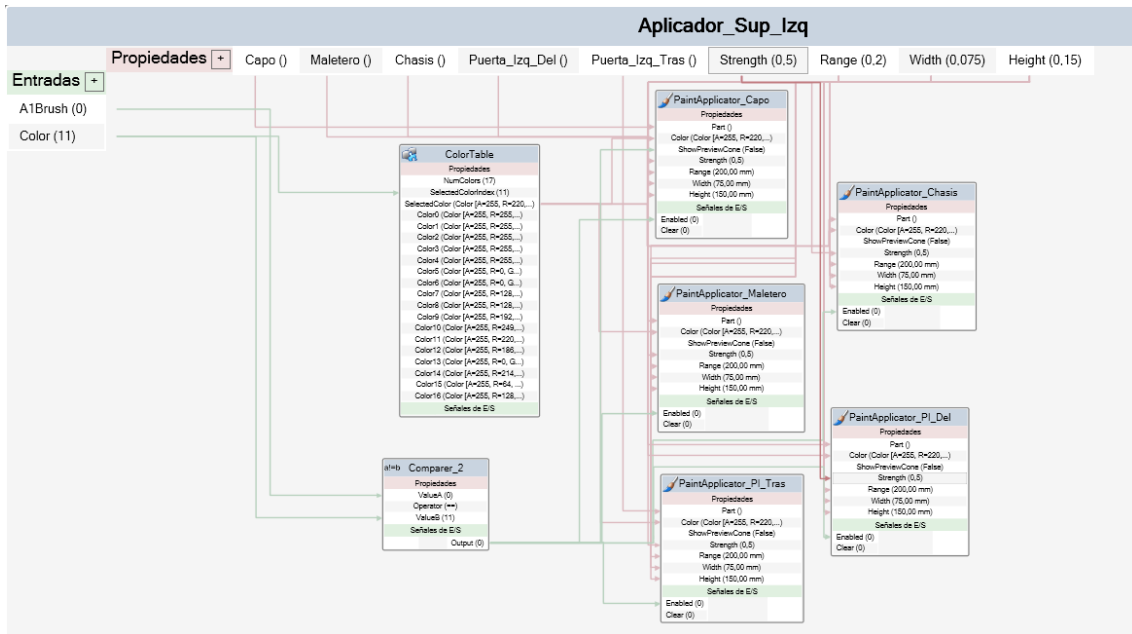


Figura 4.134 - Lógica del componente inteligente Aplicador_Sup_Izq

La señal de entrada **Color** toma el valor de la variable **Color** del controlador especificador en el nombre del aplicador, obtenida mediante el componente inteligente **Obtener_Color**. Este valor se pasa a la propiedad **ValueB** del componente inteligente **Comparar**.

El valor de la entrada **A1Brush** se pasa a la propiedad **ValueA** del componente inteligente **Comparar**. Cuando el valor de la señal **A1Brush** es igual al valor de la variable **Color**, es decir, cuando el robot que está realizando una trayectoria de pintado llega a la instrucción **SetBrush** de inicio, donde se especifica el color de la trayectoria, la salida **Output** del componente inteligente **Comparar** se activa. Esta salida está conectada a las entradas **Enabled** de los componentes inteligentes **PaintApplicator**, de forma que al activarse, activa estos componentes inteligentes, activando así la simulación del cono de patrón de pintado de los atomizadores.

Cuando el robot llega a la instrucción **SetBrush** final, el valor de **A1Brush** pasa a ser 1, con lo cual nunca es posible que sea igual al valor de la variable **Color**, lo que hace que la salida **Output** del componente inteligente **Comparar** se desactive, desactivando así los componentes inteligentes **PaintApplicator**.

Por otro lado, el valor de la entrada **Color** también se pasa a la propiedad **SelectedColorIndex** del componente inteligente **ColorTable**. En este componente se ha especificado 17 colores diferentes, que serán los posibles colores de pintado, como se muestra en la Figura 4.135.

- **S_Sup_Der.** Esta entrada está conectada a la señal analógica de salida **Señal_Luminosa** del controlador **Cont_Sup_Der**.
- **S_Inf_Izq.** Esta entrada está conectada a la señal analógica de salida **Señal_Luminosa** del controlador **Cont_Inf_Izq**.
- **S_Inf_Der.** Esta entrada está conectada a la señal analógica de salida **Señal_Luminosa** del controlador **Cont_Inf_Der**.
- **S_Puerta_Izq.** Esta entrada está conectada a la señal analógica de salida **Señal_Luminosa** del controlador **Cont_Puertas_Izq**.
- **S_Puerta_Der.** Esta entrada está conectada a la señal analógica de salida **Señal_Luminosa** del controlador **Cont_Puertas_Der**.

Este componente inteligente recibe el valor de todas las señales analógicas de salida **Señal_Luminosa** de todos los controladores de robot y realiza una serie de comparaciones y operaciones lógicas para en función del valor de todas estas señales, activar una señal de salida u otra.

Este componente inteligente **Comparador_Señales** está formado por los siguientes componentes inteligentes, mostrados en la Figura 4.137:

- 18 componentes inteligentes Comparer.
- 10 componentes inteligentes LogicGate [OR].
- 5 componentes inteligentes LogicGate [AND].

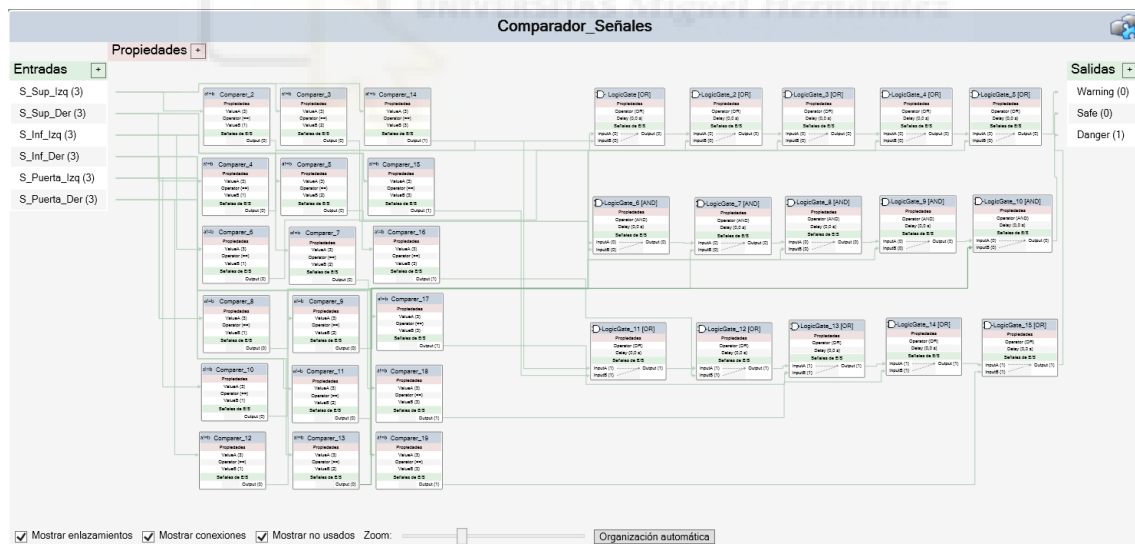


Figura 4.137 - Lógica del componente inteligente Comparador_Señales

Para cada una de las entradas del componente inteligente se realiza una comparación del valor de la misma con 1, 2 y 3, activándose la salida del componente inteligente **Comparer** destinado a esa entrada cuya comparación es cierta.

Cada una de las salidas de los componentes inteligentes **Comparer** que hacen la comparación con el valor 1, se conecta a un grupo de componentes inteligentes **LogicGate [OR]**, de modo que

si alguna de las señales de entrada tiene valor 1, se activa la salida **Warning** del componente inteligente **Comparador_Señales**.

Cada una de las salidas de los componentes inteligentes **Comparar** que hacen la comparación con el valor 2, se conecta a un grupo de componentes inteligentes **LogicGate [AND]**, de modo que si todas las señales de entrada tienen valor 2, se activa la salida **Safe** del componente inteligente **Comparador_Señales**.

Cada una de las salidas de los componentes inteligentes **Comparar** que hacen la comparación con el valor 3, se conecta a un grupo de componentes inteligentes **LogicGate [OR]**, de modo que si alguna de las señales de entrada tiene valor 3, se activa la salida **Danger** del componente inteligente **Comparador_Señales**.

Una vez se ha activado una u otra salida se pasa al siguiente componente inteligente, **Señales_Luminosas**. Este componente inteligente tiene las siguientes entradas:

- **Safe**. Esta entrada está conectada a la salida **Safe** del componente inteligente **Comparador_Señales**.
- **Warning**. Esta entrada está conectada a la salida **Warning** del componente inteligente **Comparador_Señales**.
- **Danger**. Esta entrada está conectada a la salida **Danger** del componente inteligente **Comparador_Señales**.

Este componente inteligente **Señales_Luminosas** está formado por los siguientes componentes inteligentes, mostrados en la Figura 4.138:

- 12 componentes inteligentes Highlighter.
- 12 componentes inteligentes LightControl.

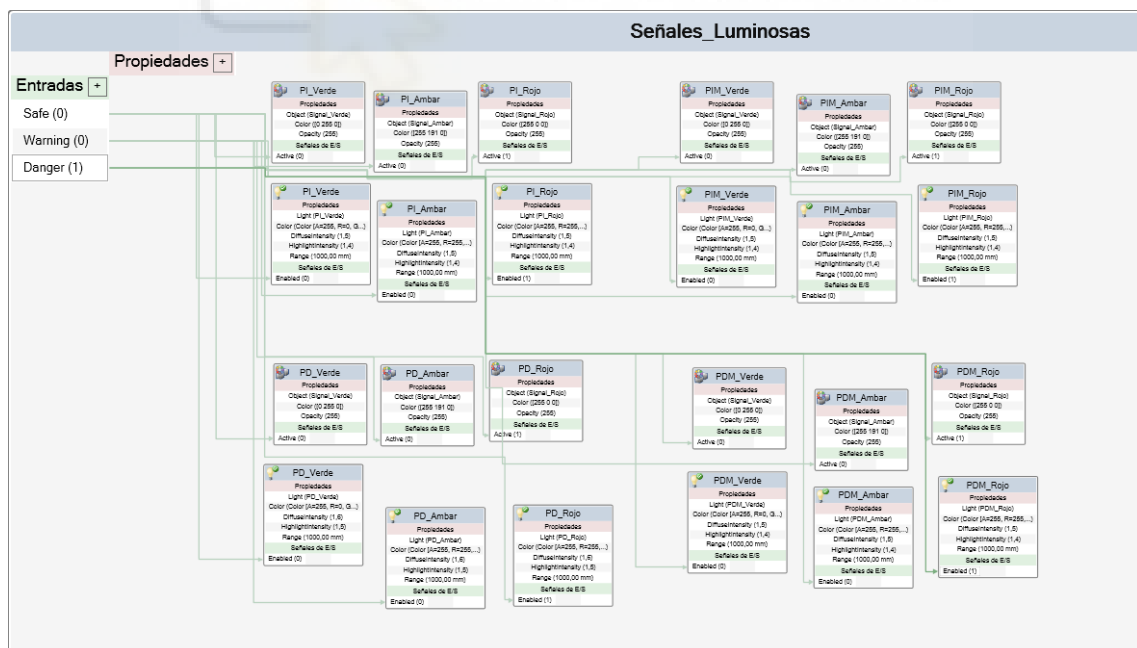


Figura 4.138 - Lógica del componente inteligente Señales_Luminosas

Puesto que en la estación se tiene 4 componentes **Torre de señalización luminosa industrial**, uno encima de cada puerta, en este componente inteligente se tienen 4 grupos, uno para cada

componente **Torre**, formados cada uno de ellos por 3 componentes inteligentes **Highlighter** y 3 componentes inteligentes **LightControl**.

Si la señal de entrada activada es **Safe**, se activará el **Highlighter Verde** y el **LightControl Verde** en cada uno de los grupos. Los componentes **Highlighter Verde** cambiarán la apariencia del componente **Signal_Verde** de cada uno de los componentes **Torre de señalización luminosa industrial**, para pasar de una apariencia de cristal verde transparente a un componente verde totalmente opaco.

Los componentes **LightControl Verde** activarán las fuentes de luz puntual de color verde, previamente creadas, situadas en el componente **Signal_Verde** de cada uno de los componentes **Torre de señalización luminosa industrial**, de modo que a la vez que cambia de apariencia se simula la iluminación de ese componente con su color verde.

Esto indicará que es seguro acceder al interior de la estación ya que todos los robots se encuentran parados.

Si la señal de entrada activada es **Warning**, se activará el **Highlighter Ámbar** y el **LightControl Ámbar** en cada uno de los grupos. Los componentes **Highlighter Ámbar** cambiarán la apariencia del componente **Signal_Ambar** de cada uno de los componentes **Torre de señalización luminosa industrial**, para pasar de una apariencia de cristal ámbar transparente a un componente ámbar totalmente opaco.

Los componentes **LightControl Ámbar** activarán las fuentes de luz puntual de color ámbar, previamente creadas, situadas en el componente **Signal_Ambar** de cada uno de los componentes **Torre de señalización luminosa industrial**, de modo que a la vez que cambia de apariencia se simula la iluminación de ese componente con su color ámbar.

Esto indicará que se tiene un cierto grado de peligro para acceder al interior de la estación ya que uno de los robots se encuentra en movimiento para realizar tareas de mantenimiento o limpieza.

Si la señal de entrada activada es **Danger**, se activará el **Highlighter Rojo** y el **LightControl Rojo** en cada uno de los grupos. Los componentes **Highlighter Rojo** cambiarán la apariencia del componente **Signal_Rojo** de cada uno de los componentes **Torre de señalización luminosa industrial**, para pasar de una apariencia de cristal rojo transparente a un componente rojo totalmente opaco.

Los componentes **LightControl Rojo** activarán las fuentes de luz puntual de color rojo, previamente creadas, situadas en el componente **Signal_Rojo** de cada uno de los componentes **Torre de señalización luminosa industrial**, de modo que a la vez que cambia de apariencia se simula la iluminación de ese componente con su color rojo.

Esto indicará que se corre un alto peligro si se accede a la estación ya que uno o varios de los robots se encuentran en movimiento para realizar la aplicación de la capa base.

4.3.9 Creación de señales de E/S

La estación robótica diseñada y programada en el presente proyecto está formada por seis robots, con sus tracks y herramientas, y multitud de componentes inteligentes. Para que todos ellos puedan ejecutarse de manera conjunta, sin causar errores y evitando colisionar entre sí o con otros componentes de la estación, es necesario una buena comunicación entre todos ellos. Para desempeñar esta labor se han programado en cada uno de los controladores de robot una serie de señales de entras y salida, en su mayoría digitales, mostrada en la siguiente Tabla 4.

Cont_Sup_Izq	Cont_Sup_Der	Cont_Inf_Izq	Cont_Inf_Der	Cont_Puertas_Izq	Cont_Puertas_Der
Abrir_Capo(Do)	Abrir_Maletero(Do)	Abrir_P(Do)	Abrir_P(Do)	Abrir(Di)	Abrir(Di)
Cambio_Color(Do)	Cambio_Color(Do)	Avance_Inf_P(Do)	Avance_Inf_P(Do)	Abrir_Puerta_Del(Do)	Abrir_Puerta_Del(Do)
Cerrar_Capo(Do)	Cerrar_Maletero(Do)	Cambio_Color(Do)	Cambio_Color(Do)	Abrir_Puerta_Tras(Do)	Abrir_Puerta_Tras(Do)
Clock_ON(Do)	Close_TMant(Do)	Close_TMant(Do)	Close_TMant(Do)	Avance(Di)	Avance(Di)
Close_TMant(Do)	Continuar(Di)	Finalizado(Do)	Continuar(Do)	Cerrar_Puerta_Del(Do)	Cerrar_Puerta_Del(Do)
Continuar(Di)	Continuar2(Do)	Get_Contador(Do)	Finalizado(Do)	Cerrar_Puerta_Tras(Do)	Cerrar_Puerta_Tras(Do)
Continuar2(Di)	Continuar3(Di)	Init_Pint1(Di)	Finalizado2(Do)	Close_TMant(Do)	Close_TMant(Do)
Continuar3(Do)	Finalizado(Do)	Init_Pint2(Di)	Init_Pint1(Di)	Finalizado(Di)	Finalizado(Di)
Finalizado(Do)	Inic_Pint_Inf(Do)	Limpieza_Atom(Di)	Init_Pint2(Di)	Init_Pint_P(Do)	Init_Pint_P(Do)
Inic_Pint_Inf(Do)	Limpieza_Atom(Di)	Mantenimiento(Di)	Limpieza_Atom(Di)	Mantenimiento(Di)	Mantenimiento(Di)
Limpieza_Atom(Di)	Mantenimiento(Di)	OK_PosP(Di)	Mantenimiento(Di)	OK_PosP(Di)	OK_PosP(Di)
Mantenimiento(Di)	OK_PosP(Di)	Ok_TMant(Di)	OK_PosP(Di)	Ok_TMant(Di)	Ok_TMant(Di)
OK_PosP(Di)	Ok_TMant(Di)	Open_TMant(Do)	Ok_TMant(Di)	Open_TMant(Do)	Open_TMant(Do)
Ok_TMant(Di)	Open_TMant(Do)	P_Puerta(Di)	Open_TMant(Do)	P_Puerta(Di)	P_Puerta(Di)
Open_TMant(Do)	P_Puerta(Di)	Paro_Propio(Di)	P_Puerta(Di)	Paro_Propio(Di)	Paro_Propio(Di)
P_Puerta(Di)	Paro_Propio(Di)	Paro_Resto(Do)	Paro_Propio(Di)	Paro_Resto(Do)	Paro_Resto(Do)
Paro_Propio(Di)	Paro_Resto(Do)	Pos_Parada(Do)	Paro_Resto(Do)	Pos_Parada(Do)	Pos_Parada(Do)
Paro_Resto(Do)	Pos_Parada(Do)	RUN_Limp(Do)	Pos_Parada(Do)	Señal_Luminosa(Ao)	Señal_Luminosa(Ao)
Pos_Parada(Do)	RUN_Limp(Do)	Señal_Luminosa(Ao)	RUN_Limp(Do)	Stop_Conveyor(Do)	Stop_Conveyor(Do)
RUN_Limp(Do)	Señal_Luminosa(Ao)	Stop_Conveyor(Do)	Señal_Luminosa(Ao)	Volver(Di)	Volver(Di)
Señal_Luminosa(Ao)	Stop_Conveyor(Do)	Volver(Do)	Stop_Conveyor(Do)		
Stop_Conveyor(Do)			Volver(Do)		

Tabla 4.4 - Señales de E/S de todos los controladores virtuales de la estación

Todas estas señales se han creado según lo expuesto en el apartado [3.1.21 Señales de E/S del sistema de robot](#). En el parámetro **Assigned to Device** de todas las señales creadas no se especifica ningún dispositivo de E/S, por lo que no están conectadas a ningún dispositivo, creándose así señales simuladas. En el parámetro **Access Level** de todas las señales creadas se indica **All**, de modo que todos los clientes, locales y remotos, tienen acceso de escritura. De esta forma se asegura que se puede cambiar el valor de cualquier señal en cualquier momento si se requiere. Esto se hace así ya que el presente proyecto es una simulación, en una estación real se deberían especificar los dispositivos de E/S y dotar a las señales de una cierta seguridad de acceso.

La tarea de cada una de estas señales de E/S creadas se ha expuesto a lo largo de los anteriores apartados [4.3.7.2 Trayectorias, procedimientos y rutinas](#), [4.3.7.3 Rutinas TRAP](#) y [4.3.8 Desarrollo de la lógica de estación](#), por lo que no volverán a exponerse.

4.3.10 Configuración de la simulación

Como se ha mencionado anteriormente, para comprobar el correcto funcionamiento de los programas de robot mientras se lleva a cabo su desarrollo, estos se ejecutan, y por lo tanto, se simulan de forma que pueda comprobarse gráficamente el desempeño de los robots y demás componentes de la estación y así poder cambiar o corregir el código programado hasta ese momento.

Para poder simular la estación primero se debe configurar como se expone en el apartado [3.1.22 Simulación de la estación](#).

En este apartado se expone la configuración final para la simulación de la estación de aplicación de la capa base diseñada.

La configuración de la simulación se puede observar en la siguiente Figura 4.139.

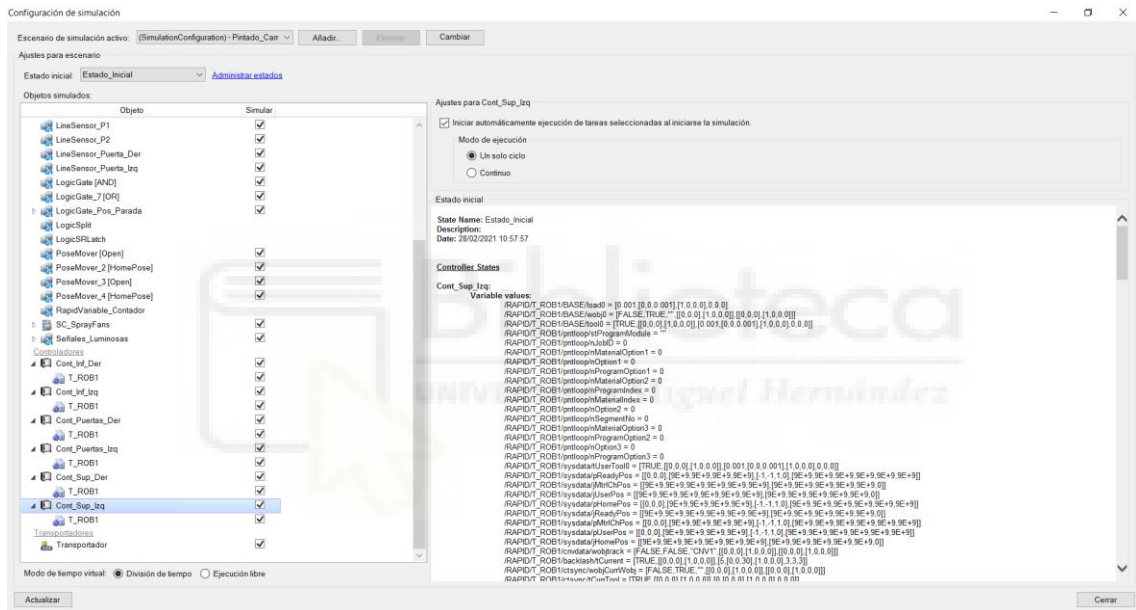


Figura 4.139 - Configuración final de la simulación

El estado inicial de la simulación, es decir, como están configurados y posicionados todos los componentes de la estación al inicio de la simulación, se fija como el estado inicial, guardado previamente, **Estado_Inicial**, quedando posicionados los componentes de la estación como se muestra en la Figura 4.140. A las paredes de la estación se le da una apariencia semitransparente para que no dificulten la visión del interior de la estación.

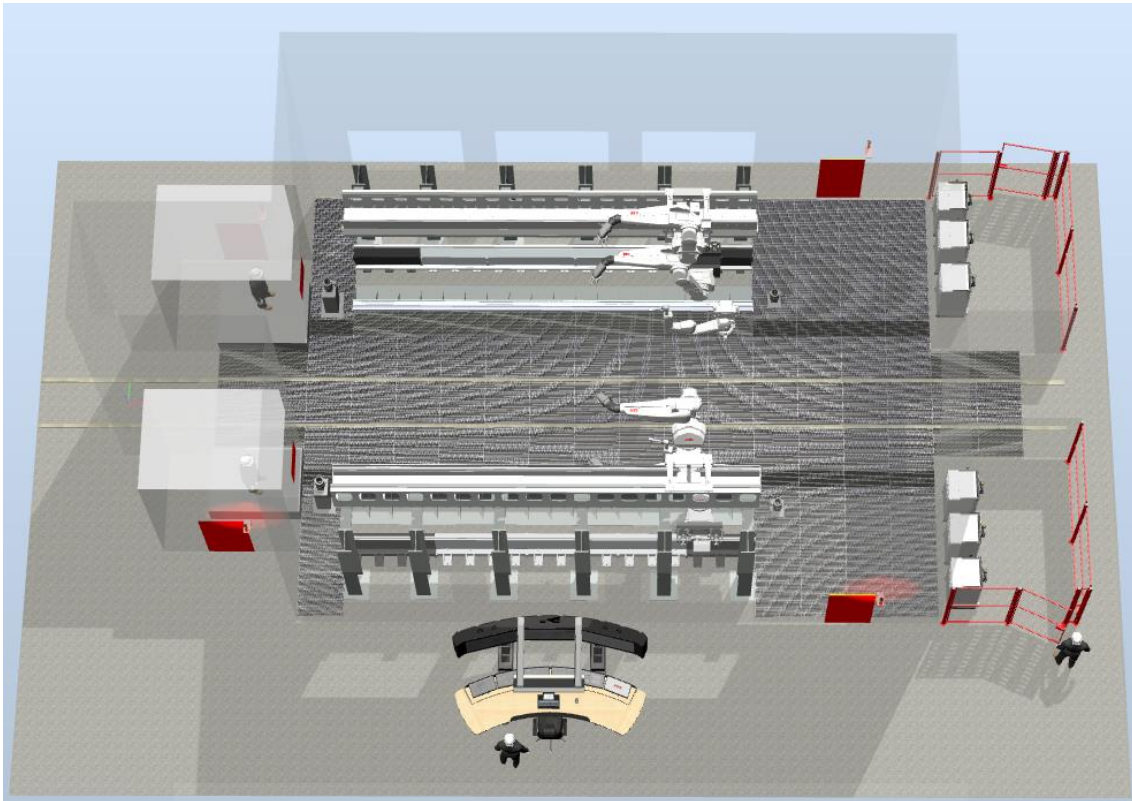


Figura 4.140 - Estado_Inicial de la estación

En la sección **Objetos simulados** se seleccionan todos los componentes inteligentes, controladores y el transportador, ya que se necesita simularlos todos.

Para cada uno de los controladores virtuales de robot se seleccionan las opciones **Iniciar automáticamente ejecución de tareas seleccionadas al iniciarse la simulación** y el **Modo de ejecución** igual a **Un solo ciclo**, ya que mediante el bucle programado en el procedimiento **main** de cada uno de los controladores se realiza el pintado de todas las carrocerías programadas.

Para cada una de las tareas de los controladores virtuales se selecciona como **Punto de entrada** de la simulación el procedimiento **main**.

Por último se selecciona el **Modo de tiempo virtual** igual a **División de tiempo**.

Por otro lado, en el **Simulador de E/S**, para cada uno de los controladores virtuales se crea una **Lista de usuarios** con todas las señales que se han creado para ese controlador en particular. De este modo se puede observar y manipular la activación y desactivación de las señales durante la simulación. En la siguiente Figura 4.141 se muestra la lista de usuarios final para el controlador **Cont_Sup_Izq**.

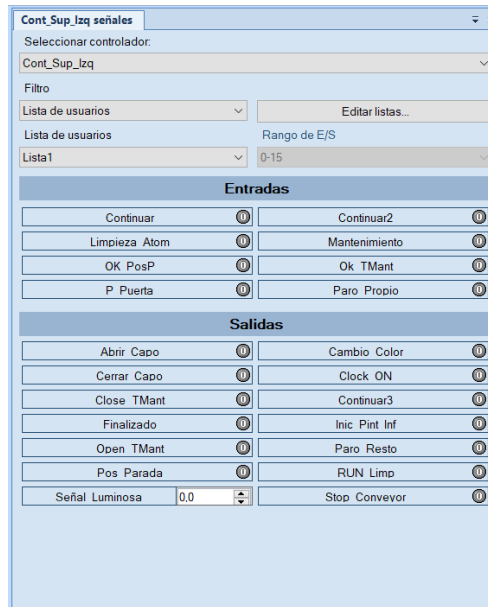


Figura 4.141 - Lista de usuarios de señales de E/S para el controlador Cont_Sup_Izq

También se crean 3 cronómetros, mostrados en la siguiente Figura 4.142.

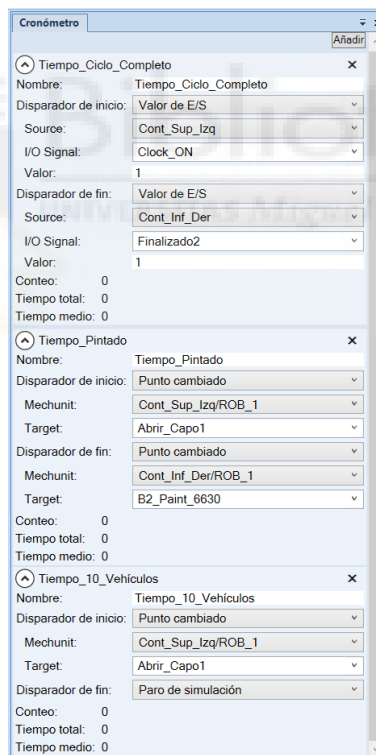


Figura 4.142 - Cronómetros de la estación

El primero de ellos se denomina **Tiempo_Ciclo_Completo** y se inicia cuando se activa la señal **Clock_ON** del controlador **Cont_Sup_Izq**, desactivándose cuando se activa la señal **Finalizado2** del controlador **Cont_Inf_Der**. Este cronometro se emplea para determinar el tiempo de un ciclo completo de pintado, incluyendo el cambio de color y la limpieza del atomizador.

El segundo de ellos se denomina **Tiempo_Pintado** y se inicia cuando en el controlador **Cont_Sup_Izq** se alcanza el objetivo **Abrir_Capo1**, desactivándose cuando en el controlador

Cont_Inf_Der se alcanza el objetivo **B2_Paint_6630**. Este cronometro se emplea para determinar el tiempo de un ciclo de pintado, pero en este caso refiriéndose solo al tiempo que tardan los robots en aplicar la capa base en toda la carrocería.

El tercero de ellos se denomina **Tiempo_10_Vehículos** y se inicia cuando en el controlador **Cont_Sup_Izq** se alcanza el objetivo **Abrir_Capo1**, desactivándose cuando se detiene la simulación, configurada en este caso para pintar 10 carrocerías. Este cronometro se emplea para determinar el tiempo de 10 ciclos completos de pintado.

Por último, para comprobar que no existen colisiones entre los diferentes componentes de la estación durante la simulación, se crean cuatro conjuntos de colisión, según lo expuesto en el apartado [3.1.23 Detección de colisiones](#). Se crea un conjunto de colisión para cada uno de los robots de aplicación de pintura IRB5500, como se muestra en la Figura 4.143. En el conjunto de objetos A se añaden el robot y su herramienta, y en el conjunto de objetos B se añaden aquellos componentes de la estación que pueden colisionar con ese robot y su herramienta. Para los robots de asistencia IRB5350 no se crean conjuntos de colisiones ya que se incluyen en los conjuntos de colisión de los robots de pintado, conociéndose así si existe alguna colisión entre ellos.

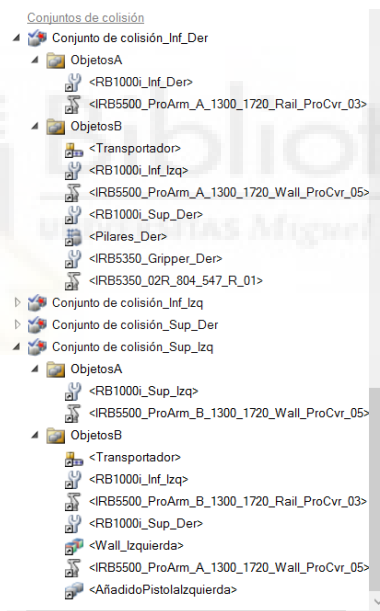


Figura 4.143 - Conjuntos de colisión de la estación

5. RESULTADOS OBTENIDOS

En este apartado se expondrá la ejecución de los programas de robot de los controladores virtuales de robot a lo largo de la simulación de la estación, de modo que se entienda como funciona la estación y por qué se ha diseñado y programado de esa forma.

Por otro lado, se expondrá la ejecución de las rutinas TRAP empleadas para las tareas de mantenimiento, limpieza y parada, a lo largo de su simulación.

5.1 Simulación del proceso de aplicación de la capa base de la carrocería

La simulación de la estación comienza con los robots y componentes posicionados como aparecen en la Figura 5.1. Al iniciarse, se coloca en el transportador la primera de las carrocerías. El transportador generará una nueva carrocería cada 7900 mm. Esta distancia se ha establecido después de varias pruebas, de modo que los robots tengan tiempo suficiente para completar el proceso de pintado y al terminar esperen en su posición inicial el mínimo tiempo a la siguiente carrocería.

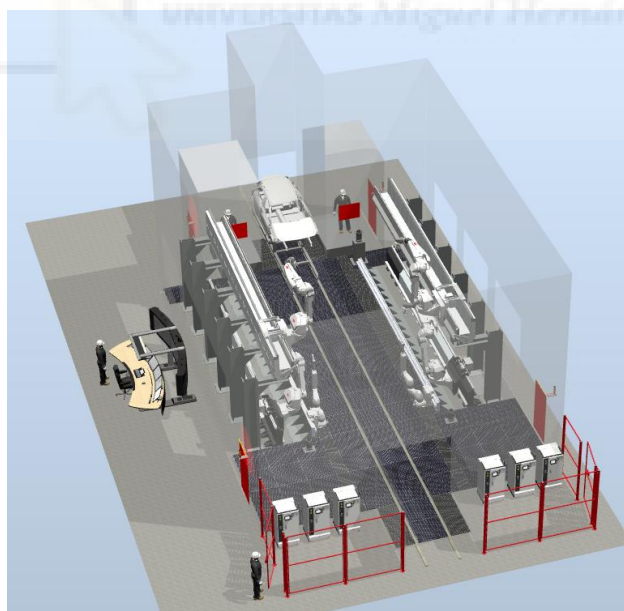


Figura 5.1 - Inicio de la simulación

En este punto, en cada uno de los controladores de los robots de pintado, IRB5500, se ha abierto y leído el archivo con los códigos de color **date+"_Lista_Colores.txt"**, guardado en la carpeta **HOME** de cada uno de los controladores, mediante la ejecución del procedimiento **Lectura_Colores**, obteniéndose así la secuencia de color de pintado y por lo tanto el número de

ciclos de pintado que se realizarán. Este archivo contiene una serie de números, entre el 2 y el 16, separados por espacios. Cada número indica un color de pintado diferente.

Se resetean todas las señales digitales de salida creadas para cada controlador.

Se han conectado también los identificadores y las señales con las rutinas TRAP y se ha establecido el valor de la variable **Color**, siendo este valor el código de color con el que se pintará la primera carrocería.

Por otro lado se ha pasado desde el controlador **Cont_Inf_Izq** el valor de la variable **Contador** a los controladores de los robots de asistencia IRB5350, haciendo así que estos también conozcan el número de ciclos de pintado a ejecutar, ya que en los controladores de estos robots, al no realizarse el pintado en sí, no se lee el archivo con los códigos de color.

Por último, se le comunica al componente inteligente **Aplicadores_Pintura** que debe obtener la variable **Color** de cada uno de los controladores de los robots de aplicación, para pasársela a los componentes inteligentes **PaintApplicator** que simularán el patrón de pintura.

Tras esto, los robots esperan en su posición hasta que la carrocería está posicionada por completo dentro de la **Ventana de inicio** del transportador. Cuando la carrocería llega a esta posición, el robot superior izquierdo comienza su movimiento para realizar la trayectoria de abertura del capó. Al mismo tiempo el robot superior derecho se posiciona en una posición intermedia de espera. (Figura 5.2)

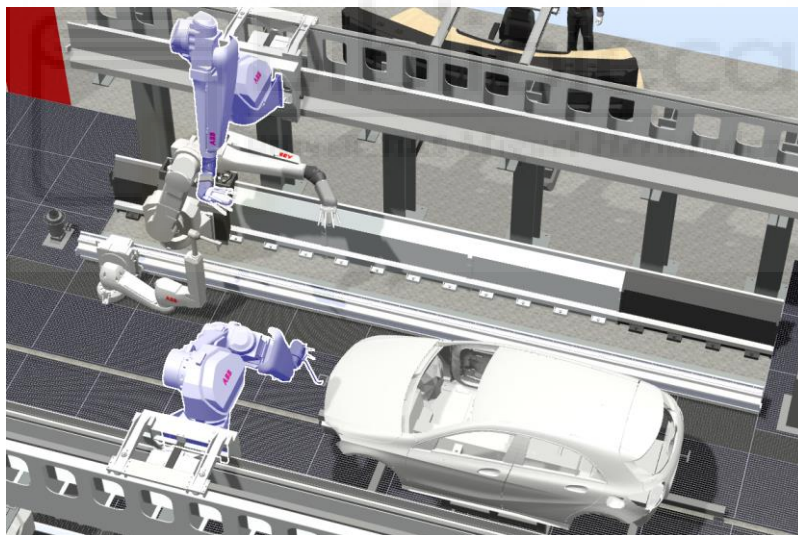


Figura 5.2 - Simulación - Abertura del capó

El robot superior izquierdo realiza la abertura del capó, en combinación con el componente inteligente **Abertura_Capó**. (Figura 5.3)

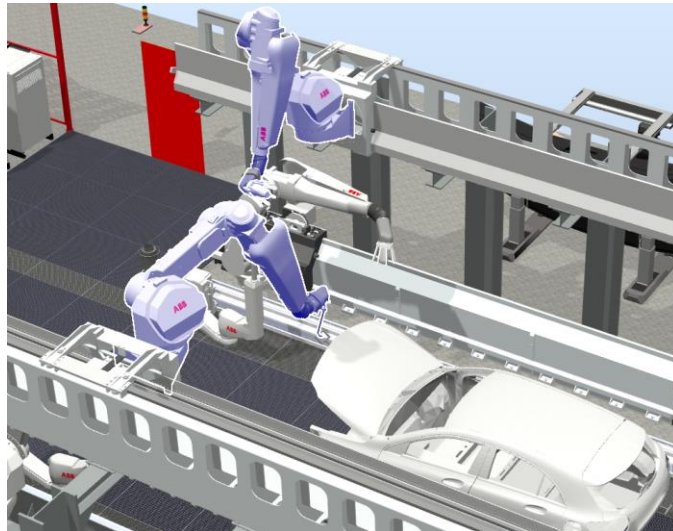


Figura 5.3 - Simulación - Abertura del capó 2

Cuando termina de abrir el capó, este robot se retira a una posición intermedia de espera y comunica a los robots inferiores de pintado que pueden comenzar la trayectoria de pintado de la zona del chasis situada debajo del capó. Primero comienza el robot inferior izquierdo, mientras el derecho espera unos segundos para no colisionar entre ellos mientras realizan el pintado de esa zona. (Figura 5.4)

Cabe destacar que los robots del lado izquierdo de la estación realizarán el proceso de aplicación de la capa base sobre la parte izquierda de la carrocería y los robots del lado derecho de la estación sobre la parte derecha de la carrocería. Los lados izquierdo y derecho se denotan desde la posición que se tendría desde dentro del vehículo.

Como ya se ha comentado en algunos apartados de este proyecto, una vez abierto el capó se mantendría en su posición abierta gracias a un elemento auxiliar hidráulico o neumático, permitiendo así que el robot superior izquierdo pueda realizar tareas de pintado mientras está abierto.

Cuando los robots de aplicación están realizando una trayectoria de pintado y alcanzan la instrucción **SetBrush** inicial, es decir el punto de disparo del atomizador, se activa la simulación del patrón de pintado, mostrando un cono del color especificado en la variable **Color**, en combinación con el componente inteligente **Aplicadores_Pintura**. Cuando alcanzan la instrucción **SetBrush** final se desactiva la simulación del patrón de pintado hasta la próxima trayectoria de pintado, donde se procede de la misma forma.

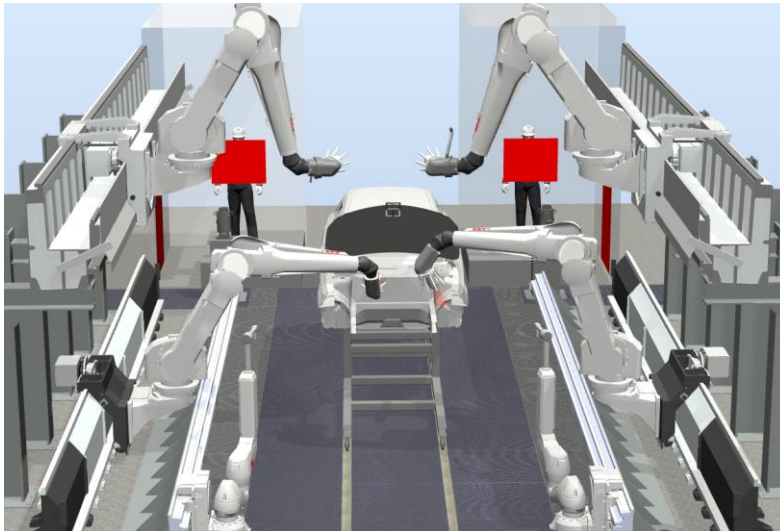


Figura 5.4 - Simulación - Pintado de la zona del chasis debajo de la carrocería

Cuando el robot inferior izquierdo termina esta zona, se mueve a una posición intermedia orientada al pintado interior de la puerta delantera izquierda. Cuando alcanza esa posición le comunica al robot IRB5350 izquierdo que puede comenzar su movimiento, y por lo tanto abrir la puerta delantera izquierda. (Figura 5.5)

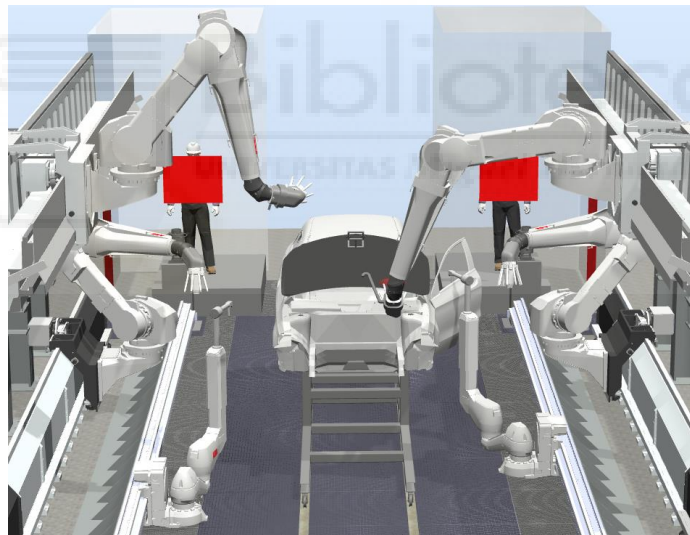


Figura 5.5 - Simulación - Abertura puerta izquierda delantera

Paralelamente a esto, cuando el robot inferior derecho termina de pintar la zona en la que estaba, les comunica a los robots superiores que pueden continuar su movimiento, y realizar el pintado de la superficie interior del capó. Primero comienza el robot superior izquierdo, mientras el derecho espera unos segundos para evitar que colisionen entre ellos al realizar la trayectoria de pintado.

Tras esto el robot inferior derecho, se mueve a una posición intermedia orientada al pintado interior de la puerta delantera derecha. Cuando alcanza esa posición le comunica al robot IRB5350 derecho que puede comenzar su movimiento, y por lo tanto abrir la puerta delantera derecha. (Figura 5.6)

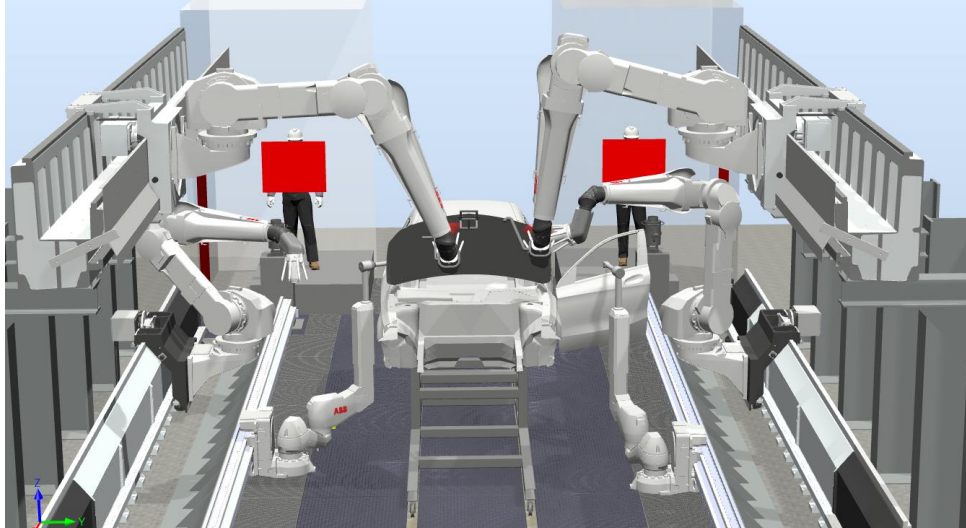


Figura 5.6 - Simulación - Abertura puerta derecha delantera y pintado de la superficie interior del capó

Cuando los robots de asistencia IRB5350 abren las puertas delanteras, cada uno en su momento, en combinación con los componentes inteligentes **Abertura_PIs** y **Abertura_PDs**, les comunican a los robots inferiores de aplicación que pueden comenzar las trayectorias de pintado interior de las puertas delanteras. Al mismo tiempo estos robots de asistencia se quedan manteniendo las puertas abiertas, desplazándose mediante su track pero sin variar la posición de sus ejes. (Figura 5.7)

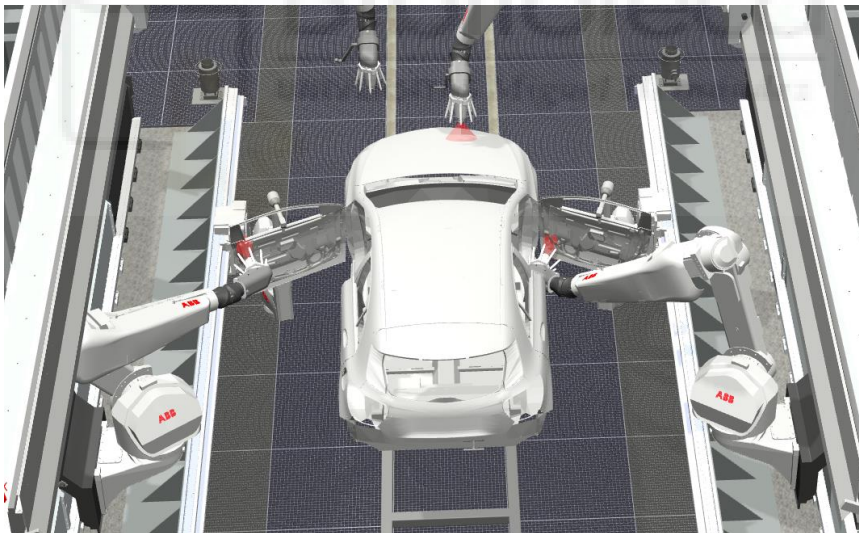


Figura 5.7 - Simulación - Pintado interior de las puertas delanteras

Paralelamente a esto, cuando el robot superior izquierdo termina el pintado de la superficie interior del capó, se mueve a una posición intermedia orientada a realizar la trayectoria de cierre del capó. Cuando el robot superior derecho termina de pintar esa superficie, se mueve a una posición intermedia orientada al pintado de la superficie exterior del capó, y le comunica al robot superior izquierdo que puede realizar la trayectoria para cerrar el capó, en combinación con el componente inteligente **Cierre_Capó**. (Figura 5.8)

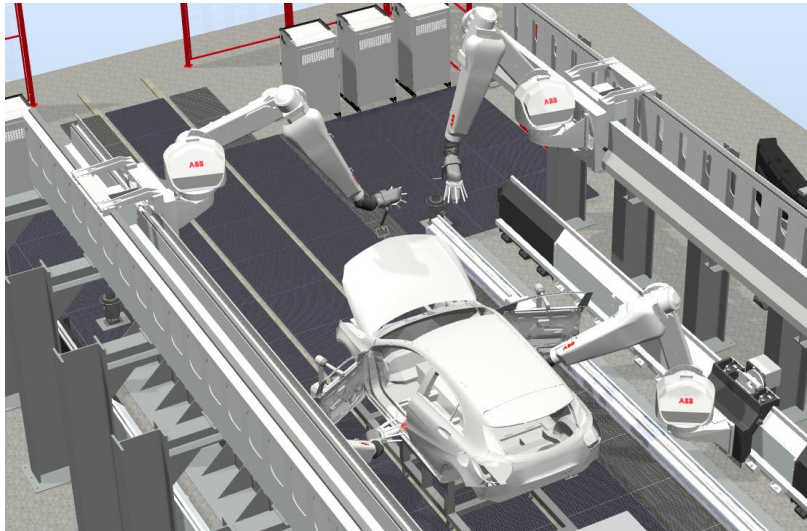


Figura 5.8 - Simulación - Cierre del capó

Cuando el robot superior izquierdo termina la trayectoria de cierre del capó, se mueve a una posición intermedia orientada al pintado de la superficie exterior del capó, y le comunica al robot superior derecho que puede comenzar el pintado de la superficie superior del capó. Tras esperar unos segundos, para evitar posibles colisiones, el robot superior izquierdo comienza el pintado de su parte de esta superficie. (Figura 5.9)

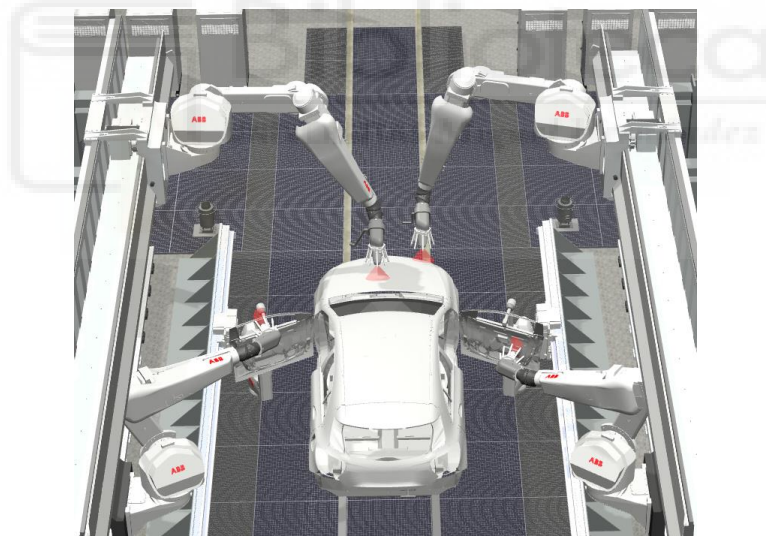


Figura 5.9 - Simulación - Pintado de la superficie exterior del capó

Paralelamente a esto, cuando los robots inferiores de aplicación terminan el pintado interior de las puertas, cada uno en su momento, se desplazan a una posición intermedia orientada al pintado del interior de las puertas traseras. Cuando alcanzan esta posición les comunican a sus respectivos robots de asistencia que pueden cerrar las puertas delanteras. (Figura 5.10)

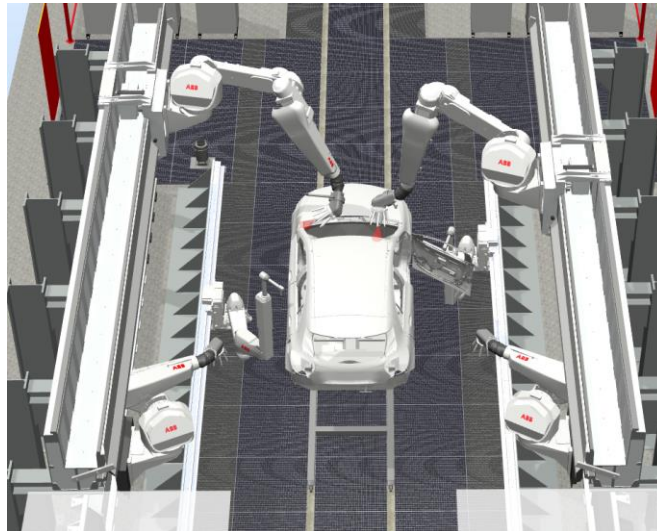


Figura 5.10 - Simulación - Cierre puertas delanteras, pintado barra y pintado techo

Cuando los robots de asistencia cierran las puertas delanteras, cada uno en su momento, en combinación con los componentes inteligentes **Abertura_PIs** y **Abertura_PDs**, se desplazan para realizar la apertura de las puertas traseras. (Figura 5.11)

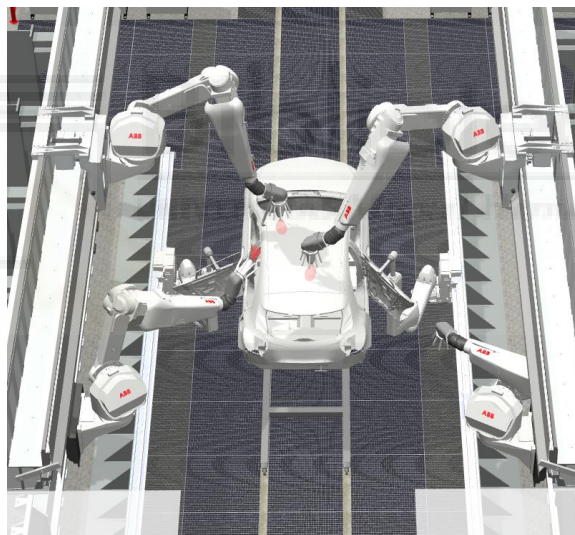


Figura 5.11 - Simulación - Abertura puertas traseras

Cuando los robots de asistencia IRB5350 abren las puertas traseras, cada uno en su momento, en combinación con los componentes inteligentes **Abertura_PIs** y **Abertura_PDs**, les comunican a los robots inferiores de aplicación que pueden comenzar las trayectorias de pintado interior de las puertas traseras. Al mismo tiempo estos robots de asistencia se quedan manteniendo las puertas abiertas, desplazándose mediante su track pero sin variar la posición de sus ejes. (Figura 5.12)

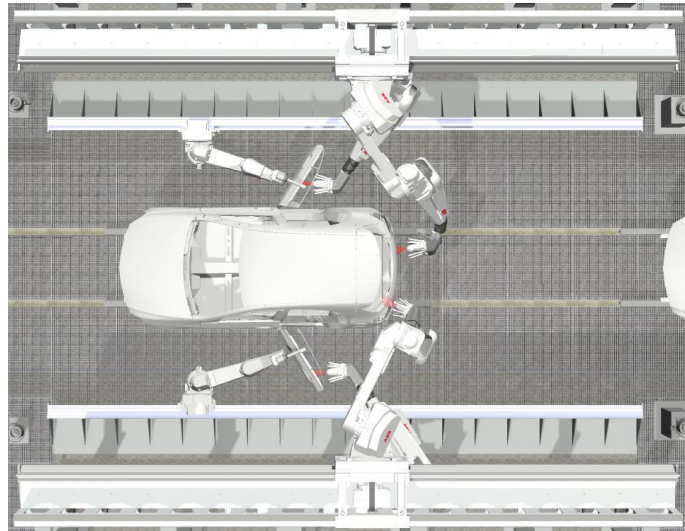


Figura 5.12 - Simulación - Pintado interior puertas traseras

Paralelamente a este proceso de pintado de las puertas, cuando los robots superiores terminan de pintar la superficie exterior del capó, cada uno es su momento, pasan a pintar la barra de su lado que une el pilar central de su lado del chasis con el techo. Una vez pintada esta barra, comienzan a pintar el techo de la carrocería. El robot superior izquierdo esperará unos segundos antes de comenzar el pintado de la barra y del techo, para así evitar posibles colisiones con el robot superior derecho durante el pintado del techo de la carrocería. (Figura 5.10 y 5.11)

Cuando terminan de pintar el techo de la carrocería pasan a pintar la superficie exterior del maletero. Primero comienza el robot superior derecho, y tras esperar unos segundos para evitar colisiones, comienza el robot superior izquierdo. (Figura 5.13)

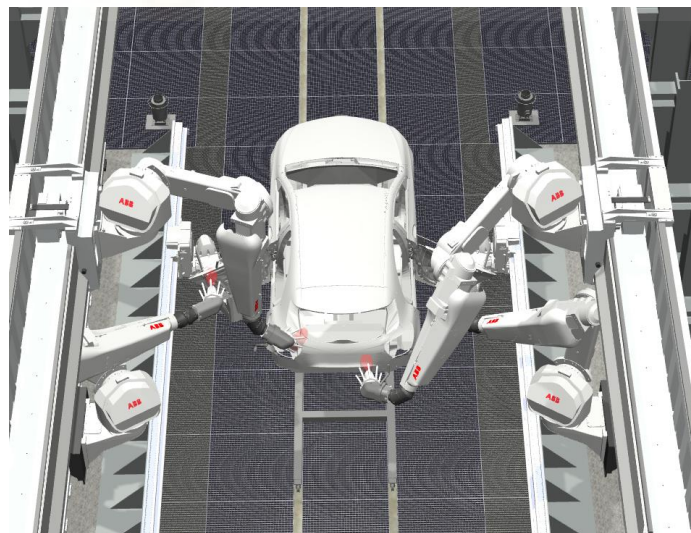


Figura 5.13 - Simulación - Pintado exterior del maletero

Cuando el robot superior derecho termina de pintar esta zona se desplaza a una posición intermedia orientada a realizar la trayectoria de apertura del maletero. Cuando el robot superior izquierdo termina de pintar el exterior del maletero le comunica al robot superior derecho que puede abrir el maletero, en combinación con el componente inteligente **Abertura_Maletero**.

Paralelamente a esto, cuando los robots inferiores de aplicación terminan de pintar el interior de las puertas se desplazan a una posición intermedia orientada a realizar la trayectoria de pintado de la superficie interior del maletero, y esperan a que el robot superior derecho realice la trayectoria de abertura del maletero. Al mismo tiempo también comunican a los robots de asistencia que pueden cerrar las puertas traseras. (Figura 5.14)

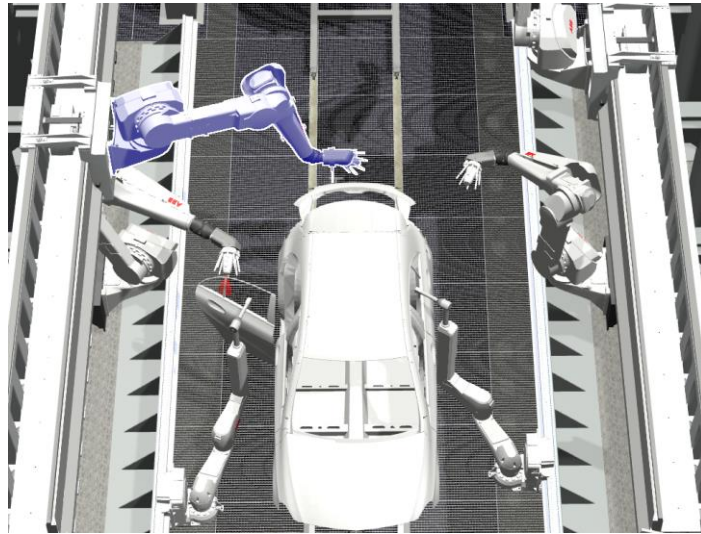


Figura 5.14 - Simulación - Cierre puertas traseras y abertura maletero

Por otro lado, cuando los robots de asistencia cierran las puertas, se desplazan a una posición intermedia de espera, donde no entorpecen el proceso de pintado.

Cuando el robot superior derecho abre el maletero se desplaza a una posición intermedia de reposo orientada a realizar posteriormente la trayectoria de cierre del maletero, en combinación con el componente inteligente **Abertura_Maletero**. Una vez abierto el maletero, este se mantendría en su posición abierta gracias a un elemento auxiliar hidráulico o neumático, permitiendo así que los robots inferiores puedan realizar las tareas de pintado de su superficie interior. Al mismo tiempo les comunica a los robots inferiores de aplicación que pueden comenzar el pintado de la superficie interior del maletero. (Figura 5.15)

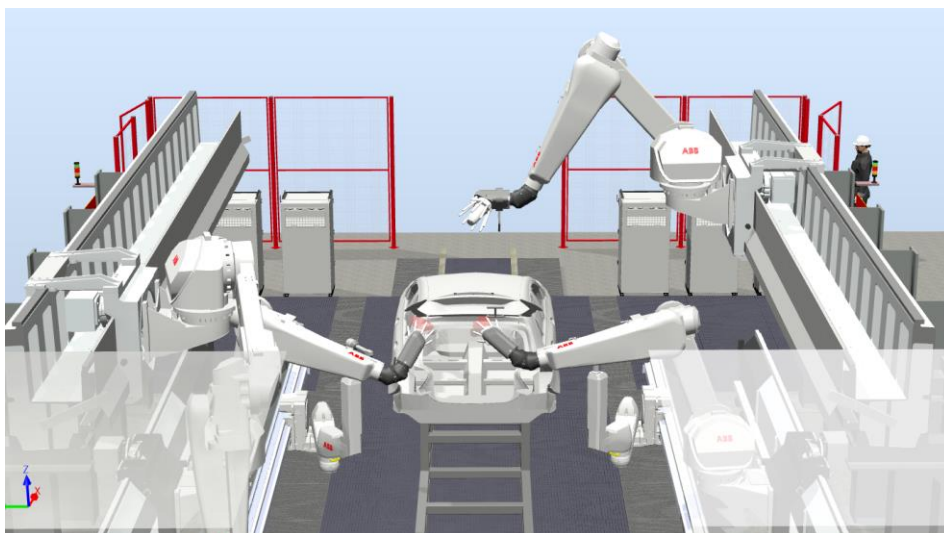


Figura 5.15 - Simulación - Pintado de la superficie interior del maletero

En este punto, el robot superior izquierdo ya ha terminado de ejecutar todos los procedimientos de pintado de la carrocería, por lo que en el caso de que el siguiente color de la secuencia de colores sea distinto al color con el que estaba pintado, se desplazada hasta la zona de limpieza y realiza la limpieza del atomizador. (Figura 5.16)

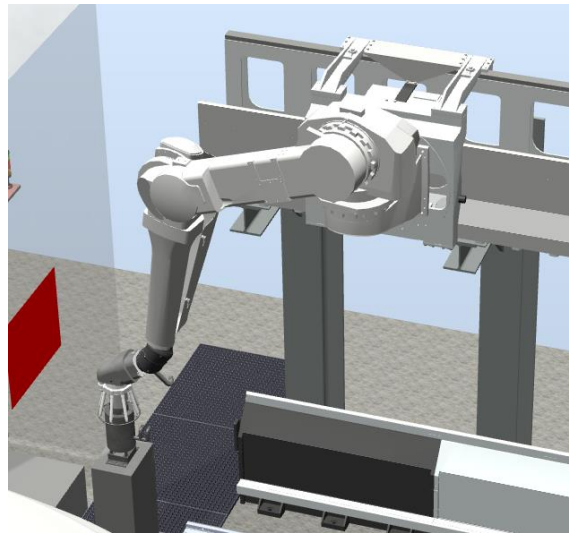


Figura 5.16 - Simulación - Limpieza del atomizador superior izquierdo

Una vez limpio, se desplaza a un punto lateral a la zona de pintado, cambia de color y expulsa un poco de pintura del nuevo color para dejar el atomizador listo para la siguiente carrocería. Tras esto se posiciona en su posición inicial (Figura 5.17). Si no se cambia de color para la siguiente carrocería, al terminar los procesos de pintado se desplaza a su posición inicial para esperar a la siguiente carrocería.

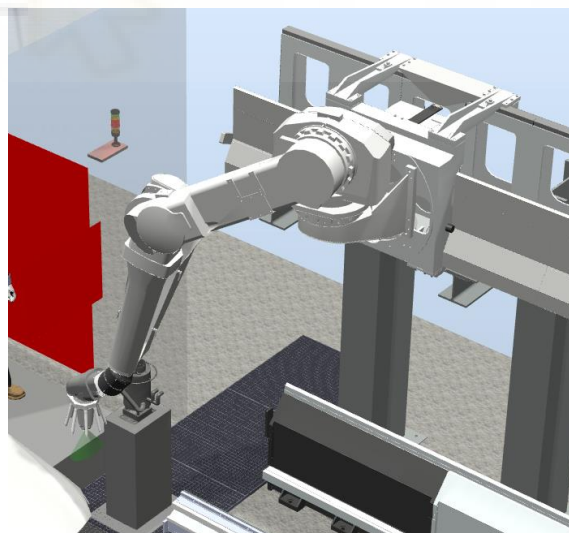


Figura 5.17 - Simulación – Cambio de color del atomizador superior izquierdo

Paralelamente a esto, cuando los robots inferiores han realizado aproximadamente la mitad del pintado de la superficie interior del maletero, comunican a los robots de asistencia IRB5350 que se desplacen hasta una nueva posición intermedia de espera, para que así no entorpezcan los posteriores procesos de pintado. (Figura 5.18)

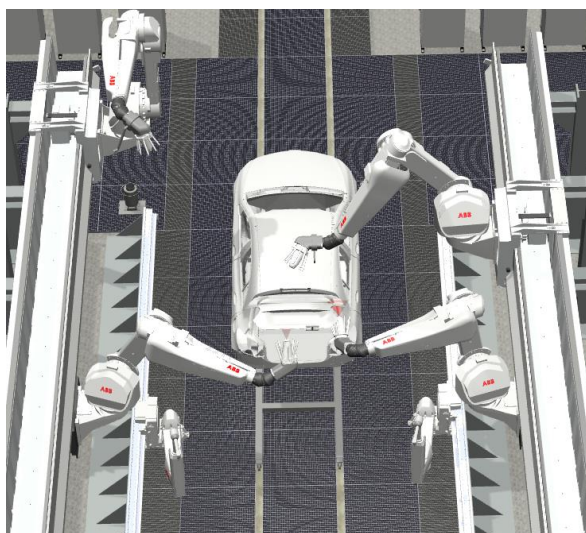


Figura 5.18 - Simulación - Desplazamiento de los robots IRB5350 a su nueva posición intermedia

Cuando el robot inferior izquierdo termina el pintado de su parte de la superficie interior del maletero se desplaza a una posición intermedia orientada al pintado de la zona de la rueda delantera izquierda y comienza el pintado de la misma (Figura 5.19).

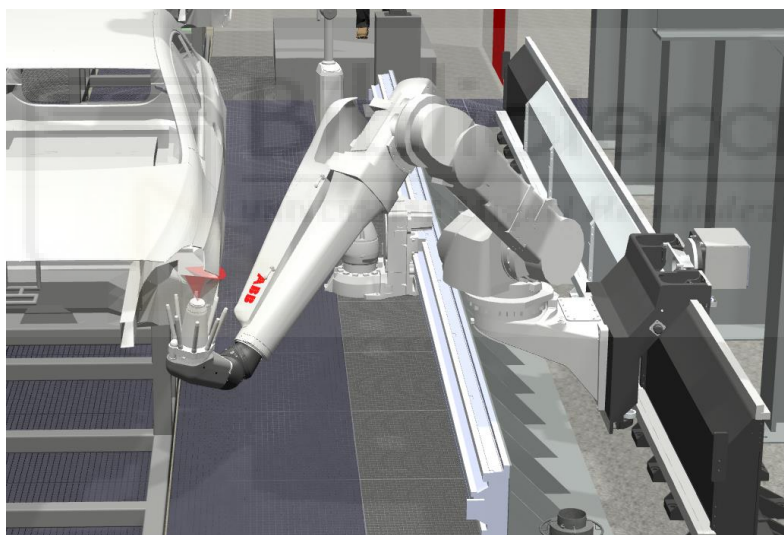


Figura 5.19 - Simulación - Pintado de la zona de la rueda delantera izquierda

Cuando el robot inferior derecho termina el pintado de su parte de la superficie interior del maletero, comunica al robot superior derecho que puede realizar la trayectoria de cierre del maletero, en combinación con el componente inteligente **Abertura_Maletero**. Al mismo tiempo se desplaza a una posición intermedia orientada al pintado de la zona de la rueda delantera derecha y comienza el pintado de la misma. (Figura 5.20 y Figura 5.21)

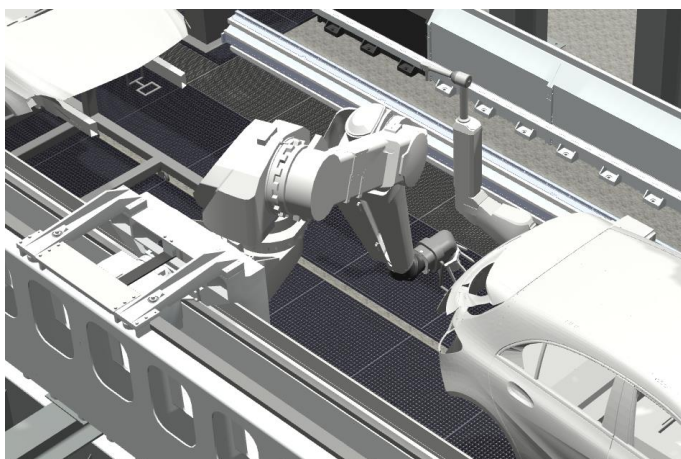


Figura 5.20 - Simulación - Cierre del maletero

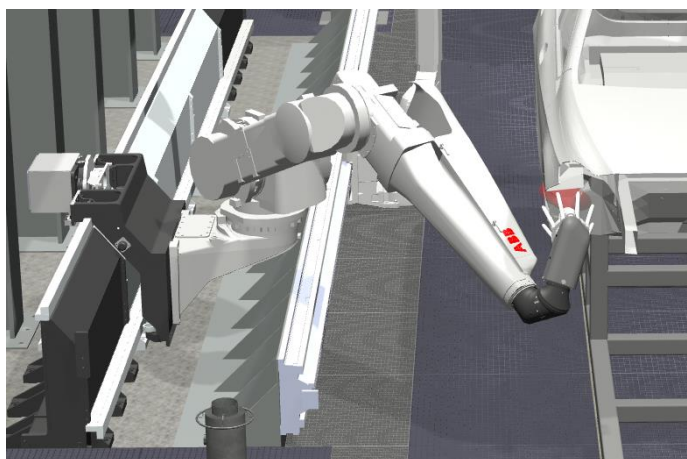


Figura 5.21 - Simulación - Pintado de la zona de la rueda delantera derecha

Con el cierre del maletero el robot superior derecho termina sus procesos de pintado y en el caso de que el siguiente color sea diferente al actual, realiza un proceso de limpieza del atomizador y cambio de color como el expuesto anteriormente para el robot superior izquierdo, pero en su respectivo limpiador del atomizador. Si no se cambia de color, el robot se desplaza hasta su posición inicial para esperar a la siguiente carrocería.

Paralelamente a esto, cuando los robots inferiores terminan de pintar las zonas de las ruedas delanteras, cada uno es su momento, comienza a pintar su lateral de la carrocería. (Figura 5.22)

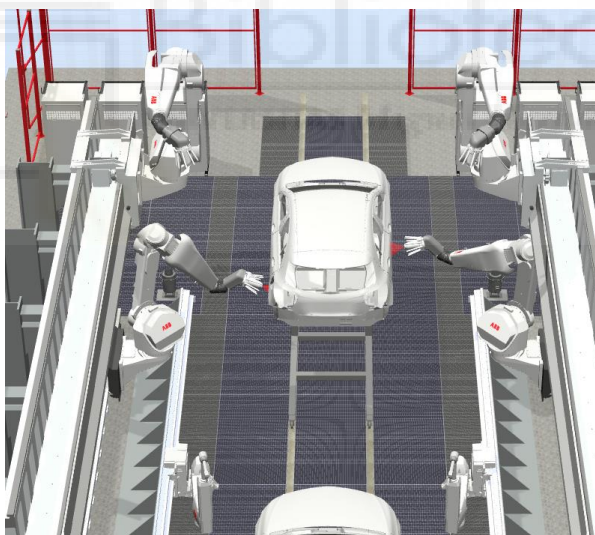


Figura 5.22 - Simulación - Pintado de los laterales de la carrocería

Cuando estos robots terminan el pintado de los laterales de la carrocería, comienzan a pintar la zona de las ruedas traseras. (Figura 4.23)

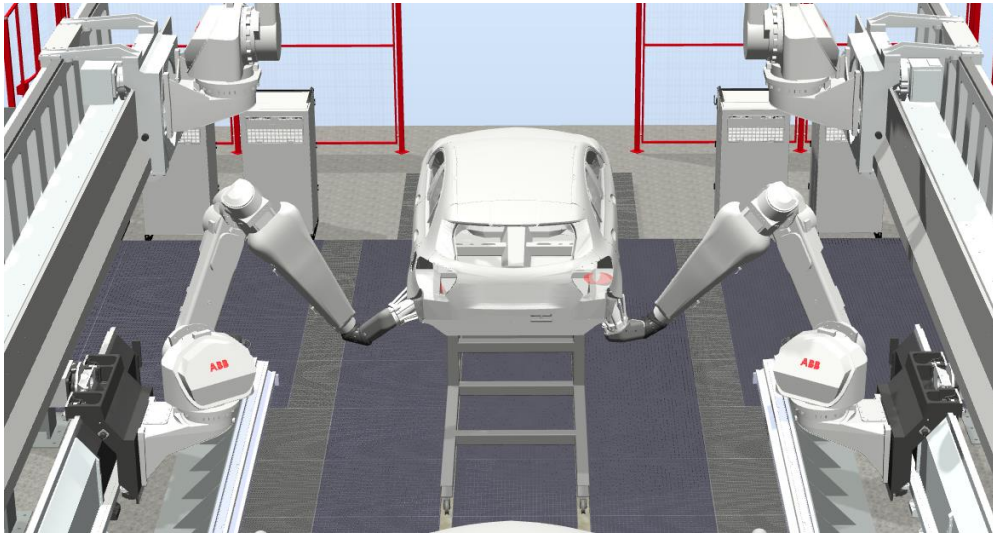


Figura 5.23 - Pintado de la zona de las ruedas traseras

Hecho esto, los robots inferiores terminan sus procesos de pintado y comunican a los robots de asistencia IRB5350 que pueden volver a su posición inicial para esperar a la siguiente carrocería.

Si el siguiente color es diferente al actual, cada uno de los robots inferiores de aplicación realiza un proceso de limpieza del atomizador y cambio de color como el expuesto anteriormente para el robot superior izquierdo, pero en su respectivo limpiador del atomizador. Si no se cambia de color, estos robots se desplazan hasta su posición inicial para esperar a la siguiente carrocería. (Figura 5.24 y Figura 5.25)

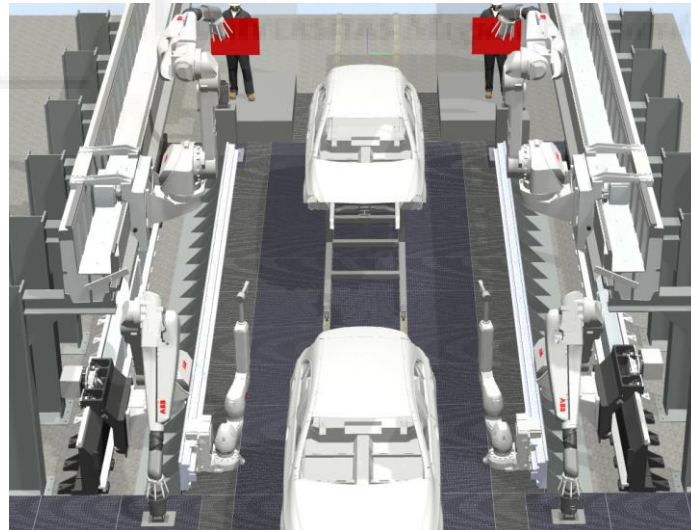


Figura 5.24 - Simulación - Limpieza de los atomizadores inferiores

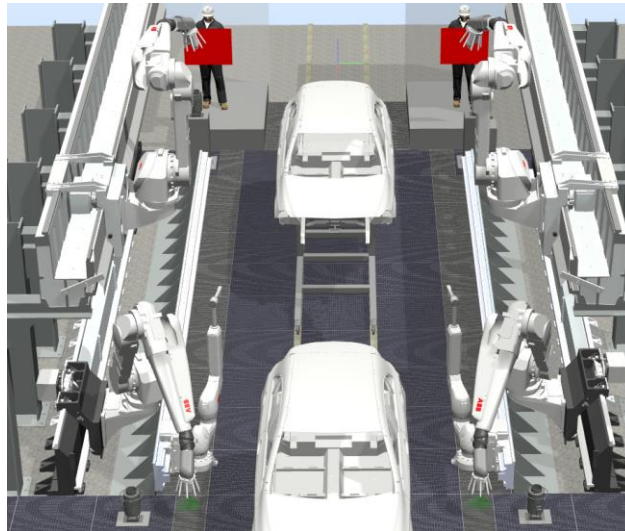


Figura 5.25 - Simulación - Cambio de color de los atomizadores inferiores

Tras esto se comenzaría un nuevo ciclo de pintado de la misma forma que el expuesto, hasta que se emplearán todos los colores de la secuencia de colores. La carrocería saldría de la estación en dirección a una etapa de secado de la misma.

Durante todo el proceso de pintado la señal analógica de salida **Señales_Luminosas** de todos los controladores tiene un valor igual a 3, lo cual indica peligro de acceso al interior de la estación, ya que todos los robots se encuentran en movimiento. Esta señal está conectada al conjunto de componentes inteligentes **Señales_Luminosas_Global**, expuesto en el apartado [4.3.8 Desarrollo de la lógica de estación](#), de forma que al tener ese valor hace que en las torres de señalización luminosa, situadas encima de las puertas, se ilumine su componente rojo, indicando el peligro, como se muestra en la Figura 5.26 y la Figura 5.27.



Figura 5.26 - Simulación - Torres de señalización luminosa

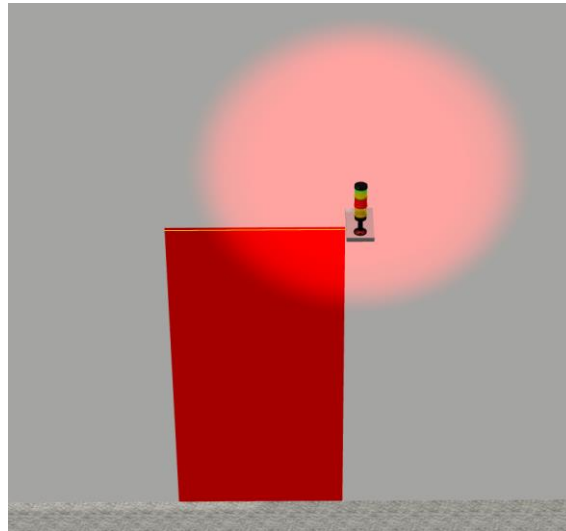


Figura 5.27 - Simulación - Torres de señalización luminosa 2

El tiempo de un ciclo completo de pintado, incluyendo el cambio de color y la limpieza del atomizador, es de 201,201 segundos aproximadamente. (Figura 5.28)

Cronómetro	
<input checked="" type="checkbox"/> Tiempo_Ciclo_Completo	<input type="button" value="Añadir"/>
Inicio:	Cont_Sup_Izq/Clock_ON = 1
Paro:	Cont_Inf_Der/Finalizado2 = 1
Conteo:	2
Tiempo total:	201,201
Tiempo medio:	100,601

Figura 5.28 - Simulación - Tiempo de un ciclo completo de pintado

El tiempo de un ciclo de pintado, pero en este caso refiriéndose solo al tiempo que tardan los robots en aplicar la capa base en toda la carrocería, sin tener en cuenta el cambio de color y la limpieza, es de 182.016 segundos aproximadamente. (Figura 5.29)

<input checked="" type="checkbox"/> Tiempo_Pintado	<input type="button" value="x"/>
Inicio:	Cont_Sup_Izq/T_ROB1/Abrir_Capo1
Paro:	Cont_Inf_Der/T_ROB1/B2_Paint_6630
Conteo:	1
Tiempo total:	182,016
Tiempo medio:	182,016

Figura 5.29 - Simulación - Tiempo de un ciclo de pintado sencillo

El tiempo de 10 ciclos completos de pintado, en el cual se ejecutan 6 procesos de cambio de color y limpieza, es de 1975.848 segundos aproximadamente, o lo que es lo mismo 32.931 minutos. Por lo que se puede afirmar que se pueden pintar aproximadamente 20 carrocerías por hora, siendo este uno de los requisitos iniciales. (Figura 5.30)

<input checked="" type="checkbox"/> Tiempo_10_Vehiculos	<input type="button" value="x"/>
Inicio:	Cont_Sup_Izq/T_ROB1/Abrir_Capo1
Paro:	Paro de simulación
Conteo:	9
Tiempo total:	1975,848
Tiempo medio:	197,483

Figura 5.30 - Simulación - Tiempo 10 ciclos de pintado completos

En el video del siguiente enlace puede observarse todo este proceso de aplicación de la capa base sobre la carrocería del vehículo.

<https://youtu.be/7NG9GEtgv2U>

5.2 Simulación de la rutina TRAP de limpieza

Si en cualquier momento durante la ejecución de los programas de robot, es decir, durante la simulación, se requiere la limpieza automática de alguno de los atomizadores de los robots de aplicación, debido a que se observa que no está funcionando correctamente, mediante la activación de la rutina TRAP **Limpieza_Atomizador** de su controlador se puede llevar a cabo.

A continuación se expone la simulación de esta rutina para el controlador del robot de aplicación inferior derecho **Cont_Inf_Der**. Para el resto de robots de aplicación se ejecutará de la misma forma, realizando la limpieza en sus respectivos limpiadores del atomizador.

Abriendo el **Simulador de E/S** de la pestaña **Simulación** se tiene acceso a las señales de los controladores de robot. Se selecciona el controlador **Cont_Inf_Der** y se selecciona la lista de usuarios **Lista1**, observándose así las señales de entrada y salida creadas durante el desarrollo del proyecto para ese controlador. (Figura 5.31)



Figura 5.31 - Señales de E/S del controlador Cont_Inf_Der

En el momento que se quiera ejecutar la rutina TRAP **Limpieza_Atomizador**, para llevar a cabo la limpieza automática del atomizador, se activa manualmente la señal digital de entrada **Limpieza_Atom**. En este ejemplo se activará cuando el robot está ejecutando la trayectoria de pintado del interior de la puerta derecha delantera, por ser una trayectoria más compleja y con menor hueco de maniobra para realizar la trayectoria de salida, uno de los casos más complejos en lo referente a evitar colisionar con la carrocería.

Al activarla, la ejecución del programa de robot se detiene, y se activa la ejecución de la rutina TRAP **Limpieza_Atomizador**. Esto hace que el robot se detenga lo más rápido posible y guarde la ubicación (posición y orientación) en la que se ha parado.

Al mismo tiempo activa la señal digital de salida **Paro_Resto**, mediante la cual se activará en el resto de robots de la estación la rutina TRAP **Parada**, mediante la cual los robots se detienen lo más rápido posible, guardan la ubicación del punto de parada, realizan una trayectoria de salida desde ese punto de paro hasta una posición de reposo, donde no entorpecen el proceso de limpieza del atomizador, y esperan para retomar su movimiento.

También se activa la señal **Stop_Conveyor**, que detiene el movimiento del transportador, y se cambia el color de la iluminación de las torres de señalización a ámbar para indicar que existe un cierto grado de peligro de acceso al interior de la estación ya que alguno de los robots está realizando una tarea de limpieza o mantenimiento.

Seguidamente el robot inferior derecho realizará una trayectoria de salida desde ese punto de paro hasta una posición intermedia cercana a la zona de limpieza del atomizador. Esta secuencia de acciones se muestra en las Figuras 5.32, 5.33, 5.34 y 5.35.

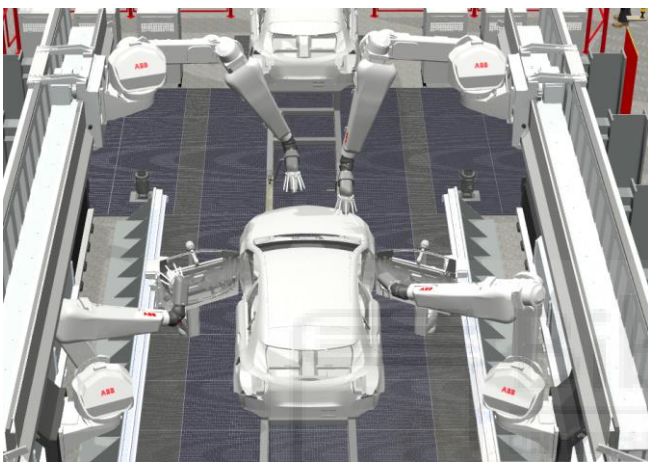


Figura 5.32 - Paro del robot

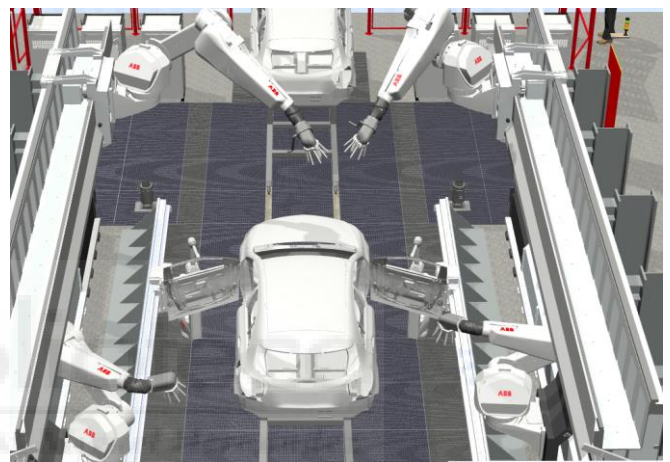


Figura 5.33 - Ejecución de la trayectoria de salida

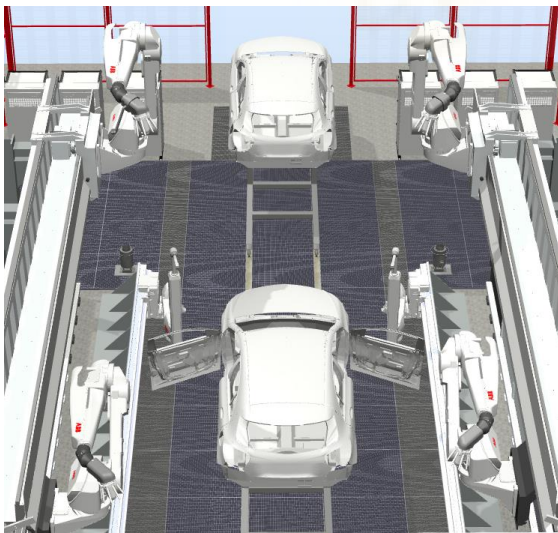


Figura 5.34 - Ejecución de la trayectoria de salida 2

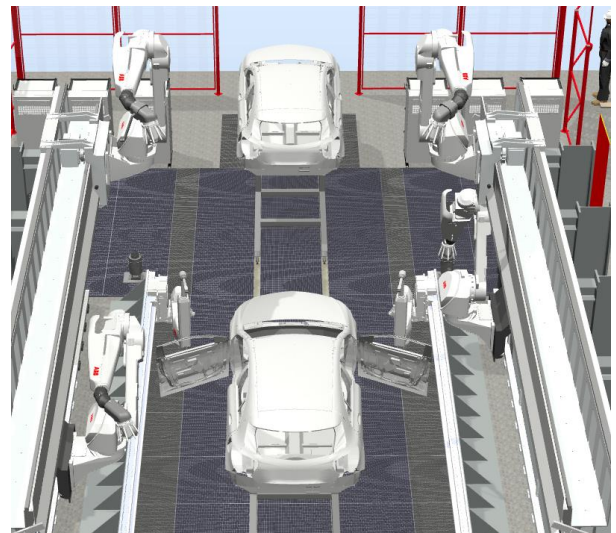


Figura 5.35 - Limpieza del atomizador inferior derecho

Una vez en esa posición, realizará el proceso de limpieza del atomizador y comprobará que funciona correctamente, expulsando un poco de pintura.

Después de esto volverá a un posición intermedia de reposo y esperará a que se desactive la señal digital de entrada **Limpieza_Atom**. En este punto se cambia la iluminación de las torres de

señalización a verde, indicando que todos los robots están en reposo y no hay peligro. (Figura 5.36 y 5.37)

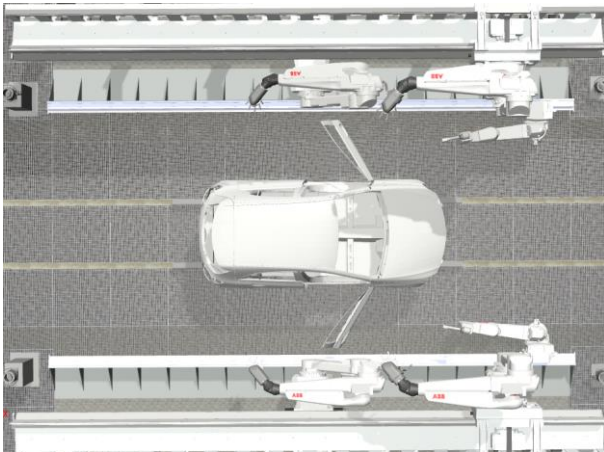


Figura 5.36 - Posición intermedia de reposo



Figura 5.37 - Iluminación de las torres de señalización en verde

Cuando se desactive esa señal, el robot realizará la trayectoria de salida en sentido inverso hasta el punto donde se detuvo. Al mismo tiempo se desactiva la señal de salida digital **Paro_Resto**, para que el resto de robots de la estación retomen el movimiento, hasta posicionarse de nuevo en el punto de paro. (Figura 5.38)

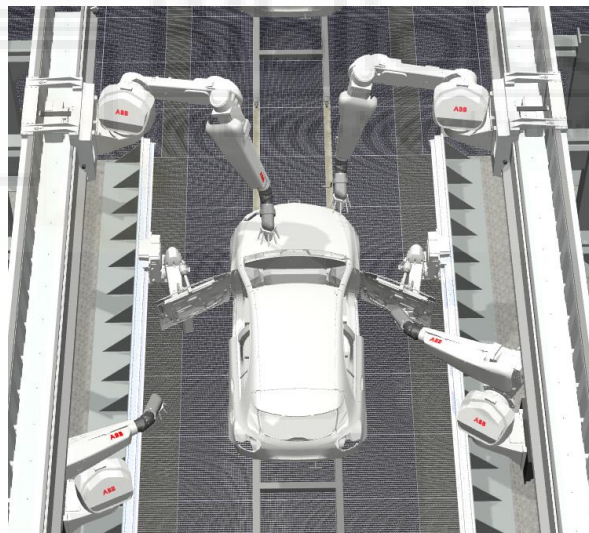


Figura 5.38 - Reposicionamiento en el punto de paro

Cuando todos están posicionados en el punto donde se detuvieron, se reanuda el movimiento del transportador y se reanuda la ejecución del programa de robot donde se detuvo.

En los videos de los siguientes enlaces pueden observarse dos ejemplos de esta rutina TRAP de limpieza, el primero para el robot inferior derecho y el segundo para el robot superior izquierdo.

https://youtu.be/5_hiHX8gzts

<https://youtu.be/rWHQHTOT3NQ>

5.3 Simulación de la rutina TRAP de mantenimiento

Si en cualquier momento durante la ejecución de los programas de robot, es decir, durante la simulación, se requiere realizar una operación de mantenimiento o una limpieza manual de alguno de los atomizadores de los robots de aplicación, debido a que se observa que no está funcionando correctamente, mediante la activación de la rutina TRAP **Mantenimiento_Atomizador** de su controlador se puede llevar a cabo.

A continuación se expone la simulación de esta rutina para el controlador del robot de aplicación inferior derecho **Cont_Inf_Der**. Para el resto de robots de aplicación se ejecutará de la misma forma, realizando el mantenimiento en sus respectivas zonas de mantenimiento.

Abriendo el **Simulador de E/S** de la pestaña **Simulación** se tiene acceso a las señales de los controladores de robot. Se selecciona el controlador **Cont_Inf_Der** y se selecciona la lista de usuarios **Lista1**, observándose así las señales de entrada y salida creadas durante el desarrollo del proyecto para ese controlador. (Figura 5.31)

En el momento que se quiera ejecutar la rutina TRAP **Mantenimiento_Atomizador**, para llevar a cabo el mantenimiento del atomizador, se activa manualmente la señal digital de entrada **Mantenimiento**. En este ejemplo se activará cuando el robot está ejecutando la trayectoria de pintado del interior de la puerta derecha delantera, por ser una trayectoria más compleja y con menor hueco de maniobra para realizar la trayectoria de salida, uno de los casos más complejos en lo referente a evitar colisionar con la carrocería.

Al activarla, la ejecución del programa de robot se detiene, y se activa la ejecución de la rutina TRAP **Mantenimiento_Atomizador**. Esto hace que el robot se detenga lo más rápido posible y guarde la ubicación (posición y orientación) en la que se ha parado.

Al mismo tiempo activa la señal digital de salida **Paro_Resto**, mediante la cual se activará en el resto de robots de la estación la rutina TRAP **Parada**, mediante la cual los robots se detienen lo más rápido posible, guardan la ubicación del punto de parada, realizan una trayectoria de salida desde ese punto de paro hasta una posición de reposo, donde no entorpecen el proceso de limpieza del atomizador, y esperan para retomar su movimiento.

También se activa la señal **Stop_Conveyor**, que detiene el movimiento del transportador, y se cambia el color de la iluminación de las torres de señalización a ámbar para indicar que existe un cierto grado de peligro de acceso al interior de la estación ya que alguno de los robots está realizando una tarea de limpieza o mantenimiento.

Seguidamente el robot inferior derecho realizará una trayectoria de salida desde ese punto de paro hasta una posición intermedia cercana a la zona de limpieza del atomizador. Esta secuencia de acciones se muestra en las Figuras 5.32, 5.33, 5.34 y 5.35.

Una vez en esa posición, realizará el proceso de limpieza del atomizador simple. Tras esto se posicionará orientándose hacia la zona de mantenimiento.

Se ejecuta el procedimiento **Mantenimiento_Atom**, de forma que el robot se traslada hasta una posición cercana a la trampilla de mantenimiento. Al llegar a esa posición, activa la abertura de la trampilla de la zona de mantenimiento y espera a que se abra completamente. (Figura 5.39)

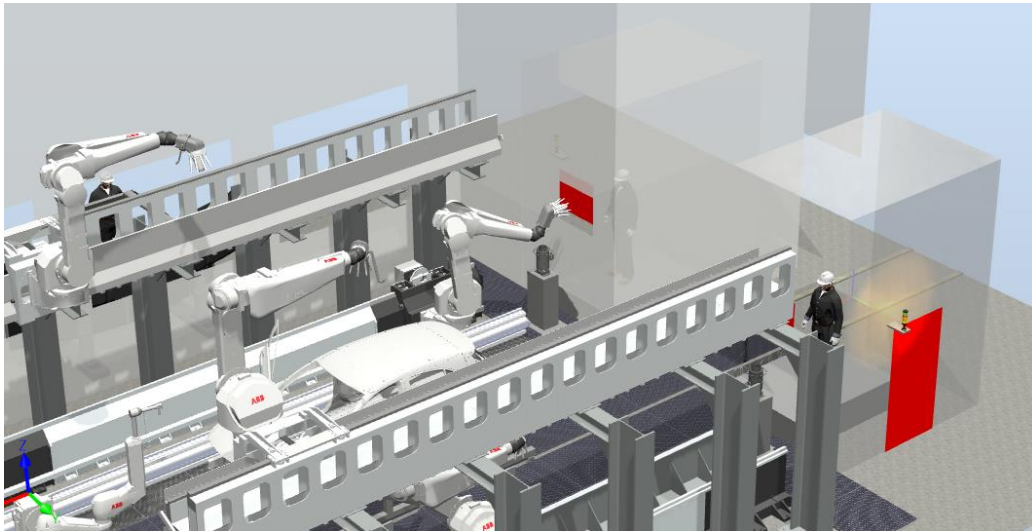


Figura 5.39 - Mantenimiento atomizador inferior derecho

Una vez abierta por completo la trampilla, el robot introduce el atomizador a través del hueco y espera hasta que la señal **Mantenimiento** se desactive, permitiendo así realizar el mantenimiento o limpieza manual del atomizador desde la zona de mantenimiento, sin acceder al interior de la estación. En este punto se cambia la iluminación de las torres de señalización a verde, indicando así que todos los robots están detenidos. (Figura 5.40)

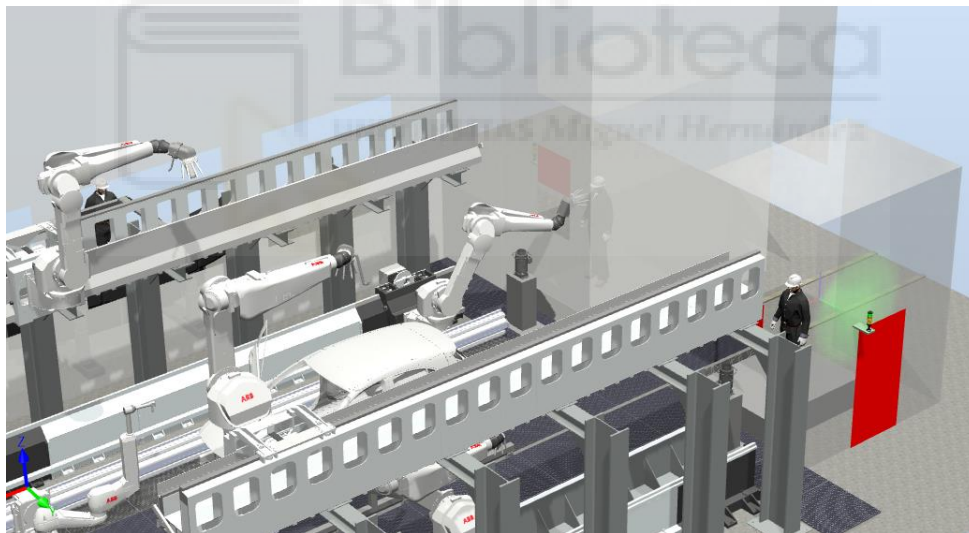


Figura 5.40 - Mantenimiento atomizador inferior derecho 2

Cuando se termina de realizar el mantenimiento del atomizador se desactiva manualmente la señal **Mantenimiento**, de modo que el robot saca el atomizador a través del hueco de la trampilla y una vez fuera activa el cierre de la misma.

Hecho esto, se traslada a una posición intermedia y desactiva la señal de salida digital **Paro_Resto**, para que el resto de robots de la estación retomen el movimiento, hasta posicionarse de nuevo en el punto de paro. Seguidamente se traslada el mismo hasta la posición donde se detuvo, realizando la trayectoria de salida en sentido inverso. (Figura 5.38)

Cuando todos están posicionados en el punto donde se detuvieron, se reanuda el movimiento del transportador y se reanuda la ejecución del programa de robot donde se detuvo. (Figura 5.41)

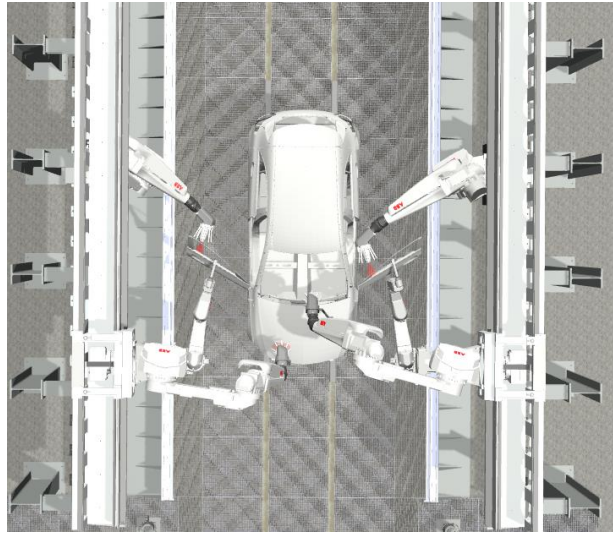


Figura 5.41 - Reanudación del movimiento de todos los robots donde se detuvieron

En los videos de los siguientes enlaces pueden observarse tres ejemplos de esta rutina TRAP de mantenimiento, el primero para el robot inferior izquierdo, el segundo para el robot superior derecho y el tercero para el robot de asistencia izquierdo.

https://youtu.be/_VRDjiozjkr4

<https://youtu.be/xtRkbFfciD8>

<https://youtu.be/qr9vOmDGYeo>

5.4 Simulación de la rutina TRAP de seguridad de puertas

Si en cualquier momento durante la ejecución de los programas de robot, es decir, durante la simulación, se abre una de las puertas de acceso al interior de la estación todos los robots de la estación se detienen inmediatamente, se trasladan a una posición de reposo y esperan a que la puerta se cierre. De esta forma si alguien entra de manera no prevista se tiene la seguridad de que no habrá un accidente.

Esto se lleva a cabo mediante la activación de la rutina TRAP **Paro_Puerta** del controlador **Cont_Sup_Izq**. Esta rutina se activa al activarse la salida de uno de los sensores que están colocados sobre las puertas de acceso a la estación, es decir, cuando una de las puertas corta el sensor de seguridad se activa la rutina TRAP **Paro_Puerta**.

A continuación se expone la simulación de esta rutina. Al rotarse, es decir, abrirse una de las puertas de acceso a la estación, esta corta el sensor colocado en la parte superior de la misma, activando la señal de entrada **P_Puerta** del contralor (Figura 5.42)

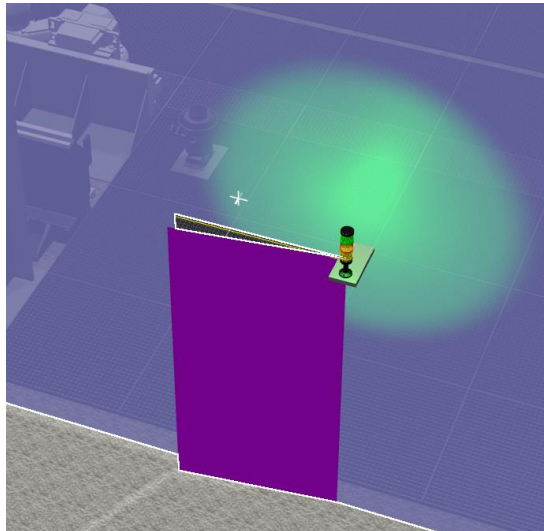


Figura 5.42 - Paro debido a la abertura de una de las puertas de acceso a la estación

Al activarse esta señal, la ejecución del programa de robot superior izquierdo se detiene, y se activa la ejecución de la rutina TRAP **Paro_Puerta**. Esto hace que el robot se detenga lo más rápido posible y guarde la ubicación (posición y orientación) en la que se ha parado.

Al mismo tiempo activa la señal digital de salida **Paro_Resto**, mediante la cual se activará en el resto de robots de la estación la rutina TRAP **Parada**, mediante la cual los robots se detienen lo más rápido posible, guardan la ubicación del punto de parada, realizan una trayectoria de salida desde ese punto de paro hasta una posición de reposo y esperan para retomar su movimiento.

También se activa la señal **Stop_Conveyor**, que detiene el movimiento del transportador, y se cambia el color de la iluminación de las torres de señalización a ámbar para indicar que existe un cierto grado de peligro de acceso al interior de la estación.

Seguidamente el robot superior izquierdo, donde se activó la rutina TRAP **Paro_Puerta**, realizará una trayectoria de salida desde ese punto de paro hasta una posición de reposo. Cuando todos los robots están en la posición de reposo se cambia la iluminación de las torres de señalización a verde. (Figura 5.43)

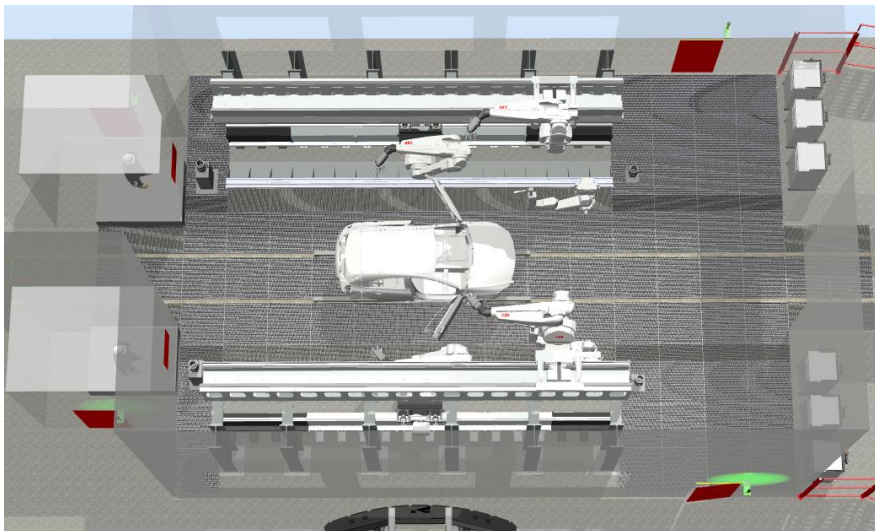


Figura 5.43 - Robots esperando en la posición de reposo

Los robots esperarán en estas posiciones de reposo hasta que se cierre la puerta y se desactive así la señal digital de entrada **P_Puerta**.

Cuando se cierra la puerta, el robot comienza su trayectoria de regreso al punto donde se detuvo y al mismo tiempo desactiva la señal de salida digital **Paro_Resto**, para que el resto de robots de la estación retomen el movimiento, hasta posicionarse de nuevo en el punto de paro.

Cuando todos están posicionados en el punto donde se detuvieron, se reanuda el movimiento del transportador y se reanuda la ejecución del programa de robot donde se detuvo. (Figura 5.41)

En el video del siguiente enlace puede observarse la rutina TRAP Paro_Puerta de seguridad.

<https://youtu.be/V4G81-wPPW0>

5.5 Simulación de la rutina TRAP paro de emergencia

Para dotar a la estación de un sistema de paro de emergencia se han creado dos señales de entrada de la estación, **Emergency_Stop** y **Reset_Stop**. Activando la primera de ellas se realiza una parada de emergencia de la estación, deteniendo el movimiento todos los robots y posicionándolos en una posición intermedia de reposo hasta que se active la señal de **Reset_Stop**.

A continuación se expone la simulación de esta rutina. Cuando se pulsa en el **Simulador de E/S**, de la pestaña **Simulación**, la señal de entrada de la estación **Emergency_Stop**, esta se activa momentáneamente para después desactivarse automáticamente. Al activarse momentáneamente se activa, momentáneamente también, la señal de entrada **Set** del componente inteligente **LogicSRLatch**, lo cual produce la activación permanente de su señal de salida **Output**. Esta señal de salida está conectada con la señal digital de entrada **P_Puerta**, del controlador **Cont_Sup_Izq**. Esta señal activa en ese controlador la rutina TRAP **Paro_Puerta**, expuesta en el apartado anterior, mediante la cual se detiene el movimiento de los robots y el transportador y se repositionan los robots en posiciones intermedias de reposo hasta que se desactive dicha señal **P_Puerta**.

Cuando se quiere reiniciar el movimiento de la estación, se pulsa en el **Simulador de E/S**, de la pestaña **Simulación**, la señal de entrada de la estación **Reset_Stop**, la cual se activa momentáneamente para activar momentáneamente la señal de entrada **Reset** del **LogicSRLatch**, que desactiva permanentemente su señal de salida **Output**, y por lo tanto desactiva la señal **P_Puerta** del controlador **Cont_Sup_Izq**, reanudándose así el movimiento del transportador y todos los robots de la estación en el punto y procedimiento donde se detuvieron.

En el video del siguiente enlace puede observarse la ejecución de una parada de emergencia.

<https://youtu.be/go0UV4yrOm8>



6. CONCLUSIONES Y LÍNEAS FUTURAS DE TRABAJO

6.1 Conclusiones

Una vez finalizado el presente proyecto de fin de máster, se concluye que se han alcanzado los objetivos impuestos al comienzo del mismo y se han acatado los requisitos de diseño exigidos para la estación robótica de aplicación de la capa base.

El objetivo principal del presente proyecto era el diseño y programación de una estación robótica completa, en este caso particular, una estación robótica de pintado en seguimiento para la aplicación de la capa base del recubrimiento de la carrocería del modelo de automóvil Mercedes Benz Clase A A45 AMG, de modo que mediante su desarrollo se adquiriera un nivel relativamente avanzado de habilidades y conocimientos acerca del software de programación robótica RobotStudio.

Analizando la estación diseñada y programada en el presente proyecto se puede concluir que efectivamente se ha alcanzado ese nivel de habilidades y conocimientos, resultado esto muy útil de cara a realizar futuros trabajos de programación de estaciones robóticas fuera de línea.

Aun así, RobotStudio es un software muy potente, con un gran número de características y herramientas, en muchos aspectos, además de diversos PowerPacs para diversas aplicaciones, con lo que abarcar el conocimiento del software al completo no ha sido posible en el presente proyecto, ya que esto conllevaría realizar un estudio en profundidad mucho más extenso.

Además, el diseño y programación de la estación permiten llevar a cabo dicha aplicación de la capa base a base de agua, tanto externa como internamente, en seguimiento, permitiendo además, gracias a su diseño, su reprogramación para emplearse con otros modelos de vehículo, siempre que no sean de grandes dimensiones.

Por otro lado, mediante la programación de las rutinas TRAP de limpieza, mantenimiento y seguridad se ha conseguido dotar a la estación robótica de pintado de un alto grado de flexibilidad en estos aspectos. Uno de los aspectos más relevantes de la estación del presente proyecto es que el mantenimiento de las herramientas de los robots puede llevarse a cabo sin necesidad de acceder al interior de la estación de pintado, característica que no todas las estaciones de pintado robótico poseen. Esto evitaría la contaminación del ambiente interior de la estación, mejorando así la calidad del pintado.

También se ha conseguido diseñar una estación de pintado muy compacta, que permite realizar la aplicación completa (interior y exterior) de la capa base de la carrocería del vehículo en tan solo 14 metros. Esto se ha conseguido gracias a la utilización de seis robots de pintura, tres a cada lado de la misma, y un diseño que se ha optimizado en varias etapas.

El tiempo de ciclo de pintado obtenido se considera adecuado y correcto, si bien siempre podría reducirse. Para poder reducirlo se tendría que superar la limitación de no conocerse muchos de los aspectos que afectarían al desarrollo de una estación real de pintado robótico, impuesta al no desarrollarse el presente proyecto en el marco de una empresa real.

Para el desarrollo del presente proyecto se han empleado muchas de las herramientas y características que ofrece RobotStudio, por lo que también se ha tenido que hacer frente a un gran número de dificultades, que iban apareciendo a medida que el proyecto aumentaba su dificultad.

Estas dificultades se han conseguido superar mediante sucesivas etapas de prueba y error, programando muchas estaciones auxiliares en las que probar ciertas herramientas o cambios, estudiando en profundidad los manuales expuestos en la bibliografía y visualizando video tutoriales en internet. Cabe señalar que la gran mayoría de estos video tutoriales solo tratan aspectos básicos de RobotStudio y que los manuales, puesto que son manuales técnicos, no están orientados al aprendizaje, resultado en algunos momentos muy poco didácticos.

6.2 Líneas futuras de trabajo

La principal línea de trabajo futura que podría seguirse sería la investigación orientada a implementar la estación diseñada en una estación real, es decir, conocer todos los aspectos y señales que intervendrían en una estación real de pintado para así poder mejorar la programación de la estación de este proyecto.

Por ejemplo, el atomizador empleado en el presente proyecto contiene un gran número de sensores orientados al control de calidad del pintado y mejora de las trayectorias, que en la simulación de RobotStudio no pueden conocerse. Tampoco se tiene conocimiento de las señales necesarias para llevar a cabo el proceso de cambio de color o el proceso de limpieza del atomizador. A lo largo del desarrollo del proyecto se han intentado conocer todos estos aspectos y algunos otros, pero ha sido imposible encontrar información acerca de ellos. Por lo que llevando la investigación a una estación de pintado real sería posible conocerlos y así mejorar el diseño de la presente estación.

Otra línea de trabajo sería investigar la optimización de las trayectorias de pintado programadas sobre la carrocería, es decir, optimizar el tamaño del patrón de pintado y las velocidades de aplicación para conseguir un acabado óptimo. Este es un campo de investigación bastante amplio, motivo por el cual no se ha desarrollado en profundidad en el presente proyecto.

Relativo a esto último, el atomizador seleccionado en este proyecto RB1000i, cuenta con una función denominada Super Pattern Control, mediante la cual se controla y adapta el flujo del aire de los dos anillos de aire de conformado, lo cual permite un gran control sobre el tamaño del patrón de pintado. Este sería un aspecto interesante a conocer.

Puesto que los robots superiores de la estación finalizan sus procedimientos de pintado antes que los robots inferiores, podría estudiarse la posibilidad de incluir entre las sucesivas carrocerías otros componentes del vehículo, como por ejemplo los parachoques, para que estos robots superiores realicen el pintado de los mismos en lugar de estar esperando a la siguiente carrocería.

Por último, se podría indagar más en la utilización del Painting PowerPac, ya que parece una herramienta muy potente, pero de la cual no se ha encontrado mucha información relevante.

7. Bibliografía

- Hans-Joachim Streitberger y Karl-Friedrich Dössel, 2008. **Automotive Paints and Coatings**. Weinheim, Alemania: WILEY-VCH Verlag GmbH & Co. KGaA.
- Mei Wen y Karel Dusek, 2017. **Protective Coatings. Film Formation and Properties**. Cham, Suiza:Springer International Publishing AG.
- Massimo Callegari, Luca Carbonari, Giacomo Palmieri y Matteo Palpacelli, 2013. **Parallel Wrists for Enhancing Grasping Performance**. Ancona, Italia: Publicaciones de la Universidad Politécnica de Marche.
- Nelson K. Akafuah, Sadegh Poozesh, Ahmad Salaimah, Gabriela Patrick, Kevin Lawler y Kozo Saito, 2016. **Evolution of the Automotive Body Coating Process—A Review**. MDPI, Coatings, 6, 24.
- Barrientos, Antonio, Peñín, Luis Felipe, Balaguer, Carlos y Aracil, Rafael, 2007. **Fundamentos de robótica**. Madrid, España: McGRAW-HILL/INTERAMERICANA DE ESPAÑA, S. A. U.
- ABB Robotics, 2018. **Application manual - Conveyor tracking**. Västerås, Suecia: ABB.
- ABB Robotics, 2007. **Manual del operador - Introducción a RAPID**. Västerås, Suecia: ABB.
- ABB Robotics, 2018. **Manual de referencia técnica - Parámetros del sistema**. Västerås, Suecia: ABB.
- ABB Robotics, 2018. **Manual de referencia técnica - Instrucciones, funciones y tipos de datos de RAPID**. Västerås, Suecia: ABB.
- ABB Robotics, 2020. **Manual del operador - RobotStudio**. Västerås, Suecia: ABB.
- ABB Robotics, 2018. **Manual de referencia técnica - Descripción general de RAPID**. Västerås, Suecia: ABB.
- ABB Robotics, 2019. **Operating manual - Painting PowerPac**. Västerås, Suecia: ABB.

Páginas web empleadas

<https://grabcad.com/library>

<https://3dwarehouse.sketchup.com/>

<https://new.abb.com/products/robotics/es/robotstudio/tutoriales>

<https://forums.robotstudio.com/>

<https://new.abb.com/products/robotics/es/>

<https://www.automotivemanufacturingsolutions.com/robotics/history-in-the-making/39755.article>

https://www.robotics.org/content-detail.cfm/Industrial-Robotics-Industry-Insights/The-Art-of-Industrial-Painting-with-Robots/content_id/4447

<http://www.lechler.eu/es/Home/Refinish/Ciclos-especiales-y-defectos-de-pintura/Manual-de-Defectos-de-Pinturas>

https://www.robotics.org/content-detail.cfm/Industrial-Robotics-Tech-Papers/Advances-in-Robotic-Finishing/content_id/783

<https://ifr.org/>

Video tutoriales

https://www.youtube.com/watch?v=-T8YE9S8bHg&list=PLUqHJL1jGsj88HEj3k1njQw9sTPiCzjW4&index=2&t=10434s&ab_channel=Ariza

https://www.youtube.com/watch?v=00ScFQ1rbe4&list=PLUqHJL1jGsj88HEj3k1njQw9sTPiCzjW4&index=12&ab_channel=AutoMotoTV

<https://www.youtube.com/playlist?list=PLWNJtOoGWRb2UTVIB2emftHZX0GzbNTkM>

<https://www.youtube.com/playlist?list=PLUqHJL1jGsj9cnT-nnsP6v6Yx5f3KQgDV>

<https://www.youtube.com/playlist?list=PLVdvHpsfqw1bX194j7Slup6ri5se2668p>

Videos finales del presente proyecto

Video de la aplicación de la capa base completo: <https://youtu.be/7NG9GEtgv2U>

Video de la rutina TRAP Limpieza_Atomizador para el robot inferior derecho: https://youtu.be/5_hiHX8gzts

Video de la rutina TRAP Limpieza_Atomizador para el robot superior izquierdo: <https://youtu.be/rWHQHT0T3NQ>

En los videos de los siguientes enlaces pueden observarse tres ejemplos de esta rutina TRAP de mantenimiento, el primero para el robot inferior izquierdo, el segundo para el robot superior derecho y el tercero para el robot de asistencia izquierdo.

Video de la rutina TRAP Mantenimiento_Atomizador para el robot inferior izquierdo: <https://youtu.be/VRDjozjkr4>

Video de la rutina TRAP Mantenimiento_Atomizador para el robot superior derecho: <https://youtu.be/xtRkbFfciD8>

Video del proceso de mantenimiento para el robot de asistencia izquierdo: <https://youtu.be/qr9vOmDGYeo>

Video de la rutina TRAP Paro_Puerta de seguridad: <https://youtu.be/V4G81-wPPW0>

Video de la ejecución de una parada de emergencia: <https://youtu.be/go0UV4yrOm8>



8. Anexos

8.1 Código RAPID de los programas de robot

8.1.1 Controlador virtual Cont Inf Izq

```
MODULE Module1
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! Módulo: Module1
!
! Descripción: Procesos para la aplicación de la capa base sobre la carrocería empleando el robot inferior izquierdo
!
! Autor: Arturo López López
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

!!!!!!!!!!!!!!!!!!!!!!!!!!!! DEFINICIÓN DE OBJETIVOS !!!!!!!!!!!!!!!!!!!!!!!

LOCAL CONST robtarget Paint_10:=[[3650,3.292,1125],[0.720581562,0.02943484,-
0.692071672,0.030538552],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_20:=[[3665.05,1.52,1326.48],[0.684076315,0.001149285,-0.729403902,-
0.002867515],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_30:=[[3767.72,3.22,1390.35],[0.684076315,0.001149285,-0.729403902,-
0.002867515],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_40:=[[3701.4,7.75,1460.75],[0.684076315,0.001149285,-0.729403902,-
0.002867515],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_50:=[[3744.817122877,5.269462202,1722.473137285],[0.684076315,0.001149285,-
0.729403902,-0.002867515],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_60:=[[3803.32,7.23,1763.22],[0.684076315,0.001149285,-0.729403902,-
0.002867515],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_70:=[[3620.08,2.45,1844.6],[0.617901459,0.000894991,-0.786249479,-
0.00295677],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_80:=[[3602.8,185.09,1850.97],[0.617901459,0.000894991,-0.786249479,-
0.00295677],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_90:=[[3799.66,178.48,1765.56],[0.684076315,0.001149285,-0.729403902,-
0.002867515],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_100:=[[3735.427122877,181.569462202,1725.673137285],[0.684076315,0.001149285,-
0.729403902,-0.002867515],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_110:=[[3701.56,187.78,1443.38],[0.684076315,0.001149285,-0.729403902,-
0.002867515],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_120:=[[3767.78,187.63,1389.69],[0.684076315,0.001149285,-0.729403902,-
0.002867515],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_130:=[[3665.52,189.11,1327.69],[0.684076315,0.001149285,-0.729403902,-
0.002867515],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_140:=[[3644.74825352,186.781228313,1179.923008902],[0.721440494,-0.060078527,-
0.689859862,0.002748554],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_150:=[[3685.402407322,360.000338508,1189.819207554],[0.738787794,-0.079269426,-
0.660914846,-0.105359005],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_160:=[[3770.2,379.18,1354.85],[0.661064437,-0.175941854,-0.703807899,-
0.191553428],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_170:=[[3699.964541187,390.079425998,1739.128208616],[0.678373959,-0.088150424,-
0.722789464,-0.098049297],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_180:=[[3805.84,377.6,1763.14],[0.67388323,-0.117656781,-0.717824679,-
0.129483609],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_190:=[[3637.333338031,372.111505033,1861.695220472],[0.563848782,-0.163319536,-
0.785161725,0.197287472],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_200:=[[3583.097685649,630.03721161,1856.039669937],[0.569780653,-0.136448387,-
0.793522865,0.164478899],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_210:=[[3690.04,624,1758.65],[0.397919544,-0.226994567,-
0.857545657,0.233984933],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];
```

LOCAL CONST robtarget Paint_220:=[[3690.832287021,767.169160792,1807.446103911],[0.506564902,-0.023059184,-0.857041022,0.091328862],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_230:=[[4135.78,715.03,1701.56],[0.49031036,-0.134728211,-0.846699089,0.156667519],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_240:=[[4203.94,720,1664.69],[0.49031036,-0.134728211,-0.846699089,0.156667519],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_250:=[[4085.92,664.64,1646.12],[0.150158992,-0.627020222,-0.761900657,0.061687167],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_260:=[[4078.3,579.98,1576.41],[0.191466237,-0.615276175,-0.764383745,0.022212565],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_270:=[[4083.45,505.34,1450.93],[0.322321929,-0.554891807,-0.720943396,-0.26161857],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_280:=[[4081.7,467.9,1422.27],[0.137362766,-0.963133863,-0.219864002,0.07186413],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_290:=[[4544.32,471.96,1422.27],[0.137362766,-0.963133863,-0.219864002,0.07186413],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_300:=[[4570.92,437.57,1365.62],[0.298836239,-0.640856382,-0.640856382,-0.298836239],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_310:=[[3822.93,437.57,1364.24],[0.5,-0.500000001,-0.704416026,-0.061628417],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_320:=[[3855.49,482.16,1422.27],[0.5,-0.500000001,-0.704416026,-0.061628417],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_330:=[[3841.78,490.9,1544.21],[0.5,-0.500000001,-0.704416026,-0.061628417],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_340:=[[3848.94,488.36,1680.33],[0.5,-0.500000001,-0.704416026,-0.061628417],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_350:=[[3934.55,523.6,1640.31],[0.5,-0.500000001,-0.704416026,-0.061628417],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_360:=[[3958.66,648.12,1688.68],[0.5,-0.500000001,-0.704416026,-0.061628417],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_370:=[[3940.15,598.08,1750.94],[0.5,-0.500000001,-0.704416026,-0.061628417],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_380:=[[3827.68,499.89,1750.94],[0.436703447,-0.50346862,-0.745313382,-0.017816031],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_390:=[[3843.61,691.22,1750.81],[0.436703447,-0.50346862,-0.745313382,-0.017816031],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_400:=[[3875.032,1150,1756.043],[0.436702463,-0.503469106,-0.745313706,-0.017812856],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];

CONST robtarget Pos_Int1:=[[2335.776012341,1815.077101406,1392.253270232],[0.141544932,0.157712638,0.97668622,0.034289702],[1,-1,0,-1],[520,9E+09,9E+09,9E+09,9E+09,0]];

CONST robtarget Aprox_Bajo_Capo:=[[4400,3.292,1125],[0.720581562,0.02943484,-0.692071672,0.030538552],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_410:=[[2376.844785733,764.733770226,1695.386831978],[0.438730793,0.245963269,-0.77778473,0.376932367],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_420:=[[2312.67,691.56,1968.28],[0.432030574,0.298901899,-0.759522255,0.383579433],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_430:=[[2287.07,641.5,2087.13],[0.397401149,0.331547218,-0.766101023,0.381101025],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_440:=[[2272.42,604.53,2151.99],[0.386767188,0.31445532,-0.735768585,0.458446926],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_450:=[[2281.82,587.9,2176.45],[0.410723784,0.257647336,-0.696505169,0.528965379],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_460:=[[2304.5,573.68,2190.46],[0.412775619,0.170089774,-0.630027655,0.635413968],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_470:=[[2336.23,567.14,2192.78],[0.638517464,0.266583223,-0.47153446,0.546702924],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_480:=[[2407.04,566.68,2197.39],[0.637652054,0.412211971,-0.366091928,0.53801287],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_490:=[[2463.04,572.32,2196.55],[0.63513548,0.417737099,-0.288175931,0.58228281],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_500:=[[2595.18,581.84,2178.51],[0.598059876,0.483365502,-0.331658894,0.546520406],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_510:=[[2731.77,602.35,2153.38],[0.576667863,0.518141883,-0.294111048,0.559000765],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_520:=[[2833.1,619.19,2116.2],[0.322458593,0.438974945,-0.448936304,0.708362653],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_530:=[[2932.22,642.86,2077.02],[0.256025856,0.553017094,-0.395689027,0.687061169],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_540:=[[3012.13,659.95,2038.07],[0.271183932,0.524669574,-0.360918946,0.721746928],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_550=[[3087.81,679.95,2010.34],[0.270015207,0.498349667,-0.291928851,0.770400509],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_560=[[3155.97,690.01,1970.86],[0.293350652,0.529569017,-0.270018443,0.748726981],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_570=[[3213.63,704.91,1941.12],[0.239667875,0.537987626,-0.236411406,0.772811925],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_580=[[3260.17,716.25,1905.39],[0.209325259,-0.333546242,0.48081389,-0.783420604],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_590=[[3305.23,724.18,1863.21],[0.420505352,-0.305487723,0.666518693,-0.534420558],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_600=[[3322.19,721.62,1815.28],[0.616858223,0.167525985,0.731626369,-0.2369469],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_610=[[3330.05,717.35,1745.04],[0.652518914,0.137440968,0.73244113,-0.137328212],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_620=[[3377.6,745.75,1718.38],[0.652518914,0.137440968,0.73244113,-0.137328212],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_630=[[3396.65,754.15,1643.3],[0.652518914,0.137440968,0.73244113,-0.137328212],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_640=[[3394.31,742.72,1552.78],[0.652518914,0.137440968,0.73244113,-0.137328212],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_650=[[3387.82,736.58,1443.32],[0.652518914,0.137440968,0.73244113,-0.137328212],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_660=[[3374.99,737.58,1338.62],[0.652518914,0.137440968,0.73244113,-0.137328212],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_670=[[3351.94,732.23,1279.11],[0.444927124,0.231849192,0.864266934,-0.036448235],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_680=[[3308.57,740.63,1231.18],[0.349584618,0.22082684,0.907782792,-0.070402448],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_690=[[3228.25,735.35,1203.88],[0.355191364,0.128678803,0.912550478,-0.156628497],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_700=[[3102.27,731.2,1201.02],[0.36168534,0.168361222,0.906069041,-0.140986192],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_710=[[2895.11,731.82,1207.34],[0.316612848,0.240567762,0.89810252,-0.187870488],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_720=[[2771.43,727.35,1214.32],[0.202619176,0.184721358,0.933157891,-0.232464707],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_730=[[2696.01,727.64,1228.88],[0.145662622,0.15593862,0.945912442,-0.244366936],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_740=[[2638.29,732.33,1246.29],[0.133580783,0.157196716,0.949509723,-0.23638243],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_750=[[2566.98,733.13,1278.61],[0.140352237,0.230419675,0.92051683,-0.282589436],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_760=[[2482.94,731.02,1342.75],[0.168061556,-0.137117549,-0.955501053,0.199929561],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_770=[[2420.89,714.7,1418.55],[0.226769787,-0.138275422,-0.926052553,0.268108262],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_780=[[2383.26,720.19,1510.7],[0.282930228,-0.000277414,-0.919666298,0.27233125],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_790=[[2372.88,731.15,1577.19],[0.301074077,-0.039762137,-0.902412483,0.305655169],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_800=[[2353.85,734.31,1795.6],[0.243119091,-0.002008995,-0.9173333,0.315259714],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_810=[[2303.248479984,711.72228907,1984.419803286],[0.322718167,-0.038930407,-0.909704645,0.258408333],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_820=[[2288.079349258,498.236478167,2343.126025944],[0.357631061,0.373943438,-0.671108714,0.53092318],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_830=[[2235.19,608.61,2199.66],[0.280379965,0.383288266,-0.72132341,0.504152476],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_840=[[2237.69,635.07,2154.24],[0.2886697,0.375195931,-0.742715896,0.473572503],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_850=[[2256.91,690.12,2027.68],[0.35026964,0.310373163,-0.722570523,0.508794181],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_860=[[2282.18,723.31,1927.08],[0.34493694,0.310532924,-0.73347342,0.496592945],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_870=[[2311.74,763.66,1778.33],[0.357797683,0.284474273,-0.741098393,0.491760488],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_880=[[2330.96,767.1633.07],[0.503394696,0.323219008,-0.681639468,0.421296676],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_890=[[2355.31,775.05,1460.57],[0.475954137,0.364469468,-0.678588585,0.424437508],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_900:=[[2382.67,784.39,1382.12],[0.040423178,0.134681434,-0.749793268,0.646557757],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_910:=[[2410.65,794.68,1299.71],[0.007386624,0.161700054,-0.721148798,0.67360444],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_920:=[[2527.54,815.96,1207.32],[0.019186782,-0.185836716,0.723676974,-0.664370544],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_930:=[[2701.35,816.61,1185.09],[0.07262804,-0.223337992,0.749193643,-0.619317524],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_940:=[[2987.85,814.99,1170.52],[0.14571116,-0.279372701,0.744213748,-0.588952502],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_950:=[[3283.038833624,819.909514841,1181.509713198],[0.437388433,0.072135606,0.885025829,-0.14218683],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_960:=[[3343.713701534,964.537353666,1204.65430373],[0.425565978,0.547234322,0.593886076,0.408322818],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_970:=[[3297.3,1095.74,1207.99],[0.52482024,0.470194445,0.499976149,0.503492552],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_980:=[[3214.5,1234.07,1198.47],[0.476728358,0.39947443,0.461262672,0.632761408],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_990:=[[3170.47,1308.29,1195.17],[0.476728358,0.39947443,0.461262672,0.632761408],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1000:=[[3118.49,1401.16,1199.68],[0.476728358,0.39947443,0.461262672,0.632761408],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1010:=[[3079.25,1467.57,1198.17],[0.476728358,0.39947443,0.461262672,0.632761408],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1020:=[[2971.98,1653.62,1204.22],[0.476728358,0.39947443,0.461262672,0.632761408],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1030:=[[2881.65,1810.48,1214.32],[0.476728358,0.39947443,0.461262672,0.632761408],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1040:=[[2906.17,1820.93,1332.86],[0.365050148,0.478002035,0.608135257,0.51809647],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1050:=[[2910.59,1786.45,1331.55],[0.365050148,0.478002035,0.608135257,0.51809647],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1060:=[[2921.4,1742.89,1327.68],[0.365050148,0.478002035,0.608135257,0.51809647],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1070:=[[2944.59,1705.9,1323.62],[0.365050148,0.478002035,0.608135257,0.51809647],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1080:=[[2930.67,1664.44,1321.03],[0.365050148,0.478002035,0.608135257,0.51809647],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1090:=[[2916.09,1582.45,1318.44],[0.365050148,0.478002035,0.608135257,0.51809647],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1100:=[[2978.66,1474.13,1306.52],[0.365050148,0.478002035,0.608135257,0.51809647],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1110:=[[3131.53,1210.56,1299.3],[0.365050148,0.478002035,0.608135257,0.51809647],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1120:=[[3245.34,1014.57,1306.7],[0.365050148,0.478002035,0.608135257,0.51809647],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1130:=[[3324.29,970.04,1306.55],[0.365050148,0.478002035,0.608135257,0.51809647],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1140:=[[3376.46433182,931.228804395,1304.818798891],[0.391229189,0.500146133,0.591633183,0.496753202],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_1150:=[[3377.412450513,927.101752603,1450.603214624],[0.386051384,0.495792156,0.595024731,0.501098829],[0,0,0,0],[520,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_1160:=[[3319.37,949.27,1437.42],[0.365050148,0.478002035,0.608135257,0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_1170:=[[3255.35,981.24,1435.02],[0.365050148,0.478002035,0.608135257,0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_1180:=[[3200.79,1075.98,1439.54],[0.365050148,0.478002035,0.608135257,0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_1190:=[[3129.23,1195.85,1445.25],[0.365050148,0.478002035,0.608135257,0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_1200:=[[2876.75,1629.39,1467.22],[0.365050148,0.478002035,0.608135257,0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_1210:=[[2856.45,1651.51,1463.14],[0.365050148,0.478002035,0.608135257,0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_1220:=[[2933.49,1741.08,1467.49],[0.365050148,0.478002035,0.608135257,0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_1230:=[[2908.85,1811.46,1468.94],[0.365050148,0.478002035,0.608135257,0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_1240:=[[2904.4,1837.92,1638.31],[0.365050148,0.478002035,0.608135257,0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_1250:=[[2875.66,1643,1625.84],[0.365050148,0.478002035,0.608135257,0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_1260:=[[2962.59,1492.85,1615.86],[0.365050148,0.478002035,0.608135257,0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_1270:=[[3111.04,1237.2,1605.02],[0.365050148,0.478002035,0.608135257,0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_1280:=[[3262.46,979.01,1586.35],[0.365050148,0.478002035,0.608135257,0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_1290:=[[3336.31,937.7,1591.36],[0.365050148,0.478002035,0.608135257,0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_1300:=[[3387.29744835,917.671099178,1602.495721842],[0.365050148,0.478002035,0.608135257,0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_1310:=[[3350.62744835,904.261099178,1748.365721842],[0.365050148,0.478002035,0.608135257,0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_1320:=[[3320.44,952.24,1743.21],[0.365050148,0.478002035,0.608135257,0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_1330:=[[3230.1,1033.35,1744.11],[0.365050148,0.478002035,0.608135257,0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_1340:=[[3118.01,1220.54,1749.96],[0.365050148,0.478002035,0.608135257,0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_1350:=[[2991.38,1437.59,1757.48],[0.365050148,0.478002035,0.608135257,0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_1360:=[[2873.83,1638.56,1768.11],[0.365050148,0.478002035,0.608135257,0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_1370:=[[2812.22,1726.57,1767.85],[0.365050148,0.478002035,0.608135257,0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg
 Paint_1380:=[[2844.86,1792.44,1776.8],[0.365050148,0.478002035,0.608135257,0.51809647],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg
 Paint_1390:=[[2874.83,1845.41,1783.45],[0.365050148,0.478002035,0.608135257,0.51809647],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg
 Paint_1400:=[[2787.95,1806.28,1874.62],[0.365050148,0.478002035,0.608135257,0.51809647],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg
 Paint_1410:=[[2892.17,1589.53,1844.89],[0.365050148,0.478002035,0.608135257,0.51809647],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg
 Paint_1420:=[[3049.78,1323.62,1819.36],[0.365050148,0.478002035,0.608135257,0.51809647],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg
 Paint_1430:=[[3168.23,1129.43,1813.01],[0.365050148,0.478002035,0.608135257,0.51809647],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg
 Paint_1440:=[[3289.29,960.08,1811.49],[0.365050148,0.478002035,0.608135257,0.51809647],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg
 Paint_1450:=[[3319.94,912.45,1833.66],[0.391229189,0.500146134,0.591633183,0.496753201],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg
 Paint_1460:=[[3248.13,1005.04,1865.47],[0.365050148,0.478002035,0.608135257,0.51809647],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg
 Paint_1470:=[[3203.26,1045.31,1920.23],[0.365050148,0.478002035,0.608135257,0.51809647],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg
 Paint_1480:=[[3164.62,1068.04,1982.87],[0.365050148,0.478002035,0.608135257,0.51809647],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg
 Paint_1490:=[[3094.04,1148.06,2035.47],[0.365050148,0.478002035,0.608135257,0.51809647],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg
 Paint_1500:=[[2957.02,1299.12,2126.97],[0.365050148,0.478002035,0.608135257,0.51809647],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg
 Paint_1510:=[[2849.68,1426.6,2178.43],[0.365050148,0.478002035,0.608135257,0.51809647],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg
 Paint_1520:=[[2765.55,1543.19,2200.63],[0.365050148,0.478002035,0.608135257,0.51809647],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg
 Paint_1530:=[[2678.52,1673.19,2214.09],[0.365050148,0.478002035,0.608135257,0.51809647],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg
 Paint_1540:=[[2614.94,1789.27,2217.83],[0.365050148,0.478002035,0.608135257,0.51809647],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg Paint_1550:=[[2684.85,1792.19,2111.69],[0.632754631,-0.025937671,0.717872832,0.289149462],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg Paint_1560:=[[2843.224551317,1805.820051646,1770.373560375],[0.585122193,-0.028421261,0.767434682,0.260515373],[0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

CONST robtarg
 Pos_Int2:=[[1390.775857389,1815.07664145,1392.253348344],[0.141544911,0.157713135,0.976686147,0.034289587],[1,-1,0,-1],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg Paint_1570:=[[1619.842639859,755.539719945,1482.736824235],[0.153522091,0.213265029,-0.906771817,0.32971786],[0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg Paint_1580:=[[1467.53,740.97,1746.37],[0.237108005,0.236712924,-0.871535672,0.358011672],[0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg Paint_1590:=[[1391.02,731.54,1833.22],[0.353380695,0.26646993,-0.78791197,0.428147857],[0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg Paint_1600:=[[1334.77,708.3,1893.12],[0.312706039,0.277578082,-0.79579159,0.438042335],[0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg Paint_1610:=[[1312.05,681.42,1948.58],[0.365850854,0.266629806,-0.766022398,0.456367599],[0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg Paint_1620:=[[1369.37,638.53,2035.07],[0.438629496,0.155740044,-0.652702074,0.597770196],[0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1630:=[[1437.71,612.85,2082.92],[0.499311366,0.219997819,-0.546283767,0.635502294],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1640:=[[1547.3,587.39,2133.11],[0.465987207,0.258368317,-0.468761174,0.704531545],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1650:=[[1661.5,570.98,2161.72],[0.459154169,0.282960903,-0.445032549,0.714882232],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1660:=[[1935.15,560.76,2200.3],[0.006723192,-0.131791865,0.504700472,-0.853148953],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1670:=[[2086.9,560.22,2201.72],[0.133696082,0.358118123,-0.320754631,0.866598658],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1680:=[[2110.79,564.17,2195.56],[0.648755363,0.079261,0.584035751,-0.481390085],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1690:=[[2137.87,577.01,2176.45],[0.803818591,0.252327548,0.511569825,-0.168827712],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1700:=[[2148.07,591.01,2165.6],[0.769196984,0.262151373,0.558149969,-0.16757467],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1710:=[[2155.09,605.5,2139.13],[0.783183623,0.230227337,0.53845328,-0.209013995],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1720:=[[2167.93,641.46,2033.23],[0.765892645,0.223643677,0.577680688,-0.17227009],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1730:=[[2185.66,686.26,1912.42],[0.726837273,0.192841925,0.642485594,-0.147417206],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1740:=[[2203.05,712.38,1800.75],[0.701269947,0.211276851,0.664303285,-0.149277257],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1750:=[[2238.17,728.4,1593.87],[0.729816216,0.183610873,0.641572716,-0.148458038],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1760:=[[2238.453765957,743.265948705,1435.648237738],[0.727752991,0.170447002,0.651580073,-0.129486725],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1770:=[[2263.41,741.7,1367.66],[0.598895449,0.186705468,0.772267652,-0.100339337],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1780:=[[2236.73,740.53,1304.07],[0.492593182,0.185415336,0.841245807,-0.123606644],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1790:=[[2174.06,742.15,1261.31],[0.477855373,0.15094105,0.829371162,-0.247011167],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1800:=[[2109.81,748.37,1249.15],[0.309158917,0.146373712,0.907325264,-0.244451151],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1810:=[[2063.6,754.04,1245.73],[0.293737544,0.107897225,0.900838055,-0.30094392],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1820:=[[1946.86,755.18,1247.07],[0.201744651,0.140036322,0.929051003,-0.276682414],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1830:=[[1869.98,748.43,1276],[0.157996106,0.111845642,0.914783948,-0.354539294],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1840:=[[1795.69,749.47,1297.77],[0.029255099,0.09888371,0.895140628,-0.433692757],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1850:=[[1743.74,748.91,1345.09],[0.001082167,0.074845784,0.940318123,-0.331961993],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1860:=[[1685.03,754.17,1402.26],[0.067202984,-0.126876088,-0.956234196,0.254955643],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1870:=[[1620.52,760.21,1484.22],[0.120504776,-0.049482158,-0.928979691,0.346448913],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1880:=[[1471.919173256,735.158285713,1737.728696076],[0.120504776,-0.049482158,-0.928979691,0.346448913],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1890:=[[1239.900079169,623.409553496,2070.421470566],[0.261770844,0.231225282,-0.79272959,0.499590524],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1900:=[[1266.88,776.43,1862.47],[0.190633218,0.083121418,-0.895000014,0.394619793],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1910:=[[1311.37,789.82,1776.57],[0.147681825,0.201022586,-0.874122379,0.416761401],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1920:=[[1347.58,801.53,1721.92],[0.147681825,0.201022586,-0.874122379,0.416761401],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1930:=[[1392.44,806.77,1671.84],[0.147681825,0.201022586,-0.874122379,0.416761401],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1940:=[[1489.24,809.37,1567.2],[0.147681825,0.201022586,-0.874122379,0.416761401],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1950:=[[1549.39,807.88,1497.65],[0.147681825,0.201022586,-0.874122379,0.416761401],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1960:=[[1613.66,806.77,1398.67],[0.147681825,0.201022586,-0.874122379,0.416761401],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1970:=[[1640.76,808.81,1332.13],[0.147681825,0.201022586,-0.874122379,0.416761401],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1980:=[[1677.58,809.97,1238.11],[0.050229921,-0.18769274,0.662563316,-0.723365912],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1990:=[[1681.96,809.95,1209.37],[0.050229921,-0.18769274,0.662563316,-0.723365912],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_2000:=[[1716.26,809.95,1246.73],[0.050229921,-0.18769274,0.662563316,-0.723365912],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_2010:=[[1765.43,809.73,1218.47],[0.050229921,-0.18769274,0.662563316,-0.723365912],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_2020:=[[1857.25,809.64,1205.24],[0.050229921,-0.18769274,0.662563316,-0.723365912],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_2030:=[[1991.33,811.44,1196.45],[0.050229921,-0.18769274,0.662563316,-0.723365912],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_2040:=[[2125.952707943,827.917998067,1208.914386834],[0.496905618,0.12056434,0.80073215,-0.312052992],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_2060:=[[2218.059050944,991.439924277,1247.161699943],[0.518907904,0.525951332,0.507199004,0.4436879],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_2070:=[[2184.15,1088.1,1221.64],[0.539381073,0.411645744,0.460743005,0.572129114],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_2080:=[[2111.47,1217.61,1221.41],[0.556823437,0.387726476,0.435348558,0.591681901],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_2090:=[[1970.58,1473.37,1227.49],[0.556823437,0.387726476,0.435348558,0.591681901],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_2100:=[[1978.89,1491.96,1355.34],[0.437499143,0.518631527,0.57365296,0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_2110:=[[1996.67,1443.1,1359.97],[0.437499143,0.518631527,0.57365296,0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_2120:=[[1934.84,1353.23,1382.99],[0.437499143,0.518631527,0.57365296,0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_2130:=[[1982.88,1272.87,1367.78],[0.437499143,0.518631527,0.57365296,0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_2140:=[[2025.19,1194.73,1369.63],[0.437499143,0.518631527,0.57365296,0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_2150:=[[2083.08,1090.1,1371.55],[0.437499143,0.518631527,0.57365296,0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_2160:=[[2143.37,993.81,1381.23],[0.508633013,0.574085766,0.511640183,0.387223857],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_2170:=[[2182.6,934.79,1380.32],[0.530466338,0.590429831,0.488966954,0.361814036],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_2180:=[[2254.67,900.47,1368.6],[0.499626193,0.567240342,0.520439128,0.397184059],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_2190:=[[2316.5,896.55,1356.21],[0.457252829,0.534329027,0.558035007,0.440464837],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_2200:=[[2305.121123706,892.893668788,1514.901127581],[0.437499143,0.518631527,0.57365296,0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_2210:=[[2237.04,917.93,1519.23],[0.437499143,0.518631527,0.57365296,0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_2220:=[[2179,944.42,1523.83],[0.437499143,0.518631527,0.57365296,0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_2230:=[[2123.17,1031.35,1515.91],[0.437499143,0.518631527,0.57365296,0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_2240:=[[2030.11,1181.14,1515.45],[0.437499143,0.518631527,0.57365296,0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
Paint_2250:=[[1928.82,1362.77,1518.7],[0.437499143,0.518631527,0.57365296,0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
Paint_2260:=[[1898.62,1405.21,1518.97],[0.437499143,0.518631527,0.57365296,0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
Paint_2270:=[[1961.23,1509.04,1507.13],[0.437499143,0.518631527,0.57365296,0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
Paint_2280:=[[1929.9,1586.78,1505.33],[0.437499143,0.518631527,0.57365296,0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
Paint_2290:=[[1857.79,1693.74,1634.09],[0.437499143,0.518631527,0.57365296,0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
Paint_2300:=[[1902.81,1619.81,1632.43],[0.437499143,0.518631527,0.57365296,0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
Paint_2310:=[[1842.56,1482.22,1643.48],[0.437499143,0.518631527,0.57365296,0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
Paint_2320:=[[1920.98,1358.45,1644.91],[0.437499143,0.518631527,0.57365296,0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
Paint_2330:=[[2050.08,1142.08,1645.8],[0.437499143,0.518631527,0.57365296,0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
Paint_2340:=[[2169.48,963.7,1660.12],[0.437499143,0.518631527,0.57365296,0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
Paint_2350:=[[2235.81,928.37,1654.58],[0.437499143,0.518631527,0.57365296,0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
Paint_2360:=[[2282.320534461,916.570757353,1653.746223018],[0.41721243,0.502302154,0.588572006,0.476664796],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
Paint_2370:=[[2225.762809264,928.034171969,1816.262818952],[0.437499143,0.518631527,0.57365296,0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
Paint_2380:=[[2190.44,960.59,1813.31],[0.437499143,0.518631527,0.57365296,0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
Paint_2390:=[[2088,1053.02,1803.46],[0.437499143,0.518631527,0.57365296,0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
Paint_2400:=[[2014.17,1165.19,1792.16],[0.437499143,0.518631527,0.57365296,0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
Paint_2410:=[[1906.79,1334.52,1789.93],[0.437499143,0.518631527,0.57365296,0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
Paint_2420:=[[1819.99,1485.37,1790.23],[0.437499143,0.518631527,0.57365296,0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
Paint_2430:=[[1782.67,1536.8,1789.24],[0.437499143,0.518631527,0.57365296,0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
Paint_2440:=[[1796.19,1631.91,1799.7],[0.437499143,0.518631527,0.57365296,0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
Paint_2450:=[[1774.37,1766.64,1783.4],[0.437499143,0.518631527,0.57365296,0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
Paint_2460:=[[1671.77,1763.05,1910.47],[0.437499143,0.518631527,0.57365296,0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
Paint_2470:=[[1719.32,1587.39,1893.81],[0.437499143,0.518631527,0.57365296,0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
Paint_2480:=[[1792.92,1489.7,1888.6],[0.437499143,0.518631527,0.57365296,0.458844332],[0,0,0],[[-425,9E+09,9E+09,9E+09,9E+09,0]]];
LOCAL CONST robtarget
Paint_2490:=[[1873.27,1353.14,1881.78],[0.437499143,0.518631527,0.57365296,0.458844332],[0,0,0],[[-425,9E+09,9E+09,9E+09,9E+09,0]]];
LOCAL CONST robtarget
Paint_2500:=[[1937.05,1254.22,1880.69],[0.437499143,0.518631527,0.57365296,0.458844332],[0,0,0],[[-425,9E+09,9E+09,9E+09,9E+09,0]]];
LOCAL CONST robtarget
Paint_2510:=[[2022.03,1124.66,1877.86],[0.437499143,0.518631527,0.57365296,0.458844332],[0,0,0],[[-425,9E+09,9E+09,9E+09,9E+09,0]]];
LOCAL CONST robtarget
Paint_2520:=[[2091.34,1008.73,1874.51],[0.437499143,0.518631527,0.57365296,0.458844332],[0,0,0],[[-425,9E+09,9E+09,9E+09,9E+09,0]]];
LOCAL CONST robtarget
Paint_2530:=[[2192.24,915.87,1881.82],[0.22860438,0.698680537,0.331514535,0.59134056],[0,0,0],[[-425,9E+09,9E+09,9E+09,9E+09,0]]];
LOCAL CONST robtarget
Paint_2540:=[[2166.12,921.23,1919.44],[0.22860438,0.698680537,0.331514535,0.59134056],[0,0,0],[[-425,9E+09,9E+09,9E+09,9E+09,0]]];
LOCAL CONST robtarget
Paint_2550:=[[2118.88,917.65,2028.53],[0.22860438,0.698680537,0.331514535,0.59134056],[0,0,0],[[-425,9E+09,9E+09,9E+09,9E+09,0]]];
LOCAL CONST robtarget
Paint_2560:=[[2018.24,889.98,2216.85],[0.22860438,0.698680537,0.331514535,0.59134056],[0,0,0],[[-425,9E+09,9E+09,9E+09,9E+09,0]]];
LOCAL CONST robtarget
Paint_2570:=[[2002.46,892.33,2234.49],[0.437499143,0.518631527,0.57365296,0.458844332],[0,0,0],[[-425,9E+09,9E+09,9E+09,9E+09,0]]];
LOCAL CONST robtarget
Paint_2580:=[[1898.58,1069.18,2227.5],[0.437499143,0.518631527,0.57365296,0.458844332],[0,0,0],[[-425,9E+09,9E+09,9E+09,9E+09,0]]];
LOCAL CONST robtarget
Paint_2590:=[[1793.64,1264.98,2202.96],[0.437499143,0.518631527,0.57365296,0.458844332],[0,0,0],[[-425,9E+09,9E+09,9E+09,9E+09,0]]];
LOCAL CONST robtarget
Paint_2600:=[[1730.49,1401.28,2169.75],[0.437499143,0.518631527,0.57365296,0.458844332],[0,0,0],[[-425,9E+09,9E+09,9E+09,9E+09,0]]];
LOCAL CONST robtarget
Paint_2610:=[[1680.07,1553,2099.27],[0.437499143,0.518631527,0.57365296,0.458844332],[0,0,0],[[-425,9E+09,9E+09,9E+09,9E+09,0]]];
LOCAL CONST robtarget
Paint_2620:=[[1654.59,1672.99,2012.77],[0.437499143,0.518631527,0.57365296,0.458844332],[0,0,0],[[-425,9E+09,9E+09,9E+09,9E+09,0]]];
LOCAL CONST robtarget
Paint_2630:=[[1657.85,1741.53,1937.3],[0.437499143,0.518631527,0.57365296,0.458844332],[0,0,0],[[-425,9E+09,9E+09,9E+09,9E+09,0]]];
LOCAL CONST robtarget
Paint_2640:=[[1666.86,1751.23,1923.46],[0.437499143,0.518631527,0.57365296,0.458844332],[0,0,0],[[-425,9E+09,9E+09,9E+09,9E+09,0]]];
LOCAL CONST robtarget
Paint_2650:=[[1843.15,1470.73,1925.72],[0.437499143,0.518631527,0.57365296,0.458844332],[0,0,0],[[-425,9E+09,9E+09,9E+09,9E+09,0]]];
LOCAL CONST robtarget
Paint_2660:=[[1811.14,1451.1,1937.94],[0.437499143,0.518631527,0.57365296,0.458844332],[0,0,0],[[-425,9E+09,9E+09,9E+09,9E+09,0]]];
LOCAL CONST robtarget
Paint_2670:=[[1656.56470442,1400.452204417,2284.216955119],[0.437499143,0.518631527,0.57365296,0.458844332],[0,0,0],[[-425,9E+09,9E+09,9E+09,9E+09,0]]];
CONST jointtarget Pos_Int3:=[[180,-20,20,0,45,0],[4060.94488189,9E+09,9E+09,9E+09,9E+09,0]]];
CONST robtarget Pos_Int4:=[[1656.565,1400.452,2600],[0,0.499999999,0.866025404,0],[0,0,0],[[-425,9E+09,9E+09,9E+09,9E+09,0]]];
CONST jointtarget Pos_Inicial2:=[[0,20,-20,0,-45,0],[6750,9E+09,9E+09,9E+09,9E+09,0]]];
LOCAL CONST robtarget
Paint_2680:=[[4231.829478651,900.010594169,1404.051579205],[0.41663936,0.574681469,0.588034335,-0.38777374],[0,0,0],[[1200,9E+09,9E+09,9E+09,9E+09,0]]];
LOCAL CONST robtarget Paint_2690:=[[4231.75,820.32,1631.44],[0.442673914,0.540103586,0.579783134,-0.419737346],[0,0,0],[[1200,9E+09,9E+09,9E+09,9E+09,0]]];
LOCAL CONST robtarget Paint_2700:=[[4237.58,797.81,1674.49],[0.226214202,0.601270087,0.708036941,-0.293232173],[0,0,0],[[1200,9E+09,9E+09,9E+09,9E+09,0]]];

LOCAL CONST robtarget Paint_2710:=[[4240.57,773.69,1698.23],[0.224156636,0.598226059,0.703045466,-0.312420322],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_2720:=[[4083.12,788.98,1727.28],[0.300730427,0.589935623,0.675028652,-0.325382068],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_2730:=[[4083.25,828.07,1683.34],[0.173492132,0.57846167,0.721713526,-0.33824867],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_2740:=[[4081.23,846.57,1642.37],[0.296666499,0.501477083,0.696171768,-0.419350203],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_2750:=[[4079.71,883.86,1574.96],[0.340884011,0.551047191,0.629917234,-0.428193137],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_2760:=[[4130.4,884.01,1507.41],[0.516173759,0.440884236,0.552490702,-0.4836732],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_2770:=[[4038.15,886.36,1554.08],[0.500418087,0.458762321,0.537398957,-0.500321129],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_2780:=[[3976.65,887.18,1572.33],[0.505095755,0.453535679,0.533850017,-0.504170433],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_2790:=[[3931.17,887.67,1584],[0.50590259,0.452626293,0.532997293,-0.505080087],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_2800:=[[3931.47,852.69,1673.14],[0.361692007,0.580739621,0.646522712,-0.337533356],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_2810:=[[3933.95,835.46,1702.68],[0.258774284,0.573505365,0.727617475,-0.273313513],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_2820:=[[3932.76,800.75,1744.61],[0.300702798,0.543420646,0.70660302,-0.339092909],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_2830:=[[3784.69,804.36,1765.27],[0.359043802,0.583408839,0.652982916,-0.323009268],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_2840:=[[3783.68,845.95,1712.7],[0.290964983,0.591596615,0.702368565,-0.268386331],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_2850:=[[3782.45,862.91,1672.27],[0.383125061,0.523098857,0.640199886,-0.411979223],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_2860:=[[3779.84,888.46,1596.44],[0.478488869,0.48242459,0.584574717,-0.443381683],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_2870:=[[3777.12,887.19,1558.29],[0.514038214,0.449826172,0.525846815,-0.506859209],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_2880:=[[3697.41,885.96,1522.62],[0.468843147,0.491755523,0.555758769,-0.479056156],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_2890:=[[3649.43,884.41,1485.86],[0.469237111,0.492680285,0.547247175,-0.487445586],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_2900:=[[3630.66,883.17,1464.69],[0.469673488,0.492887747,0.542237672,-0.492388861],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_2910:=[[3632.34,887.47,1528.62],[0.437624609,0.524145109,0.579386799,-0.4450478],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_2920:=[[3631.42,866.74,1664.01],[0.443026369,0.520787159,0.555972954,-0.47265468],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_2930:=[[3634.28,855.75,1718.35],[0.257618534,0.637410251,0.654542862,-0.314506764],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_2940:=[[3632.49,803.16,1788.3],[0.334755817,0.594646196,0.616245384,-0.393161635],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_2950:=[[3636.9,790.36,1813.81],[0.342200514,0.589729765,0.610329829,-0.403255641],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_2960:=[[3560.42,760.91,1871.02],[0.183152331,0.598376408,0.708335845,-0.326590307],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_2970:=[[3514.06,702.43,1903.51],[0.213429312,0.593239508,0.716821918,-0.29779381],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_2980:=[[3484.88,703.94,1908.91],[0.133433026,0.591783032,0.726031662,-0.323830969],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_2990:=[[3486.88,753.81,1891.74],[0.206742531,0.632429385,0.672409564,-0.324277623],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3000:=[[3484.68,804.13,1800.6],[0.329303473,0.562175119,0.641023452,-0.405718242],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3010:=[[3483.72,861.01,1726.07],[0.255318205,0.604533603,0.678316128,-0.330513187],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3020:=[[3482.49,874,1640.43],[0.418116203,0.502265717,0.567045298,-0.501365756],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3030:=[[3477.36,881.98,1488.48],[0.421499369,0.497747227,0.573299962,-0.495896293],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3040:=[[3479.95,876.85,1321.92],[0.382776552,0.539286849,0.571984408,-0.485268629],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3050:=[[3477.38,839.73,1159.72],[0.339941184,0.575590136,0.595012883,-0.446201363],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3060:=[[3474.73,841.93,1100.94],[0.339941184,0.575590136,0.595012883,-0.446201363],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3070:=[[3470.61,823.61,1057.83],[0.339941184,0.575590136,0.595012883,-0.446201363],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3080:=[[3324.78,827.99,1066.27],[0.339941184,0.575590136,0.595012883,-0.446201363],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3090:=[[3325.31,835.88,1136.25],[0.375126113,0.581133999,0.595504097,-0.408581137],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3100:=[[3332.5,880.01,1310.44],[0.402892327,0.565600882,0.566459089,-0.443731356],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3110:=[[3333.8,880.43,1451.42],[0.453497051,0.520046308,0.541563496,-0.480209582],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3120:=[[3337.95,876.39,1658.49],[0.443469232,0.52869091,0.540226655,-0.481638997],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3130:=[[3333.11,869.65,1727.5],[0.475013431,0.500848657,0.515196274,-0.508021321],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3140:=[[3327.4,810.66,1797.22],[0.456677227,0.519762811,0.533635508,-0.486339055],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3150:=[[3333.8,775.43,1882.55],[0.37939577,0.581438193,0.592350514,-0.4087901],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3160:=[[3333.93,751.11,1931.37],[0.314312498,0.621520407,0.627804583,-0.347536247],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3170:=[[3336.62,721.57,1954.99],[0.234249026,0.634959604,0.667849302,-0.309727308],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3180:=[[3335.64,689.77,1961.81],[0.346736107,0.618242904,0.630561621,-0.316135772],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3190:=[[3180.12,659.83,2042.4],[0.393116595,0.583100667,0.556453997,-0.442506388],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3200:=[[3172.89,698.16,2034.59],[0.346498706,0.559460933,0.598970186,-0.456263988],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3210:=[[3171.42,722.05,2011.32],[0.459364446,0.515351455,0.525921715,-0.496793249],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3220:=[[3177.57,761.84,1882.67],[0.459364446,0.515351455,0.525921715,-0.496793249],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3230:=[[3211.65,843.27,1768.17],[0.410451493,0.478258629,0.620126584,-0.467163008],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3240:=[[3221.89,874.63,1728.58],[0.414664221,0.473776814,0.615003985,-0.474720142],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3250:=[[3224.58,878.43,1667.91],[0.405639653,0.483045751,0.581381705,-0.513924691],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3260:=[[3230.33,883.92,1623.47],[0.418739899,0.471202713,0.568964949,-0.528113423],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3270:=[[3223.7,877.98,1391.57],[0.384854067,0.500537148,0.596394938,-0.495644013],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3280:=[[3220.91,883.65,1324.92],[0.366943937,0.515109172,0.607264563,-0.480878818],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3290:=[[3220.32,878.56,1288],[0.394822383,0.502380019,0.562610892,-0.524593734],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3300:=[[3212.77,846.23,1174.1],[0.332886334,0.559282293,0.595436049,-0.47100522],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3310:=[[3178.96,837.53,1141.15],[0.371698183,0.523828419,0.603150353,-0.472920607],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3320:=[[3179.65,839.7,1121.18],[0.354562996,0.536316576,0.615879031,-0.455348912],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3330:=[[3178.86,828.2,1070.73],[0.361060321,0.593789351,0.597699582,-0.399756002],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3340:=[[3027.03,828.86,1075.21],[0.361060321,0.593789351,0.597699582,-0.399756002],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3350:=[[3027.62,839.11,1144.94],[0.367671916,0.57132544,0.600299296,-0.421954215],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3360:=[[3036.8,858.3,1207.34],[0.386787961,0.575787083,0.581381474,-0.425276251],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3370:=[[3037.65,885.35,1307.7],[0.406327863,0.547973985,0.565203847,-0.463860744],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3380:=[[3044.04,882.2,1356.41],[0.396795238,0.553845319,0.585284343,-0.439603389],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3390:=[[3043.53,879.59,1438.99],[0.409927116,0.544974372,0.572696526,-0.454952066],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3400:=[[3055.35,880.48,1687.07],[0.449076236,0.511667507,0.538575019,-0.496451252],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3410:=[[3050.16,877.92,1733.68],[0.478986123,0.488536851,0.521316701,-0.51003229],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3420:=[[3051.52,805.16,1822.67],[0.458984054,0.505805699,0.536988504,-0.495113704],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3430:=[[3036.5,709.73,2044.67],[0.495933191,0.475874779,0.565684911,-0.455624897],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3440:=[[3038.84,678.83,2095.21],[0.307682905,0.539227899,0.620845397,-0.478660105],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3450:=[[3037.91,626.31,2115.44],[0.309945134,0.588396397,0.568959561,-0.483744469],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3460:=[[2888.29,590.88,2183.94],[0.38082821,0.559794209,0.546359767,-0.493042921],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3470:=[[2887.55,634.24,2172.11],[0.433280565,0.555213453,0.534760812,-0.466944158],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3480:=[[2885.56,674.13,2109.19],[0.388495042,0.592250014,0.573847708,-0.411108662],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3490:=[[2906.2,806.13,1831.28],[0.364237975,0.602560311,0.591448527,-0.39298907],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3500:=[[2915.97,874.53,1746.97],[0.370636553,0.601227589,0.600154964,-0.375456989],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3510:=[[2912.49,882.39,1699.28],[0.438241278,0.47771232,0.551169258,-0.525307502],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3520:=[[2904.89,880.27,1444.72],[0.40725062,0.462270079,0.567359175,-0.546403581],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3530:=[[2901.5,886.42,1339.05],[0.341196249,0.517838811,0.631965888,-0.46480878],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3540:=[[2899.01,867.6,1235.82],[0.341717252,0.543479008,0.592265588,-0.486910013],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3550:=[[2880.53,840.5,1143.3],[0.344721771,0.53435313,0.613237621,-0.468586441],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3560:=[[2878.42,823.49,1074.33],[0.334645382,0.610661307,0.612879934,-0.373474795],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3570:=[[2720.17,815.11,1073.34],[0.361060321,0.593789351,0.597699582,-0.399756002],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3580:=[[2727.37,841.74,1153.9],[0.391479056,0.55529263,0.585312041,-0.442497523],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3590:=[[2729.43,879.38,1279.65],[0.355946741,0.582171191,0.605332857,-0.409817952],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3600:=[[2732.56,885.2,1354.29],[0.380051195,0.563999478,0.598419222,-0.423509283],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3610:=[[2744.54,883.49,1482.59],[0.414490992,0.539289284,0.5720226,-0.458426037],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3620:=[[2757.23,884.79,1692.92],[0.455331522,0.504215514,0.540488129,-0.496298804],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3630:=[[2756.88,880.01,1742.74],[0.467154322,0.497054801,0.539795227,-0.493279309],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3640:=[[2759.29,800.49,1847.11],[0.483186975,0.478613526,0.521653843,-0.515108444],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3650:=[[2731.18,645.85,2160.72],[0.435114073,0.527154498,0.585045934,-0.436468938],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3660:=[[2739.79,619.27,2212.41],[0.404254263,0.532139632,0.582122169,-0.463184287],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3670:=[[2740.55,596.52,2229.46],[0.427620081,0.473716274,0.547042398,-0.541736626],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3680:=[[2739,563.78,2236.53],[0.334497005,0.520997222,0.607126217,-0.498067671],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3690:=[[2589.55,541.34,2270.97],[0.317074496,0.581023373,0.61866912,-0.42323058],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3700:=[[2592.38,579.62,2261.23],[0.396032677,0.545720353,0.589167415,-0.445229347],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3710:=[[2589.15,624.86,2195.78],[0.412095699,0.538191412,0.598996166,-0.426298877],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3720:=[[2593.08,795.52,1862.15],[0.408195998,0.534741469,0.590509625,-0.445786912],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3730:=[[2599.26,883.13,1738.65],[0.380158545,0.554993207,0.610523479,-0.417998926],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3740:=[[2600.06,885.74,1690.46],[0.457039979,0.495607485,0.54322602,-0.500393014],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3750:=[[2594.35,878.03,1434.11],[0.418992699,0.52967662,0.568829391,-0.469383554],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3760:=[[2593.19,886.85,1357.25],[0.398767944,0.545907719,0.582009624,-0.451922214],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3770:=[[2590.55,884.07,1308.17],[0.370776724,0.56509526,0.602836649,-0.42400465],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3780:=[[2583,850.82,1189.43],[0.341812917,0.59056842,0.611869492,-0.400010745],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3790:=[[2577.9,842.31,1152.18],[0.330918489,0.59526209,0.622552435,-0.385466554],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3800:=[[2575.51,832.85,1087.85],[0.386787961,0.575787083,0.581381474,-0.425276251],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3810:=[[2429.05,833.79,1091.45],[0.361060321,0.593789351,0.597699582,-0.399756002],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3820:=[[2426.68,843.39,1161.45],[0.405014253,0.561074428,0.576324246,-0.434752004],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3830:=[[2434.82,859.64,1226.06],[0.395437512,0.564197123,0.589502139,-0.421661012],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3840:=[[2440.92,887.51,1348.67],[0.448257463,0.518900208,0.560905611,-0.46388869],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3850:=[[2443.54,881.65,1414.34],[0.410099911,0.551380281,0.562219558,-0.460116308],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3860:=[[2445.75,881.51,1491.92],[0.440836877,0.520490227,0.55648112,-0.474427586],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3870:=[[2447.42,885.49,1689.9],[0.465737639,0.498328462,0.536300129,-0.497131137],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3880:=[[2451.72,881.49,1742.58],[0.3998649,0.553113667,0.594273251,-0.425455798],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3890:=[[2436.46,794.63,1870.38],[0.454727902,0.507343853,0.550092906,-0.482931201],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3900:=[[2439.32,778.23,1893.77],[0.425555004,0.53239448,0.574115809,-0.453707059],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3910:=[[2441.23,614.29,2209.81],[0.402362018,0.552518618,0.602048148,-0.412754178],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3920:=[[2440.5,581.94,2265.89],[0.359992003,0.579065412,0.62675103,-0.377189812],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3930:=[[2441.7,565.8,2280.14],[0.417615983,0.537027788,0.586667767,-0.43933925],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3940:=[[2443.41,532.2,2289.67],[0.388717026,0.548370785,0.604351002,-0.427724704],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3950:=[[2288.69,524.6,2300.05],[0.37748138,0.572679521,0.614889146,-0.38917517],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3960:=[[2290.07,592.34,2249.71],[0.429162842,0.538423587,0.583624753,-0.430466543],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3970:=[[2280.95,647.57,2162.89],[0.449536676,0.521463002,0.564303463,-0.455581734],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3980:=[[2294.7,756.53,1922.21],[0.47306465,0.500025154,0.538116664,-0.486431021],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_3990:=[[2299.02,821.33,1847.49],[0.440232061,0.524995377,0.561650672,-0.463814735],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_4000:=[[2317.2,870.78,1772.99],[0.44989484,0.517434114,0.556916005,-0.468722876],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_4010:=[[2318.46,883.77,1699.2],[0.44989484,0.517434114,0.556916005,-0.468722876],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_4020:=[[2319.98,888.26,1621.37],[0.44989484,0.517434114,0.556916005,-0.468722876],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_4030:=[[2319.71,885.09,1555.7],[0.399644426,0.556317011,0.59400769,-0.42184189],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_4040:=[[2304.58,881.6,1450.46],[0.399644426,0.556317011,0.59400769,-0.42184189],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_4050:=[[2301,886.44,1376.59],[0.399644426,0.556317011,0.59400769,-0.42184189],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_4060:=[[2296.55,870.66,1288.39],[0.399644426,0.556317011,0.59400769,-0.42184189],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_4070:=[[2282.6,848.07,1189.52],[0.373353802,0.574188005,0.610874571,-0.397174183],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_4080:=[[2277.46,844.32,1162.84],[0.37187833,0.577409078,0.59667825,-0.415066658],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_4090:=[[2280.654595682,870.918228525,959.34171887],[0.361059756,0.593789617,0.59769993,-0.399755597],[1,-2,1,1],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_4100:=[[2127.536,870.918,945.701],[0.346045007,0.622619301,0.611395681,-0.344664156],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_4110:=[[2127.12,846.16,1164.35],[0.417709184,0.576700821,0.563071515,-0.419387255],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_4120:=[[2124.75,858.6,1241.5],[0.419393183,0.573658703,0.565833804,-0.418159248],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_4130:=[[2132.89,882.51,1393.45],[0.441239313,0.548451606,0.570128237,-0.423630143],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_4140:=[[2144.25,885.17,1471.12],[0.441259481,0.549868786,0.532495588,-0.468383216],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_4150:=[[2163.27,881.62,1704.54],[0.488964454,0.505149205,0.518420778,-0.4868038],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_4160:=[[2157.71,859.85,1800.07],[0.46263283,0.527476893,0.537546816,-0.467741823],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_4170:=[[2147.99,789.3,1885.4],[0.417623137,0.562598831,0.571479767,-0.427181866],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_4180:=[[2143.03,636.48,2176.41],[0.434296563,0.554123571,0.572907509,-0.419655274],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_4190:=[[2139.64,580.72,2261.12],[0.451407492,0.539925471,0.556957093,-0.441033512],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_4200:=[[2140.48,540.48,2300.32],[0.324573577,0.61886836,0.637388611,-0.324637805],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_4210:=[[2140.64,514.67,2305.85],[0.211138826,0.663167889,0.682533298,-0.223107695],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_4220:=[[1990.53,504.47,2307.17],[0.181228773,0.677746407,0.691902513,-0.17054868],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_4230:=[[1991.07,547.35,2292.4],[0.361067965,0.607787159,0.619005348,-0.342136045],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_4240:=[[1988.71,584.62,2248.6],[0.466160889,0.536139622,0.543243807,-0.447363943],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_4250:=[[1986.51,778.23,1899.34],[0.395859754,0.58354543,0.588071167,-0.396159172],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_4260:=[[1988.26,831.41,1847.23],[0.404741462,0.575444355,0.578567034,-0.412684298],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_4270:=[[1991.51,867.54,1769.55],[0.431881381,0.555174027,0.570512033,-0.424000345],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_4280:=[[1995.1,878.93,1701.47],[0.487490751,0.509390572,0.521324271,-0.480723432],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_4290:=[[1991.04,881.33,1543.74],[0.458848611,0.535770293,0.533811268,-0.466319284],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_4300:=[[1991.89,887.15,1501.96],[0.435309316,0.555274709,0.535337068,-0.464316724],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_4310:=[[1989.39,886.64,1472.77],[0.461846562,0.522058726,0.55260486,-0.456924839],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_4320:=[[1991.32,875.19,1377.34],[0.447209983,0.536433809,0.557095847,-0.449317501],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_4330:=[[1990.97,860.16,1243.83],[0.421041138,0.563378399,0.561826726,-0.435522524],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_4340:=[[1976.91,848.88,1170.71],[0.399099467,0.580342937,0.575739849,-0.415265357],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_4350:=[[1976.89,833.84,1090.63],[0.372959253,0.611571453,0.583938115,-0.381966009],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_4360:=[[1824.46,833.95,1084.62],[0.395333186,0.599911703,0.566730776,-0.40327887],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_4370:=[[1825.25,851.99,1168.58],[0.41002136,0.5872422,0.564995058,-0.409645783],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_4380:=[[1832.61,869.41,1292.85],[0.416617501,0.581012469,0.563615666,-0.413753249],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_4390:=[[1837.1,878.12,1446.19],[0.467775051,0.534378137,0.542824986,-0.448294037],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_4400:=[[1844.13,885.37,1516.66],[0.469434297,0.531017807,0.54828497,-0.443886382],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_4410:=[[1843.96,884.14,1554.54],[0.421615236,0.571070335,0.550207485,-0.439762423],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_4420:=[[1846.41,879.02,1627.6],[0.440406296,0.548091544,0.567764218,-0.428114175],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_4430:=[[1848.59,873.1,1727.65],[0.432819887,0.553689395,0.573939295,-0.420343532],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_4440:=[[1832.18,776.99,1900.2],[0.461092685,0.528716296,0.547358823,-0.456345191],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_4450:=[[1838.45,597.53,2220.03],[0.480787213,0.536397009,0.550150296,-0.422441186],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_4460:=[[1838.52,559.34,2273.56],[0.460430419,0.543378381,0.55281186,-0.43260006],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_4470:=[[1838.12,544.55,2287.61],[0.394812306,0.590784085,0.603743316,-0.361374344],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_4480:=[[1841.22,493.69,2304.37],[0.35207856,0.613796905,0.629598417,-0.320780111],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_4490:=[[1690.02,492.7,2297.94],[0.363494547,0.617415978,0.623535066,-0.312847002],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_4500:=[[1689.77,524.89,2291],[0.38046223,0.607898188,0.612495459,-0.332502027],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_4510:=[[1688.49,596.28,2212.33],[0.508520334,0.511408045,0.499225243,-0.480253097],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_4520:=[[1688.55,771.45,1903.8],[0.416113156,0.581911852,0.571409277,-0.402144099],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_4530:=[[1692.31,830.26,1846.06],[0.412143905,0.585420669,0.57671998,-0.393464239],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_4540:=[[1695.24,851.45,1796.32],[0.432808064,0.570663017,0.56782016,-0.405710694],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_4550:=[[1696.34,874.97,1650.5],[0.462803015,0.547760111,0.544632881,-0.434910629],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_4560:=[[1696.63,882.64,1593.8],[0.440250133,0.565920774,0.531318832,-0.451235854],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_4570:=[[1698.8,883.26,1557.22],[0.481690252,0.522431738,0.534195701,-0.457902318],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_4580:=[[1697.83,880.27,1505.47],[0.481190702,0.524708931,0.525669304,-0.465626276],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_4590:=[[1699.72,880.07,1388.58],[0.461972436,0.545154565,0.520788639,-0.467083677],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_4600:=[[1697.81,858.92,1212.31],[0.413027436,0.582320477,0.566251828,-0.411910266],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_4610:=[[1683.86,856.99,1156.4],[0.386805473,0.6018984,0.578676769,-0.391449919],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_4620:=[[1680.95,841.92,1083.06],[0.372464622,0.616419249,0.585262424,-0.372512162],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_4630:=[[1595.95,859.1,1093.29],[0.408989328,0.594733908,0.557676223,-0.409898205],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_4640:=[[1588.02,868.65,1264.9],[0.415174213,0.56667436,0.601262969,-0.38077997],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_4650:=[[1555.71,878.97,1363.96],[0.455931779,0.537706979,0.573766849,-0.416880105],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_4660:=[[1528.63,880.9,1401.29],[0.475628088,0.524300249,0.553243364,-0.439100161],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_4670:=[[1534.59,883.73,1519.04],[0.458515938,0.544534151,0.518979828,-0.473186677],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_4680:=[[1532.85,879.55,1619.23],[0.50097684,0.500150194,0.523564676,-0.474080182],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_4690:=[[1533.82,875.89,1648.93],[0.436384218,0.553554635,0.549409194,-0.448659802],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_4700:=[[1541.01,849.77,1782.5],[0.437228601,0.553303929,0.572623411,-0.418076957],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_4710:=[[1537.42,788.16,1884.08],[0.409231045,0.569916513,0.589118932,-0.400829145],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_4720:=[[1538.57,754.6,1918.07],[0.450714006,0.54086686,0.563227726,-0.432544162],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_4730:=[[1536.79,611.71,2176.47],[0.460711696,0.533240926,0.555621903,-0.441229134],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_4740:=[[1537.41,567.65,2240.78],[0.420394435,0.566417053,0.595031659,-0.385198087],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_4750:=[[1539.5,514.76,2284.19],[0.328834938,0.6209117,0.65736863,-0.272401779],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_4760:=[[1539.06,488.13,2288.08],[0.322747536,0.628550155,0.660795418,-0.253195865],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_4770:=[[1390.22,482.81,2274.71],[0.365760086,0.620358308,0.638866286,-0.270601179],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_4780:=[[1390.59,511.18,2270.09],[0.358966557,0.619725444,0.636869016,-0.285449193],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_4790:=[[1394.75,584.61,2205.69],[0.437656896,0.573945858,0.572093881,-0.389552544],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_4800:=[[1388.21,650.67,2094.26],[0.481187678,0.540376155,0.532127014,-0.439650852],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_4810:=[[1388.66,744.85,1923.44],[0.439099619,0.57004095,0.560697299,-0.409711335],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_4820:=[[1388,758.23,1907.29],[0.414884005,0.585117146,0.575512779,-0.392803041],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_4830:=[[1385.13,803.49,1867.65],[0.402058621,0.593191344,0.583744967,-0.38172596],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_4840:=[[1386.04,840.98,1793.48],[0.440448414,0.565993309,0.566310359,-0.406139565],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_4850:=[[1383.71,859.36,1720.1],[0.449257531,0.559799884,0.55338585,-0.422558707],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_4860:=[[1387,868.9,1683.85],[0.436426377,0.569613772,0.553719058,-0.42245399],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_4870:=[[1385.53,872.21,1659.32],[0.475710767,0.537253446,0.531828536,-0.449684789],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_4880:=[[1382.73,877.41,1624.82],[0.438542709,0.567337746,0.530687692,-0.451861425],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_4890:=[[1379.67,884.85,1534.01],[0.475740269,0.529080374,0.530626176,-0.460631107],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_4900:=[[1229.05,885.62,1579.59],[0.478100903,0.528123545,0.521318979,-0.469820785],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_4910:=[[1234.13,870.6,1664.44],[0.422978229,0.572860731,0.572681255,-0.406147979],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_4920:=[[1233.27,844.64,1756.68],[0.435650297,0.563527701,0.569694315,-0.409992362],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_4930:=[[1231.38,817,1836.93],[0.391238177,0.592914404,0.602429551,-0.363955812],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_4940:=[[1231.56,756.42,1901.72],[0.44177377,0.558981023,0.565943851,-0.414829736],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_4950:=[[1235.17,697.64,1985.32],[0.43745579,0.563687252,0.579455118,-0.393853881],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_4960:=[[1239.18,628.51,2114],[0.409469359,0.581953959,0.607986388,-0.352160455],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_4970:=[[1238.1,564.32,2208.18],[0.405890909,0.585495957,0.616808112,-0.334656253],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_4980:=[[1238.58,525.67,2240.83],[0.373828942,0.605630436,0.642670996,-0.283615389],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_4990:=[[1240,475.9,2256.41],[0.348707525,0.626271443,0.657623133,-0.231773502],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_5000:=[[1184.19,488.12,2246.64],[0.360145648,0.620482189,0.65081131,-0.248478579],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_5010:=[[1165.33,557.05,2201.99],[0.329210959,0.633051096,0.659830341,-0.235563951],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_5020:=[[1119.6,590.44,2147.97],[0.385220512,0.608987473,0.636458596,-0.275063392],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_5030:=[[1083.13,602.03,2117.71],[0.404836801,0.598562575,0.632715879,-0.27838934],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_5040:=[[1086.77,694.56,1965.14],[0.457004938,0.570636156,0.572527829,-0.371123631],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_5050:=[[1085.22,735.8,1911.6],[0.435432608,0.578861498,0.576987368,-0.377363733],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_5060:=[[1085.6,796.76,1852.98],[0.399583401,0.600565214,0.601139164,-0.343927659],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_5070:=[[1081.33,853.41,1688.11],[0.483999202,0.543946018,0.561143827,-0.39368148],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_5080:=[[1081.35,861.15,1664.69],[0.455722677,0.562445,0.586388357,-0.363484744],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_5090:=[[1082.24,885.7,1592.06],[0.462810464,0.559201708,0.571396377,-0.382891766],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_5100:=[[1080.76,884.97,1558.7],[0.476825671,0.559032944,0.525807064,-0.428539821],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_5110:=[[987.56,882.41,1516],[0.446713409,0.580390296,0.556523304,-0.392270375],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_5120:=[[932.76,879.1,1515.9],[0.506736276,0.527813692,0.61160528,-0.300948556],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_5130:=[[933.12,853.87,1590.32],[0.498012423,0.533634038,0.613962115,-0.300447768],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_5140:=[[933.47,832.95,1665.62],[0.521704848,0.531242674,0.559951273,-0.36340039],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_5150:=[[934.39,804.01,1796.53],[0.469283943,0.567635861,0.578110623,-0.351212496],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_5160:=[[934.3,737.95,1888.52],[0.454615808,0.573265584,0.584004885,-0.351609628],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_5170:=[[936.26,659.09,1967.47],[0.531235684,0.539365694,0.569113293,-0.3209102],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_5180:=[[935.96,637.7,1993.96],[0.552515602,0.529164811,0.562517094,-0.313505395],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_5190:=[[885.68,640.65,1959.97],[0.565217636,0.526561075,0.568774823,-0.282413983],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_5200:=[[853.8,702.14,1894.37],[0.514124078,0.533300471,0.584925378,-0.330347307],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_5210:=[[830.38,796.31,1724.75],[0.580931723,0.486220974,0.53994173,-0.36683842],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_5220:=[[781.96,779.99,1691.45],[0.603715548,0.449209571,0.582435971,-0.307419319],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_5230:=[[782.32,782.69,1653.01],[0.618658056,0.429439678,0.590522319,-0.290046828],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_5240:=[[783.03,753.44,1596.18],[0.539339271,0.514423698,0.561544591,-0.359373178],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_5250:=[[783.22,760.19,1548.1],[0.614712475,0.438188985,0.559047749,-0.342906111],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_5260:=[[775.7,748.45,1446.33],[0.607295843,0.392782251,0.638879339,-0.262196591],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_5270:=[[783.21,750.58,1415.94],[0.58965544,0.397949341,0.657079988,-0.249376571],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_5280:=[[842.3,790.23,1421.03],[0.588885036,0.399755158,0.656221474,-0.250566567],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_5290:=[[973.973294736,861.715612246,1593.498139036],[0.611052265,0.417934158,0.593787121,-0.315219011],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

CONST robtarget Pos_Int5:=[[300,1250,1590.32],[0.579227965,0.579227965,0.405579788,0.405579788],[0,0,0,-1],[-300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_5300:=[[786.56,-98.29118776,1325.046169734],[0.487298903,0.512386357,0.487298903,0.512386357],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_5310:=[[786.56,733.69,1366.83],[0.487298903,0.512386357,0.487298903,0.512386357],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_5320:=[[736.99,654.38,1413.25],[0.547325722,0.359083621,0.382462403,0.652085898],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_5330:=[[726.19,390.61,1410.2],[0.539263833,0.41054445,0.30112502,0.670799147],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_5340:=[[717.33,62.54,1409.96],[0.575659449,0.315956011,0.343894753,0.671211141],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_5350:=[[655.59,63.11,1421.72],[0.575653611,0.315951873,0.343963157,0.671183046],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_5360:=[[641.46,62.73,1476.74],[0.527325538,0.476665121,0.522293361,0.471092119],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_5370:=[[650.91,340.17,1470.51],[0.549353422,0.499487212,0.499885338,0.445912538],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_5380:=[[665.36,451.83,1481.5],[0.566204383,0.515453411,0.481137761,0.42688035],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_5390:=[[682.71,526.26,1478.15],[0.602108707,0.53218426,0.440833909,0.399888089],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_5400:=[[752.36,728.43,1478.59],[0.620667614,0.542422837,0.417171904,0.382775106],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_5410:=[[746.67,730.41,1579.73],[0.618059604,0.543346697,0.41978299,0.382830165],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_5420:=[[684.92,548.35,1582.31],[0.603556986,0.542108392,0.436415625,0.389074361],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_5430:=[[649.42,393.32,1575.11],[0.484097431,0.561159391,0.543096313,0.394710285],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];

```

LOCAL CONST robtarget
Paint_5440:=[[662.94,428.45,1624.37],[0.600832921,0.300855015,0.715480113,0.191244006],[0,0,0,0],[-
300,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_5450:=[[654.4,438.46,1675],[0.639769185,0.251241259,0.682254905,0.249201654],[0,0,0,0],[-
300,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget
Paint_5460:=[[651.96,504.53,1677.7],[0.695569386,0.239232167,0.623187148,0.265685865],[0,0,0,0],[-
300,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget
Paint_5470:=[[660.11,448.95,1732.73],[0.610848208,0.212636646,0.706362064,0.287580872],[0,0,0,0],[-
300,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget
Paint_5480:=[[669.98,435.64,1769.54],[0.544698234,0.080587005,0.736872487,0.392209774],[0,0,0,0],[-
300,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_5490:=[[682.3,430.06,1807.4],[0.56895347,0.058599526,0.694500017,0.436494869],[0,0,0,0],[-
300,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget
Paint_5500:=[[708.72,443.81,1845.98],[0.474794678,0.102426696,0.843498176,0.229324253],[0,0,0,0],[-
300,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget
Paint_5510:=[[721.84,485.24,1867.18],[0.420367202,0.05999377,0.892978715,0.149268808],[0,0,0,0],[-
300,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_5520:=[[762.01,539.3,1878.92],[0.449817228,0.014975471,0.89167558,0.048526873],[0,0,0,0],[-
300,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_5530:=[[845.4,656.59,1886.06],[0.30132462,-0.029932568,0.948027548,-
0.097730662],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_5540:=[[845.19,567.12,1897.57],[0.379325251,-0.041497419,0.920901987,-
0.079560344],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_5550:=[[889.8,572.89,1923.35],[0.454837785,-0.04604597,0.881609937,-
0.117329777],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_5560:=[[951.95,563.69,1972.57],[0.474249274,-0.048600814,0.871282617,-
0.116585541],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_5570:=[[989.6,556.55,2005.68],[0.558073249,-0.058710267,0.816581246,-
0.135286444],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_5580:=[[1022.56,550.39,2035.87],[0.569247694,-0.06030638,0.8081101,-
0.138846205],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_5590:=[[1075.81,538.08,2087.56],[0.560072085,-0.055365828,0.808072725,-
0.173989527],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_5600:=[[1099.64,526.21,2111.65],[0.64358757,0.19382955,0.73731962,0.067712059],[0,0,0,0],[-
300,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget
Paint_5610:=[[1110.11,490.83,2135.78],[0.641933346,0.401598764,0.634709442,0.154220417],[0,0,0,0],[-
300,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget
Paint_5620:=[[1105.15,428.73,2157.58],[0.65883986,0.373215792,0.627378339,0.18175927],[0,0,0,0],[-
300,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget
Paint_5630:=[[1104.39,388.12,2167.99],[0.65883986,0.373215792,0.627378339,0.18175927],[0,0,0,0],[-
300,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget
Paint_5640:=[[1105.38,312.23,2177.88],[0.65883986,0.373215792,0.627378339,0.18175927],[0,0,0,0],[-
300,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_5650:=[[1114.84,243.6,2190.41],[0.65883986,0.373215792,0.627378339,0.18175927],[0,0,0,0],[-
300,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget
Paint_5660:=[[1123.51,166.97,2190.05],[0.65883986,0.373215792,0.627378339,0.18175927],[0,0,0,0],[-
300,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_5670:=[[1134.82,8.92,2189.29],[0.65883986,0.373215792,0.627378339,0.18175927],[0,0,0,0],[-
300,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget
Paint_5680:=[[1195.51,6.36,2193.82],[0.688860674,0.314373916,0.609149611,0.235747246],[0,0,0,0],[-
300,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget
Paint_5690:=[[1188.45,259.46,2196.43],[0.704837915,0.306185908,0.592925651,0.240609384],[0,0,0,0],[-
300,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget
Paint_5700:=[[1178.71,388.94,2197.24],[0.693430509,0.274296244,0.60760462,0.273372138],[0,0,0,0],[-
300,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget
Paint_5710:=[[1142.891763549,506.58559787,2188.980652214],[0.69157784,0.308350579,0.60706916,0.241054034],[0,0,0,0],[-
300,9E+09,9E+09,9E+09,9E+09,0]];

```

LOCAL CONST robtarget
 Paint_5720:=[[1141.11,537.44,2172.79],[0.705959588,0.09719886,0.701387208,0.015146782],[0,0,0],[
 -300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_5730:=[[1068.82,566.56,2114.6],[0.606280395,0.160979045,0.74575826,-
 0.224397965],[0,0,0],[
 -300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_5740:=[[1029.14,587.88,2069.19],[0.59061356,0.079684171,0.803001032,0.00392399],[0,0,0],[
 -300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_5750:=[[969.94,601.94,2017.48],[0.608909779,0.109269416,0.784883712,0.035307697],[0,0,0],[
 -300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_5760:=[[769.466482243,637.168500177,1873.577979676],[0.577092357,0.136111123,0.804001265,0.044945957],[0,0,0],[
 -300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_5770:=[[1205.535989659,-
 140.903958016,2240.905859],[0.546593133,0.443962211,0.44922121,0.549848893],[0,0,0],[
 -300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_5780:=[[1184.17,442.88,2235.82],[0.54297943,0.421997402,0.45256553,0.567693555],[0,0,0],[
 -300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_5790:=[[1178.56,491.4,2219.88],[0.506746447,0.407328817,0.487034709,0.583171043],[0,0,0],[
 -300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_5800:=[[1126.74,483.69,2304.63],[0.546547891,0.352128001,0.423408709,0.630885361],[0,0,0],[
 -300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_5810:=[[1135.12,276.16,2311.73],[0.546547891,0.352128001,0.423408709,0.630885361],[0,0,0],[
 -300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_5820:=[[1148.3,5.3,2309.08],[0.546547891,0.352128001,0.423408709,0.630885361],[0,0,0],[
 -300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_5830:=[[1091.88,6.21,2359.68],[0.546547891,0.352128001,0.423408709,0.630885361],[0,0,0],[
 -300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_5840:=[[1085.83,385.53,2352.43],[0.546547891,0.352128001,0.423408709,0.630885361],[0,0,0],[
 -300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_5850:=[[1073.89,453.99,2346.21],[0.546547891,0.352128001,0.423408709,0.630885361],[0,0,0],[
 -300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_5860:=[[1037.93,515.43,2341.88],[0.546547891,0.352128001,0.423408709,0.630885361],[0,0,0],[
 -300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_5870:=[[917.57,557.43,2390.9],[0.182509412,0.484800095,0.032451381,0.854754988],[0,0,0],[
 -300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_5880:=[[785.46,548.59,2478.65],[0.182509412,0.484800095,0.032451381,0.854754988],[0,0,0],[
 -300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_5890:=[[716.92,678.99,2477.9],[0.55117352,0.26124459,0.414338066,0.675487218],[0,0,0],[
 -300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_5900:=[[717.81,574.41,2523.55],[0.6190905,0.24549175,0.255880738,0.70070379],[0,0,0],[
 -300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_5910:=[[662.63,458.24,2575.95],[0.6190905,0.24549175,0.255880738,0.70070379],[0,0,0],[
 -300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_5920:=[[642.75,377.15,2550.91],[0.645837199,0.162584009,0.162231445,0.72810831],[0,0,0],[
 -300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_5930:=[[613.53,173.3,2596.76],[0.645837199,0.162584009,0.162231445,0.72810831],[0,0,0],[
 -300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_5940:=[[610.22,73.3,2574.8],[0.645837199,0.162584009,0.162231445,0.72810831],[0,0,0],[
 -300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_5950:=[[615.47,13.72,2567.05],[0.645837199,0.162584009,0.162231445,0.72810831],[0,0,0],[
 -300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_5960:=[[492.08,6.09,2570.75],[0.645837199,0.162584009,0.162231445,0.72810831],[0,0,0],[
 -300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_5970:=[[498.8,170.8,2564.01],[0.645837199,0.162584009,0.162231445,0.72810831],[0,0,0],[
 -300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_5980:=[[509.97,297.15,2551.53],[0.645837199,0.162584009,0.162231445,0.72810831],[0,0,0],[
 -300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_5990:=[[539.71,432.81,2613.41],[0.645837199,0.162584009,0.162231445,0.72810831],[0,0,0],[
 -300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_6000:=[[466.12,499.59,2597.98],[0.645837199,0.162584009,0.162231445,0.72810831],[0,0,0],[
 -300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_6010:=[[444.63,371.47,2589.49],[0.645837199,0.162584009,0.162231445,0.72810831],[0,0,0],[
 -300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_6020:=[[434.08,249.72,2539.69],[0.645837199,0.162584009,0.162231445,0.72810831],[0,0,0],[
 -300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_6030:=[[433.87,140.72,2548.11],[0.645837199,0.162584009,0.162231445,0.72810831],[0,0,0],[
 -300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_6040:=[[434.29,13.81,2552.52],[0.645837199,0.162584009,0.162231445,0.72810831],[0,0,0],[
 -300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_6050:=[[369.41,9.07,2532.11],[0.645837199,0.162584009,0.162231445,0.72810831],[0,0,0],[
 -300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_6060:=[[373.31,165.08,2528.94],[0.645837199,0.162584009,0.162231445,0.72810831],[0,0,0],[
 -300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_6070:=[[379.53,271.12,2522.29],[0.645837199,0.162584009,0.162231445,0.72810831],[0,0,0],[
 -300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_6080:=[[383.48,368.7,2577.68],[0.645837199,0.162584009,0.162231445,0.72810831],[0,0,0],[
 -300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_6090:=[[397.72,526.18,2575.1],[0.645837199,0.162584009,0.162231445,0.72810831],[0,0,0],[
 -300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_6100:=[[319.22,461.36,2564.42],[0.645837199,0.162584009,0.162231445,0.72810831],[0,0,0],[
 -300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_6110:=[[311,379.55,2567.87],[0.645837199,0.162584009,0.162231445,0.72810831],[0,0,0],[
 -300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_6120:=[[306.51,261.97,2574.53],[0.645837199,0.162584009,0.162231445,0.72810831],[0,0,0],[
 -300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_6130:=[[302.599528639,-
 142.382276444,2579.362452413],[0.645837199,0.162584009,0.162231445,0.72810831],[0,0,0],[
 -300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_6140:=[[4270.160778555,787.838079383,1420.40718973],[0.314725234,0.140197302,0.159800499,0.925071102],[0,0,0],[
 1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_6150:=[[4082.89,797.47,1545.98],[0.358932885,0.082853506,0.083948436,0.925880738],[0,0,0],[1200,9E+09,9E+09,9E+0
 9,9E+09,0]];

LOCAL CONST robtarget
 Paint_6160:=[[4016.2,796,1576.31],[0.357140692,0.042298129,0.048182432,0.931847545],[0,0,0],[1200,9E+09,9E+09,9E+09,9E
 +09,0]];

LOCAL CONST robtarget Paint_6170:=[[3948.91,793.27,1591.66],[0.419368529,-
 0.040798411,0.061586504,0.904805299],[0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_6180:=[[3844.59,797.46,1601.15],[0.354543968,-
 0.142229106,0.052972918,0.922639326],[0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_6190:=[[3733.52,795.14,1559.96],[0.239814912,-0.348934199,-
 0.035122807,0.90526246],[0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_6200:=[[3615.18,785.21,1473.09],[0.135174927,-0.452072198,-
 0.013847506,0.881570595],[0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_6210:=[[3542.98,773.7,1369.61],[0.152662931,-0.590785185,-
 0.042958041,0.791088807],[0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_6220:=[[3521.11,748.98,1201.29],[0.149918163,-0.676052515,-
 0.15899322,0.703703558],[0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_6230:=[[3515.26,736.2,1080.01],[0.388361388,-0.593136742,-
 0.372542279,0.598812566],[0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_6240:=[[3515.34,597.89,1075.93],[0.310819699,-0.524278955,-
 0.439362147,0.65991181],[0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_6250:=[[3531.02,588.78,1261.08],[0.005833382,-0.631188444,-
 0.280349824,0.723167405],[0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_6260:=[[3613.89,589.57,1447.45],[0.078739423,0.384514486,0.269128511,-
 0.879499038],[0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_6270:=[[3704.11,613.63,1535.18],[0.358826931,-0.069994512,-
 0.330664831,0.870060211],[0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_6280:=[[4006.88,631.86,1573.73],[0.410878752,0.23295112,-0.182973287,0.862225727],[0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_6290:=[[4060.67,622.09,1545.77],[0.349842366,0.294031619,-0.152217728,0.876347813],[0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_6300:=[[3969.56,517.75,1466.53],[0.023821703,0.044895486,0.482213476,-0.874578233],[0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_6310:=[[3856.07,512.75,1461.1],[0.025757344,0.043813667,0.519903082,-0.852711972],[0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_6320:=[[3832.42,497.94,1359.61],[0.027643954,0.042648447,0.556603023,-0.829222525],[0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_6330:=[[4027.63,497.94,1348.15],[0.029477942,0.041402043,0.592243439,-0.804154606],[0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_6340:=[[4717.972556058,497.94,1367.260889413],[0.029477942,0.041402043,0.592243439,-0.804154606],[0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

CONST jointtarget Pos_Int3_1:=[[180,-20,20,0,45,0],[6545,9E+09,9E+09,9E+09,9E+09,0]];

CONST robtarget Pos_Int6:=[[1000,1250,2500],[0.664463024,0.241844763,0.664463024,0.241844763],[0,0,0],[-1,-300,9E+09,9E+09,9E+09,0]];

CONST robtarget Pos_Int6_2:=[[2154.115447064,2072.991980913,1696.026532984],[0.101402602,0.391811429,0.908417772,-0.104778183],[2,0,-1,0],[-300,9E+09,9E+09,9E+09,0]];

CONST robtarget Aprox_Bajos1:=[[4270.163604977,1359.27550049,1420.405936761],[0.314724899,0.140198888,0.159800047,0.925071054],[1,1,0,-1],[1200,9E+09,9E+09,9E+09,9E+09,0]];

CONST robtarget Pos_Int7:=[[807.117,900,1396.54],[0.69636424,0.69636424,-0.122787804,-0.122787804],[0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_6350:=[[807.116851231,790.874413283,1396.53980705],[0.908925139,0.006395513,-0.280840554,0.30812785],[0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_6360:=[[1029.92,807.46,1583.72],[0.90147148,-0.01187204,-0.205467783,0.38077712],[0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_6370:=[[1160.06,806.02,1630.29],[0.821541675,-0.011309553,-0.103154292,0.560625153],[0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_6380:=[[1294.91,804.06,1624.03],[0.688469514,0.08042346,0.028365557,0.720234122],[0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_6390:=[[1418.03,799.87,1567.47],[0.672305652,0.217540184,0.156761812,0.690005155],[0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_6400:=[[1539.52,792.85,1402.64],[0.792879737,0.152823629,0.574573699,0.133610349],[0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_6410:=[[1571.2,793.93,1261.77],[0.750829085,0.152239996,0.637965814,0.077963386],[0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_6420:=[[1579.72,805.08,1083.34],[0.7160612,0.055830281,0.695766155,-0.006985356],[0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_6430:=[[1569.45,699.14,1079.9],[0.707220948,0.11376911,0.695319005,0.058537183],[0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_6440:=[[1562.51,690.06,1244.02],[0.711481841,0.166746302,0.680495201,-0.053995761],[0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_6450:=[[1553.88,691.46,1325.26],[0.732517831,0.186503614,0.650991701,-0.069597657],[0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_6460:=[[1506.07,683.72,1463.24],[0.810045,0.219227864,0.529779278,-0.122882701],[0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_6470:=[[1366.17,688.32,1597.85],[0.758944789,0.285018709,0.047447215,0.583537406],[0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_6480:=[[1234.39,684.91,1633.33],[0.923831262,0.184068332,0.018161046,0.335148959],[0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_6490:=[[1103.58,687.8,1616.55],[0.911015775,0.121688088,-0.168616663,0.356104884],[0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_6500:=[[986.34,687.66,1552.79],[0.923246472,0.123383924,-0.177388371,0.317688093],[0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_6510:=[[924.64,682.36,1489.62],[0.915007465,0.14542603,-0.194449416,0.322183229],[0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_6520:=[[989.01,620.6,1465.46],[0.87543383,0.445483474,0.030737092,-0.184973821],[0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_6530:=[[1139.79,620.85,1513.9],[0.873259878,0.453127917,0.068893724,-0.165366051],[0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_6540:=[[1273.91,619.5,1515.71],[0.848029829,0.4749594,0.21948706,-0.084168932],[0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_6550:=[[1442.4,623.87,1409.54],[0.825673261,0.480487842,0.292562515,-0.042453205],[0,0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_6560:=[[1499.48,602.38,1289.54],[0.825673261,0.480487842,0.292562515,-0.042453205],[0,0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_6570:=[[1483.59,530.64,1251.01],[0.825673261,0.480487842,0.292562515,-0.042453205],[0,0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_6580:=[[1242.48,530.64,1315.44],[0.791319308,0.581074925,0.15658419,-0.107921622],[0,0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_6590:=[[1058.63,530.64,1345.44],[0.797396254,0.575814395,0.121918293,-0.133165037],[0,0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_6600:=[[941.98,530.64,1335.69],[0.804987799,0.56201715,0.051956695,-0.182843835],[0,0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_6610:=[[838.98,530.64,1332.85],[0.829863974,0.517801372,0.025815452,-0.206254907],[0,0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_6620:=[[818.86,638.07,1311.29],[0.80195531,0.56945777,0.087020318,-0.158154964],[0,0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_6630:=[[638.833779579,619.768101632,1296.671502562],[0.80195531,0.56945777,0.087020318,-0.158154964],[0,0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];

CONST robtarget Limpieza_1:=[[-8474.76,0,-150],[0,1,0,0],[1,-2,1,0],[6400,9E+09,9E+09,9E+09,9E+09,0]];

CONST robtarget Limpieza_2:=[[-8474.76,0,-290],[0,1,0,0],[1,-2,1,0],[6400,9E+09,9E+09,9E+09,9E+09,0]];

CONST jointtarget Pos_Int8:=[[0,0,0,0,-45,0],[6670,9E+09,9E+09,9E+09,9E+09,0]];

CONST robtarget Pos_Int9:=[[255.344923914,763.925476211,1148.354086804],[0.747986491,0.617246508,0.053526284,-0.23802919],[1,-1,-2,1],[-480,9E+09,9E+09,9E+09,9E+09,0]];

CONST jointtarget Pos_Int10:=[[90,-30,30,0,-45,0],[6670,9E+09,9E+09,9E+09,9E+09,0]];

CONST robtarget Pos_Sal_PD1:=[[2748.618955261,1179.263669934,1392.253424999],[0.141544526,0.157714474,0.976686003,0.034289112],[1,0,-1,-1],[520,9E+09,9E+09,9E+09,9E+09,0]];

CONST robtarget Pos_Sal_PD2:=[[2500.086295605,2927.007611965,1392.253348344],[0.10095399,-0.391759141,0.908467962,0.104971566],[1,-1,0,-1],[520,9E+09,9E+09,9E+09,9E+09,0]];

CONST jointtarget Pos_Sal_PD2_3:=[[0,-45.000004667,44.999997837,-0.000050014,-44.999946611,0.000055621],[3500,9E+09,9E+09,9E+09,9E+09,0]];

CONST robtarget Pos_Sal_PT1:=[[1682.581843921,1214.372797184,1392.2533107],[0.14154428,0.15771642,0.97668574,0.034288667],[1,0,-1,-1],[-425,9E+09,9E+09,9E+09,9E+09,0]];

CONST robtarget Pos_Sal_Cap1:=[[4001.552847748,284.910998197,1899.25213869],[0.653276919,-0.218241123,-0.695989512,0.202974575],[0,-1,-1,-1],[3750,9E+09,9E+09,9E+09,9E+09,0]];

CONST robtarget Pos_Sal_Cap2:=[[4478.89911865,284.908605081,1899.252062558],[0.65327647,-0.218241759,-0.695989387,0.202975767],[1,-2,0,-1],[3750,9E+09,9E+09,9E+09,9E+09,0]];

CONST jointtarget Pos_Sal_Cap3:=[[180,-45,45,0,45,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];

CONST robtarget Pos_Sal_Malet1:=[[133.378000877,297.145125778,2145.167751278],[0.576876222,0.585143842,0.390171697,0.415435381],[0,0,-1,-1],[-300,9E+09,9E+09,9E+09,9E+09,0]];

CONST robtarget Pos_Sal_Baj1_1:=[[3937.76467587,692.52494689,914.824127148],[0.484575156,-0.306727907,-0.464307982,0.674924445],[1,-1,-1,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

CONST robtarget Pos_Sal_Baj2_1:=[[1058.88130485,855.002561054,964.619692874],[0.630625655,0.342019887,0.323799937,0.616836511],[0,2,-2,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];

CONST robtarget Pos_Sal_Malet2:=[[32.728211432,2386.802581734,1985.830539871],[0.286787613,0.736279339,0.286787585,-0.541662701],[-1,-1,0,-1],[-300,9E+09,9E+09,9E+09,9E+09,0]];

CONST robtarget Pos_Sal_PD2_2:=[[787.271788568,2386.802581734,1985.830539871],[0.286787613,0.736279339,0.286787585,-0.541662701],[-1,-1,0,-1],[520,9E+09,9E+09,9E+09,9E+09,0]];

CONST robtarget Pos_Sal_PT2_2:=[[157.728211432,2386.802581734,1985.830539871],[0.286787613,0.736279339,0.286787585,-0.541662701],[-1,-1,0,-1],[-425,9E+09,9E+09,9E+09,9E+09,0]];

CONST robtarget Pos_Sal_PD2_1:=[[1172.001687426,1522.881731348,1392.253348344],[0.135614501,0.689395223,0.70959389,-0.053098446],[0,-1,0,-1],[520,9E+09,9E+09,9E+09,9E+09,0]];

CONST robtarget Pos_Sal_PT2_1:=[[402.346210775,1472.91400404,1392.253474508],[0.139726221,0.624927109,0.766978083,-0.041076898],[0,-1,0,-1],[-425,9E+09,9E+09,9E+09,9E+09,0]];

CONST robtarget Mantenimiento_2:=[[2025.24,350,1360],[0.707106781,0,0.707106781,0],[0,-2,1,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

CONST robtarget Mantenimiento_3:=[[2675.24,350,1360],[0.707106781,0,0.707106781,0],[0,-2,1,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

CONST jointtarget Pos_Mant_Int_1:=[[180,-45,45,0,45,0],[500,9E+09,9E+09,9E+09,9E+09,0]];

CONST jointtarget Pos_Mant_Int_2:=[[90,-45,45,0,45,0],[500,9E+09,9E+09,9E+09,9E+09,0]];

CONST jointtarget Pos_Mant_Int_3:=[[0,-45,45,0,45,0],[500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_6640:=[[-8474.76,700,0],[0,1,0,0],[1,-2,1,0],[6400,9E+09,9E+09,9E+09,9E+09,0]];


```
LOCAL CONST robtarget Paint_6650:=[[-8474.76,700,-265],[0,1,0,0],[1,-2,1,0],[6400,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_6660:=[[-8474.76,700,0],[0,1,0,0],[1,-2,1,0],[6400,9E+09,9E+09,9E+09,0]];
```

!!!!!!!!!!!!!!!!!!!!!! DEFINICIÓN DE VARIABLES !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

```
VAR num Color;
VAR intnum Limp_Atom;
VAR intnum Paro_Robot;
VAR intnum Mant_Atom;
VAR num Lectura_Color;
VAR iodev infile;
VAR num Contador;
VAR num Secuencia_Colores{1000};
VAR string date;
VAR string N_Proc;
VAR string Limpieza;
```

!!!!!!!!!!!!!!!!!!!!!! PROCESOS DE PINTADO !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

```
PROC main()
```

```
Reset Abrir_P;
Reset Volver;
Reset Avance_Inf_P;
Reset Stop_Conveyor;
Reset Finalizado;
Reset Cambio_Color;
Reset Get_Contador;
Reset Pos_Parada;
Reset Paro_Resto;
Reset Open_TMant;
Reset Close_TMant;
SetAO Señal_Luminosa,3;
Lectura_Colores;
! Eliminar los avisos de trayectoria de fallo de esquina
CornerPathWarning FALSE;
```

```
! Definición de los parametros para la rutina TRAP de limpieza si se requiere la limpieza del atomizador
IDelete Limp_Atom;
CONNECT Limp_Atom WITH Limpieza_Atomizador;
ISignalDI Limpieza_Atom,1,Limp_Atom;
```

```
! Definición de los parametros para la rutina TRAP de paro del robot en caso de que otro de los robots lo solicite
IDelete Paro_Robot;
CONNECT Paro_Robot WITH Parada;
ISignalDI Paro_Propio,1,Paro_Robot;
```

```
! Definición de los parametros para la rutina TRAP de mantenimiento si se requiere realizar mantenimiento al atomizador
IDelete Mant_Atom;
CONNECT Mant_Atom WITH Mantenimiento_Atomizador;
ISignalDI Mantenimiento,1,Mant_Atom;
```

```
Set Get_Contador;
Color:=Secuencia_Colores{1};
Set Cambio_Color;
ActUnit CNV1;
FOR i1 FROM 1 TO Contador-1 DO
Principal;
Reset Cambio_Color;
N_Proc:="main";
WaitRob\ZeroSpeed;
IF Finalizado=1 AND Color<>Secuencia_Colores{i1+1} AND Secuencia_Colores{i1+1}<>0 THEN
Limpieza:="Limp1";
MoveAbsJ Pos_Int10,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atom_Color;
Color:=Secuencia_Colores{i1+1};
Set Cambio_Color;
! Purga de color para preparar el atomizador
PaintL Paint_6640,v2000,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetBrush Color\Z:=-150;
PaintL Paint_6650,v800,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
WaitTime 2;
```

```

SetBrush 1\Z:=-150;
PaintL Paint_6660,v800,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
N_Proc:="main";
MoveAbsJ Pos_Int10,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
ELSEIF Finalizado=1 AND Color<>Secuencia_Colores{i1+1} AND Secuencia_Colores{i1+1}=0 THEN
Limpieza:="Limp1";
MoveAbsJ Pos_Int10,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atom_Color;
N_Proc:="main";
MoveAbsJ Pos_Int10,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
ELSE
Color:=Secuencia_Colores{i1+1};
Set Cambio_Color;
MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
ENDIF
ENDFOR
Reset Get_Contador;
DeactUnit CNV1;
! Stop;
ENDPROC

PROC Principal()
Reset Abrir_P;
Reset Volver;
N_Proc:="Principal1";
MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_200\WObj:=WO_In_Int;
WaitWObj wobj_cnv1;
Reset Finalizado;
WaitDI Init_Pint1,1;
Pint_Bajo_Capo;
N_Proc:="Principal2_2";
MoveLDO Pos_Int1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1,Abrir_P,1;
WaitDI Init_Pint2,1;
Reset Abrir_P;
Pint_PI_Del;
N_Proc:="Principal2_3";
MoveLDO Pos_Int2,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1,Abrir_P,1;
WaitDI Init_Pint2,1;
Pint_PI_Tras;
N_Proc:="Principal3";
Reset Abrir_P;
MoveL Pos_Int5,v1500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
WaitDI Init_Pint2,1;
Pint_Malet_Inf;
N_Proc:="Principal3";
MoveL Pos_Int5,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Pos_Int6,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
N_Proc:="Principal2";
MoveL Pos_Int6_2,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveAbsJ Pos_Int3_1,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Reset Avance_Inf_P;
Pint_Bajos_1;
N_Proc:="Principal2";
MoveAbsJ Pos_Int3_1,v1000,z100,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Pint_Lateral;
Pint_Bajos_2;
N_Proc:="Principal3";
Set Finalizado;
MoveAbsJ Pos_Int8,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Set Volver;
DropWObj wobj_cnv1;
ENDPROC

PROC Lectura_Colores()
VAR string Num_CE;
! Lectura del archivo con la cadena de números referentes a los colores
date:=CDate();
Open date+"_Lista_Colores.txt",infile\Read;
Lectura_Color:=ReadNum(infile\Delim:=" ");

```

```

Contador:=1;
Num_CE:=NumToStr(Contador,0);
! Comprobación de que el archivo no está vacío o contine codigos de color incorrectos.
IF Lectura_Color>EOF_NUM THEN
  TPWrite "El archivo que contiene la Secuencia de colores está vacío/incorrecto, revíselo.";
  ErrWrite "Color error","El archivo que contiene la Secuencia de colores está vacío/incorrecto, revíselo.";
  Break;
  ! Stop;
ELSEIF Lectura_Color>16 OR Lectura_Color<2 THEN
  TPWrite "El identificador de color "+Num_CE+" es incorrecto, revíselo.";
  ErrWrite "Color error","El identificador de color "+Num_CE+" es incorrecto, revíselo."\\RL2:="Identificador superior a 16 o inferior a 2.";
  Break;
ENDIF
! Lectura de la secuencia de colores completa.
Secuencia_Colores{Contador}:=Lectura_Color;
WHILE Lectura_Color<EOF_NUM DO
  Contador:=Contador+1;
  Num_CE:=NumToStr(Contador,0);
  Lectura_Color:=ReadNum(infile\\Delim:=" ");
  ! Comprobación de que el archivo no continen ningún codigo de color incorrecto.
  IF Lectura_Color<EOF_NUM THEN
    IF Lectura_Color>16 OR Lectura_Color<2 THEN
      TPWrite "El identificador de color "+Num_CE+" es incorrecto, revíselo.";
      ErrWrite "Color error","El identificador de color "+Num_CE+" es incorrecto, revíselo."\\RL2:="Identificador superior a 16 o inferior a 2.";
      Break;
    ENDIF
    Secuencia_Colores{Contador}:=Lectura_Color;
  ENDWHILE
  Secuencia_Colores{Contador}:=0;
ENDPROC

PROC Limpieza_Atom_Color()
  N_Proc:="Limpieza_Atom_Color";
  MoveL Limpieza_1,v1500,fine,RB1000_WSC_70_700_0\\WObj:=WO_In_Int;
  MoveL Limpieza_2,v100,fine,RB1000_WSC_70_700_0\\WObj:=WO_In_Int;
  IF Limpieza="Limp1" THEN
    Set RUN_Limp;
    WaitTime 5;
    MoveL Limpieza_1,v100,fine,RB1000_WSC_70_700_0\\WObj:=WO_In_Int;
  ELSEIF Limpieza="Limp2" THEN
    Set RUN_Limp;
    WaitTime 5;
    MoveL Limpieza_1,v100,fine,RB1000_WSC_70_700_0\\WObj:=WO_In_Int;
    ! Purga de color para preparar el atomizador
    PaintL Paint_6640,v2000,z50,RB1000_WSC_70_700_0\\WObj:=WO_In_Int;
    SetBrush Color\\Z:=-150;
    PaintL Paint_6650,v800,z50,RB1000_WSC_70_700_0\\WObj:=WO_In_Int;
    WaitTime 2;
    SetBrush 1\\Z:=-150;
    PaintL Paint_6660,v800,z50,RB1000_WSC_70_700_0\\WObj:=WO_In_Int;
  ENDIF
  Reset RUN_Limp;
ENDPROC

PROC Mantenimiento_Atom()
  MoveAbsJ Pos_Mant_Int_3,v2500,z50,RB1000_WSC_70_700_0\\WObj:=WO_In_Int;
  MoveLDO Mantenimiento_2,v500,fine,RB1000_WSC_70_700_200\\WObj:=WO_In_Int,Open_TMant,1;
  WaitDI Ok_TMant,1;
  VelSet 100,100;
  MoveL Mantenimiento_3,v100,fine,RB1000_WSC_70_700_0\\WObj:=WO_In_Int;
  SetAO Señal_Luminosa,2;
  WaitDI Mantenimiento,0;
  SetAO Señal_Luminosa,1;
  Reset Open_TMant;
  MoveLDO Mantenimiento_2,v100,fine,RB1000_WSC_70_700_200\\WObj:=WO_In_Int,Close_TMant,1;
  VelSet 100,MaxRobSpeed();
  MoveAbsJ Pos_Mant_Int_3,v2500,z50,RB1000_WSC_70_700_0\\WObj:=WO_In_Int;
  Reset Close_TMant;

```



```

PaintL Paint_6540,v800,z100,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_6550,v800,z100,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_6560,v800,z100,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_6570,v800,z100,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_6580,v800,z100,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_6590,v800,z100,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_6600,v800,z100,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_6610,v800,z100,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_6620,v800,z100,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
SetBrush 1\X:=787.58;
PaintL Paint_6630,v800,z100,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Pos_Int9,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;

```

ENDPROC

!!!!!!!!!!!!!!!!!!!!!!!!!!!! RUTINAS TRAP !!!!!!!!!!!!!!!!!!!!!!!!!!!!!

TRAP Limpieza_Atomicador

VAR robtarget Punt_Paro;

VAR jointtarget jointPunt_Paro;

VAR string N_PROG2;

StopMove\Quick;

Set Paro_Resto;

Set Stop_Conveyor;

SetAO Señal_Luminosa,1;

StorePath;

N_PROG2:=N_Proc;

Limpieza:="Limp2";

Punt_Paro:=CRobT(\Tool:=RB1000_WSC_70_700_200\WObj:=WO_In_Int);

jointPunt_Paro:=CJointT(\TaskRef:=T_ROB1Id);

TEST N_Proc

CASE "main":

MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;

MoveAbsJ Pos_Int10,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;

Limpieza_Atomicolor;

MoveAbsJ Pos_Int10,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;

MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;

SetAO Señal_Luminosa,2;

WaitDI Limpieza_Atomicolor,0;

Reset Paro_Resto;

SetAO Señal_Luminosa,1;

CASE "Principal1":

MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;

MoveAbsJ Pos_Int10,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;

Limpieza_Atomicolor;

MoveAbsJ Pos_Int10,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;

MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;

SetAO Señal_Luminosa,2;

WaitDI Limpieza_Atomicolor,0;

Reset Paro_Resto;

SetAO Señal_Luminosa,1;

CASE "Principal2":

MoveAbsJ Pos_Sal_Cap3,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;

Limpieza_Atomicolor;

MoveAbsJ Pos_Sal_Cap3,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;

SetAO Señal_Luminosa,2;

WaitDI Limpieza_Atomicolor,0;

Reset Paro_Resto;

SetAO Señal_Luminosa,1;

CASE "Principal2_2":

MoveL Pos_Sal_PD2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;

MoveL Pos_Sal_PD2_2,v2500,z50,tool0\WObj:=wobj_cnv1;

MoveAbsJ Pos_Sal_PD2_3,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;

MoveAbsJ Pos_Int8,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;

MoveAbsJ Pos_Int10,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;

Limpieza_Atomicolor;

MoveAbsJ Pos_Int10,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;

MoveAbsJ Pos_Int8,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;

MoveAbsJ Pos_Sal_PD2_3,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;

SetAO Señal_Luminosa,2;

WaitDI Limpieza_Atomicolor,0;

Reset Paro_Resto;

SetAO Señal_Luminosa,1;
MoveL Pos_Sal_PD2_2,v2500,z50,tool0\WObj:=wobj_cnv1;
MoveL Pos_Sal_PD2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
CASE "Principal2_3":
MoveL Pos_Sal_PT2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Pos_Sal_PT2_2,v2500,z50,tool0\WObj:=wobj_cnv1;
MoveAbsJ Pos_Sal_PD2_3,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Int8,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Int10,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atom_Color;
MoveAbsJ Pos_Int10,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Int8,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Sal_PD2_3,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Limpieza_Atom,0;
Reset Paro_Resto;
SetAO Señal_Luminosa,1;
MoveL Pos_Sal_PT2_2,v2500,z50,tool0\WObj:=wobj_cnv1;
MoveL Pos_Sal_PT2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
CASE "Principal3":
MoveAbsJ Pos_Sal_PD2_3,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Int8,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Int10,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atom_Color;
MoveAbsJ Pos_Int10,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Int8,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Sal_PD2_3,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Limpieza_Atom,0;
Reset Paro_Resto;
SetAO Señal_Luminosa,1;
CASE "Limpieza_Atom_Color":
SetAO Señal_Luminosa,2;
WaitDI Limpieza_Atom,0;
Reset Paro_Resto;
SetAO Señal_Luminosa,1;
CASE "Pint_Bajo_Capo":
MoveL Pos_Sal_Cap1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Pos_Sal_Cap2,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveAbsJ Pos_Sal_Cap3,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atom_Color;
MoveAbsJ Pos_Sal_Cap3,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Limpieza_Atom,0;
Reset Paro_Resto;
SetAO Señal_Luminosa,1;
MoveL Pos_Sal_Cap2,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Pos_Sal_Cap1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
CASE "Pint_PI_Del":
MoveL Pos_Sal_PD1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Pos_Int1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Pos_Sal_PD2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Pos_Sal_PD2_2,v2500,z50,tool0\WObj:=wobj_cnv1;
MoveAbsJ Pos_Sal_PD2_3,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Int8,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Int10,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atom_Color;
MoveAbsJ Pos_Int10,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Int8,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Sal_PD2_3,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Limpieza_Atom,0;
Reset Paro_Resto;
SetAO Señal_Luminosa,1;
MoveL Pos_Sal_PD2_2,v2500,z50,tool0\WObj:=wobj_cnv1;
MoveL Pos_Sal_PD2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Pos_Int1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Pos_Sal_PD1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
CASE "Pint_PI_Tras":
MoveL Pos_Sal_PT1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Pos_Int2,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;

```

MoveL Pos_Sal_PT2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Pos_Sal_PT2_2,v2500,z50,tool0\WObj:=wobj_cnv1;
MoveAbsJ Pos_Sal_PD2_3,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Int8,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Int10,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atom_Color;
MoveAbsJ Pos_Int10,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Int8,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Sal_PD2_3,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Limpieza_Atom,0;
Reset Paro_Resto;
SetAO Señal_Luminosa,1;
MoveL Pos_Sal_PT2_2,v2500,z50,tool0\WObj:=wobj_cnv1;
MoveL Pos_Sal_PT2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Pos_Int2,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Pos_Sal_PT1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
CASE "Pint_Lateral":
MoveAbsJ Pos_Sal_Cap3,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atom_Color;
MoveAbsJ Pos_Sal_Cap3,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Limpieza_Atom,0;
Reset Paro_Resto;
SetAO Señal_Luminosa,1;
CASE "Pint_Malet_Inf":
MoveL Pos_Sal_Malet1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Pos_Sal_Malet2,v2500,z50,tool0\WObj:=wobj_cnv1;
MoveAbsJ Pos_Sal_PD2_3,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Int8,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Int10,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atom_Color;
MoveAbsJ Pos_Int10,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Int8,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Sal_PD2_3,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Limpieza_Atom,0;
Reset Paro_Resto;
SetAO Señal_Luminosa,1;
MoveL Pos_Sal_Malet2,v2500,z50,tool0\WObj:=wobj_cnv1;
MoveL Pos_Sal_Malet1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
CASE "Pint_Bajos_1":
MoveL Pos_Sal_Baj1_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Offs(Pos_Sal_Baj1_1,0,400,250),v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveAbsJ Pos_Sal_Cap3,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atom_Color;
MoveAbsJ Pos_Sal_Cap3,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Limpieza_Atom,0;
Reset Paro_Resto;
SetAO Señal_Luminosa,1;
MoveL Offs(Pos_Sal_Baj1_1,0,400,250),v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Pos_Sal_Baj1_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
CASE "Pint_Bajos_2":
MoveL Pos_Sal_Baj2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveAbsJ Pos_Sal_PD2_3,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Int8,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Int10,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atom_Color;
MoveAbsJ Pos_Int10,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Int8,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Sal_PD2_3,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Limpieza_Atom,0;
Reset Paro_Resto;
SetAO Señal_Luminosa,1;
MoveL Pos_Sal_Baj2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
DEFAULT:
MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Int10,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atom_Color;

```



```

MoveAbsJ Pos_Int10,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Limpieza_Atom,0;
Reset Paro_Resto;
SetAO Señal_Luminosa,1;
ENDTEST
MoveAbsJ jointPunt_Paro,v1500,fine,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
N_Proc:=N_PROC2;
Set Pos_Parada;
WaitDI OK_PosP,1;
RestoPath;
StartMove;
WaitTime 0.25;
Reset Stop_Conveyor;
Reset Pos_Parada;
SetAO Señal_Luminosa,3;
ENDTRAP

TRAP Parada
VAR robtarget Punt_Paro;
VAR jointtarget jointPunt_Paro;
SetAO Señal_Luminosa,1;
StopMove\Quick;
StorePath;
Punt_Paro:=CRobT(\Tool:=RB1000_WSC_70_700_200\WObj:=WO_In_Int);
jointPunt_Paro:=CJointT(\TaskRef:=T_ROB1Id);
TEST N_Proc
CASE "main":
  MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  SetAO Señal_Luminosa,2;
  WaitDI Paro_Propio,0;
  SetAO Señal_Luminosa,1;
CASE "Principal1":
  MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  SetAO Señal_Luminosa,2;
  WaitDI Paro_Propio,0;
  SetAO Señal_Luminosa,1;
CASE "Principal2":
  MoveAbsJ Pos_Sal_Cap3,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  SetAO Señal_Luminosa,2;
  WaitDI Paro_Propio,0;
  SetAO Señal_Luminosa,1;
CASE "Principal2_2":
  MoveL Pos_Sal_PD2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
  MoveL Pos_Sal_PD2_2,v2500,z50,tool0\WObj:=wobj_cnv1;
  MoveAbsJ Pos_Sal_PD2_3,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  SetAO Señal_Luminosa,2;
  WaitDI Paro_Propio,0;
  SetAO Señal_Luminosa,1;
  MoveL Pos_Sal_PD2_2,v2500,z50,tool0\WObj:=wobj_cnv1;
  MoveL Pos_Sal_PD2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
CASE "Principal2_3":
  MoveL Pos_Sal_PT2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
  MoveL Pos_Sal_PT2_2,v2500,z50,tool0\WObj:=wobj_cnv1;
  MoveAbsJ Pos_Sal_PD2_3,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  SetAO Señal_Luminosa,2;
  WaitDI Paro_Propio,0;
  SetAO Señal_Luminosa,1;
  MoveL Pos_Sal_PT2_2,v2500,z50,tool0\WObj:=wobj_cnv1;
  MoveL Pos_Sal_PT2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
CASE "Principal3":
  MoveAbsJ Pos_Sal_PD2_3,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  SetAO Señal_Luminosa,2;
  WaitDI Paro_Propio,0;
  SetAO Señal_Luminosa,1;
CASE "Limpieza_Atom_Color":
  MoveL Limpieza_1,v1500,fine,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  MoveAbsJ Pos_Int10,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  SetAO Señal_Luminosa,2;

```

```

WaitDI Paro_Propio,0;
SetAO Señal_Luminosa,1;
MoveAbsJ Pos_Int10,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveL Limpieza_1,v1500,fine,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
CASE "Pint_Bajo_Capo":
MoveL Pos_Sal_Cap1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Pos_Sal_Cap2,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveAbsJ Pos_Sal_Cap3,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Paro_Propio,0;
SetAO Señal_Luminosa,1;
MoveL Pos_Sal_Cap2,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Pos_Sal_Cap1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
CASE "Pint_Pl_Del":
MoveL Pos_Sal_PD1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Pos_Int1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Pos_Sal_PD2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Pos_Sal_PD2_2,v2500,z50,tool0\WObj:=wobj_cnv1;
MoveAbsJ Pos_Sal_PD2_3,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Paro_Propio,0;
SetAO Señal_Luminosa,1;
MoveL Pos_Sal_PD2_2,v2500,z50,tool0\WObj:=wobj_cnv1;
MoveL Pos_Sal_PD2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Pos_Int1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Pos_Sal_PD1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
CASE "Pint_Pl_Tras":
MoveL Pos_Sal_PT1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Pos_Int2,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Pos_Sal_PT2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Pos_Sal_PT2_2,v2500,z50,tool0\WObj:=wobj_cnv1;
MoveAbsJ Pos_Sal_PD2_3,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Paro_Propio,0;
SetAO Señal_Luminosa,1;
MoveL Pos_Sal_PT2_2,v2500,z50,tool0\WObj:=wobj_cnv1;
MoveL Pos_Sal_PT2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Pos_Int2,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Pos_Sal_PT1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
CASE "Pint_Lateral":
MoveAbsJ Pos_Sal_Cap3,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Paro_Propio,0;
SetAO Señal_Luminosa,1;
CASE "Pint_Malet_Inf":
MoveL Pos_Sal_Malet1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Pos_Sal_Malet2,v2500,z50,tool0\WObj:=wobj_cnv1;
MoveAbsJ Pos_Sal_PD2_3,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Paro_Propio,0;
SetAO Señal_Luminosa,1;
MoveL Pos_Sal_Malet2,v2500,z50,tool0\WObj:=wobj_cnv1;
MoveL Pos_Sal_Malet1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
CASE "Pint_Bajos_1":
MoveL Pos_Sal_Baj1_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Offs(Pos_Sal_Baj1_1,0,400,250),v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveAbsJ Pos_Sal_Cap3,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Paro_Propio,0;
SetAO Señal_Luminosa,1;
MoveL Offs(Pos_Sal_Baj1_1,0,400,250),v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Pos_Sal_Baj1_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
CASE "Pint_Bajos_2":
MoveL Pos_Sal_Baj2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveAbsJ Pos_Sal_PD2_3,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Paro_Propio,0;
SetAO Señal_Luminosa,1;
MoveL Pos_Sal_Baj2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
DEFAULT:
SetAO Señal_Luminosa,2;

```

```

WaitDI Paro_Propio,0;
SetAO Señal_Luminosa,1;
ENDTEST
MoveAbsJ jointPunt_Paro,v1500,fine,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Set Pos_Parada;
WaitDI OK_PosP,1;
RestoPath;
StartMove;
WaitTime 0.25;
Reset Pos_Parada;
SetAO Señal_Luminosa,3;
ENDTRAP

TRAP Mantenimiento_Atomizador
VAR robtarget Punt_Paro;
VAR jointtarget jointPunt_Paro;
VAR string N_PROC2;
StopMove\Quick;
Set Paro_Resto;
Set Stop_Conveyor;
SetAO Señal_Luminosa,1;
StorePath;
N_PROC2:=N_Proc;
Limpieza:="Limp1";
Punt_Paro:=CRobT(\Tool:=RB1000_WSC_70_700_200\WObj:=WO_In_Int);
jointPunt_Paro:=CJointT(\TaskRef:=T_ROB1Id);
TEST N_Proc
CASE "main":
MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Int10,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atom_Color;
MoveAbsJ Pos_Int10,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Mantenimiento_Atom;
Reset Paro_Resto;
MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
CASE "Principal1":
MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Int10,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atom_Color;
MoveAbsJ Pos_Int10,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Mantenimiento_Atom;
Reset Paro_Resto;
MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
CASE "Principal2":
MoveAbsJ Pos_Sal_Cap3,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atom_Color;
MoveAbsJ Pos_Int10,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Mantenimiento_Atom;
Reset Paro_Resto;
MoveAbsJ Pos_Mant_Int_2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Mant_Int_1,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Sal_Cap3,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
CASE "Principal2_2":
MoveL Pos_Sal_PD2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Pos_Sal_PD2_2,v2500,z50,tool0\WObj:=wobj_cnv1;
MoveAbsJ Pos_Sal_PD2_3,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Int8,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Int10,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atom_Color;
MoveAbsJ Pos_Int10,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Mantenimiento_Atom;
Reset Paro_Resto;
MoveAbsJ Pos_Sal_PD2_3,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveL Pos_Sal_PD2_2,v2500,z50,tool0\WObj:=wobj_cnv1;
MoveL Pos_Sal_PD2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
CASE "Principal2_3":
MoveL Pos_Sal_PT2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;

```

MoveL Pos_Sal_PT2_2,v2500,z50,tool0\WObj:=wobj_cnv1;
 MoveAbsJ Pos_Sal_PD2_3,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveAbsJ Pos_Int8,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveAbsJ Pos_Int10,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Limpieza_Atom_Color;
 MoveAbsJ Pos_Int10,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Mantenimiento_Atom;
 Reset Paro_Resto;
 MoveAbsJ Pos_Sal_PD2_3,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveL Pos_Sal_PT2_2,v2500,z50,tool0\WObj:=wobj_cnv1;
 MoveL Pos_Sal_PT2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
CASE "Principal3":
 MoveAbsJ Pos_Sal_PD2_3,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveAbsJ Pos_Int8,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveAbsJ Pos_Int10,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Limpieza_Atom_Color;
 MoveAbsJ Pos_Int10,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Mantenimiento_Atom;
 Reset Paro_Resto;
 MoveAbsJ Pos_Sal_PD2_3,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
CASE "Limpieza_Atom_Color":
 MoveL Limpieza_1,v1500,fine,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Limpieza_Atom_Color;
 MoveAbsJ Pos_Int10,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Mantenimiento_Atom;
 Reset Paro_Resto;
 MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveAbsJ Pos_Int10,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveL Limpieza_1,v1500,fine,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
CASE "Pint_Bajo_Capo":
 MoveL Pos_Sal_Cap1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
 MoveL Pos_Sal_Cap2,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
 MoveAbsJ Pos_Sal_Cap3,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Limpieza_Atom_Color;
 MoveAbsJ Pos_Int10,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Mantenimiento_Atom;
 Reset Paro_Resto;
 MoveAbsJ Pos_Mant_Int_2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveAbsJ Pos_Mant_Int_1,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveAbsJ Pos_Sal_Cap3,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveL Pos_Sal_Cap2,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
 MoveL Pos_Sal_Cap1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
CASE "Pint_PL_Del":
 MoveL Pos_Sal_PD1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
 MoveL Pos_Int1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
 MoveL Pos_Sal_PD2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
 MoveL Pos_Sal_PD2_2,v2500,z50,tool0\WObj:=wobj_cnv1;
 MoveAbsJ Pos_Sal_PD2_3,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveAbsJ Pos_Int8,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveAbsJ Pos_Int10,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Limpieza_Atom_Color;
 MoveAbsJ Pos_Int10,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Mantenimiento_Atom;
 Reset Paro_Resto;
 MoveAbsJ Pos_Sal_PD2_3,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveL Pos_Sal_PD2_2,v2500,z50,tool0\WObj:=wobj_cnv1;
 MoveL Pos_Sal_PD2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
 MoveL Pos_Int1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
 MoveL Pos_Sal_PD1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
CASE "Pint_PL_Tras":
 MoveL Pos_Sal_PT1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
 MoveL Pos_Int2,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
 MoveL Pos_Sal_PT2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
 MoveL Pos_Sal_PT2_2,v2500,z50,tool0\WObj:=wobj_cnv1;
 MoveAbsJ Pos_Sal_PD2_3,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveAbsJ Pos_Int8,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;

```

MoveAbsJ Pos_Int10,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atom_Color;
MoveAbsJ Pos_Int10,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Mantenimiento_Atom;
Reset Paro_Resto;
MoveAbsJ Pos_Sal_PD2_3,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveL Pos_Sal_PT2_2,v2500,z50,tool0\WObj:=wobj_cnv1;
MoveAbsJ Pos_Sal_PT2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Pos_Int2,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Pos_Sal_PT1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
CASE "Pint_Lateral":
MoveAbsJ Pos_Sal_Cap3,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atom_Color;
MoveAbsJ Pos_Int10,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Mantenimiento_Atom;
Reset Paro_Resto;
MoveAbsJ Pos_Mant_Int_2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Mant_Int_1,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Sal_Cap3,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
CASE "Pint_Malet_Inf":
MoveL Pos_Sal_Malet1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Pos_Sal_Malet2,v2500,z50,tool0\WObj:=wobj_cnv1;
MoveAbsJ Pos_Sal_PD2_3,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Int8,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Int10,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atom_Color;
MoveAbsJ Pos_Int10,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Mantenimiento_Atom;
Reset Paro_Resto;
MoveAbsJ Pos_Sal_PD2_3,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveL Pos_Sal_Malet2,v2500,z50,tool0\WObj:=wobj_cnv1;
MoveL Pos_Sal_Malet1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
CASE "Pint_Bajos_1":
MoveL Pos_Sal_Baj1_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Offs(Pos_Sal_Baj1_1,0,400,250),v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveAbsJ Pos_Sal_Cap3,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atom_Color;
MoveAbsJ Pos_Int10,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Mantenimiento_Atom;
Reset Paro_Resto;
MoveAbsJ Pos_Mant_Int_2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Mant_Int_1,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Sal_Cap3,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveL Offs(Pos_Sal_Baj1_1,0,400,250),v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Pos_Sal_Baj1_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
CASE "Pint_Bajos_2":
MoveL Pos_Sal_Baj2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveAbsJ Pos_Sal_PD2_3,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Int8,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Int10,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atom_Color;
MoveAbsJ Pos_Int10,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Mantenimiento_Atom;
Reset Paro_Resto;
MoveAbsJ Pos_Sal_PD2_3,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveL Pos_Sal_Baj2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
DEFAULT:
MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Int10,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atom_Color;
MoveAbsJ Pos_Int10,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Mantenimiento_Atom;
Reset Paro_Resto;
MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
ENDTEST

```

```
MoveAbsJ jointPunt_Paro,v1500,fine,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
N_Proc:=N_PROC2;
Set Pos_Parada;
WaitDI OK_PosP,1;
RestoPath;
StartMove;
WaitTime 0.25;
Reset Stop_Conveyor;
Reset Pos_Parada;
SetAO Señal_Luminosa,3;
ENDTRAP
ENDMODULE
```

8.1.2 Controlador virtual Cont Inf Der

MODULE Module1

!!

! Módulo: Module1

!

! Descripción: Procesos para la aplicación de la capa base sobre la carrocería empleando el robot inferior derecho

!

! Autor: Arturo López López

!!

!!!!!!!!!!!!!!!!!!!!!!!!!!!! DEFINICIÓN DE OBJETIVOS !!!!!!!!!!!!!!!!!!!!!

```
LOCAL CONST robtarget Paint_10:=[[3670,-44.188,1125],[0.683515034,0.015082674,-0.728808674,-
0.037651394],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_20:=[[3666.4,-67.79,1329.92],[0.683516783,0.015085068,-0.728807297,-
0.037645342],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_30:=[[3767.82,-63.85,1389.62],[0.683516783,0.015085068,-0.728807297,-
0.037645342],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_40:=[[3716.475995084,-67.37626514,1723.959381776],[0.683516783,0.015085068,-
0.728807297,-0.037645342],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_50:=[[3795.09,-64.85,1768.11],[0.683516783,0.015085068,-0.728807297,-
0.037645342],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_60:=[[3639.255995084,-71.94626514,1848.849381776],[0.683516783,0.015085068,-
0.728807297,-0.037645342],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_70:=[[3628.405995084,-228.97626514,1849.959381776],[0.683516783,0.015085068,-
0.728807297,-0.037645342],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_80:=[[3804.21,-216.89,1762.93],[0.683516783,0.015085068,-0.728807297,-
0.037645342],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_90:=[[3709.155995084,-217.64626514,1711.059381776],[0.683516783,0.015085068,-
0.728807297,-0.037645342],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_100:=[[3701.95,-219.16,1436.52],[0.683516783,0.015085068,-0.728807297,-
0.037645342],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_110:=[[3767.04,-219.42,1394.36],[0.683516783,0.015085068,-0.728807297,-
0.037645342],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_120:=[[3662.61,-216.04,1320.24],[0.683516783,0.015085068,-0.728807297,-
0.037645342],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_130:=[[3674.830176468,-204.083365245,1220.958128206],[0.732266438,-0.095820126,-
0.67393604,0.020361294],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_140:=[[3686.009477724,-325.141582597,1231.515165891],[0.744917995,0.113123404,-
0.651425172,0.089137659],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_150:=[[3765.42,-385.5,1403.73],[0.725579213,0.203062866,-
0.6357064,0.167862],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_160:=[[3679.209809228,-390.011334201,1717.385217575],[0.742830264,0.126106786,-
0.64977029,0.10049302],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_170:=[[3797.5,-383.22,1764.46],[0.67476541,0.110065412,-
0.726953812,0.064151392],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_180:=[[3624.658624629,-384.854241752,1865.671536427],[0.577148634,0.24743938,-
0.745713831,-0.222674853],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_190:=[[3656.240511924,-624.09060764,1879.288260446],[0.564046231,0.243515482,-
0.755672887,-0.226959351],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_200:=[[3712.636234542,-769.22911296,1838.709340674],[0.333820947,0.578421732,-
0.650918584,-0.36096658],[0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];
```

LOCAL CONST robtarget Paint_210:=[[4147.13,-715.35,1699.37],[0.071068035,0.66440665,-0.741331032,-0.062780876],[0,0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_220:=[[4204.32,-717.18,1664.62],[0.071158317,0.664418509,-0.741322341,-0.062655611],[0,0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_230:=[[4069.97,-650.6,1638.02],[0.317124094,0.590967317,-0.737919008,0.075269361],[0,0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_240:=[[4082.22,-576.19,1573.52],[0.330995036,0.585789521,-0.725490699,0.144762457],[0,0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_250:=[[4066.3,-515.4,1484.66],[0.32105575,0.591320038,-0.731219367,0.112169762],[0,0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_260:=[[4038.35,-473.17,1422.27],[0.09752227,0.719676656,-0.68219171,-0.08467224],[0,0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_270:=[[4554.13,-465.06,1422.27],[0,-0.627211767,0.778848765,0],[0,0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_280:=[[4535.88,-436.84,1362.88],[0.383022222,0.821393805,-0.383022222,0.178606194],[0,0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_290:=[[3831.26,-436.84,1365.29],[0.54191397,0.498095685,-0.662044028,0.141165299],[0,0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_300:=[[3822.16,-490.46,1538.98],[0.546152606,0.431903145,-0.700987106,0.154253305],[0,0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_310:=[[3834.12,-485.98,1694.61],[0.591010486,0.43872163,-0.673173073,0.071189535],[0,0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_320:=[[3918.34,-508.94,1671.34],[0.515102982,0.525774648,-0.653748699,0.175620544],[0,0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_330:=[[3955.04,-573.06,1678.19],[0.595588754,0.245927981,-0.747203758,0.162726793],[0,0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_340:=[[3922.25,-578,1759.58],[0.595588754,0.245927981,-0.747203758,0.162726793],[0,0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_350:=[[3834.44,-519.78,1761.47],[0.322317073,0.679420096,-0.650457536,-0.106794342],[0,0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_360:=[[3844.37,-878,1755.85],[0.32232008,0.679417376,-0.650458679,-0.106795604],[0,0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_370:=[[3862.234,-1150,1749.525],[0.32232008,0.679417376,-0.650458679,-0.106795604],[0,0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];

CONST jointarget Pos_Inicial2:=[[0,19.999999418,-19.999999422,0],[6750,9E+09,9E+09,9E+09,9E+09,0]];

CONST robtarget Pos_Int1:=[[2335.776,-1815.077,1392.253],[0.143891977,-0.228053839,0.962694568,-0.022488102],[-2,0,-1,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

CONST robtarget Pos_Int2:=[[1390.776,-1815.077,1392.253],[0.143891977,-0.228053839,0.962694568,-0.022488102],[-2,0,-1,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

CONST jointarget Pos_Int3:=[[-180,-20,20,0,-45,0],[4061,9E+09,9E+09,9E+09,9E+09,0]];

CONST robtarget Pos_Int5:=[[300,-1250,1590.32],[0.560985526,-0.560985526,0.430459335,-0.430459335],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];

CONST robtarget Aprox_Bajo_Capo:=[[4400,-44.188,1125],[0.683515034,0.015082674,-0.728808674,-0.037651394],[0,0,0,0],[3750,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget PD_Paint_410:=[[2376.844785733,-764.733770226,1695.386831978],[0.438730793,-0.245963269,-0.777778473,-0.376932367],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget PD_Paint_420:=[[2312.67,-691.56,1968.28],[0.432030574,-0.298901899,-0.759522255,-0.383579433],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget PD_Paint_430:=[[2287.07,-641.5,2087.13],[0.397401149,-0.331547218,-0.766101023,-0.381101025],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget PD_Paint_440:=[[2272.42,-604.53,2151.99],[0.386767188,-0.31445532,-0.735768585,-0.458446926],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget PD_Paint_450:=[[2281.82,-587.9,2176.45],[0.410723784,-0.257647336,-0.696505169,-0.528965379],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget PD_Paint_460:=[[2304.5,-573.68,2190.46],[0.412775619,-0.170089774,-0.630027655,-0.635413968],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget PD_Paint_470:=[[2336.23,-567.14,2192.78],[0.638517464,-0.266583223,-0.47153446,-0.546702924],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget PD_Paint_480:=[[2407.04,-566.68,2197.39],[0.637652054,-0.412211971,-0.366091928,-0.53801287],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget PD_Paint_490:=[[2463.04,-572.32,2196.55],[0.63513548,-0.417737099,-0.288175931,-0.58228281],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget PD_Paint_500:=[[2595.18,-581.84,2178.51],[0.598059876,-0.483365502,-0.331658894,-0.546520406],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget PD_Paint_510:=[[2731.77,-602.35,2153.38],[0.576667863,-0.518141883,-0.294111048,-0.559000765],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget PD_Paint_520:=[[2833.1,-619.19,2116.2],[0.322458593,-0.438974945,-0.448936304,-0.708362653],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget PD_Paint_530:=[[2932.22,-642.86,2077.02],[0.256025856,-0.553017094,-0.395689027,-0.687061169],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget PD_Paint_540=[[3012.13,-659.95,2038.07],[0.271183932,-0.524669574,-0.360918946,-0.721746928],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_550=[[3087.81,-679.95,2010.34],[0.270015207,-0.498349667,-0.291928851,-0.770400509],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_560=[[3155.97,-690.01,1970.86],[0.293350652,-0.529569017,-0.270018443,-0.748726981],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_570=[[3213.63,-704.91,1941.12],[0.239667875,-0.537987626,-0.236411406,-0.772811925],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_580=[[3260.17,-716.25,1905.39],[0.209325259,0.333546242,0.48081389,0.783420604],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_590=[[3305.23,-724.18,1863.21],[0.420505352,0.305487723,0.666518693,0.534420558],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_600=[[3322.19,-721.62,1815.28],[0.616858223,-0.167525985,0.731626369,0.2369469],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_610=[[3330.05,-717.35,1745.04],[0.652518914,-0.137440968,0.73244113,0.137328212],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_620=[[3377.6,-745.75,1718.38],[0.652518914,-0.137440968,0.73244113,0.137328212],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_630=[[3396.65,-754.15,1643.3],[0.652518914,-0.137440968,0.73244113,0.137328212],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_640=[[3394.31,-742.72,1552.78],[0.652518914,-0.137440968,0.73244113,0.137328212],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_650=[[3387.82,-736.58,1443.32],[0.652518914,-0.137440968,0.73244113,0.137328212],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_660=[[3374.99,-737.58,1338.62],[0.652518914,-0.137440968,0.73244113,0.137328212],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_670=[[3351.94,-732.23,1279.11],[0.444927124,-0.231849192,0.864266934,0.036448235],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_680=[[3308.57,-740.63,1231.18],[0.349584618,-0.22082684,0.907782792,0.070402448],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_690=[[3228.25,-735.35,1203.88],[0.355191364,-0.128678803,0.912550478,0.156628497],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_700=[[3102.27,-731.2,1201.02],[0.36168534,-0.168361222,0.906069041,0.140986192],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_710=[[2895.11,-731.82,1207.34],[0.316612848,-0.240567762,0.89810252,0.187870488],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_720=[[2771.43,-727.35,1214.32],[0.202619176,-0.184721358,0.933157891,0.232464707],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_730=[[2696.01,-727.64,1228.88],[0.145662622,-0.15593862,0.945912442,0.244366936],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_740=[[2638.29,-732.33,1246.29],[0.133580783,-0.157196716,0.949509723,0.23638243],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_750=[[2566.98,-733.13,1278.61],[0.140352237,-0.230419675,0.92051683,0.282589436],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_760=[[2482.94,-731.02,1342.75],[0.168061556,0.137117549,-0.955501053,-0.199929561],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_770=[[2420.89,-714.7,1418.55],[0.226769787,0.138275422,-0.926052553,-0.268108262],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_780=[[2383.26,-720.19,1510.7],[0.282930228,0.000277414,-0.919666298,-0.27233125],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_790=[[2372.88,-731.15,1577.19],[0.301074077,0.039762137,-0.902412483,-0.305655169],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_800=[[2353.85,-734.31,1795.6],[0.243119091,0.002008995,-0.9173333,-0.315259714],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_810=[[2303.248479984,-711.72228907,1984.419803286],[0.322718167,0.038930407,-0.909704645,-0.258408333],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_820=[[2288.079349258,-498.236478167,2343.126025944],[0.357631061,-0.373943438,-0.671108714,-0.53092318],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_830=[[2235.19,-608.61,2199.66],[0.280379965,-0.383288266,-0.72132341,-0.504152476],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_840=[[2237.69,-635.07,2154.24],[0.2886697,-0.375195931,-0.742715896,-0.473572503],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_850=[[2256.91,-690.12,2027.68],[0.327765863,-0.334050685,-0.685318689,-0.557958756],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_860=[[2282.18,-723.31,1927.08],[0.310587811,-0.344887521,-0.677547278,-0.570541406],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_870=[[2311.74,-763.66,1778.33],[0.302963492,-0.342283386,-0.644446131,-0.612979927],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_880=[[2330.96,-767,1633.07],[0.402586704,-0.442493722,-0.549373562,-0.583362615],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget PD_Paint_890:=[[2355.31,-775.05,1460.57],[0.373643123,-0.468786894,-0.555750878,-0.575995336],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget PD_Paint_900:=[[2382.67,-784.39,1382.12],[0.040423178,-0.134681434,-0.749793268,-0.646557757],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget PD_Paint_910:=[[2410.65,-794.68,1299.71],[0.007386624,-0.161700054,-0.721148798,-0.67360444],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget PD_Paint_920:=[[2527.54,-815.96,1207.32],[0.019186782,0.185836716,0.723676974,0.664370544],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget PD_Paint_930:=[[2701.35,-816.61,1185.09],[0.07262804,0.223337992,0.749193643,0.619317524],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget PD_Paint_940:=[[2987.85,-814.99,1170.52],[0.14571116,0.279372701,0.744213748,0.588952502],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget PD_Paint_950:=[[3301.881,-812.04,1167.086],[0.437388433,-0.072135606,0.885025829,0.14218683],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget PD_Paint_960:=[[3367.523,-965.018,1197.046],[0.425565978,-0.547234322,0.593886076,-0.408322818],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget PD_Paint_970:=[[3297.3,-1095.74,1207.99],[0.537369856,-0.415864378,0.480480605,-0.554462663],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget PD_Paint_980:=[[3214.5,-1234.07,1198.47],[0.476728358,-0.39947443,0.461262672,-0.632761408],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget PD_Paint_990:=[[3170.47,-1308.29,1195.17],[0.476728358,-0.39947443,0.461262672,-0.632761408],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget PD_Paint_1000:=[[3118.49,-1401.16,1199.68],[0.476728358,-0.39947443,0.461262672,-0.632761408],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget PD_Paint_1010:=[[3079.25,-1467.57,1198.17],[0.476728358,-0.39947443,0.461262672,-0.632761408],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget PD_Paint_1020:=[[2971.98,-1653.62,1204.22],[0.476728358,-0.39947443,0.461262672,-0.632761408],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget PD_Paint_1030:=[[2881.65,-1810.48,1214.32],[0.476728358,-0.39947443,0.461262672,-0.632761408],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget PD_Paint_1040:=[[2906.17,-1820.93,1332.86],[0.365050148,-0.478002035,0.608135257,-0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget PD_Paint_1050:=[[2910.59,-1786.45,1331.55],[0.365050148,-0.478002035,0.608135257,-0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget PD_Paint_1060:=[[2921.4,-1742.89,1327.68],[0.365050148,-0.478002035,0.608135257,-0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget PD_Paint_1070:=[[2944.59,-1705.9,1323.62],[0.365050148,-0.478002035,0.608135257,-0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget PD_Paint_1080:=[[2930.67,-1664.44,1321.03],[0.365050148,-0.478002035,0.608135257,-0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget PD_Paint_1090:=[[2916.09,-1582.45,1318.44],[0.365050148,-0.478002035,0.608135257,-0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget PD_Paint_1100:=[[2978.66,-1474.13,1306.52],[0.365050148,-0.478002035,0.608135257,-0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget PD_Paint_1110:=[[3131.53,-1210.56,1299.3],[0.365050148,-0.478002035,0.608135257,-0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget PD_Paint_1120:=[[3245.34,-1014.57,1306.7],[0.365050148,-0.478002035,0.608135257,-0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget PD_Paint_1130:=[[3324.29,-970.04,1306.55],[0.365050148,-0.478002035,0.608135257,-0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget PD_Paint_1140:=[[3400.46,-936.14,1299.81],[0.391229189,-0.500146133,0.591633183,-0.496753202],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget PD_Paint_1150:=[[3401.32,-932.44,1445.61],[0.386051384,-0.495792156,0.595024731,-0.501098829],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget PD_Paint_1160:=[[3319.37,-949.27,1437.42],[0.365050148,-0.478002035,0.608135257,-0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget PD_Paint_1170:=[[3255.35,-981.24,1435.02],[0.365050148,-0.478002035,0.608135257,-0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget PD_Paint_1180:=[[3200.79,-1075.98,1439.54],[0.365050148,-0.478002035,0.608135257,-0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget PD_Paint_1190:=[[3129.23,-1195.85,1445.25],[0.365050148,-0.478002035,0.608135257,-0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget PD_Paint_1200:=[[2876.75,-1629.39,1467.22],[0.365050148,-0.478002035,0.608135257,-0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget PD_Paint_1210:=[[2856.45,-1651.51,1463.14],[0.365050148,-0.478002035,0.608135257,-0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget PD_Paint_1220:=[[2933.49,-1741.08,1467.49],[0.365050148,-0.478002035,0.608135257,-0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget PD_Paint_1230:=[[2908.85,-1811.46,1468.94],[0.365050148,-0.478002035,0.608135257,-0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget PD_Paint_1240:=[[2904.4,-1837.92,1638.31],[0.365050148,-0.478002035,0.608135257,-0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1250:=[[2875.66,-1643,1625.84],[0.365050148,-0.478002035,0.608135257,-0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1260:=[[2962.59,-1492.85,1615.86],[0.365050148,-0.478002035,0.608135257,-0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1270:=[[3111.04,-1237.2,1605.02],[0.365050148,-0.478002035,0.608135257,-0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1280:=[[3262.46,-979.01,1586.35],[0.365050148,-0.478002035,0.608135257,-0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1290:=[[3336.31,-937.7,1591.36],[0.365050148,-0.478002035,0.608135257,-0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1300:=[[3410.78,-924.7,1597.58],[0.365050148,-0.478002035,0.608135257,-0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1310:=[[3374.11,-911.29,1743.45],[0.365050148,-0.478002035,0.608135257,-0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1320:=[[3320.44,-952.24,1743.21],[0.365050148,-0.478002035,0.608135257,-0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1330:=[[3230.1,-1033.35,1744.11],[0.365050148,-0.478002035,0.608135257,-0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1340:=[[3118.01,-1220.54,1749.96],[0.365050148,-0.478002035,0.608135257,-0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1350:=[[2991.38,-1437.59,1757.48],[0.365050148,-0.478002035,0.608135257,-0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1360:=[[2873.83,-1638.56,1768.11],[0.365050148,-0.478002035,0.608135257,-0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1370:=[[2812.22,-1726.57,1767.85],[0.365050148,-0.478002035,0.608135257,-0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1380:=[[2844.86,-1792.44,1776.8],[0.365050148,-0.478002035,0.608135257,-0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1390:=[[2874.83,-1845.41,1783.45],[0.365050148,-0.478002035,0.608135257,-0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1400:=[[2787.95,-1806.28,1874.62],[0.365050148,-0.478002035,0.608135257,-0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1410:=[[2892.17,-1589.53,1844.89],[0.365050148,-0.478002035,0.608135257,-0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1420:=[[3049.78,-1323.62,1819.36],[0.365050148,-0.478002035,0.608135257,-0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1430:=[[3168.23,-1129.43,1813.01],[0.365050148,-0.478002035,0.608135257,-0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1440:=[[3289.29,-960.08,1811.49],[0.365050148,-0.478002035,0.608135257,-0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1450:=[[3319.94,-912.45,1833.66],[0.391229189,-0.500146134,0.591633183,-0.496753201],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1460:=[[3248.13,-1005.04,1865.47],[0.365050148,-0.478002035,0.608135257,-0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1470:=[[3203.26,-1045.31,1920.23],[0.365050148,-0.478002035,0.608135257,-0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1480:=[[3164.62,-1068.04,1982.87],[0.365050148,-0.478002035,0.608135257,-0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1490:=[[3094.04,-1148.06,2035.47],[0.365050148,-0.478002035,0.608135257,-0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1500:=[[2957.02,-1299.12,2126.97],[0.365050148,-0.478002035,0.608135257,-0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1510:=[[2849.68,-1426.6,2178.43],[0.365050148,-0.478002035,0.608135257,-0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1520:=[[2765.55,-1543.19,2200.63],[0.365050148,-0.478002035,0.608135257,-0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1530:=[[2678.52,-1673.19,2214.09],[0.365050148,-0.478002035,0.608135257,-0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1540:=[[2614.94,-1789.27,2217.83],[0.365050148,-0.478002035,0.608135257,-0.51809647],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1550:=[[2684.85,-1792.19,2111.69],[0.632754631,0.025937671,0.717872832,-0.289149462],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1560:=[[2843.224551317,-1805.820051646,1770.373560375],[0.585122193,0.028421261,0.767434682,-0.260515373],[0,0,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1570:=[[1619.842639859,-755.539719945,1482.736824235],[0.153522091,-0.213265029,-0.906771817,-0.32971786],[0,0,0,0],[425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1580:=[[1467.53,-740.97,1746.37],[0.237108005,-0.236712924,-0.871535672,-0.358011672],[0,0,0,0],[425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget PD_Paint_1590:=[[1391.02,-731.54,1833.22],[0.353380695,-0.26646993,-0.78791197,-0.428147857],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1600:=[[1334.77,-708.3,1893.12],[0.312706039,-0.277578082,-0.79579159,-0.438042335],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1610:=[[1312.05,-681.42,1948.58],[0.365850854,-0.266629806,-0.766022398,-0.456367599],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1620:=[[1369.37,-638.53,2035.07],[0.438629496,-0.155740044,-0.652702074,-0.597770196],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1630:=[[1437.71,-612.85,2082.92],[0.499311366,-0.219997819,-0.546283767,-0.635502294],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1640:=[[1547.3,-587.39,2133.11],[0.465987207,-0.258368317,-0.468761174,-0.704531545],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1650:=[[1661.5,-570.98,2161.72],[0.459154169,-0.282960903,-0.445032549,-0.714882232],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1660:=[[1935.15,-560.76,2200.3],[0.006723192,0.131791865,0.504700472,0.853148953],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1670:=[[2086.9,-560.22,2201.72],[0.133696082,-0.358118123,-0.320754631,-0.866598658],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1680:=[[2110.79,-564.17,2195.56],[0.648755363,-0.079261,0.584035751,0.481390085],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1690:=[[2137.87,-577.01,2176.45],[0.803818591,-0.252327548,0.511569825,0.168827712],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1700:=[[2148.07,-591.01,2165.6],[0.769196984,-0.262151373,0.558149969,0.16757467],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1710:=[[2155.09,-605.5,2139.13],[0.783183623,-0.230227337,0.53845328,0.209013995],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1720:=[[2167.93,-641.46,2033.23],[0.757621033,-0.250236707,0.571316641,0.192325913],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1730:=[[2185.66,-686.26,1912.42],[0.715748601,-0.230617366,0.633889869,0.180840273],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1740:=[[2203.05,-712.38,1800.75],[0.689251505,-0.247688937,0.655580312,0.183839625],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1750:=[[2238.17,-728.4,1593.87],[0.719206578,-0.22155487,0.632923769,0.181831903],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1760:=[[2262.82,-733.35,1423.08],[0.72136115,-0.195741383,0.646664126,0.152147662],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1770:=[[2263.41,-741.7,1367.66],[0.595545775,-0.197129199,0.770398869,0.113801984],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1780:=[[2236.73,-740.53,1304.07],[0.489282214,-0.193984033,0.838960448,0.138269582],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1790:=[[2174.06,-742.15,1261.31],[0.475148309,-0.159257787,0.824933906,0.261448069],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1800:=[[2109.81,-748.37,1249.15],[0.309158917,-0.146373712,0.907325264,0.244451151],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1810:=[[2063.6,-754.04,1245.73],[0.293737544,-0.107897225,0.900838055,0.30094392],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1820:=[[1946.86,-755.18,1247.07],[0.201744651,-0.140036322,0.929051003,0.276682414],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1830:=[[1869.98,-748.43,1276],[0.157996106,-0.111845642,0.914783948,0.354539294],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1840:=[[1795.69,-749.47,1297.77],[0.029255099,-0.09888371,0.895140628,0.433692757],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1850:=[[1743.74,-748.91,1345.09],[0.001082167,-0.074845784,0.940318123,0.331961993],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1860:=[[1685.03,-754.17,1402.26],[0.067202984,0.126876088,-0.956234196,-0.254955643],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1870:=[[1620.52,-760.21,1484.22],[0.120504776,0.049482158,-0.928979691,-0.346448913],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1880:=[[1471.919173256,-735.158285713,1737.728696076],[0.120504776,0.049482158,-0.928979691,-0.346448913],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1890:=[[1239.900079169,-623.409553496,2070.421470566],[0.261770844,-0.231225282,-0.79272959,-0.499590524],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1900:=[[1266.88,-776.43,1862.47],[0.190633218,-0.083121418,-0.895000014,-0.394619793],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1910:=[[1311.37,-789.82,1776.57],[0.147681825,-0.201022586,-0.874122379,-0.416761401],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1920:=[[1347.58,-801.53,1721.92],[0.147681825,-0.201022586,-0.874122379,-0.416761401],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1930:=[[1392.44,-806.77,1671.84],[0.147681825,-0.201022586,-0.874122379,-0.416761401],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget PD_Paint_1940:=[[1489.24,-809.37,1567.2],[0.147681825,-0.201022586,-0.874122379,-0.416761401],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1950:=[[1549.39,-807.88,1497.65],[0.147681825,-0.201022586,-0.874122379,-0.416761401],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1960:=[[1613.66,-806.77,1398.67],[0.147681825,-0.201022586,-0.874122379,-0.416761401],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1970:=[[1640.76,-808.81,1332.13],[0.147681825,-0.201022586,-0.874122379,-0.416761401],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1980:=[[1677.58,-809.97,1238.11],[0.050229921,0.18769274,0.662563316,0.723365912],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_1990:=[[1681.96,-809.95,1209.37],[0.050229921,0.18769274,0.662563316,0.723365912],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2000:=[[1716.26,-809.95,1246.73],[0.050229921,0.18769274,0.662563316,0.723365912],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2010:=[[1765.43,-809.73,1218.47],[0.050229921,0.18769274,0.662563316,0.723365912],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2020:=[[1857.25,-809.64,1205.24],[0.050229921,0.18769274,0.662563316,0.723365912],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2030:=[[1991.33,-811.44,1196.45],[0.050229921,0.18769274,0.662563316,0.723365912],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2040:=[[2143.966,-812.429,1201.129],[0.496905618,-0.12056434,0.80073215,0.312052992],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2060:=[[2241.06,-987.72,1220.57],[0.518907904,-0.525951332,0.507199004,-0.4436879],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2070:=[[2184.15,-1088.1,1221.64],[0.539381073,-0.411645744,0.460743005,-0.572129114],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2080:=[[2111.47,-1217.61,1221.41],[0.556823437,-0.387726476,0.435348558,-0.591681901],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2090:=[[1970.58,-1473.37,1227.49],[0.556823437,-0.387726476,0.435348558,-0.591681901],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2100:=[[1978.89,-1491.96,1355.34],[0.437499143,-0.518631527,0.57365296,-0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2110:=[[1996.67,-1443.1,1359.97],[0.437499143,-0.518631527,0.57365296,-0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2120:=[[1934.84,-1353.23,1382.99],[0.437499143,-0.518631527,0.57365296,-0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2130:=[[1982.88,-1272.87,1367.78],[0.437499143,-0.518631527,0.57365296,-0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2140:=[[2025.19,-1194.73,1369.63],[0.437499143,-0.518631527,0.57365296,-0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2150:=[[2083.08,-1090.1,1371.55],[0.437499143,-0.518631527,0.57365296,-0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2160:=[[2143.37,-993.81,1381.23],[0.508633013,-0.574085766,0.511640183,-0.387223857],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2170:=[[2182.6,-934.79,1380.32],[0.530466338,-0.590429831,0.488966954,-0.361814036],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2180:=[[2254.67,-900.47,1368.6],[0.508633013,-0.574085766,0.511640183,-0.387223857],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2190:=[[2316.5,-896.55,1356.21],[0.457252829,-0.534329027,0.558035007,-0.440464837],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2200:=[[2314.9,-893.62,1512.94],[0.437499143,-0.518631527,0.57365296,-0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2210:=[[2237.04,-917.93,1519.23],[0.437499143,-0.518631527,0.57365296,-0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2220:=[[2179,-944.42,1523.83],[0.437499143,-0.518631527,0.57365296,-0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2230:=[[2123.17,-1031.35,1515.91],[0.437499143,-0.518631527,0.57365296,-0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2240:=[[2030.11,-1181.14,1515.45],[0.437499143,-0.518631527,0.57365296,-0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2250:=[[1928.82,-1362.77,1518.7],[0.437499143,-0.518631527,0.57365296,-0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2260:=[[1898.62,-1405.21,1518.97],[0.437499143,-0.518631527,0.57365296,-0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2270:=[[1961.23,-1509.04,1507.13],[0.437499143,-0.518631527,0.57365296,-0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2280:=[[1929.9,-1586.78,1505.33],[0.437499143,-0.518631527,0.57365296,-0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2290:=[[1857.79,-1693.74,1634.09],[0.437499143,-0.518631527,0.57365296,-0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget PD_Paint_2300:=[[1902.81,-1619.81,1632.43],[0.437499143,-0.518631527,0.57365296,-0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2310:=[[1842.56,-1482.22,1643.48],[0.437499143,-0.518631527,0.57365296,-0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2320:=[[1920.98,-1358.45,1644.91],[0.437499143,-0.518631527,0.57365296,-0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2330:=[[2050.08,-1142.08,1645.8],[0.437499143,-0.518631527,0.57365296,-0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2340:=[[2169.48,-963.7,1660.12],[0.437499143,-0.518631527,0.57365296,-0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2350:=[[2235.81,-928.37,1654.58],[0.437499143,-0.518631527,0.57365296,-0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2360:=[[2306.57,-920.12,1648.81],[0.41721243,-0.502302154,0.588572006,-0.476664796],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2370:=[[2250.21,-929.85,1811.36],[0.437499143,-0.518631527,0.57365296,-0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2380:=[[2190.44,-960.59,1813.31],[0.437499143,-0.518631527,0.57365296,-0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2390:=[[2088,-1053.02,1803.46],[0.437499143,-0.518631527,0.57365296,-0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2400:=[[2014.17,-1165.19,1792.16],[0.437499143,-0.518631527,0.57365296,-0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2410:=[[1906.79,-1334.52,1789.93],[0.437499143,-0.518631527,0.57365296,-0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2420:=[[1819.99,-1485.37,1790.23],[0.437499143,-0.518631527,0.57365296,-0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2430:=[[1782.67,-1536.8,1789.24],[0.437499143,-0.518631527,0.57365296,-0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2440:=[[1796.19,-1631.91,1799.7],[0.437499143,-0.518631527,0.57365296,-0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2450:=[[1774.37,-1766.64,1783.4],[0.437499143,-0.518631527,0.57365296,-0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2460:=[[1671.77,-1763.05,1910.47],[0.437499143,-0.518631527,0.57365296,-0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2470:=[[1719.32,-1587.39,1893.81],[0.437499143,-0.518631527,0.57365296,-0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2480:=[[1792.92,-1489.7,1888.6],[0.437499143,-0.518631527,0.57365296,-0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2490:=[[1873.27,-1353.14,1881.78],[0.437499143,-0.518631527,0.57365296,-0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2500:=[[1937.05,-1254.22,1880.69],[0.437499143,-0.518631527,0.57365296,-0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2510:=[[2022.03,-1124.66,1877.86],[0.437499143,-0.518631527,0.57365296,-0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2520:=[[2091.34,-1008.73,1874.51],[0.437499143,-0.518631527,0.57365296,-0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2530:=[[2192.24,-915.87,1881.82],[0.22860438,-0.698680537,0.331514535,-0.59134056],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2540:=[[2166.12,-921.23,1919.44],[0.22860438,-0.698680537,0.331514535,-0.59134056],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2550:=[[2118.88,-917.65,2028.53],[0.22860438,-0.698680537,0.331514535,-0.59134056],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2560:=[[2018.24,-889.98,2216.85],[0.22860438,-0.698680537,0.331514535,-0.59134056],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2570:=[[2002.46,-892.33,2234.49],[0.437499143,-0.518631527,0.57365296,-0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2580:=[[1898.58,-1069.18,2227.5],[0.437499143,-0.518631527,0.57365296,-0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2590:=[[1793.64,-1264.98,2202.96],[0.437499143,-0.518631527,0.57365296,-0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2600:=[[1730.49,-1401.28,2169.75],[0.437499143,-0.518631527,0.57365296,-0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2610:=[[1680.07,-1553,2099.27],[0.437499143,-0.518631527,0.57365296,-0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2620:=[[1654.59,-1672.99,2012.77],[0.437499143,-0.518631527,0.57365296,-0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2630:=[[1657.85,-1741.53,1937.3],[0.437499143,-0.518631527,0.57365296,-0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2640:=[[1666.86,-1751.23,1923.46],[0.437499143,-0.518631527,0.57365296,-0.458844332],[0,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget PD_Paint_2650:=[[1843.15,-1470.73,1925.72],[0.437499143,-0.518631527,0.57365296,-0.458844332],[0,0,0,0],[-425.9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2660:=[[1811.14,-1451.1,1937.94],[0.437499143,-0.518631527,0.57365296,-0.458844332],[0,0,0,0],[-425.9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget PD_Paint_2670:=[[1656.56470442,-1400.452204417,2284.216955119],[0.437499143,-0.518631527,0.57365296,-0.458844332],[0,0,0,0],[-425.9E+09,9E+09,9E+09,9E+09,0]];
 CONST robtarget Pos_Int4:=[[1656.565,-1400.452,2600],[0,-0.499999999,0.866025404,0],[0,0,0,0],[-425.9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_2680:=[[4231.829478651,-900.010594169,1404.051579205],[0.41663936,-0.574681469,0.588034335,0.38777374],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_2690:=[[4231.75,-820.32,1631.44],[0.442673914,-0.540103586,0.579783134,0.419737346],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_2700:=[[4237.58,-797.81,1674.49],[0.226214202,-0.601270087,0.708036941,0.293232173],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_2710:=[[4240.57,-773.69,1698.23],[0.224156636,-0.598226059,0.703045466,0.312420322],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_2720:=[[4083.12,-788.98,1727.28],[0.300730427,-0.589935623,0.675028652,0.325382068],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_2730:=[[4083.25,-828.07,1683.34],[0.173492132,-0.57846167,0.721713526,0.33824867],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_2740:=[[4081.23,-846.57,1642.37],[0.296666499,-0.501477083,0.696171768,0.419350203],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_2750:=[[4079.71,-883.86,1574.96],[0.340884011,-0.551047191,0.629917234,0.428193137],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_2760:=[[4130.4,-884.01,1507.41],[0.516173759,-0.440884236,0.552490702,0.4836732],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_2770:=[[4038.15,-886.36,1554.08],[0.500418087,-0.458762321,0.537398957,0.500321129],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_2780:=[[3976.65,-887.18,1572.33],[0.505095755,-0.453535679,0.533850017,0.504170433],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_2790:=[[3931.17,-887.67,1584],[0.50590259,-0.452626293,0.532997293,0.505080087],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_2800:=[[3931.47,-852.69,1673.14],[0.361692007,-0.580739621,0.646522712,0.337533356],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_2810:=[[3933.95,-835.46,1702.68],[0.258774284,-0.573505365,0.727617475,0.273313513],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_2820:=[[3932.76,-800.75,1744.61],[0.300702798,-0.543420646,0.70660302,0.339092909],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_2830:=[[3784.69,-804.36,1765.27],[0.359043802,-0.583408839,0.652982916,0.323009268],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_2840:=[[3783.68,-845.95,1712.7],[0.290964983,-0.591596615,0.702368565,0.268386331],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_2850:=[[3782.45,-862.91,1672.27],[0.383125061,-0.523098857,0.640199886,0.411979223],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_2860:=[[3779.84,-888.46,1596.44],[0.478488869,-0.48242459,0.584574717,0.443381683],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_2870:=[[3777.12,-887.19,1558.29],[0.514038214,-0.449826172,0.525846815,0.506859209],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_2880:=[[3697.41,-885.96,1522.62],[0.515496625,-0.44813175,0.512781566,0.52009252],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_2890:=[[3649.43,-884.41,1485.86],[0.515147256,-0.448321806,0.504268026,0.528530624],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_2900:=[[3630.66,-883.17,1464.69],[0.515145366,-0.448097644,0.499239552,0.53347317],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_2910:=[[3632.34,-887.47,1528.62],[0.486456202,-0.483362107,0.53904056,0.489036514],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_2920:=[[3631.42,-866.74,1664.01],[0.443026369,-0.520787159,0.555972954,0.47265468],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_2930:=[[3634.28,-855.75,1718.35],[0.257618534,-0.637410251,0.654542862,0.314506764],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_2940:=[[3632.49,-803.16,1788.3],[0.334755817,-0.594646196,0.616245384,0.393161635],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_2950:=[[3636.9,-790.36,1813.81],[0.342200514,-0.589729765,0.610329829,0.403255641],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_2960:=[[3560.42,-760.91,1871.02],[0.183152331,-0.598376408,0.708335845,0.326590307],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_2970:=[[3514.06,-702.43,1903.51],[0.213429312,-0.593239508,0.716821918,0.29779381],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_2980:=[[3484.88,-703.94,1908.91],[0.133433026,-0.591783032,0.726031662,0.323830969],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_2990:=[[3486.88,-753.81,1891.74],[0.206742531,-0.632429385,0.672409564,0.324277623],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_3000:=[[3484.68,-804.13,1800.6],[0.329303473,-0.562175119,0.641023452,0.405718242],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_3010:=[[3483.72,-861.01,1726.07],[0.255318205,-0.604533603,0.678316128,0.330513187],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_3020:=[[3482.49,-874.1640.43],[0.418116203,-0.502265717,0.567045298,0.501365756],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_3030:=[[3477.36,-881.98,1488.48],[0.396091202,-0.518904174,0.591139852,0.473712881],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_3040:=[[3479.95,-876.85,1321.92],[0.357462624,-0.559940688,0.588136484,0.461283399],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_3050:=[[3477.38,-839.73,1159.72],[0.313663538,-0.594505332,0.609274589,0.420669789],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_3060:=[[3474.73,-841.93,1100.94],[0.313663538,-0.594505332,0.609274589,0.420669789],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_3070:=[[3470.61,-823.61,1057.83],[0.313663538,-0.594505332,0.609274589,0.420669789],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_3080:=[[3324.78,-827.99,1066.27],[0.313663538,-0.594505332,0.609274589,0.420669789],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_3090:=[[3325.31,-835.88,1136.25],[0.348793552,-0.598402947,0.61130008,0.382843549],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_3100:=[[3332.5,-880.01,1310.44],[0.377800264,-0.584417845,0.583493862,0.418637858],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_3110:=[[3333.8,-880.43,1451.42],[0.429442754,-0.540497787,0.560829311,0.457068427],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_3120:=[[3337.95,-876.39,1658.49],[0.443469232,-0.52869091,0.540226655,0.481638997],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_3130:=[[3333.11,-869.65,1727.5],[0.475013431,-0.500848657,0.515196274,0.508021321],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_3140:=[[3327.4,-810.66,1797.22],[0.456677227,-0.519762811,0.533635508,0.486339055],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_3150:=[[3333.8,-775.43,1882.55],[0.37939577,-0.581438193,0.592350514,0.4087901],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_3160:=[[3333.93,-751.11,1931.37],[0.314312498,-0.621520407,0.627804583,0.347536247],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_3170:=[[3336.62,-721.57,1954.99],[0.234249026,-0.634959604,0.667849302,0.309727308],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_3180:=[[3335.64,-689.77,1961.81],[0.346736107,-0.618242904,0.630561621,0.316135772],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_3190:=[[3180.12,-659.83,2042.4],[0.393116595,-0.583100667,0.556453997,0.442506388],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_3200:=[[3172.89,-698.16,2034.59],[0.346498706,-0.559460933,0.598970186,0.456263988],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_3210:=[[3171.42,-722.05,2011.32],[0.459364446,-0.515351455,0.525921715,0.496793249],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_3220:=[[3177.57,-761.84,1882.67],[0.459364446,-0.515351455,0.525921715,0.496793249],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_3230:=[[3211.65,-843.27,1768.17],[0.410451493,-0.478258629,0.620126584,0.467163008],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_3240:=[[3221.89,-874.63,1728.58],[0.414664221,-0.473776814,0.615003985,0.474720142],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_3250:=[[3224.58,-878.43,1667.91],[0.405639653,-0.483045751,0.581381705,0.513924691],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_3260:=[[3230.33,-883.92,1623.47],[0.418739899,-0.471202713,0.568964949,0.528113423],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_3270:=[[3223.7,-877.98,1391.57],[0.384854067,-0.500537148,0.596394938,0.495644013],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_3280:=[[3220.91,-883.65,1324.92],[0.366943937,-0.515109172,0.607264563,0.480878818],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_3290:=[[3220.32,-878.56,1288],[0.394822383,-0.502380019,0.562610892,0.524593734],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_3300:=[[3212.77,-846.23,1174.1],[0.332886334,-0.559282293,0.595436049,0.47100522],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_3310:=[[3178.96,-837.53,1141.15],[0.371698183,-0.523828419,0.603150353,0.472920607],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_3320:=[[3179.65,-839.7,1121.18],[0.354562996,-0.536316576,0.615879031,0.455348912],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_3330:=[[3178.86,-828.2,1070.73],[0.361060321,-0.593789351,0.597699582,0.399756002],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3340:=[[3027.03,-828.86,1075.21],[0.361060321,-0.593789351,0.597699582,0.399756002],[0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3350:=[[3027.62,-839.11,1144.94],[0.367671916,-0.57132544,0.600299296,0.421954215],[0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3360:=[[3036.8,-858.3,1207.34],[0.386787961,-0.575787083,0.581381474,0.425276251],[0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3370:=[[3037.65,-885.35,1307.7],[0.406327863,-0.547973985,0.565203847,0.463860744],[0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3380:=[[3044.04,-882.2,1356.41],[0.396795238,-0.553845319,0.585284343,0.439603389],[0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3390:=[[3043.53,-879.59,1438.99],[0.409927116,-0.544974372,0.572696526,0.454952066],[0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3400:=[[3055.35,-880.48,1687.07],[0.449076236,-0.511667507,0.538575019,0.496451252],[0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3410:=[[3050.16,-877.92,1733.68],[0.478986123,-0.488536851,0.521316701,0.51003229],[0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3420:=[[3051.52,-805.16,1822.67],[0.458984054,-0.505805699,0.536988504,0.495113704],[0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3430:=[[3036.5,-709.73,2044.67],[0.495933191,-0.475874779,0.565684911,0.455624897],[0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3440:=[[3038.84,-678.83,2095.21],[0.307682905,-0.539227899,0.620845397,0.478660105],[0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3450:=[[3037.91,-626.31,2115.44],[0.309945134,-0.588396397,0.568959561,0.483744469],[0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3460:=[[2888.29,-590.88,2183.94],[0.38082821,-0.559794209,0.546359767,0.493042921],[0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3470:=[[2887.55,-634.24,2172.11],[0.433280565,-0.555213453,0.534760812,0.466944158],[0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3480:=[[2885.56,-674.13,2109.19],[0.388495042,-0.592250014,0.573847708,0.411108662],[0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3490:=[[2906.2,-806.13,1831.28],[0.364237975,-0.602560311,0.591448527,0.39298907],[0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3500:=[[2915.97,-874.53,1746.97],[0.370636553,-0.601227589,0.600154964,0.375456989],[0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3510:=[[2912.49,-882.39,1699.28],[0.438241278,-0.47771232,0.551169258,0.525307502],[0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3520:=[[2904.89,-880.27,1444.72],[0.40725062,-0.462270079,0.567359175,0.546403581],[0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3530:=[[2901.5,-886.42,1339.05],[0.341196249,-0.517838811,0.631965888,0.46480878],[0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3540:=[[2899.01,-867.6,1235.82],[0.341717252,-0.543479008,0.592265588,0.486910013],[0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3550:=[[2880.53,-840.5,1143.3],[0.344721771,-0.53435313,0.613237621,0.468586441],[0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3560:=[[2878.42,-823.49,1074.33],[0.334645382,-0.610661307,0.612879934,0.373474795],[0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3570:=[[2720.17,-815.11,1073.34],[0.361060321,-0.593789351,0.597699582,0.399756002],[0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3580:=[[2727.37,-841.74,1153.9],[0.391479056,-0.55529263,0.585312041,0.442497523],[0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3590:=[[2729.43,-879.38,1279.65],[0.355946741,-0.582171191,0.605332857,0.409817952],[0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3600:=[[2732.56,-885.2,1354.29],[0.380051195,-0.563999478,0.598419222,0.423509283],[0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3610:=[[2744.54,-883.49,1482.59],[0.414490992,-0.539289284,0.5720226,0.458426037],[0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3620:=[[2757.23,-884.79,1692.92],[0.455331522,-0.504215514,0.540488129,0.496298804],[0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3630:=[[2756.88,-880.01,1742.74],[0.467154322,-0.497054801,0.539795227,0.493279309],[0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3640:=[[2759.29,-800.49,1847.11],[0.483186975,-0.478613526,0.521653843,0.515108444],[0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3650:=[[2731.18,-645.85,2160.72],[0.435114073,-0.527154498,0.585045934,0.436468938],[0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3660:=[[2739.79,-619.27,2212.41],[0.404254263,-0.532139632,0.582122169,0.463184287],[0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3670:=[[2740.55,-596.52,2229.46],[0.427620081,-0.473716274,0.547042398,0.541736626],[0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3680:=[[2739,-563.78,2236.53],[0.334497005,-0.520997222,0.607126217,0.498067671],[0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];



LOCAL CONST robtarget Lat_Paint_3690:=[[2589.55,-541.34,2270.97],[0.317074496,-0.581023373,0.61866912,0.42323058],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3700:=[[2592.38,-579.62,2261.23],[0.396032677,-0.545720353,0.589167415,0.445229347],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3710:=[[2589.15,-624.86,2195.78],[0.412095699,-0.538191412,0.598996166,0.426298877],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3720:=[[2593.08,-795.52,1862.15],[0.408195998,-0.534741469,0.590509625,0.445786912],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3730:=[[2599.26,-883.13,1738.65],[0.380158545,-0.554993207,0.610523479,0.417998926],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3740:=[[2600.06,-885.74,1690.46],[0.457039979,-0.495607485,0.54322602,0.500393014],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3750:=[[2594.35,-878.03,1434.11],[0.4434059,-0.508698262,0.550011787,0.492040974],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3760:=[[2593.19,-886.85,1357.25],[0.423775308,-0.525675566,0.564061666,0.475304244],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3770:=[[2590.55,-884.07,1308.17],[0.396719192,-0.546062591,0.586089828,0.4482502],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3780:=[[2583,-850.82,1189.43],[0.368176959,-0.57258106,0.596377458,0.425390256],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3790:=[[2577.9,-842.31,1152.18],[0.357758884,-0.577881717,0.607525441,0.411064643],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3800:=[[2575.51,-832.85,1087.85],[0.411779328,-0.556688772,0.563956674,0.449986962],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3810:=[[2429.05,-833.79,1091.45],[0.386787961,-0.575787083,0.581381474,0.425276251],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3820:=[[2426.68,-843.39,1161.45],[0.42976768,-0.541576793,0.558109239,0.458811939],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3830:=[[2434.82,-859.64,1226.06],[0.420774865,-0.545267537,0.571692321,0.445869617],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3840:=[[2440.92,-887.51,1348.67],[0.47229718,-0.49817179,0.540819037,0.48608128],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3850:=[[2443.54,-881.65,1414.34],[0.43423326,-0.530785498,0.543796143,0.483729249],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3860:=[[2445.75,-881.51,1491.92],[0.464690664,-0.499300595,0.536722439,0.496679501],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3870:=[[2447.42,-885.49,1689.9],[0.465737639,-0.498328462,0.536300129,0.497131137],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3880:=[[2451.72,-881.49,1742.58],[0.3998649,-0.553113667,0.594273251,0.425455798],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3890:=[[2436.46,-794.63,1870.38],[0.454727902,-0.507343853,0.550092906,0.482931201],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3900:=[[2439.32,-778.23,1893.77],[0.425555004,-0.53239448,0.574115809,0.453707059],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3910:=[[2441.23,-614.29,2209.81],[0.402362018,-0.552518618,0.602048148,0.412754178],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3920:=[[2440.5,-581.94,2265.89],[0.359992003,-0.579065412,0.62675103,0.377189812],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3930:=[[2441.7,-565.8,2280.14],[0.417615983,-0.537027788,0.586667767,0.43933925],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3940:=[[2443.41,-532.2,2289.67],[0.388717026,-0.548370785,0.604351002,0.427724704],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3950:=[[2288.69,-524.6,2300.05],[0.37748138,-0.572679521,0.614889146,0.38917517],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3960:=[[2290.07,-592.34,2249.71],[0.429162842,-0.538423587,0.583624753,0.430466543],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3970:=[[2280.95,-647.57,2162.89],[0.449536676,-0.521463002,0.564303463,0.455581734],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3980:=[[2294.7,-756.53,1922.21],[0.47306465,-0.500025154,0.538116664,0.486431021],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_3990:=[[2299.02,-821.33,1847.49],[0.440232061,-0.524995377,0.561650672,0.463814735],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_4000:=[[2317.2,-870.78,1772.99],[0.44989484,-0.517434114,0.556916005,0.468722876],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_4010:=[[2318.46,-883.77,1699.2],[0.44989484,-0.517434114,0.556916005,0.468722876],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_4020:=[[2319.98,-888.26,1621.37],[0.44989484,-0.517434114,0.556916005,0.468722876],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_4030:=[[2319.71,-885.09,1555.7],[0.44989484,-0.517434114,0.556916005,0.468722876],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];



LOCAL CONST robtarget Lat_Paint_4040:=[[2304.58,-881.6,1450.46],[0.44989484,-0.517434114,0.556916005,0.468722876],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_4050:=[[2301,-886.44,1376.59],[0.44989484,-0.517434114,0.556916005,0.468722876],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_4060:=[[2296.55,-870.66,1288.39],[0.44989484,-0.517434114,0.556916005,0.468722876],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_4070:=[[2282.6,-848.07,1189.52],[0.44989484,-0.517434114,0.556916005,0.468722876],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_4080:=[[2277.46,-844.32,1162.84],[0.397551124,-0.558754559,0.57988924,0.439857837],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_4090:=[[2280.654730309,-870.917987222,959.343108938],[0.361060321,-0.593789351,0.597699582,0.399756002],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_4100:=[[2127.536,-870.918,945.701],[0.398016067,-0.590209554,0.578908162,0.397619457],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_4110:=[[2127.12,-846.16,1164.35],[0.465194591,-0.537954292,0.524523104,0.468054148],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_4120:=[[2124.75,-858.6,1241.5],[0.489661348,-0.51417024,0.506251222,0.489459324],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_4130:=[[2132.89,-882.51,1393.45],[0.511881117,-0.488464696,0.507657424,0.491593228],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_4140:=[[2144.25,-885.17,1471.12],[0.485990404,-0.506954082,0.492010984,0.514525099],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_4150:=[[2163.27,-881.62,1704.54],[0.488964454,-0.505149205,0.518420778,0.4868038],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_4160:=[[2157.71,-859.85,1800.07],[0.46263283,-0.527476893,0.537546816,0.467741823],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_4170:=[[2147.99,-789.3,1885.4],[0.417623137,-0.562598831,0.571479767,0.427181866],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_4180:=[[2143.03,-636.48,2176.41],[0.434296563,-0.554123571,0.572907509,0.419655274],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_4190:=[[2139.64,-580.72,2261.12],[0.451407492,-0.539925471,0.556957093,0.441033512],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_4200:=[[2140.48,-540.48,2300.32],[0.324573577,-0.61886836,0.637388611,0.324637805],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_4210:=[[2140.64,-514.67,2305.85],[0.211138826,-0.663167889,0.682533298,0.223107695],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_4220:=[[1990.53,-504.47,2307.17],[0.181228773,-0.677746407,0.691902513,0.17054868],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_4230:=[[1991.07,-547.35,2292.4],[0.361067965,-0.607787159,0.619005348,0.342136045],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_4240:=[[1988.71,-584.62,2248.6],[0.466160889,-0.536139622,0.543243807,0.447363943],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_4250:=[[1986.51,-778.23,1899.34],[0.395859754,-0.58354543,0.588071167,0.396159172],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_4260:=[[1988.26,-831.41,1847.23],[0.404741462,-0.575444355,0.578567034,0.412684298],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_4270:=[[1991.51,-867.54,1769.55],[0.431881381,-0.555174027,0.570512033,0.424000345],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_4280:=[[1995.1,-878.93,1701.47],[0.487490751,-0.509390572,0.521324271,0.480723432],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_4290:=[[1991.04,-881.33,1543.74],[0.458848611,-0.535770293,0.533811268,0.466319284],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_4300:=[[1991.89,-887,1501.96],[0.435309316,-0.555274709,0.535337068,0.464316724],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_4310:=[[1989.39,-886.64,1472.77],[0.485511272,-0.50163106,0.531933438,0.47926183],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_4320:=[[1991.32,-875.19,1377.34],[0.471084518,-0.516324289,0.53705859,0.472288764],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_4330:=[[1990.97,-860.16,1243.83],[0.445146938,-0.543844962,0.542926435,0.459682224],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_4340:=[[1976.91,-848.88,1170.71],[0.423833032,-0.561676959,0.557783398,0.44018432],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_4350:=[[1976.89,-833.84,1090.63],[0.398075301,-0.594328249,0.567114081,0.408278834],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_4360:=[[1824.46,-833.95,1084.62],[0.419677366,-0.581749943,0.548947183,0.429062819],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_4370:=[[1825.25,-851.99,1168.58],[0.434275849,-0.568814777,0.546572427,0.434871036],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_4380:=[[1832.61,-869.41,1292.85],[0.440805543,-0.56241181,0.544906628,0.438702855],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg Lat_Paint_4390:=[[1837.1,-878.12,1446.19],[0.467775051,-0.534378137,0.542824986,0.448294037],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4400:=[[1844.13,-885.37,1516.66],[0.469434297,-0.531017807,0.54838497,0.443886382],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4410:=[[1843.96,-884.14,1554.54],[0.445213665,-0.551344635,0.531293211,0.464253605],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4420:=[[1846.41,-879.02,1627.6],[0.464752654,-0.528895804,0.54801358,0.451614123],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4430:=[[1848.59,-873.1,1727.65],[0.457442819,-0.534827278,0.554513694,0.444095051],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4440:=[[1832.18,-776.99,1900.2],[0.461092685,-0.528716296,0.547358823,0.456345191],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4450:=[[1838.45,-597.53,2220.03],[0.480787213,-0.536397009,0.550150296,0.422441186],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4460:=[[1838.52,-559.34,2273.56],[0.460430419,-0.543378381,0.55281186,0.43260006],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4470:=[[1838.12,-544.55,2287.61],[0.394812306,-0.590784085,0.603743316,0.361374344],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4480:=[[1841.22,-493.69,2304.37],[0.35207856,-0.613796905,0.629598417,0.320780111],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4490:=[[1690.02,-492.7,2297.94],[0.363494547,-0.617415978,0.623535066,0.312847002],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4500:=[[1689.77,-524.89,2291],[0.38046223,-0.607898188,0.612495459,0.332502027],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4510:=[[1688.49,-596.28,2212.33],[0.508520334,-0.511408045,0.499225243,0.480253097],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4520:=[[1688.55,-771.45,1903.8],[0.416113156,-0.581911852,0.571409277,0.402144099],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4530:=[[1692.31,-830.26,1846.06],[0.412143905,-0.585420669,0.57671998,0.393464239],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4540:=[[1695.24,-851.45,1796.32],[0.432808064,-0.570663017,0.56782016,0.405710694],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4550:=[[1696.34,-874.97,1650.5],[0.462803015,-0.547760111,0.544632881,0.434910629],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4560:=[[1696.63,-882.64,1593.8],[0.440250133,-0.565920774,0.531318832,0.451235854],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4570:=[[1698.8,-883.26,1557.22],[0.481690252,-0.522431738,0.534195701,0.457902318],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4580:=[[1697.83,-880.27,1505.47],[0.481190702,-0.524708931,0.525669304,0.465626276],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4590:=[[1699.72,-880.07,1388.58],[0.438816259,-0.565009603,0.540443918,0.442859809],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4600:=[[1697.81,-858.92,1212.31],[0.413027436,-0.582320477,0.566251828,0.411910266],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4610:=[[1683.86,-856.99,1156.4],[0.411678846,-0.58425072,0.561253779,0.417331785],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4620:=[[1680.95,-841.92,1083.06],[0.397638907,-0.599583802,0.568458705,0.399045443],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4630:=[[1595.95,-859.1,1093.29],[0.408989328,-0.594733908,0.557676223,0.409898205],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4640:=[[1588.02,-868.65,1264.9],[0.441005781,-0.549525622,0.582581055,0.40513554],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4650:=[[1555.71,-878.97,1363.96],[0.455931779,-0.537706979,0.573766849,0.416880105],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4660:=[[1528.63,-880.9,1401.29],[0.475628088,-0.524300249,0.553243364,0.439100161],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4670:=[[1534.59,-883.73,1519.04],[0.458515938,-0.544534151,0.518979828,0.473186677],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4680:=[[1532.85,-879.55,1619.23],[0.50097684,-0.500150194,0.523564676,0.474080182],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4690:=[[1533.82,-875.89,1648.93],[0.436384218,-0.55354635,0.549409194,0.448659802],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4700:=[[1541.01,-849.77,1782.5],[0.437228601,-0.553303929,0.572623411,0.418076957],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4710:=[[1537.42,-788.16,1884.08],[0.409231045,-0.569916513,0.589118932,0.400829145],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4720:=[[1538.57,-754.6,1918.07],[0.450714006,-0.54086686,0.563227726,0.432544162],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4730:=[[1536.79,-611.71,2176.47],[0.460711696,-0.533240926,0.555621903,0.441229134],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg Lat_Paint_4740:=[[1537.41,-567.65,2240.78],[0.420394435,-0.566417053,0.595031659,0.385198087],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4750:=[[1539.5,-514.76,2284.19],[0.328834938,-0.6209117,0.65736863,0.272401779],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4760:=[[1539.06,-488.13,2288.08],[0.322747536,-0.628550155,0.660795418,0.253195865],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4770:=[[1390.22,-482.81,2274.71],[0.365760086,-0.620358308,0.638866286,0.270601179],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4780:=[[1390.59,-511.18,2270.09],[0.358966557,-0.619725444,0.636869016,0.285449193],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4790:=[[1394.75,-584.61,2205.69],[0.437656896,-0.573945858,0.572093881,0.389552544],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4800:=[[1388.21,-650.67,2094.26],[0.481187678,-0.540376155,0.532127014,0.439650852],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4810:=[[1388.66,-744.85,1923.44],[0.439099619,-0.57004095,0.560697299,0.409711335],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4820:=[[1388,-758.23,1907.29],[0.414884005,-0.585117146,0.575512779,0.392803041],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4830:=[[1385.13,-803.49,1867.65],[0.402058621,-0.593191344,0.583744967,0.38172596],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4840:=[[1386.04,-840.98,1793.48],[0.440448414,-0.565993309,0.566310359,0.406139565],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4850:=[[1383.71,-859.36,1720.1],[0.449257531,-0.559799884,0.55338585,0.422558707],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4860:=[[1387,-868.9,1683.85],[0.436426377,-0.569613772,0.553719058,0.42245399],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4870:=[[1385.53,-872.21,1659.32],[0.475710767,-0.537253446,0.531828536,0.449684789],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4880:=[[1382.73,-877.41,1624.82],[0.438542709,-0.567337746,0.530687692,0.451861425],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4890:=[[1379.67,-884.85,1534.01],[0.475740269,-0.529080374,0.530626176,0.460631107],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4900:=[[1229.05,-885.62,1579.59],[0.478100903,-0.528123545,0.521318979,0.469820785],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4910:=[[1234.13,-870.6,1664.44],[0.422978229,-0.572860731,0.572681255,0.406147979],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4920:=[[1233.27,-844.64,1756.68],[0.435650297,-0.563527701,0.569694315,0.409992362],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4930:=[[1231.38,-817,1836.93],[0.391238177,-0.592914404,0.602429551,0.363955812],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4940:=[[1231.56,-756.42,1901.72],[0.44177377,-0.558981023,0.565943851,0.414829736],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4950:=[[1235.17,-697.64,1985.32],[0.43745579,-0.563687252,0.579455118,0.393853881],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4960:=[[1239.18,-628.51,2114],[0.409469359,-0.581953959,0.607986388,0.352160455],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4970:=[[1238.1,-564.32,2208.18],[0.405890909,-0.585495957,0.616808112,0.334656253],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4980:=[[1238.58,-525.67,2240.83],[0.373828942,-0.605630436,0.642670996,0.283615389],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_4990:=[[1240,-475.9,2256.41],[0.348707525,-0.626271443,0.657623133,0.231773502],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_5000:=[[1184.19,-488.12,2246.64],[0.360145648,-0.620482189,0.65081131,0.248478579],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_5010:=[[1165.33,-557.05,2201.99],[0.329210959,-0.633051096,0.659830341,0.235563951],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_5020:=[[1119.6,-590.44,2147.97],[0.385220512,-0.608987473,0.636458596,0.275063392],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_5030:=[[1083.13,-602.03,2117.71],[0.404836801,-0.598562575,0.632715879,0.27838934],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_5040:=[[1086.77,-694.56,1965.14],[0.457004938,-0.570636156,0.572527829,0.371123631],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_5050:=[[1085.22,-735.8,1911.6],[0.435432608,-0.578861498,0.576987368,0.377363733],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_5060:=[[1085.6,-796.76,1852.98],[0.399583401,-0.600565214,0.601139164,0.343927659],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_5070:=[[1081.33,-853.41,1688.11],[0.483999202,-0.543946018,0.561143827,0.39368148],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg Lat_Paint_5080:=[[1081.35,-861.15,1664.69],[0.455722677,-0.562445,0.586388357,0.363484744],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Lat_Paint_5090:=[[1082.24,-885.7,1592.06],[0.462810464,-0.559201708,0.571396377,0.382891766],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_5100:=[[1080.76,-884.97,1558.7],[0.476825671,-0.559032944,0.525807064,0.428539821],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_5110:=[[987.56,-882.41,1516],[0.446713409,-0.580390296,0.556523304,0.392270375],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_5120:=[[932.76,-879.1,1515.9],[0.506736276,-0.527813692,0.61160528,0.300948556],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_5130:=[[933.12,-853.87,1590.32],[0.498012423,-0.533634038,0.613962115,0.300447768],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_5140:=[[933.47,-832.95,1665.62],[0.521704848,-0.531242674,0.559951273,0.36340039],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_5150:=[[934.39,-804.01,1796.53],[0.469283943,-0.567635861,0.578110623,0.351212496],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_5160:=[[934.3,-737.95,1888.52],[0.454615808,-0.573265584,0.584004885,0.351609628],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_5170:=[[936.26,-659.09,1967.47],[0.531235684,-0.539365694,0.569113293,0.3209102],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_5180:=[[935.96,-637.7,1993.96],[0.552515602,-0.529164811,0.562517094,0.313505395],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_5190:=[[885.68,-640.65,1959.97],[0.565217636,-0.526561075,0.568774823,0.282413983],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_5200:=[[853.8,-702.14,1894.37],[0.514124078,-0.533300471,0.584925378,0.330347307],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_5210:=[[830.38,-796.31,1724.75],[0.580931723,-0.486220974,0.53994173,0.36683842],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_5220:=[[781.96,-779.99,1691.45],[0.603715548,-0.449209571,0.582435971,0.307419319],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_5230:=[[782.32,-782.69,1653.01],[0.618658056,-0.429439678,0.590522319,0.290046828],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_5240:=[[783.03,-753.44,1596.18],[0.539339271,-0.514423698,0.561544591,0.359373178],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_5250:=[[783.22,-760.19,1548.1],[0.614712475,-0.438188985,0.559047749,0.342906111],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_5260:=[[775.7,-748.45,1446.33],[0.607295843,-0.392782251,0.638879339,0.262196591],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_5270:=[[783.21,-750.58,1415.94],[0.58965544,-0.397949341,0.657079988,0.249376571],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_5280:=[[842.3,-790.23,1421.03],[0.588885036,-0.399755158,0.656221474,0.250566567],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Lat_Paint_5290:=[[973.973294736,-861.715612246,1593.498139036],[0.611052265,-0.417934158,0.593787121,0.315219011],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget M_Paint_5300:=[[786.56,98.29118776,1325.046169734],[0.487298903,-0.512386357,0.487298903,-0.512386357],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget M_Paint_5310:=[[786.56,-733.69,1366.83],[0.487298903,-0.512386357,0.487298903,-0.512386357],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget M_Paint_5320:=[[736.99,-654.38,1413.25],[0.547325722,-0.359083621,0.382462403,-0.652085898],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget M_Paint_5330:=[[726.19,-390.61,1410.2],[0.539263833,-0.41054445,0.30112502,-0.670799147],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget M_Paint_5340:=[[717.33,-62.54,1409.96],[0.575659449,-0.315956011,0.343894753,-0.671211141],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget M_Paint_5350:=[[655.59,-63.11,1421.72],[0.575653611,-0.315951873,0.343963157,-0.671183046],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget M_Paint_5360:=[[641.46,-62.73,1476.74],[0.527325538,-0.476665121,0.522293361,-0.471092119],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget M_Paint_5370:=[[650.91,-340.17,1470.51],[0.549353422,-0.499487212,0.499885338,-0.445912538],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget M_Paint_5380:=[[665.36,-451.83,1481.5],[0.566204383,-0.515453411,0.481137761,-0.42688035],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget M_Paint_5390:=[[682.71,-526.26,1478.15],[0.602108707,-0.53218426,0.440833909,-0.399888089],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget M_Paint_5400:=[[752.36,-728.43,1478.59],[0.620667614,-0.542422837,0.417171904,-0.382775106],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget M_Paint_5410:=[[746.67,-730.41,1579.73],[0.618059604,-0.543346697,0.41978299,-0.382830165],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget M_Paint_5420:=[[684.92,-548.35,1582.31],[0.603556986,-0.542108392,0.436415625,-0.389074361],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget M_Paint_5430:=[[649.42,-393.32,1575.11],[0.484097431,-0.561159391,0.543096313,-0.394710285],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg M_Paint_5440:=[[662.94,-428.45,1624.37],[0.600832921,-0.300855015,0.715480113,-0.191244006],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg M_Paint_5450:=[[654.4,-438.46,1675],[0.639769185,-0.251241259,0.682254905,-0.249201654],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg M_Paint_5460:=[[651.96,-504.53,1677.7],[0.695569386,-0.239232167,0.623187148,-0.265685865],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg M_Paint_5470:=[[660.11,-448.95,1732.73],[0.610848208,-0.212636646,0.706362064,-0.287580872],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg M_Paint_5480:=[[669.98,-435.64,1769.54],[0.544698234,-0.080587005,0.736872487,-0.392209774],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg M_Paint_5490:=[[682.3,-430.06,1807.4],[0.56895347,-0.058599526,0.694500017,-0.436494869],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg M_Paint_5500:=[[708.72,-443.81,1845.98],[0.474794678,-0.102426696,0.843498176,-0.229324253],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg M_Paint_5510:=[[721.84,-485.24,1867.18],[0.420367202,-0.05999377,0.892978715,-0.149268808],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg M_Paint_5520:=[[762.01,-539.3,1878.92],[0.449817228,-0.014975471,0.89167558,-0.048526873],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg M_Paint_5530:=[[845.4,-566.59,1886.06],[0.30132462,0.029932568,0.948027548,0.097730662],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg M_Paint_5540:=[[845.19,-567.12,1897.57],[0.379325251,0.041497419,0.920901987,0.079560344],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg M_Paint_5550:=[[889.8,-572.89,1923.35],[0.454837785,0.04604597,0.881609937,0.117329777],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg M_Paint_5560:=[[951.95,-563.69,1972.57],[0.474249274,0.048600814,0.871282617,0.116585541],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg M_Paint_5570:=[[989.6,-566.55,2005.68],[0.558073249,0.058710267,0.816581246,0.135286444],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg M_Paint_5580:=[[1022.56,-550.39,2035.87],[0.569247694,0.06030638,0.8081101,0.138846205],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg M_Paint_5590:=[[1075.81,-538.08,2087.56],[0.560072085,0.055365828,0.808072725,0.173989527],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg M_Paint_5600:=[[1099.64,-526.21,2111.65],[0.64358757,-0.19382955,0.73731962,-0.067712059],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg M_Paint_5610:=[[1110.11,-490.83,2135.78],[0.641933346,-0.401598764,0.634709442,-0.154220417],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg M_Paint_5620:=[[1105.15,-428.73,2157.58],[0.65883986,-0.373215792,0.627378339,-0.18175927],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg M_Paint_5630:=[[1104.39,-388.12,2167.99],[0.65883986,-0.373215792,0.627378339,-0.18175927],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg M_Paint_5640:=[[1105.38,-312.23,2177.88],[0.65883986,-0.373215792,0.627378339,-0.18175927],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg M_Paint_5650:=[[1114.84,-243.6,2190.41],[0.65883986,-0.373215792,0.627378339,-0.18175927],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg M_Paint_5660:=[[1123.51,-166.97,2190.05],[0.65883986,-0.373215792,0.627378339,-0.18175927],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg M_Paint_5670:=[[1134.82,-8.92,2189.29],[0.65883986,-0.373215792,0.627378339,-0.18175927],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg M_Paint_5680:=[[1195.51,-6.36,2193.82],[0.688860674,-0.314373916,0.609149611,-0.235747246],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg M_Paint_5690:=[[1188.45,-259.46,2196.43],[0.688860674,-0.314373916,0.609149611,-0.235747246],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg M_Paint_5700:=[[1178.71,-388.94,2197.24],[0.69941253,-0.290141517,0.600551073,-0.25686265],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg M_Paint_5710:=[[1167.6,-503.24,2190.8],[0.686091049,-0.320373313,0.611183673,-0.230422503],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg M_Paint_5720:=[[1141.11,-537.44,2172.79],[0.705959588,-0.09719886,0.701387208,-0.015146782],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg M_Paint_5730:=[[1068.82,-566.56,2114.6],[0.606280395,-0.160979045,0.74575826,0.224397965],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg M_Paint_5740:=[[1029.14,-587.88,2069.19],[0.59061356,-0.079684171,0.803001032,-0.00392399],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg M_Paint_5750:=[[969.94,-601.94,2017.48],[0.608909779,-0.109269416,0.784883712,-0.035307697],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg M_Paint_5760:=[[769.466482243,-637.168500177,1873.577979676],[0.577092357,-0.136111123,0.804001265,-0.044945957],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg M_Paint_5770:=[[1205.535989659,140.903958016,2240.905859],[0.546593133,-0.443962211,0.44922121,-0.549848893],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg M_Paint_5780:=[[1184.17,-442.88,2235.82],[0.54297943,-0.421997402,0.45256553,-0.567693555],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget M_Paint_5790:=[[1178.56,-491.4,2219.88],[0.506746447,-0.407328817,0.487034709,-0.583171043],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget M_Paint_5800:=[[1126.74,-483.69,2304.63],[0.546547891,-0.352128001,0.423408709,-0.630885361],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget M_Paint_5810:=[[1135.12,-276.16,2311.73],[0.546547891,-0.352128001,0.423408709,-0.630885361],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget M_Paint_5820:=[[1148.3,-5.3,2309.08],[0.546547891,-0.352128001,0.423408709,-0.630885361],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget M_Paint_5830:=[[1091.88,-6.21,2359.68],[0.546547891,-0.352128001,0.423408709,-0.630885361],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget M_Paint_5840:=[[1085.83,-385.53,2352.43],[0.546547891,-0.352128001,0.423408709,-0.630885361],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget M_Paint_5850:=[[1073.89,-453.99,2346.21],[0.546547891,-0.352128001,0.423408709,-0.630885361],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget M_Paint_5860:=[[1037.93,-515.43,2341.88],[0.546547891,-0.352128001,0.423408709,-0.630885361],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget M_Paint_5870:=[[917.57,-557.43,2390.9],[0.182509412,-0.484800095,0.032451381,-0.854754988],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget M_Paint_5880:=[[785.46,-548.59,2478.65],[0.182509412,-0.484800095,0.032451381,-0.854754988],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget M_Paint_5890:=[[716.92,-678.99,2477.9],[0.55117352,-0.26124459,0.414338066,-0.675487218],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget M_Paint_5900:=[[717.81,-574.41,2523.55],[0.6190905,-0.24549175,0.255880738,-0.70070379],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget M_Paint_5910:=[[662.63,-458.24,2575.95],[0.6190905,-0.24549175,0.255880738,-0.70070379],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget M_Paint_5920:=[[642.75,-377.15,2550.91],[0.645837199,-0.162584009,0.162231445,-0.72810831],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget M_Paint_5930:=[[613.53,-173.3,2596.76],[0.645837199,-0.162584009,0.162231445,-0.72810831],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget M_Paint_5940:=[[610.22,-73.3,2574.8],[0.645837199,-0.162584009,0.162231445,-0.72810831],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget M_Paint_5950:=[[615.47,-13.72,2567.05],[0.645837199,-0.162584009,0.162231445,-0.72810831],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget M_Paint_5960:=[[492.08,-6.09,2570.75],[0.645837199,-0.162584009,0.162231445,-0.72810831],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget M_Paint_5970:=[[498.8,-170.8,2564.01],[0.645837199,-0.162584009,0.162231445,-0.72810831],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget M_Paint_5980:=[[509.97,-297.15,2551.53],[0.645837199,-0.162584009,0.162231445,-0.72810831],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget M_Paint_5990:=[[539.71,-432.81,2613.41],[0.645837199,-0.162584009,0.162231445,-0.72810831],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget M_Paint_6000:=[[466.12,-499.59,2597.98],[0.645837199,-0.162584009,0.162231445,-0.72810831],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget M_Paint_6010:=[[444.63,-371.47,2589.49],[0.645837199,-0.162584009,0.162231445,-0.72810831],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget M_Paint_6020:=[[434.08,-249.72,2539.69],[0.645837199,-0.162584009,0.162231445,-0.72810831],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget M_Paint_6030:=[[433.87,-140.72,2548.11],[0.645837199,-0.162584009,0.162231445,-0.72810831],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget M_Paint_6040:=[[434.29,-13.81,2552.52],[0.645837199,-0.162584009,0.162231445,-0.72810831],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget M_Paint_6050:=[[369.41,-9.07,2532.11],[0.645837199,-0.162584009,0.162231445,-0.72810831],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget M_Paint_6060:=[[373.31,-165.08,2528.94],[0.645837199,-0.162584009,0.162231445,-0.72810831],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget M_Paint_6070:=[[379.53,-271.12,2522.29],[0.645837199,-0.162584009,0.162231445,-0.72810831],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget M_Paint_6080:=[[383.48,-368.7,2577.68],[0.645837199,-0.162584009,0.162231445,-0.72810831],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget M_Paint_6090:=[[397.72,-526.18,2575.1],[0.645837199,-0.162584009,0.162231445,-0.72810831],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget M_Paint_6100:=[[319.22,-461.36,2564.42],[0.645837199,-0.162584009,0.162231445,-0.72810831],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget M_Paint_6110:=[[311,-379.55,2567.87],[0.645837199,-0.162584009,0.162231445,-0.72810831],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget M_Paint_6120:=[[306.51,-261.97,2574.53],[0.645837199,-0.162584009,0.162231445,-0.72810831],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget M_Paint_6130:=[[302.599528639,142.382276444,2579.362452413],[0.645837199,-0.162584009,0.162231445,-0.72810831],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg B_Paint_6140:=[[4270.160778555,-787.838079383,1420.40718973],[0.314725234,-0.140197302,0.159800499,-0.925071102],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg B_Paint_6150:=[[4082.89,-797.47,1545.98],[0.358932885,-0.082853506,0.083948436,-0.925880738],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg B_Paint_6160:=[[4016.2,-796,1576.31],[0.357140692,-0.042298129,0.048182432,-0.931847545],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg B_Paint_6170:=[[3948.91,-793.27,1591.66],[0.419368529,0.040798411,0.061586504,-0.904805299],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg B_Paint_6180:=[[3844.59,-797.46,1601.15],[0.354543968,0.142229106,0.052972918,-0.922639326],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg B_Paint_6190:=[[3733.52,-795.14,1559.96],[0.239814912,0.348934199,-0.035122807,-0.90526246],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg B_Paint_6200:=[[3615.18,-785.21,1473.09],[0.135174927,0.452072198,-0.013847506,-0.881570595],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg B_Paint_6210:=[[3542.98,-773.7,1369.61],[0.152662931,0.590785185,-0.042958041,-0.791088807],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg B_Paint_6220:=[[3521.11,-748.98,1201.29],[0.149918163,0.676052515,-0.15899322,-0.703703558],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg B_Paint_6230:=[[3515.26,-736.2,1080.01],[0.388361388,0.593136742,-0.372542279,-0.598812566],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg B_Paint_6240:=[[3515.34,-597.89,1075.93],[0.310819699,0.524278955,-0.439362147,-0.65991181],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg B_Paint_6250:=[[3531.02,-588.78,1261.08],[0.005833382,0.631188444,-0.280349824,-0.723167405],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg B_Paint_6260:=[[3613.89,-589.57,1447.45],[0.078739423,-0.384514486,0.269128511,0.879499038],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg B_Paint_6270:=[[3704.11,-613.63,1535.18],[0.358826931,0.069994512,-0.330664831,-0.870060211],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg B_Paint_6280:=[[4006.88,-631.86,1573.73],[0.410878752,-0.23295112,-0.182973287,-0.862225726],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg B_Paint_6290:=[[4060.67,-622.09,1545.77],[0.349842366,-0.294031619,-0.152217728,-0.876347813],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg B_Paint_6300:=[[3969.56,-517.75,1466.53],[0.023821703,-0.044895486,0.482213476,0.874578233],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg B_Paint_6310:=[[3856.07,-512.75,1461.1],[0.025757344,-0.043813667,0.519903082,0.852711972],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg B_Paint_6320:=[[3832.42,-497.94,1359.61],[0.027643954,-0.042648447,0.556603023,0.829222525],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg B_Paint_6330:=[[4027.63,-497.94,1348.15],[0.029477942,-0.041402043,0.592243439,0.804154606],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg B_Paint_6340:=[[4717.972556058,-497.94,1367.260889413],[0.029477942,-0.041402043,0.592243439,0.804154606],[0,0,0,0],[1200,9E+09,9E+09,9E+09,9E+09,0]];

CONST jointtarget Pos_Int3_1:=[-180,-20,20,0,-45,0],[6545,9E+09,9E+09,9E+09,9E+09,0]];

CONST robtarg Pos_Int6:=[[1000,-1250,2500],[0.640856382,-0.298836238,0.640856382,-0.298836238],[0,0,0,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];

CONST robtarg Pos_Int6_2:=[[2154.698541113,-2072.991980913,1694.614861711],[0.100953942,-0.391759634,0.908467742,0.104971677],[-3,-1,0,-1],[-300,9E+09,9E+09,9E+09,9E+09,0]];

CONST robtarg Aprox_Bajos1:=[[4270.163626376,-1414.118435667,1420.405685912],[0.314725377,-0.140199019,0.159800255,-0.925070835],[-2,1,0,-1],[1200,9E+09,9E+09,9E+09,9E+09,0]];

CONST robtarg Pos_Int7:=[[807.117,-900,1396.54],[0.69636424,-0.69636424,-0.122787804,0.122787804],[0,0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg B2_Paint_6350:=[[807.116851231,-790.874413283,1396.53980705],[0.908925139,-0.006395513,-0.280840554,-0.30812785],[0,0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg B2_Paint_6360:=[[1029.92,-807.46,1583.72],[0.90147148,0.01187204,-0.205467783,-0.38077712],[0,0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg B2_Paint_6370:=[[1160.06,-806.02,1630.29],[0.824641246,0.0308682,-0.074420062,-0.559888938],[0,0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg B2_Paint_6380:=[[1294.91,-804.06,1624.03],[0.687060173,-0.055238659,0.052375517,-0.722602114],[0,0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg B2_Paint_6390:=[[1418.03,-799.87,1567.47],[0.666425193,-0.193326832,0.180129446,-0.697176865],[0,0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg B2_Paint_6400:=[[1539.52,-792.85,1402.64],[0.785319501,-0.136435055,0.585949324,-0.146021053],[0,0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg B2_Paint_6410:=[[1571.2,-793.93,1261.77],[0.753371684,-0.139113035,0.636508,-0.08908555],[0,0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg B2_Paint_6420:=[[1579.72,-805.08,1083.34],[0.7160612,-0.055830281,0.695766155,0.006985356],[0,0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg B2_Paint_6430:=[[1569.45,-699.14,1079.9],[0.707220948,-0.11376911,0.695319005,-0.058537183],[0,0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg B2_Paint_6440:=[[1562.51,-690.06,1244.02],[0.714283603,-0.154303835,0.681333914,0.042111258],[0,0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarg B2_Paint_6450:=[[1553.88,-691.46,1325.26],[0.735661202,-0.17369101,0.652107199,0.058225685],[0,0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg B2_Paint_6460:=[[1506.07,-683.72,1463.24],[0.81374768,-0.20505724,0.531843189,0.113618062],[0,0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg B2_Paint_6470:=[[1366.17,-688.32,1597.85],[0.75800113,-0.274791167,0.060685401,-0.588422793],[0,0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg B2_Paint_6480:=[[1234.39,-684.91,1633.33],[0.923373604,-0.178191142,0.034281359,-0.338310349],[0,0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg B2_Paint_6490:=[[1103.58,-687.8,1616.55],[0.911015775,-0.121688088,-0.168616663,-0.356104884],[0,0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg B2_Paint_6500:=[[986.34,-687.66,1552.79],[0.923246472,-0.123383924,-0.177388371,-0.317688093],[0,0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg B2_Paint_6510:=[[924.64,-682.36,1489.62],[0.915007465,-0.14542603,-0.194449416,-0.322183229],[0,0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg B2_Paint_6520:=[[989.01,-620.6,1465.46],[0.87543383,-0.445483474,0.030737092,0.184973821],[0,0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg B2_Paint_6530:=[[1139.79,-620.85,1513.9],[0.873259878,-0.453127917,0.068893724,0.165366051],[0,0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg B2_Paint_6540:=[[1273.91,-619.5,1515.71],[0.848029829,-0.4749594,0.21948706,0.084168932],[0,0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg B2_Paint_6550:=[[1442.4,-623.87,1409.54],[0.825673261,-0.480487842,0.292562515,0.042453205],[0,0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg B2_Paint_6560:=[[1499.48,-602.38,1289.54],[0.825673261,-0.480487842,0.292562515,0.042453205],[0,0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg B2_Paint_6570:=[[1483.59,-530.64,1251.01],[0.825673261,-0.480487842,0.292562515,0.042453205],[0,0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg B2_Paint_6580:=[[1242.48,-530.64,1315.44],[0.791319308,-0.581074925,0.15658419,0.107921622],[0,0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg B2_Paint_6590:=[[1058.63,-530.64,1345.44],[0.797396254,-0.575814395,0.121918293,0.133165037],[0,0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg B2_Paint_6600:=[[941.98,-530.64,1335.69],[0.804987799,-0.56201715,0.051956695,0.182843835],[0,0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg B2_Paint_6610:=[[838.98,-530.64,1332.85],[0.829863974,-0.517801372,0.025815452,0.206254907],[0,0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg B2_Paint_6620:=[[818.86,-638.07,1311.29],[0.80195531,-0.56945777,0.087020318,0.158154964],[0,0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarg B2_Paint_6630:=[[638.833779579,-619.768101632,1296.671502562],[0.80195531,-0.56945777,0.087020318,0.158154964],[0,0,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];
 CONST jointtarget Pos_Int8:=[[0,0,0,0,44.999994422,0],[6670,9E+09,9E+09,9E+09,9E+09,0]];
 CONST jointtarget Pos_Int9:=[[-90,-30,30,0,44.999994422,0],[6670,9E+09,9E+09,9E+09,9E+09,0]];
 CONST robtarg Limpieza_2:=[[-8474.76,0,-290],[0,1,0,0],[-2,1,-2,-1],[6670,9E+09,9E+09,9E+09,9E+09,0]];
 CONST robtarg Limpieza_1:=[[-8474.76,0,-150],[0,1,0,0],[-3,1,-2,-1],[6670,9E+09,9E+09,9E+09,9E+09,0]];
 CONST robtarg Pos_Sal_Cap1:=[[4480,-228.12,1848.391],[0.653281482,0.27059805,-0.653281482,-0.27059805],[-2,-1,1,-1],[3750,9E+09,9E+09,9E+09,9E+09,0]];
 CONST jointtarget Pos_Sal_Int2:=[[-180,-45,45,0,-45,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];
 CONST robtarg Pos_Sal_PD1:=[[2748.619,-1179.264,1392.253],[0.141544676,-0.157712987,0.976686233,-0.034288778],[-2,-1,0,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 CONST robtarg Pos_Sal_PD2_1:=[[1172.002,-1522.882,1392.253],[0.135613846,-0.689395125,0.709594071,0.053098969],[-1,0,-1,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 CONST robtarg Pos_Sal_PD2_2:=[[787.271811091,-2386.802457234,1985.830432144],[0.286787956,-0.736279172,0.286787956,0.541662549],[0,0,-1,0],[520,9E+09,9E+09,9E+09,9E+09,0]];
 CONST jointtarget Pos_Sal_Int1:=[[0,-45,45,0,45,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];
 CONST robtarg Pos_Sal_PT1:=[[1682.582,-1214.373,1392.253],[0.141544676,-0.157712987,0.976686233,-0.034288778],[-2,-1,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 CONST robtarg Pos_Sal_PT2_1:=[[402.346,-1472.914,1392.253],[0.139725674,-0.62492657,0.766978596,0.041077386],[-1,0,-1,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 CONST robtarg Pos_Sal_PT2_2:=[[-157.728188909,-2386.802457234,1985.830432144],[0.286787956,-0.736279172,0.286787956,0.541662549],[0,0,-1,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
 CONST robtarg Pos_Sal_Malet1:=[[133.378,-297.145,2145.168],[0.591311481,-0.570551212,0.411216842,-0.394617227],[-1,-1,-1,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];
 CONST robtarg Pos_Sal_Malet2:=[[-32.728188909,-2386.802457234,1985.830432144],[0.286787956,-0.736279172,0.286787956,0.541662549],[0,0,-1,0],[-300,9E+09,9E+09,9E+09,9E+09,0]];
 CONST robtarg Pos_Sal_Baj1_1:=[[3937.765,-692.525,914.824],[0.46818888,0.331201992,-0.44717531,-0.686395407],[-2,0,0,-1],[1200,9E+09,9E+09,9E+09,9E+09,0]];
 CONST robtarg Pos_Sal_Baj2_1:=[[1058.881,-855.003,964.62],[0.635068602,-0.333697338,0.33237484,-0.612258869],[-1,-2,0,0],[-480,9E+09,9E+09,9E+09,9E+09,0]];
 CONST jointtarget Pos_Mant_Int_3:=[[0,-45,45,0,-45,0],[500,9E+09,9E+09,9E+09,9E+09,0]];
 CONST robtarg Mantenimiento_2:=[[2025.24,-350,1360],[0.707106781,0,0.707106781,0],[-1,1,-2,-1],[0,9E+09,9E+09,9E+09,9E+09,0]];
 CONST robtarg Mantenimiento_3:=[[2675.24,-350,1360],[0.707106781,0,0.707106781,0],[-1,1,-2,-1],[0,9E+09,9E+09,9E+09,9E+09,0]];

```

CONST jointtarget Pos_Mant_Int_1:=[-180,-45,45,0,-45,0],[500,9E+09,9E+09,9E+09,0];
CONST jointtarget Pos_Mant_Int_2:=[-90,-45,45,0,-45,0],[500,9E+09,9E+09,9E+09,0];
LOCAL CONST robtarget Paint_380:=[[-8474.76,-700,0],[0,1,0,0],[-2,1,-2,-1],[6670,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_390:=[[-8474.76,-700,-265],[0,1,0,0],[-2,1,-2,-1],[6670,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_400:=[[-8474.76,-700,0],[0,1,0,0],[-2,1,-2,-1],[6670,9E+09,9E+09,9E+09,0]];

```

!!!!!!!!!!!!!!!!!!!!!! DEFINICIÓN DE VARIABLES !!!!!!!!!!!!!!!!!!!!!!!

```

VAR num Color;
VAR intnum Limp_Atom;
VAR intnum Paro_Robot;
VAR intnum Mant_Atom;
VAR num Lectura_Color;
VAR iodev infile;
VAR num Contador;
VAR num Secuencia_Colores{1000};
VAR string date;
VAR string N_Proc;
VAR string Limpieza;

```

!!!!!!!!!!!!!!!!!!!!!! PROCESOS DE PINTADO !!!!!!!!!!!!!!!!!!!!!!!

PROC main()

```

Reset Continuar;
Reset Abrir_P;
Reset Volver;
Reset Avance_Inf_P;
Reset Stop_Conveyor;
Reset Finalizado;
Reset Finalizado2;
Reset Cambio_Color;
Reset Pos_Parada;
Reset Paro_Resto;
Reset Open_TMant;
Reset Close_TMant;
SetAO Señal_Luminosa,3;
Lectura_Colores;
! Eliminar los avisos de trayectoria de fallo de esquina
CornerPathWarning FALSE;

```

! Definición de los parametros para la rutina TRAP de limpieza si se requiere la limpieza del atomizador

```

IDelete Limp_Atom;
CONNECT Limp_Atom WITH Limpieza_Atomicador;
ISignalDI Limpieza_Atomicador,1,Limp_Atom;

```

! Definición de los parametros para la rutina TRAP de paro del robot en caso de que otro de los robots lo solicite

```

IDelete Paro_Robot;
CONNECT Paro_Robot WITH Parada;
ISignalDI Paro_Propio,1,Paro_Robot;

```

! Definición de los parametros para la rutina TRAP de mantenimiento si se requiere realizar mantenimiento al atomizador

```

IDelete Mant_Atom;
CONNECT Mant_Atom WITH Mantenimiento_Atomicador;
ISignalDI Mantenimiento,1,Mant_Atom;

```

```

Color:=Secuencia_Colores{1};
Set Cambio_Color;
ActUnit CNV1;
FOR i1 FROM 1 TO Contador-1 DO
  Reset Finalizado2;
  Principal;
  Reset Cambio_Color;
  N_Proc:="main";
  WaitRob\ZeroSpeed;
  IF Finalizado=1 AND Color<>Secuencia_Colores{i1+1} AND Secuencia_Colores{i1+1}<>0 THEN
    Limpieza:="Limp1";
    MoveAbsJ Pos_Int9,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
    Limpieza_Atomicador_Color;
    Color:=Secuencia_Colores{i1+1};
    Set Cambio_Color;
    ! Purga de color para preparar el atomizador

```

```

PaintL Paint_380,v2000,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetBrush Color\Z:=-150;
PaintL Paint_390,v800,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
WaitTime 2;
SetBrush 1\Z:=-150;
PaintL Paint_400,v800,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
N_Proc:="main";
MoveAbsJ Pos_Int9,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
ELSEIF Finalizado=1 AND Color<>Secuencia_Colores{i1+1} AND Secuencia_Colores{i1+1}=0 THEN
  Limpieza:="Limp1";
  MoveAbsJ Pos_Int9,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  Limpieza_Atom_Color;
  N_Proc:="main";
  MoveAbsJ Pos_Int9,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
ELSE
  Color:=Secuencia_Colores{i1+1};
  Set Cambio_Color;
  MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
ENDIF
WaitRob\ZeroSpeed;
Set Finalizado2;
ENDFOR
DeactUnit CNV1;
! Stop;
ENDPROC

PROC Principal()
  N_Proc:="Principal1";
  Reset Abrir_P;
  Reset Volver;
  Reset Continuar;
  Reset Finalizado2;
  MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  WaitWObj wobj_cnv1;
  Reset Finalizado;
  WaitDI Init_Pint1,1;
  WaitTime 4.5;
  Pint_Bajo_Capo;
  N_Proc:="Principal2_2";
  MoveLDO Pos_Int1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1,Abrir_P,1;
  Reset Continuar;
  WaitDI Init_Pint2,1;
  Reset Abrir_P;
  Pint_PD_Del;
  N_Proc:="Principal2_3";
  MoveLDO Pos_Int2,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1,Abrir_P,1;
  WaitDI Init_Pint2,1;
  Pint_PD_Tras;
  N_Proc:="Principal3";
  Reset Abrir_P;
  MoveL Pos_Int5,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
  WaitDI Init_Pint2,1;
  WaitTime 1;
  Pint_Maletero_Inf;
  N_Proc:="Principal3";
  MoveLDO Pos_Int5,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1,Continuar,1;
  MoveL Pos_Int6,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
  N_Proc:="Principal2";
  MoveL Pos_Int6_2,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
  MoveAbsJ Pos_Int3_1,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  Reset Avance_Inf_P;
  Pint_Bajos_1_1;
  N_Proc:="Principal2";
  MoveAbsJ Pos_Int3_1,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  Pint_Lat;
  Pint_Bajos_2_2;
  N_Proc:="Principal3";
  Set Finalizado;
  MoveAbsJ Pos_Int8,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;

```

```

Set Volver;
Reset Continuar;
DropWObj wobj_cnv1;

```

ENDPROC

PROC Lectura_Colores()

```

VAR string Num_CE;
! Lectura del archivo con la cadena de números referentes a los colores
date:=CDate();
Open date+"_Lista_Colores.txt",infile\Read;
Lectura_Color:=ReadNum(infile\Delim:=" ");
Contador:=1;
Num_CE:=NumToStr(Contador,0);
! Comprobación de que el archivo no está vacío o contine codigos de color incorrectos.
IF Lectura_Color>EOF_NUM THEN
  TPWrite "El archivo que contiene la Secuencia de colores está vacío/incorrecto, revíselo.";
  ErrWrite "Color error","El archivo que contiene la Secuencia de colores está vacío/incorrecto, revíselo.";
  Break;
  ! Stop;
ELSEIF Lectura_Color>16 OR Lectura_Color<2 THEN
  TPWrite "El identificador de color "+Num_CE+" es incorrecto, revíselo.";
  ErrWrite "Color error","El identificador de color "+Num_CE+" es incorrecto, revíselo."\\RL2:="Identificador superior a 16 o inferior a 2.";
  Break;
ENDIF
! Lectura de la secuencia de colores completa.
Secuencia_Colores{Contador}:=Lectura_Color;
WHILE Lectura_Color<EOF_NUM DO
  Contador:=Contador+1;
  Num_CE:=NumToStr(Contador,0);
  Lectura_Color:=ReadNum(infile\Delim:=" ");
  ! Comprobación de que el archivo no contiene ningún código de color incorrecto.
  IF Lectura_Color<EOF_NUM THEN
    IF Lectura_Color>16 OR Lectura_Color<2 THEN
      TPWrite "El identificador de color "+Num_CE+" es incorrecto, revíselo.";
      ErrWrite "Color error","El identificador de color "+Num_CE+" es incorrecto, revíselo."\\RL2:="Identificador superior a 16 o inferior a 2.";
      Break;
    ENDIF
  ENDIF
  Secuencia_Colores{Contador}:=Lectura_Color;
ENDWHILE
Secuencia_Colores{Contador}:=0;
ENDPROC

```

PROC Limpieza_Atom_Color()

```

N_Proc:="Limpieza_Atom_Color";
MoveL Limpieza_1,v1500,fine,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveL Limpieza_2,v100,fine,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
IF Limpieza="Limp1" THEN
  Set RUN_Limp;
  WaitTime 5;
  MoveL Limpieza_1,v100,fine,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
ELSEIF Limpieza="Limp2" THEN
  Set RUN_Limp;
  WaitTime 5;
  MoveL Limpieza_1,v100,fine,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  ! Purga de color para preparar el atomizador
  PaintL Paint_380,v2000,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  SetBrush Color\Z:=-150;
  PaintL Paint_390,v800,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  WaitTime 2;
  SetBrush 1\Z:=-150;
  PaintL Paint_400,v800,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
ENDIF
Reset RUN_Limp;
ENDPROC

```

PROC Mantenimiento_Atom()

```

MoveAbsJ Pos_Mant_Int_3,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;

```

```

MoveLDO Mantenimiento_2,v500,fine,RB1000_WSC_70_700_200\WObj:=WO_In_Int,Open_TMant,1;
WaitDI Ok_TMant,1;
VelSet 100,100;
MoveL Mantenimiento_3,v100,fine,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Mantenimiento,0;
SetAO Señal_Luminosa,1;
Reset Open_TMant;
MoveLDO Mantenimiento_2,v100,fine,RB1000_WSC_70_700_200\WObj:=WO_In_Int,Close_TMant,1;
VelSet 100,MaxRobSpeed();
MoveAbsJ Pos_Mant_Int_3,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Reset Close_TMant;

```

ENDPROC

PROC Pint_Bajo_Capo()

```

N_Proc:="Pint_Bajo_Capo";
MoveL Aprox_Bajo_Capo,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_10,v2000,z20,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
SetBrush Color\Z:=1198.7;
PaintL Paint_20,v1500,z20,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_30,v1500,z20,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_40,v1500,z20,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_50,v1500,z20,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_60,v1500,z20,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_70,v1500,z20,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_80,v1500,z20,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_90,v1500,z20,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_100,v1500,z20,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_110,v1500,z20,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_120,v1500,z20,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_130,v1500,z20,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_140,v1500,z20,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_150,v1500,z20,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_160,v1500,z20,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_170,v1500,z20,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_180,v1500,z20,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_190,v1500,z20,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_200,v1500,z20,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_210,v1500,z20,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_220,v1500,z20,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_230,v1500,z20,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_240,v1500,z20,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_250,v1500,z20,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_260,v1500,z20,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_270,v1500,z20,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_280,v1500,z20,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_290,v1500,z20,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_300,v1500,z20,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_310,v1500,z20,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_320,v1500,z20,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_330,v1500,z20,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_340,v1500,z20,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_350,v1500,z20,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
SetBrush 1\Y:=-778.90;
PaintL Paint_360,v1500,z20,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveLDO Paint_370,v1500,z20,RB1000_WSC_70_700_200\WObj:=wobj_cnv1,Continuar,1;

```

ENDPROC

PROC Pint_PD_Del()

```

N_Proc:="Pint_PD_Del";
PaintL PD_Paint_410,v2000,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
SetBrush Color\Z:=1836.67;
PaintL PD_Paint_420,v1500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL PD_Paint_430,v1500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL PD_Paint_440,v1500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL PD_Paint_450,v1500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL PD_Paint_460,v1500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL PD_Paint_470,v1500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL PD_Paint_480,v1500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL PD_Paint_490,v1500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL PD_Paint_500,v1500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;

```



```

PaintL B2_Paint_6450,v800,z100,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL B2_Paint_6460,v800,z100,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL B2_Paint_6470,v800,z100,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL B2_Paint_6480,v800,z100,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL B2_Paint_6490,v800,z100,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL B2_Paint_6500,v800,z100,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL B2_Paint_6510,v800,z100,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL B2_Paint_6520,v800,z100,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL B2_Paint_6530,v800,z100,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL B2_Paint_6540,v800,z100,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL B2_Paint_6550,v800,z100,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL B2_Paint_6560,v800,z100,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL B2_Paint_6570,v800,z100,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL B2_Paint_6580,v800,z100,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL B2_Paint_6590,v800,z100,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL B2_Paint_6600,v800,z100,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL B2_Paint_6610,v800,z100,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL B2_Paint_6620,v800,z100,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
SetBrush 1\X:=787.58;
PaintL B2_Paint_6630,v800,z100,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
ENDPROC

```

!!!!!!!!!!!!!!!!!!!!!!!!!!!! RUTINAS TRAP !!!!!!!!!!!!!!!!!!!!!!!!!!!!!

TRAP Parada

```

VAR robtarget Punt_Paro;
VAR jointtarget jointPunt_Paro;
SetAO Señal_Luminosa,1;
StopMove\Quick;
StorePath;
Punt_Paro:=CRobT(\Tool:=RB1000_WSC_70_700_200\WObj:=WO_In_Int);
jointPunt_Paro:=CJointT(\TaskRef:=T_ROB1Id);
TEST N_Proc
CASE "main":
  MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  SetAO Señal_Luminosa,2;
  WaitDI Paro_Propio,0;
  SetAO Señal_Luminosa,1;
CASE "Principal1":
  MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_200\WObj:=WO_In_Int;
  SetAO Señal_Luminosa,2;
  WaitDI Paro_Propio,0;
  SetAO Señal_Luminosa,1;
CASE "Principal2":
  MoveAbsJ Pos_Sal_Int2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  SetAO Señal_Luminosa,2;
  WaitDI Paro_Propio,0;
  SetAO Señal_Luminosa,1;
CASE "Principal2_2":
  MoveL Pos_Sal_PD2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
  MoveL Pos_Sal_PD2_2,v2500,z50,tool0\WObj:=wobj_cnv1;
  MoveAbsJ Pos_Sal_Int1,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  SetAO Señal_Luminosa,2;
  WaitDI Paro_Propio,0;
  SetAO Señal_Luminosa,1;
  MoveL Pos_Sal_PD2_2,v2500,z50,tool0\WObj:=wobj_cnv1;
  MoveL Pos_Sal_PD2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
CASE "Principal2_3":
  MoveL Pos_Sal_PT2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
  MoveL Pos_Sal_PT2_2,v2500,z50,tool0\WObj:=wobj_cnv1;
  MoveAbsJ Pos_Sal_Int1,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  SetAO Señal_Luminosa,2;
  WaitDI Paro_Propio,0;
  SetAO Señal_Luminosa,1;
  MoveL Pos_Sal_PT2_2,v2500,z50,tool0\WObj:=wobj_cnv1;
  MoveL Pos_Sal_PT2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
CASE "Principal3":
  MoveAbsJ Pos_Sal_Int1,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  SetAO Señal_Luminosa,2;
  WaitDI Paro_Propio,0;
  SetAO Señal_Luminosa,1;

```

CASE "Limpieza_Atom_Color":
 MoveL Limpieza_1,v1500,fine,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveAbsJ Pos_Int9,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 SetAO Señal_Luminosa,2;
 WaitDI Paro_Propio,0;
 SetAO Señal_Luminosa,1;
 MoveAbsJ Pos_Int9,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveL Limpieza_1,v1500,fine,RB1000_WSC_70_700_0\WObj:=WO_In_Int;

CASE "Pint_Bajo_Capo":
 MoveL Pos_Sal_Cap1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
 MoveAbsJ Pos_Sal_Int2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 SetAO Señal_Luminosa,2;
 WaitDI Paro_Propio,0;
 SetAO Señal_Luminosa,1;
 MoveL Pos_Sal_Cap1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;

CASE "Pint_PD_Del":
 MoveL Pos_Sal_PD1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
 MoveL Pos_Int1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
 MoveL Pos_Sal_PD2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
 MoveL Pos_Sal_PD2_2,v2500,z50,tool0\WObj:=wobj_cnv1;
 MoveAbsJ Pos_Sal_Int1,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 SetAO Señal_Luminosa,2;
 WaitDI Paro_Propio,0;
 SetAO Señal_Luminosa,1;
 MoveL Pos_Sal_PD2_2,v2500,z50,tool0\WObj:=wobj_cnv1;
 MoveL Pos_Sal_PD2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
 MoveL Pos_Int1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
 MoveL Pos_Sal_PD1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;

CASE "Pint_PD_Tras":
 MoveL Pos_Sal_PT1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
 MoveL Pos_Int2,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
 MoveL Pos_Sal_PT2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
 MoveL Pos_Sal_PT2_2,v2500,z50,tool0\WObj:=wobj_cnv1;
 MoveAbsJ Pos_Sal_Int1,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 SetAO Señal_Luminosa,2;
 WaitDI Paro_Propio,0;
 SetAO Señal_Luminosa,1;
 MoveL Pos_Sal_PT2_2,v2500,z50,tool0\WObj:=wobj_cnv1;
 MoveL Pos_Sal_PT2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
 MoveL Pos_Int2,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
 MoveL Pos_Sal_PT1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;

CASE "Pint_Lat":
 MoveAbsJ Pos_Sal_Int2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 SetAO Señal_Luminosa,2;
 WaitDI Paro_Propio,0;
 SetAO Señal_Luminosa,1;

CASE "Pint_Maletero_Inf":
 MoveL Pos_Sal_Malet1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
 MoveL Pos_Sal_Malet2,v2500,z50,tool0\WObj:=wobj_cnv1;
 MoveAbsJ Pos_Sal_Int1,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 SetAO Señal_Luminosa,2;
 WaitDI Paro_Propio,0;
 SetAO Señal_Luminosa,1;
 MoveL Pos_Sal_Malet2,v2500,z50,tool0\WObj:=wobj_cnv1;
 MoveL Pos_Sal_Malet1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;

CASE "Pint_Bajos_1_1":
 MoveL Pos_Sal_Baj1_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
 MoveL Offs(Pos_Sal_Baj1_1,0,-400,250),v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
 MoveAbsJ Pos_Sal_Int2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 SetAO Señal_Luminosa,2;
 WaitDI Paro_Propio,0;
 SetAO Señal_Luminosa,1;
 MoveL Offs(Pos_Sal_Baj1_1,0,-400,250),v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
 MoveL Pos_Sal_Baj1_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;

CASE "Pint_Bajos_2_2":
 MoveL Pos_Sal_Baj2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
 MoveAbsJ Pos_Sal_Int1,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 SetAO Señal_Luminosa,2;
 WaitDI Paro_Propio,0;
 SetAO Señal_Luminosa,1;


```

MoveL Pos_Sal_Baj2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
DEFAULT:
SetAO Señal_Luminosa,2;
WaitDI Paro_Propio,0;
SetAO Señal_Luminosa,1;
ENDTEST
MoveAbsJ jointPunt_Paro,v1500,fine,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Set Pos_Parada;
WaitDI OK_PosP,1;
RestoPath;
StartMove;
WaitTime 0.25;
Reset Pos_Parada;
SetAO Señal_Luminosa,3;
ENDTRAP

TRAP Limpieza_Atomicador
VAR robtarget Punt_Paro;
VAR jointtarget jointPunt_Paro;
VAR string N_PROC2;
StopMove\Quick;
Set Paro_Resto;
Set Stop_Conveyor;
SetAO Señal_Luminosa,1;
StorePath;
N_PROC2:=N_Proc;
Limpieza:="Limp2";
Punt_Paro:=CRobT(\Tool:=RB1000_WSC_70_700_200\WObj:=WO_In_Int);
jointPunt_Paro:=CJointT(\TaskRef:=T_ROB1Id);
TEST N_Proc
CASE "main":
MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Int9,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atomicolor;
MoveAbsJ Pos_Int9,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Limpieza_Atomicolor,0;
Reset Paro_Resto;
SetAO Señal_Luminosa,1;
CASE "Principal1":
MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Int9,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atomicolor;
MoveAbsJ Pos_Int9,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Limpieza_Atomicolor,0;
Reset Paro_Resto;
SetAO Señal_Luminosa,1;
CASE "Principal2":
MoveAbsJ Pos_Sal_Int2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atomicolor;
MoveAbsJ Pos_Sal_Int2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Limpieza_Atomicolor,0;
Reset Paro_Resto;
SetAO Señal_Luminosa,1;
CASE "Principal2_2":
MoveL Pos_Sal_PD2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Pos_Sal_PD2_2,v2500,z50,tool0\WObj:=wobj_cnv1;
MoveAbsJ Pos_Sal_Int1,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Int8,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Int9,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atomicolor;
MoveAbsJ Pos_Int9,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Int8,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Sal_Int1,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Limpieza_Atomicolor,0;
Reset Paro_Resto;

```

```

SetAO Señal_Luminosa,1;
MoveL Pos_Sal_PD2_2,v2500,z50,tool0\WObj:=wobj_cnv1;
MoveL Pos_Sal_PD2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
CASE "Principal2_3":
MoveL Pos_Sal_PT2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Pos_Sal_PT2_2,v2500,z50,tool0\WObj:=wobj_cnv1;
MoveAbsJ Pos_Sal_Int1,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Int8,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Int9,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atom_Color;
MoveAbsJ Pos_Int9,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Int8,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Sal_Int1,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Limpieza_Atom,0;
Reset Paro_Resto;
SetAO Señal_Luminosa,1;
MoveL Pos_Sal_PT2_2,v2500,z50,tool0\WObj:=wobj_cnv1;
MoveL Pos_Sal_PT2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
CASE "Principal3":
MoveAbsJ Pos_Sal_Int1,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Int8,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Int9,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atom_Color;
MoveAbsJ Pos_Int9,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Int8,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Sal_Int1,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Limpieza_Atom,0;
Reset Paro_Resto;
SetAO Señal_Luminosa,1;
CASE "Limpieza_Atom_Color":
SetAO Señal_Luminosa,2;
WaitDI Limpieza_Atom,0;
Reset Paro_Resto;
SetAO Señal_Luminosa,1;
CASE "Pint_Bajo_Capo":
MoveL Pos_Sal_Cap1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveAbsJ Pos_Sal_Int2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atom_Color;
MoveAbsJ Pos_Sal_Int2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Limpieza_Atom,0;
Reset Paro_Resto;
SetAO Señal_Luminosa,1;
MoveL Pos_Sal_Cap1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
CASE "Pint_PD_Del":
MoveL Pos_Sal_PD1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Pos_Int1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Pos_Sal_PD2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Pos_Sal_PD2_2,v2500,z50,tool0\WObj:=wobj_cnv1;
MoveAbsJ Pos_Sal_Int1,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Int8,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Int9,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atom_Color;
MoveAbsJ Pos_Int9,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Int8,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Sal_Int1,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Limpieza_Atom,0;
Reset Paro_Resto;
SetAO Señal_Luminosa,1;
MoveL Pos_Sal_PD2_2,v2500,z50,tool0\WObj:=wobj_cnv1;
MoveL Pos_Sal_PD2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Pos_Int1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Pos_Sal_PD1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
CASE "Pint_PD_Tras":
MoveL Pos_Sal_PT1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Pos_Int2,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Pos_Sal_PT2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Pos_Sal_PT2_2,v2500,z50,tool0\WObj:=wobj_cnv1;

```

```

MoveAbsJ Pos_Sal_Int1,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Int8,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Int9,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atom_Color;
MoveAbsJ Pos_Int9,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Int8,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Sal_Int1,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Limpieza_Atom,0;
Reset Paro_Resto;
SetAO Señal_Luminosa,1;
MoveL Pos_Sal_PT2_2,v2500,z50,tool0\WObj:=wobj_cnv1;
MoveL Pos_Sal_PT2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Pos_Int2,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Pos_Sal_PT1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
CASE "Pint_Lat":
MoveAbsJ Pos_Sal_Int2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atom_Color;
MoveAbsJ Pos_Sal_Int2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Limpieza_Atom,0;
Reset Paro_Resto;
SetAO Señal_Luminosa,1;
CASE "Pint_Maletero_Inf":
MoveL Pos_Sal_Malet1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Pos_Sal_Malet2,v2500,z50,tool0\WObj:=wobj_cnv1;
MoveAbsJ Pos_Sal_Int1,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Int8,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Int9,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atom_Color;
MoveAbsJ Pos_Int9,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Int8,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Sal_Int1,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Limpieza_Atom,0;
Reset Paro_Resto;
SetAO Señal_Luminosa,1;
MoveL Pos_Sal_Malet2,v2500,z50,tool0\WObj:=wobj_cnv1;
MoveL Pos_Sal_Malet1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
CASE "Pint_Bajos_1_1":
MoveL Pos_Sal_Baj1_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Offs(Pos_Sal_Baj1_1,0,-400,250),v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveAbsJ Pos_Sal_Int2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atom_Color;
MoveAbsJ Pos_Sal_Int2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Limpieza_Atom,0;
Reset Paro_Resto;
SetAO Señal_Luminosa,1;
MoveL Offs(Pos_Sal_Baj1_1,0,-400,250),v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Pos_Sal_Baj1_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
CASE "Pint_Bajos_2_2":
MoveL Pos_Sal_Baj2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveAbsJ Pos_Sal_Int1,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Int8,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Int9,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atom_Color;
MoveAbsJ Pos_Int9,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Int8,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Sal_Int1,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Limpieza_Atom,0;
Reset Paro_Resto;
SetAO Señal_Luminosa,1;
MoveL Pos_Sal_Baj2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
DEFAULT:
MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Int9,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atom_Color;
MoveAbsJ Pos_Int9,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;

```

```

SetAO Señal_Luminosa,2;
WaitDI Limpieza_Atom,0;
Reset Paro_Resto;
SetAO Señal_Luminosa,1;
ENDTEST
MoveAbsJ jointPunt_Paro,v1500,fine,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
N_Proc:=N_PROCC2;
Set Pos_Parada;
WaitDI OK_PosP,1;
RestoPath;
StartMove;
WaitTime 0.25;
Reset Stop_Conveyor;
Reset Pos_Parada;
SetAO Señal_Luminosa,3;
ENDTRAP

TRAP Mantenimiento_Atomizador
VAR robtarget Punt_Paro;
VAR jointtarget jointPunt_Paro;
VAR string N_PROCC2;
StopMove\Quick;
Set Paro_Resto;
Set Stop_Conveyor;
SetAO Señal_Luminosa,1;
StorePath;
N_PROCC2:=N_Proc;
Limpieza:="Limp1";
Punt_Paro:=CRobT(\Tool:=RB1000_WSC_70_700_200\WObj:=WO_In_Int);
jointPunt_Paro:=CJointT(\TaskRef:=T_ROB1Id);
TEST N_Proc
CASE "main":
MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Int9,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atom_Color;
MoveAbsJ Pos_Int9,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Mantenimiento_Atom;
Reset Paro_Resto;
MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
CASE "Principal1":
MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Int9,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atom_Color;
MoveAbsJ Pos_Int9,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Mantenimiento_Atom;
Reset Paro_Resto;
MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
CASE "Principal2":
MoveAbsJ Pos_Sal_Int2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atom_Color;
MoveAbsJ Pos_Int9,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Mantenimiento_Atom;
Reset Paro_Resto;
MoveAbsJ Pos_Mant_Int_2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Mant_Int_1,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Sal_Int2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
CASE "Principal2_2":
MoveL Pos_Sal_PD2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Pos_Sal_PD2_2,v2500,z50,tool0\WObj:=wobj_cnv1;
MoveAbsJ Pos_Sal_Int1,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Int8,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Int9,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atom_Color;
MoveAbsJ Pos_Int9,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Mantenimiento_Atom;
Reset Paro_Resto;
MoveAbsJ Pos_Sal_Int1,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;

```

MoveL Pos_Sal_PD2_2,v2500,z50,tool0\WObj:=wobj_cnv1;
 MoveL Pos_Sal_PD2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
CASE "Principal2_3":
 MoveL Pos_Sal_PT2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
 MoveL Pos_Sal_PT2_2,v2500,z50,tool0\WObj:=wobj_cnv1;
 MoveAbsJ Pos_Sal_Int1,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveAbsJ Pos_Int8,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveAbsJ Pos_Int9,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Limpieza_Atom_Color;
 MoveAbsJ Pos_Int9,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Mantenimiento_Atom;
 Reset Paro_Resto;
 MoveAbsJ Pos_Sal_Int1,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveL Pos_Sal_PT2_2,v2500,z50,tool0\WObj:=wobj_cnv1;
 MoveL Pos_Sal_PT2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
CASE "Principal3":
 MoveAbsJ Pos_Sal_Int1,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveAbsJ Pos_Int8,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveAbsJ Pos_Int9,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Limpieza_Atom_Color;
 MoveAbsJ Pos_Int9,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Mantenimiento_Atom;
 Reset Paro_Resto;
 MoveAbsJ Pos_Sal_Int1,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
CASE "Limpieza_Atom_Color":
 MoveL Limpieza_1,v1500,fine,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Limpieza_Atom_Color;
 MoveAbsJ Pos_Int9,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Mantenimiento_Atom;
 Reset Paro_Resto;
 MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveAbsJ Pos_Int9,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveL Limpieza_1,v1500,fine,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
CASE "Pint_Bajo_Capo":
 MoveL Pos_Sal_Cap1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
 MoveAbsJ Pos_Sal_Int2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Limpieza_Atom_Color;
 MoveAbsJ Pos_Int9,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Mantenimiento_Atom;
 Reset Paro_Resto;
 MoveAbsJ Pos_Mant_Int_2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveAbsJ Pos_Mant_Int_1,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveAbsJ Pos_Sal_Int2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveL Pos_Sal_Cap1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
CASE "Pint_PD_Del":
 MoveL Pos_Sal_PD1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
 MoveL Pos_Int1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
 MoveL Pos_Sal_PD2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
 MoveL Pos_Sal_PD2_2,v2500,z50,tool0\WObj:=wobj_cnv1;
 MoveAbsJ Pos_Sal_Int1,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveAbsJ Pos_Int8,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveAbsJ Pos_Int9,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Limpieza_Atom_Color;
 MoveAbsJ Pos_Int9,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Mantenimiento_Atom;
 Reset Paro_Resto;
 MoveAbsJ Pos_Sal_Int1,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveL Pos_Sal_PD2_2,v2500,z50,tool0\WObj:=wobj_cnv1;
 MoveL Pos_Sal_PD2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
 MoveL Pos_Int1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
 MoveL Pos_Sal_PD1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
CASE "Pint_PD_Tras":
 MoveL Pos_Sal_PT1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
 MoveL Pos_Int2,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
 MoveL Pos_Sal_PT2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
 MoveL Pos_Sal_PT2_2,v2500,z50,tool0\WObj:=wobj_cnv1;

MoveAbsJ Pos_Sal_Int1,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveAbsJ Pos_Int8,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveAbsJ Pos_Int9,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Limpieza_Atom_Color;
 MoveAbsJ Pos_Int9,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Mantenimiento_Atom;
 Reset Paro_Resto;
 MoveAbsJ Pos_Sal_Int1,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveL Pos_Sal_PT2_2,v2500,z50,tool0\WObj:=wobj_cnv1;
 MoveL Pos_Sal_PT2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
 MoveL Pos_Int2,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
 MoveL Pos_Sal_PT1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
CASE "Pint_Lat":
 MoveAbsJ Pos_Sal_Int2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Limpieza_Atom_Color;
 MoveAbsJ Pos_Int9,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Mantenimiento_Atom;
 Reset Paro_Resto;
 MoveAbsJ Pos_Mant_Int_2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveAbsJ Pos_Mant_Int_1,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveAbsJ Pos_Sal_Int2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
CASE "Pint_Maletero_Inf":
 MoveL Pos_Sal_Malet1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
 MoveL Pos_Sal_Malet2,v2500,z50,tool0\WObj:=wobj_cnv1;
 MoveAbsJ Pos_Sal_Int1,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveAbsJ Pos_Int8,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveAbsJ Pos_Int9,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Limpieza_Atom_Color;
 MoveAbsJ Pos_Int9,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Mantenimiento_Atom;
 Reset Paro_Resto;
 MoveAbsJ Pos_Sal_Int1,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveL Pos_Sal_Malet2,v2500,z50,tool0\WObj:=wobj_cnv1;
 MoveL Pos_Sal_Malet1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
CASE "Pint_Bajos_1_1":
 MoveL Pos_Sal_Baj1_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
 MoveL Offs(Pos_Sal_Baj1_1,0,-400,250),v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
 MoveAbsJ Pos_Sal_Int2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Limpieza_Atom_Color;
 MoveAbsJ Pos_Int9,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Mantenimiento_Atom;
 Reset Paro_Resto;
 MoveAbsJ Pos_Mant_Int_2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveAbsJ Pos_Mant_Int_1,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveAbsJ Pos_Sal_Int2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveL Offs(Pos_Sal_Baj1_1,0,-400,250),v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
 MoveL Pos_Sal_Baj1_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
CASE "Pint_Bajos_2_2":
 MoveL Pos_Sal_Baj2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
 MoveAbsJ Pos_Sal_Int1,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveAbsJ Pos_Int8,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveAbsJ Pos_Int9,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Limpieza_Atom_Color;
 MoveAbsJ Pos_Int9,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Mantenimiento_Atom;
 Reset Paro_Resto;
 MoveAbsJ Pos_Sal_Int1,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveL Pos_Sal_Baj2_1,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
DEFAULT:
 MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveAbsJ Pos_Int9,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Limpieza_Atom_Color;
 MoveAbsJ Pos_Int9,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Mantenimiento_Atom;
 Reset Paro_Resto;

```

MoveAbsJ Pos_Inicial2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
ENDTEST
MoveAbsJ jointPunt_Paro,v1500,fine,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
N_Proc:=N_PROC2;
Set Pos_Parada;
WaitDI OK_PosP,1;
RestoPath;
StartMove;
WaitTime 0.25;
Reset Stop_Conveyor;
Reset Pos_Parada;
SetAO Señal_Luminosa,3;
ENDTRAP
ENDMODULE

```

8.1.3 Controlador virtual Cont Sup Izq

```

MODULE Module1
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! Módulo: Module1
!
! Descripción: Procesos para la aplicación de la capa base sobre la carrocera empleando el robot superior izquierdo
!
! Autor: Arturo López López
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

!!!!!!!!!!!!!!!!!!!!!!!!!!!! DEFINICIÓN DE OBJETIVOS !!!!!!!!!!!!!!!!!!!!!!!

CONST jointtarget Pos_Inicial:=[[0,0,0,-45,0],[6500,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtargt Abrir_Capo1:=[[5250,104.992,1858.807],[0.502426647,0.497636297,0.497079388,-0.502829655],[1,-
2,1,0],[4100,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtargt Abrir_Capo2:=[[5063.099983881,104.992078004,1858.806812759],[0.502427429,0.497635472,0.497079405,-
0.502829673],[0,-2,1,0],[4100,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtargt Abrir_Capo3:=[[5063.1,104.992,1897.146],[0.502428459,0.497634586,0.497078492,-
0.502830425],[0,0,0,0],[4100,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtargt Abrir_Capo4:=[[5062.901,104.992,1909.425],[0.502428459,0.497634586,0.497078492,-
0.502830425],[0,0,0,0],[4100,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtargt Abrir_capo5:=[[4861.096535329,104.992,2654.520688343],[0.61232361,0.606242265,0.35664762,-
0.361016049],[0,0,0,0],[4100,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtargt Abrir_Capo6:=[[4301.444,104.992,3214.721],[0.686532723,0.679501837,0.181663609,-
0.184250938],[0,0,0,0],[4100,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtargt Abrir_Capo7:=[[5500,600,3214.721],[0.686532723,0.679501837,0.181663609,-0.184250938],[0,1,-1,-
1],[4100,9E+09,9E+09,9E+09,9E+09,0]];
CONST jointtarget Pos_Int1:=[[90.053435074,35.791215779,33.27518241,-
98.393833711,58.027620622,10.793228851],[5500,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtargt Paint_10:=[[3625,102.98,1950],[0.761472523,-0.009675371,-0.648123006,-
0.001597859],[0,0,0,0],[3900,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtargt Paint_20:=[[3800.2,103.34,2264.91],[0.783625591,-0.009612934,-0.621156006,-
0.001934292],[0,0,0,0],[3900,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtargt Paint_30:=[[3995.21,101.08,2495.65],[0.834780972,-0.009407769,-0.55049485,-
0.002764754],[0,0,0,0],[3900,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtargt Paint_40:=[[4248.45166153,100.692459062,2694.523308567],[0.834780972,-0.009407769,-
0.55049485,-0.002764754],[0,0,0,0],[3900,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtargt Paint_50:=[[4257.80166153,250.952459062,2687.433308567],[0.834780972,-0.009407769,-
0.55049485,-0.002764754],[0,0,0,0],[3900,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtargt Paint_60:=[[4001.8,252.57,2494.06],[0.834780972,-0.009407769,-0.55049485,-
0.002764754],[0,0,0,0],[3900,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtargt Paint_70:=[[3801.34,251.06,2260.82],[0.809974197,-0.009519412,-0.586383535,-
0.002351761],[0,0,0,0],[3900,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtargt Paint_80:=[[3622.3,251.43,2028.27],[0.783625591,-0.009612934,-0.621156006,-
0.001934292],[0,0,0,0],[3900,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtargt Paint_90:=[[3604.3,401.85,1997.86],[0.783625591,-0.009612934,-0.621156006,-
0.001934292],[0,0,0,0],[3900,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtargt Paint_100:=[[3807.88,400.26,2258.44],[0.783625591,-0.009612934,-0.621156006,-
0.001934292],[0,0,0,0],[3900,9E+09,9E+09,9E+09,9E+09,0]];

```

LOCAL CONST robtarget Paint_110:=[[4009.81,402.57,2485.88],[0.809974197,-0.009519412,-0.586383535,-0.002351761],[0,0,0,0],[3900,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_120:=[[4202.22,400.33,2670.6],[0.783625591,-0.009612934,-0.621156006,-0.001934292],[0,0,0,0],[3900,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_130:=[[4189,551.06,2623.74],[0.783625591,-0.009612934,-0.621156006,-0.001934292],[0,0,0,0],[3900,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_140:=[[4022.97,551.17,2470.31],[0.783625591,-0.009612934,-0.621156006,-0.001934292],[0,0,0,0],[3900,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_150:=[[3820.26,553.45,2255.59],[0.783625591,-0.009612934,-0.621156006,-0.001934292],[0,0,0,0],[3900,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_160:=[[3589,553.01,1969.13],[0.743405362,-0.128843921,-0.642016761,0.13624313],[0,0,0,0],[3900,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_170:=[[3562.9,704.15,1922.31],[0.707344444,-0.046699208,-0.70439336,0.036235564],[0,0,0,0],[3900,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_180:=[[3639.37,704.46,2017.7],[0.783625591,-0.009612934,-0.621156006,-0.001934292],[0,0,0,0],[3900,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_190:=[[3834.02,701.85,2237.86],[0.783625591,-0.009612934,-0.621156006,-0.001934292],[0,0,0,0],[3900,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_200:=[[3944.2,702.63,2344.15],[0.783625591,-0.009612934,-0.621156006,-0.001934292],[0,0,0,0],[3900,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_210:=[[4054.07,699.79,2452.36],[0.783625591,-0.009612934,-0.621156006,-0.001934292],[0,0,0,0],[3900,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_220:=[[4126.87,697.94,2515.37],[0.783625591,-0.009612934,-0.621156006,-0.001934292],[0,0,0,0],[3900,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_230:=[[4288.185422446,571.561685447,2675.575183047],[0.783625591,-0.009612934,-0.621156006,-0.001934292],[0,0,0,0],[3900,9E+09,9E+09,9E+09,9E+09,0]];
 CONST robtarget Aprox_Capo_Cierre:=[[4573.771241524,104.99200968,3349.602915356],[0.686655381,0.686657246,0.168829319,-0.168827945],[0,0,-1,-1],[4100,9E+09,9E+09,9E+09,9E+09,0]];
 CONST robtarget Cerrar_Capo1:=[[4245.053355199,104.992044882,3349.60283291],[0.68665584,0.68665677,0.168829033,-0.168828298],[0,0,-1,-1],[4100,9E+09,9E+09,9E+09,9E+09,0]];
 CONST robtarget Cerrar_Capo2:=[[4245.053124605,104.992,3245.651133019],[0.686656296,0.686656296,0.168828702,-0.168828702],[0,0,0,0],[4100,9E+09,9E+09,9E+09,9E+09,0]];
 CONST robtarget Cerrar_Capo3:=[[4301.444,104.992,3214.721],[0.686532723,0.679501837,0.181663609,-0.184250938],[0,0,0,0],[4100,9E+09,9E+09,9E+09,9E+09,0]];
 CONST robtarget Cerrar_Capo4:=[[4843.31869494,104.992,2684.668562641],[0.616000729,0.616000729,0.347193176,-0.347193176],[0,0,0,0],[4100,9E+09,9E+09,9E+09,9E+09,0]];
 CONST robtarget Cerrar_Capo5:=[[5062.074,104.992,1951.105],[0.5,0.5,0.5,-0.5],[0,0,0,0],[4100,9E+09,9E+09,9E+09,9E+09,0]];
 CONST robtarget Pos_Int2:=[[5250,650,2441.08],[0.004324243,0.866051702,-0.003050813,-0.499926437],[1,-2,1,0],[4100,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_240:=[[3536.413679937,103.556134284,1915.280094557],[0.002313989,0.998962711,-0.000817544,-0.045469549],[0,0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_250:=[[3883.68,102.89,1878.52],[0.003359889,0.998265436,-0.000786064,-0.058772552],[0,0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_260:=[[4180.12,102.62,1832.93],[0.00714906,0.995961206,-0.00054638,-0.089497867],[0,0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_270:=[[4476.31,101.42,1768.03],[0.020756118,0.992420132,0.000889678,-0.121122554],[0,0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_280:=[[4599.1,102.54,1732.52],[0.03351153,0.987972237,0.002210013,-0.15094023],[0,0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_290:=[[4573.34,252.91,1727.7],[0.049790677,0.985409166,0.004492297,-0.162694447],[0,0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_300:=[[4327.46,252.67,1796.45],[0.033276032,0.992846017,0.005040942,-0.114560378],[0,0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_310:=[[4033.05,253.33,1854.45],[0.020352403,0.996575081,0.005175706,-0.079981866],[0,0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_320:=[[3677.6,252.38,1898.97],[0.007213267,0.998883622,0.005258256,-0.046387818],[0,0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_330:=[[3659.26,402.84,1897.79],[0.011444601,0.998837144,0.005454231,-0.046514864],[0,0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_340:=[[4032.65,402.08,1847.46],[0.032098214,0.995693578,0.006768469,-0.086707505],[0,0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_350:=[[4326.16,403.93,1783.5],[0.070738934,0.989428928,0.010375895,-0.126169494],[0,0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_360:=[[4542.78,402.16,1715.82],[0.089067749,0.980401936,0.010019823,-0.175438261],[0,0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_370:=[[4495.21,550.08,1704.29],[0.064149992,0.981625743,0.005085619,-0.179638015],[0,0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_380:=[[4181.51,551.56,1798.85],[0.070906267,0.989793563,0.009629883,-0.123240699],[0,0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_390=[[3887.71,552.98,1860.83],[0.044302209,0.995365728,0.010169029,-0.084740624],[0,0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_400=[[3628.98,554.47,1896.07],[0.016856027,0.998319795,0.010043117,-0.054521522],[0,0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_410=[[3585.44,701.74,1893.75],[0.019582799,0.997304834,0.009960588,-0.070002635],[0,0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_420=[[3874.25,700.17,1847.49],[0.142015104,0.984566115,0.021780927,-0.099934315],[0,0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_430=[[4030.17,680.5,1818.96],[0.130575159,0.984073337,0.017571919,-0.119335753],[0,0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_440=[[4180.46,649.46,1788.78],[0.131723652,0.982391144,0.016076234,-0.131522141],[0,0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_450=[[4317.36,631.69,1748.76],[0.216882376,0.957749144,0.024794365,-0.187253444],[0,0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_460=[[4424.016,640.812,1701.164],[0.211790603,0.973996946,0.052891077,-0.060640116],[0,0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_470=[[4373.03,701.56,1700.5],[0.250030152,0.95614085,0.0439671,-0.146104388],[0,0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_480=[[4291.97,750.73,1703.15],[0.32333402,0.934495575,0.063597401,-0.134642127],[0,0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_490=[[4169.27,753.99,1744.23],[0.347904304,0.923391734,0.061531682,-0.15008049],[0,0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_500=[[4017.44,763.77,1776.53],[0.394882921,0.907734564,0.078449288,-0.118030288],[0,0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_510=[[3872.52,766.4,1806.09],[0.371264205,0.918659536,0.081062841,-0.107964637],[0,0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_520=[[3721.34,771.01,1830.99],[0.41740265,0.898533342,0.085994476,-0.10496576],[0,0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_530=[[3437.312313753,747.324112488,1909.699401691],[0.132921198,0.981709876,0.05014812,-0.126739259],[0,0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_540=[[3659.920787211,681.257691625,1841.66086715],[0.097191086,-0.98041757,-0.169357232,-0.025561867],[0,0,0,0],[3200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_550=[[3434.71,689.81,1912.19],[0.486043276,-0.834644026,-0.248325604,-0.073930224],[0,0,0,0],[3200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_560=[[3283.93,671.21,1981.08],[0.573856121,-0.768914918,-0.281866079,0.003242709],[0,0,0,0],[3200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_570=[[3156.82,645.15,2051.16],[0.561363776,-0.769596904,-0.302825591,0.029798961],[0,0,0,0],[3200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_580=[[3016.77,612.24,2122.21],[0.563489532,-0.769802678,-0.2988219,0.024266371],[0,0,0,0],[3200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_590=[[2672.490857459,536.773669749,2270.757088694],[0.201744323,-0.65021943,-0.72797313,0.081049635],[0,0,0,0],[3200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_600=[[2762.891334827,697.847240358,2204.522095099],[0.061192227,-0.631942848,-0.758879221,0.144934731],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_610=[[2813.56,439.01,2226.29],[0.06749067,-0.633171055,-0.758366068,0.13935685],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_620=[[2836.59,291.89,2233.59],[0.102944119,-0.636084157,-0.753761302,0.129009117],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_630=[[2846.15,143.41,2238.69],[0.110356707,-0.638362484,-0.752825054,0.116486798],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_640=[[2855.4,13.22,2238.52],[0.116512438,-0.639987979,-0.751923797,0.107008606],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_650=[[2717.97,10.94,2275.5],[0.065571313,-0.644587716,-0.759059037,0.063533127],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_660=[[2720.65,296.98,2264.77],[0.063261213,-0.641561187,-0.758256192,0.097184407],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_670=[[2720.46,539.6,2237.51],[0.04467264,-0.636590256,-0.757904983,0.135415055],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_680=[[2571.03,515.59,2270.1],[0.018327011,-0.638954725,-0.763090231,0.095363937],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_690=[[2564.72,299.2,2290.8],[0.029953282,-0.642478013,-0.763460899,0.05875594],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_700=[[2569.78,11.07,2298.76],[0.043110286,-0.643696649,-0.762856237,0.042970796],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_710=[[2415.65,10.44,2312.88],[0.022867439,-0.644384292,-0.76395898,0.024751582],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_720=[[2415.23,297.19,2305.56],[0.011729318,-0.643771108,-0.764074544,0.040140689],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_730=[[2414.85,505.13,2288.19],[0.010063097,0.642206899,0.763805558,-0.063797352],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_740:=[[2266.5,496.39,2297.52],[0.018257131,0.641958864,0.764468448,-0.056066802],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_750:=[[2263.87,297.85,2314.07],[0.003893659,-0.643446379,-0.764787011,0.032594853],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_760:=[[2264.51,5.75,2321.03],[0.010253736,-0.644066487,-0.764789289,0.01306003],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_770:=[[2112.6,6.48,2325.23],[0.003936734,-0.644094039,-0.764892586,0.008167132],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_780:=[[2113.73,300.01,2318.11],[0.015081004,0.643675179,0.764615893,-0.028589561],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_790:=[[2116.42,489.4,2302.26],[0.033419745,0.642800039,0.763949664,-0.045520783],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_800:=[[1965.79,480.28,2302.93],[0.0408361,0.642363849,0.76432592,-0.03881993],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_810:=[[1964.74,299.94,2318.71],[0.023053995,0.643093688,0.765055699,-0.024265174],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_820:=[[1963.07,7.58,2326.22],[0.002561803,0.643525219,0.765416102,-0.002630842],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_830:=[[1813.14,9.55,2323.95],[0.015215821,0.643353016,0.765381078,0.007561755],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_840:=[[1815.32,302.22,2315.84],[0.037365186,0.643373812,0.764504824,-0.014365081],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_850:=[[1817.2,474.27,2299.69],[0.057523779,0.642813588,0.763141329,-0.033121253],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_860:=[[1665.21,472.08,2292.54],[0.065332159,0.642225754,0.763108143,-0.030720545],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_870:=[[1665.88,298.67,2309.9],[0.044857414,0.642728284,0.764714933,-0.009961776],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_880:=[[1662.85,13.16,2318.24],[0.022715952,0.642652308,0.765713756,0.012823432],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_890:=[[1512.19,10.6,2309.16],[0.029231699,0.642448034,0.765552967,0.018294406],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_900:=[[1515.23,303.81,2299.7],[0.052406628,0.642877029,0.764155051,-0.005452379],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_910:=[[1514.7,465.63,2282.97],[0.072343762,0.642681219,0.762258593,-0.026250108],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_920:=[[1366.62,455.59,2269.86],[0.087693483,0.641089387,0.762265861,-0.016279067],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_930:=[[1365.46,303.8,2285.96],[0.066465551,0.641207961,0.76447653,0.003211704],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_940:=[[1364.76,8.08,2296.35],[0.042340701,0.640560951,0.76621949,0.028224585],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_950:=[[1215.55,8.02,2278.69],[0.053667643,0.639971707,0.765644312,0.036671314],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_960:=[[1214.86,300.91,2267.39],[0.078721298,0.641074408,0.763332848,0.012227945],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_970:=[[1218.911345299,608.889780439,2233.738634262],[0.097159129,0.641569509,0.760838259,-0.008591386],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];

CONST robtarget Pos_Int3:=[[4700,571.562,2675.575],[0.5,-0.5,-0.5,0.5],[0,0,0,0],[4100,9E+09,9E+09,9E+09,9E+09,0]];

CONST robtarget Pos_Int4:=[[200,604.841,1556.836],[0.340374741,0.695881078,0.553990646,0.304940854],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

CONST robtarget Pos_Int5:=[[200,604.841,2500],[0.048163096,0.773166059,0.628516432,0.069725572],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_980:=[[1224.61,690.692,2200],[0.235006131,0.752489989,0.614711234,0.02571057],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_990:=[[1136.14,519.74,2226.11],[0.155415003,0.76805817,0.619197787,0.050268516],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1000:=[[1123.37,482.58,2236.99],[0.096978336,0.771722918,0.626938611,0.044574853],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
Paint_1010:=[[1107.68,357.23,2245.54],[0.083106444,0.773187885,0.627739445,0.034885573],[0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
Paint_1020:=[[1098.37,203.66,2256.6],[0.082891903,0.772670828,0.628006293,0.041434517],[0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
Paint_1030:=[[1093.38,10.52,2259.23],[0.059208814,0.771230728,0.630634419,0.063227448],[0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
Paint_1050:=[[902.08,202.93,2218.7],[0.100271996,0.770829816,0.625962216,0.062755291],[0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
Paint_1040:=[[901.7,6.93,2220.7],[0.075051414,0.769273779,0.629072251,0.082784304],[0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
Paint_1060:=[[911.74,346.73,2208.33],[0.116563649,0.772135216,0.623175309,0.043274219],[0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
Paint_1070:=[[932.18,456.68,2196.07],[0.144324138,0.770447189,0.6198435,0.037090525],[0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
Paint_1080:=[[987.06,510.28,2194.93],[0.208175476,0.755912001,0.619574238,0.037252923],[0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
Paint_1090:=[[1110.57,579.93,2159.19],[0.221266158,0.732811867,0.640238475,0.064208642],[0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1100:=[[1039.97,545.57,2110.62],[0.098355972,0.922543705,-0.028818058,0.372033244],[0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1110:=[[1003.87,552.03,2073.4],[0.077354012,0.922114631,-0.013293628,0.378872332],[0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
Paint_1120:=[[950.56,559.44,2031.81],[0.039231783,0.944477329,0.009795864,0.326078952],[0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1130:=[[837.25,580.07,1946.51],[0.039205722,-0.952201112,-0.030501062,-0.301406104],[0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
Paint_1140:=[[855.86,667.29,1920.46],[0.286442416,0.893191137,0.208384428,0.277013112],[0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
Paint_1150:=[[785.94,609.46,1910.89],[0.297111202,0.864362377,0.321419311,0.2475727],[0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
Paint_1160:=[[722.66,504.85,1903.41],[0.280749356,0.866876463,0.326551954,0.251135059],[0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
Paint_1170:=[[666.63,351.12,1886.48],[0.349133005,0.615537019,0.676439115,0.204084409],[0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
Paint_1180:=[[636.55,203.51,1878.69],[0.354678528,0.595224158,0.676348872,0.249927083],[0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
Paint_1190:=[[620.7,53.43,1870.73],[0.356937152,0.578994985,0.67593523,0.283676298],[0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
Paint_1200:=[[588.28,52.27,1807.8],[0.40603576,0.598788941,0.544979359,0.423773836],[0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
Paint_1210:=[[597.87,204.03,1811.19],[0.431606839,0.595635851,0.563384882,0.376205988],[0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
Paint_1220:=[[619.07,353.59,1809.15],[0.450298735,0.62054909,0.545734359,0.338118154],[0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
Paint_1230:=[[643.86,457.05,1808.51],[0.461759741,0.618261312,0.568332584,0.285532774],[0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
Paint_1240:=[[633.93,478.48,1752.57],[0.502925557,0.602244976,0.47853314,0.394173702],[0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_1250:=[[642.62,353.37,1748.37],[0.408757944,0.540379563,0.539266258,0.500098765],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_1260:=[[665.39,203.98,1748.38],[0.444517696,0.57372486,0.548335274,0.415418139],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_1270:=[[655.93,47.95,1735.99],[0.429934976,0.616980212,0.516591841,0.409419351],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_1280:=[[654.81,5.71,1735.88],[0.429934976,0.616980212,0.516591841,0.409419351],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_1290:=[[636.55,10.6,1667.64],[0.452059193,0.634251605,0.49734666,0.382117375],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_1300:=[[636.57,56.17,1663.19],[0.411667152,0.638926403,0.489187861,0.427783175],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_1310:=[[642.33,204.42,1663.06],[0.424777175,0.595734415,0.523727012,0.436319695],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_1320:=[[637.83,305.27,1665.66],[0.407460729,0.57972944,0.533195142,0.462160656],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_1330:=[[624.42,349.27,1668.42],[0.44061102,0.587139034,0.500794548,0.4586224],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_1340:=[[617.54,434.06,1676.48],[0.493129167,0.579301937,0.457926278,0.459930881],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_1350:=[[629.36,500.58,1667.82],[0.510417849,0.588245125,0.440164622,0.446874029],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_1360:=[[640.51,547.68,1665.41],[0.524094821,0.595376435,0.42543277,0.435727527],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_1370:=[[625.52,502.33,1599.02],[0.523969729,0.58163846,0.428735889,0.450930108],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_1380:=[[610.33,416.18,1596.61],[0.454089342,0.592714894,0.407475659,0.525790368],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_1390:=[[608.25,352.31,1584.18],[0.465973078,0.606032685,0.4945809,0.41350116],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_1400:=[[616.91,200.45,1588.93],[0.452052138,0.618383061,0.486723181,0.419942615],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_1410:=[[614.2,52.1,1592.22],[0.518751085,0.660779058,0.389178871,0.377899661],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_1420:=[[613.48,2.73,1592.64],[0.501773597,0.677125887,0.359985681,0.400167589],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_1430:=[[586.27,2.07,1539.29],[0.445248503,0.718320541,0.268523291,0.462238697],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_1440:=[[587.04,53.55,1539.65],[0.483840955,0.692183768,0.330107239,0.421673775],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_1450:=[[591.47,203.83,1538.89],[0.478861088,0.651433264,0.389946781,0.440758742],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_1460:=[[602.06,353.39,1541.01],[0.440800225,0.649157519,0.357303043,0.506580905],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget
 Paint_1470:=[[634.321778373,604.841377838,1556.835611193],[0.482922733,0.60571534,0.474857575,0.41761806],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

CONST robtarget Cerrar_Capo7:=[[5506.850791837,104.99216803,1951.104759241],[0.499998953,0.500001004,0.500000705,-0.499999339],[1,-2,1,0],[4100,9E+09,9E+09,9E+09,9E+09,0]];

```

CONST robtarget
Cerrar_Capo6:=[[5106.460176544,104.992114584,1951.104819345],[0.499999324,0.500000681,0.500000416,-0.499999579],[1,-
2,1,0],[4100,9E+09,9E+09,9E+09,0]];
CONST robtarget Pos_Int6:=[[4000,102.98,1950],[0.761472523,-0.009675371,-0.648123006,-
0.001597859],[0,0,0,0],[3900,9E+09,9E+09,9E+09,0]];
CONST robtarget Limpieza_1:=[[-500,2350,1500],[0,1,0,0],[0,-2,1,0],[300,9E+09,9E+09,9E+09,0]];
CONST robtarget Limpieza_2:=[[-500,2350,1360],[0,1,0,0],[0,-2,1,0],[300,9E+09,9E+09,9E+09,0]];
CONST jointtarget Pos_Int7:=[[0,0,0,0,-45,0],[3435,9E+09,9E+09,9E+09,0]];
CONST robtarget Pos_Sal_Cap:=[[4750,251.886,2707.119],[0.353553391,-0.612372436,-
0.612372436,0.353553391],[0,0,0,0],[3900,9E+09,9E+09,9E+09,0]];
CONST robtarget Mantenimiento_2:=[[-1000,2200,1950],[0.707106781,0,-0.707106781,0],[0,-2,-
1,0],[0,9E+09,9E+09,9E+09,0]];
CONST robtarget Mantenimiento_3:=[[-1425,2200,1950],[0.707106781,0,-0.707106781,0],[-1,-2,-
1,0],[0,9E+09,9E+09,9E+09,0]];
CONST jointtarget Mantenimiento_1:=[[-2.485981604,94.365340328,0.85943566,-104.93583671,30.957019621,-
77.611726407],[0,9E+09,9E+09,9E+09,0]];
CONST jointtarget Pos_Mant_Int_1:=[[60,-30,30,0,45,0],[500,9E+09,9E+09,0,9E+09,0]];
CONST jointtarget Pos_Mant_Int_2:=[[45,50,20,-45,45,0],[500,9E+09,9E+09,0,9E+09,0]];
LOCAL CONST robtarget Paint_1480:=[[-500,1800,1450],[0,1,0,0],[0,-2,1,0],[300,9E+09,9E+09,0,9E+09,0]];
LOCAL CONST robtarget Paint_1490:=[[-500,1800,1185],[0,1,0,0],[0,-2,1,0],[300,9E+09,9E+09,0,9E+09,0]];
LOCAL CONST robtarget Paint_1500:=[[-500,1800,1450],[0,1,0,0],[0,-2,1,0],[300,9E+09,9E+09,0,9E+09,0]];
CONST robtarget Pos_Int_Limp:=[[279.256,1080.862,3217.652],[0,1,0,0],[0,0,0,0],[300,9E+09,9E+09,0,9E+09,0]];

```

!!!!!!!!!!!!!!!!!!!! DEFINICIÓN DE VARIABLES !!!!!!!!!!!!!!!!!!!!!

```

VAR num Color;
VAR intnum Puerta_Abierta;
VAR intnum Limp_Atomo;
VAR intnum Paro_Robot;
VAR intnum Mant_Atomo;
VAR num Lectura_Color;
VAR iodev infile;
VAR num Contador;
VAR num Secuencia_Colores{1000};
VAR string date;
VAR string N_Proc;
VAR string Limpieza;

```

!!!!!!!!!!!!!!!!!!!! PROCESOS DE PINTADO !!!!!!!!!!!!!!!!!!!!!

```

PROC main()
Reset Inic_Pint_Inf;
Reset Abrir_Capo;
Reset Cerrar_Capo;
Reset Continuar3;
Reset Stop_Conveyor;
Reset Finalizado;
Reset Cambio_Color;
Reset Pos_Parada;
Reset Paro_Resto;
Reset Open_TMant;
Reset Close_TMant;
SetAO Señal_Luminosa,3;
Reset Clock_ON;
Lectura_Colores;
! Eliminar los avisos de trayectoria de fallo de esquina
CornerPathWarning FALSE;

! Defición de los parámetros para la rutina TRAP de paro si se abre la puerta
IDelete Puerta_Abierta;
CONNECT Puerta_Abierta WITH Paro_Puerta;
ISignalDI P_Puerta,1,Puerta_Abierta;

! Definición de los parámetros para la rutina TRAP de limpieza si se requiere la limpieza del atomizador
IDelete Limp_Atomo;
CONNECT Limp_Atomo WITH Limpieza_Atomizador;
ISignalDI Limpieza_Atomo,1,Limp_Atomo;

! Definición de los parámetros para la rutina TRAP de paro del robot en caso de que otro de los robots lo solicite
IDelete Paro_Robot;
CONNECT Paro_Robot WITH Parada;

```

```
ISignalDI Paro_Propio,1,Paro_Robot;
```

```
! Definicion de los parametros para la rutina TRAP de mantenimiento si se requiere realizar mantenimiento al atomizador
```

```
IDelete Mant_Atom;
```

```
CONNECT Mant_Atom WITH Mantenimiento_Atomizador;
```

```
ISignalDI Mantenimiento,1,Mant_Atom;
```

```
Color:=Secuencia_Colores{1};
```

```
Set Cambio_Color;
```

```
ActUnit CNV1;
```

```
FOR i1 FROM 1 TO Contador-1 DO
```

```
Principal;
```

```
Reset Cambio_Color;
```

```
N_Proc:="main";
```

```
WaitRob\ZeroSpeed;
```

```
IF Finalizado=1 AND Color<>Secuencia_Colores{i1+1} AND Secuencia_Colores{i1+1}<>0 THEN
```

```
  Limpieza:="Limp1";
```

```
  Limpieza_Atom_Color;
```

```
  Color:=Secuencia_Colores{i1+1};
```

```
  Set Cambio_Color;
```

```
  ! Purga de color para preparar el atomizador
```

```
  PaintL Paint_1480,v2000,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
```

```
  SetBrush Color\Z:=1300;
```

```
  PaintL Paint_1490,v800,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
```

```
  WaitTime 2;
```

```
  SetBrush 1\Z:=1300;
```

```
  PaintL Paint_1500,v800,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
```

```
  N_Proc:="main";
```

```
  MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
```

```
ELSEIF Finalizado=1 AND Color<>Secuencia_Colores{i1+1} AND Secuencia_Colores{i1+1}=0 THEN
```

```
  Limpieza:="Limp1";
```

```
  Limpieza_Atom_Color;
```

```
  N_Proc:="main";
```

```
  MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
```

```
ELSE
```

```
  Color:=Secuencia_Colores{i1+1};
```

```
  Set Cambio_Color;
```

```
  MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
```

```
ENDIF
```

```
ENDFOR
```

```
DeactUnit CNV1;
```

```
! Stop;
```

```
ENDPROC
```

```
PROC Principal()
```

```
  Reset Inic_Pint_Inf;
```

```
  Reset Continuar3;
```

```
  N_Proc:="Principal";
```

```
  MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
```

```
  WaitWObj wobj_cnv1;
```

```
  Set Clock_ON;
```

```
  Reset Finalizado;
```

```
  Abertura_Capo;
```

```
  Reset Clock_ON;
```

```
  N_Proc:="Principal";
```

```
  MoveAbsJ Pos_Int1,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
```

```
  WaitDI Continuar,1;
```

```
  Pint_Capo_Inf;
```

```
  N_Proc:="Principal";
```

```
  MoveL Pos_Int3,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
```

```
  WaitTime 1.5;
```

```
  MoveL Aprox_Capo_Cierre,v2500,z50,tool0\WObj:=wobj_cnv1;
```

```
  WaitDI Continuar2,1;
```

```
  Cierre_Capo;
```

```
  N_Proc:="Principal";
```

```
  MoveL Pos_Int2,v2500,z50,tool0\WObj:=wobj_cnv1;
```

```
  WaitTime 3.5;
```

```
  Pint_Capo_Sup;
```

```
  Pint_Barra;
```

```
  N_Proc:="Principal";
```

```
  WaitTime 2;
```

```

Pint_Techo;
Pint_Malet_Sup;
N_Proc:="Principal";
MoveLDO Pos_Int4,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1,Continuar3,1;
Set Finalizado;
MoveLDO Pos_Int5,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1,Continuar3,0;
MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
DropWObj wobj_cnv1;

```

ENDPROC

PROC Lectura_Colores()

```
VAR string Num_CE;
```

```
! Lectura del archivo con la cadena de números referentes a los colores
```

```
date:=CDate();
```

```
Open date+"_Lista_Colores.txt",infile\Read;
```

```
Lectura_Color:=ReadNum(infile\Delim:=" ");
```

```
Contador:=1;
```

```
Num_CE:=NumToStr(Contador,0);
```

```
! Comprobación de que el archivo no está vacío o contine codigos de color incorrectos.
```

```
IF Lectura_Color>EOF_NUM THEN
```

```
TPWrite "El archivo que contiene la Secuencia de colores está vacío/incorrecto, revíselo.";
```

```
ErrWrite "Color error","El archivo que contiene la Secuencia de colores está vacío/incorrecto, revíselo.";
```

```
Break;
```

```
! Stop;
```

```
ELSEIF Lectura_Color>16 OR Lectura_Color<2 THEN
```

```
TPWrite "El identificador de color "+Num_CE+" es incorrecto, revíselo.";
```

```
ErrWrite "Color error","El identificador de color "+Num_CE+" es incorrecto, revíselo."\\RL2:="Identificador superior a 16 o inferior a 2.";
```

```
Break;
```

```
ENDIF
```

```
! Lectura de la secuencia de colores completa.
```

```
Secuencia_Colores{Contador}:=Lectura_Color;
```

```
WHILE Lectura_Color<EOF_NUM DO
```

```
Contador:=Contador+1;
```

```
Num_CE:=NumToStr(Contador,0);
```

```
Lectura_Color:=ReadNum(infile\Delim:=" ");
```

```
! Comprobación de que el archivo no continen ningún codigo de color incorrecto.
```

```
IF Lectura_Color<EOF_NUM THEN
```

```
IF Lectura_Color>16 OR Lectura_Color<2 THEN
```

```
TPWrite "El identificador de color "+Num_CE+" es incorrecto, revíselo.";
```

```
ErrWrite "Color error","El identificador de color "+Num_CE+" es incorrecto, revíselo."\\RL2:="Identificador superior a 16 o inferior a 2.";
```

```
Break;
```

```
ENDIF
```

```
ENDIF
```

```
Secuencia_Colores{Contador}:=Lectura_Color;
```

```
ENDWHILE
```

```
Secuencia_Colores{Contador}:=0;
```

ENDPROC

PROC Limpieza_Atomo_Color()

```
N_Proc:="Limpieza_Atomo_Color";
```

```
MoveL Limpieza_1,v1500,fine,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
```

```
MoveL Limpieza_2,v100,fine,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
```

```
IF Limpieza="Limp1" THEN
```

```
Set RUN_Limp;
```

```
WaitTime 5;
```

```
MoveL Limpieza_1,v100,fine,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
```

```
ELSEIF Limpieza="Limp2" THEN
```

```
Set RUN_Limp;
```

```
WaitTime 5;
```

```
MoveL Limpieza_1,v100,fine,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
```

```
! Purga de color para preparar el atomizador
```

```
PaintL Paint_1480,v2000,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
```

```
SetBrush Color\Z:=1300;
```

```
PaintL Paint_1490,v800,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
```

```
WaitTime 2;
```

```
SetBrush 1\Z:=1300;
```

```
PaintL Paint_1500,v800,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
```

```
ENDIF
```

```
Reset RUN_Limp;  
ENDPROC
```

```
PROC Mantenimiento_Atom()
```

```
MoveAbsJ Pos_Mant_Int_1,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;  
MoveAbsJ Pos_Mant_Int_2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;  
MoveAbsJ Mantenimiento_1,v2500,z10,RB1000_WSC_70_700_0\WObj:=WO_In_Int;  
MoveLDO Mantenimiento_2,v500,fine,RB1000_WSC_70_700_200\WObj:=WO_In_Int,Open_TMant,1;  
WaitDI Ok_TMant,1;  
VelSet 100,100;  
MoveL Mantenimiento_3,v100,fine,RB1000_WSC_70_700_0\WObj:=WO_In_Int;  
SetAO Señal_Luminosa,2;  
WaitDI Mantenimiento,0;  
SetAO Señal_Luminosa,1;  
Reset Open_TMant;  
MoveLDO Mantenimiento_2,v100,fine,RB1000_WSC_70_700_200\WObj:=WO_In_Int,Close_TMant,1;  
VelSet 100,MaxRobSpeed();  
MoveAbsJ Mantenimiento_1,v2500,z10,RB1000_WSC_70_700_0\WObj:=WO_In_Int;  
MoveAbsJ Pos_Mant_Int_2,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;  
MoveAbsJ Pos_Mant_Int_1,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;  
Reset Close_TMant;
```

```
ENDPROC
```

```
PROC Abertura_Capo()
```

```
N_Proc:="Abertura_Capo1";  
MoveL Abrir_Capo1,v2500,z50,tool0\WObj:=wobj_cnv1;  
N_Proc:="Abertura_Capo2";  
MoveL Abrir_Capo2,v2500,z1,tool0\WObj:=wobj_cnv1;  
N_Proc:="Abertura_Capo3";  
MoveLDO Abrir_Capo3,v50,z1,tool0\WObj:=wobj_cnv1,Abrir_Capo,1;  
MoveLDO Abrir_Capo4,v300,z1,tool0\WObj:=wobj_cnv1,Abrir_Capo,0;  
MoveC Abrir_capo5,Abrir_Capo6,v300,z1,tool0\WObj:=wobj_cnv1;  
N_Proc:="Abertura_Capo1";  
MoveLDO Abrir_Capo7,v2500,z50,tool0\WObj:=wobj_cnv1,Inic_Pint_Inf,1;
```

```
ENDPROC
```

```
PROC Pint_Capo_Inf()
```

```
N_Proc:="Pint_Capo_Inf";  
MoveL Pos_Int6,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;  
PaintL Paint_10,v2000,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;  
SetBrush Color\Z:=2026.49;  
PaintL Paint_20,v1500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;  
PaintL Paint_30,v1500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;  
PaintL Paint_40,v1500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;  
PaintL Paint_50,v1500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;  
PaintL Paint_60,v1500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;  
PaintL Paint_70,v1500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;  
PaintL Paint_80,v1500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;  
PaintL Paint_90,v1500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;  
PaintL Paint_100,v1500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;  
PaintL Paint_110,v1500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;  
PaintL Paint_120,v1500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;  
PaintL Paint_130,v1500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;  
PaintL Paint_140,v1500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;  
PaintL Paint_150,v1500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;  
PaintL Paint_160,v1500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;  
PaintL Paint_170,v1500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;  
PaintL Paint_180,v1500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;  
PaintL Paint_190,v1500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;  
PaintL Paint_200,v1500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;  
PaintL Paint_210,v1500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;  
PaintL Paint_220,v1500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;  
SetBrush 1\X:=4195.16;  
PaintL Paint_230,v1500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
```

```
ENDPROC
```

```
PROC Cierre_Capo()
```

```
N_Proc:="Cierre_Capo";  
MoveL Cerrar_Capo1,v2500,z50,tool0\WObj:=wobj_cnv1;  
MoveLDO Cerrar_Capo2,v50,z1,tool0\WObj:=wobj_cnv1,Cerrar_Capo,1;  
MoveLDO Cerrar_Capo3,v300,z1,tool0\WObj:=wobj_cnv1,Cerrar_Capo,0;
```



```
MoveC Cerrar_Capo4,Cerrar_Capo5,v300,z1,tool0\WObj:=wobj_cnv1;
MoveLDO Cerrar_Capo6,v2500,z5,tool0\WObj:=wobj_cnv1,Continuar3,1;
MoveLDO Cerrar_Capo7,v2500,z50,tool0\WObj:=wobj_cnv1,Continuar3,0;
ENDPROC
```

```
PROC Pint_Capo_Sup()
```

```
N_Proc:="Pint_Capo_Sup";
PaintL Paint_240,v2000,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
SetBrush Color\X:=3685.58;
PaintL Paint_250,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_260,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_270,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_280,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_290,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_300,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_310,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_320,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_330,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_340,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_350,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_360,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_370,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_380,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_390,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_400,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_410,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_420,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_430,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_440,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_450,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_460,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_470,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_480,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_490,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_500,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_510,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_520,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
SetBrush 1\X:=3581.4;
PaintL Paint_530,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
```

```
ENDPROC
```

```
PROC Pint_Barra()
```

```
N_Proc:="Pint_Barra";
PaintL Paint_540,v2000,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
SetBrush Color\X:=3516.87;
PaintL Paint_550,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_560,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_570,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_580,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
SetBrush 1\X:=2807.51;
PaintL Paint_590,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
```

```
ENDPROC
```

```
PROC Pint_Techo()
```

```
N_Proc:="Pint_Techo";
PaintL Paint_600,v2000,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
SetBrush Color\Y:=551.14;
PaintL Paint_610,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_620,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_630,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_640,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_650,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_660,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_670,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_680,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_690,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_700,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_710,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_720,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_730,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_740,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
```



```

PaintL Paint_1390,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_1400,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_1410,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_1420,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_1430,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_1440,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_1450,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_1460,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
SetBrush 1\Y:=456.35;
PaintL Paint_1470,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
ENDPROC

```

!!!!!!!!!!!!!!!!!!!!!! RUTINAS TRAP !!!!!!!!!!!!!!!!!!!!!!!

```

TRAP Paro_Puerta
VAR robtarget Punt_Paro;
VAR jointtarget jointPunt_Paro;
StopMove\Quick;
Set Paro_Resto;
Set Stop_Conveyor;
SetAO Señal_Luminosa,1;
StorePath;
Punt_Paro:=CRobT(\Tool:=RB1000_WSC_70_700_200\WObj:=WO_In_Int);
jointPunt_Paro:=CJointT(\TaskRef:=T_ROB1Id);
TEST N_Proc
CASE "main":
  MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  SetAO Señal_Luminosa,2;
  WaitDI P_Puerta,0;
  Reset Paro_Resto;
  SetAO Señal_Luminosa,1;
CASE "Principal":
  MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  SetAO Señal_Luminosa,2;
  WaitDI P_Puerta,0;
  Reset Paro_Resto;
  SetAO Señal_Luminosa,1;
CASE "Limpieza_Atóm_Color":
  MoveL Limpieza_1,v1500,fine,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  SetAO Señal_Luminosa,2;
  WaitDI P_Puerta,0;
  Reset Paro_Resto;
  SetAO Señal_Luminosa,1;
  MoveL Limpieza_1,v1500,fine,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
CASE "Abertura_Capo1":
  MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  SetAO Señal_Luminosa,2;
  WaitDI P_Puerta,0;
  Reset Paro_Resto;
  SetAO Señal_Luminosa,1;
CASE "Abertura_Capo2":
  MoveL Abrir_Capo1,v2500,z50,tool0\WObj:=wobj_cnv1;
  MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  SetAO Señal_Luminosa,2;
  WaitDI P_Puerta,0;
  Reset Paro_Resto;
  SetAO Señal_Luminosa,1;
  MoveL Abrir_Capo1,v2500,z50,tool0\WObj:=wobj_cnv1;
CASE "Abertura_Capo3":
  MoveL offs(Punt_Paro,150,0,-40),v1500,fine,RB1000_WSC_70_700_200\WObj:=WO_In_Int;
  MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  SetAO Señal_Luminosa,2;
  WaitDI P_Puerta,0;
  Reset Paro_Resto;
  SetAO Señal_Luminosa,1;
  MoveL offs(Punt_Paro,150,0,-40),v1500,fine,RB1000_WSC_70_700_200\WObj:=WO_In_Int;
CASE "Pint_Capo_Inf":
  MoveL Pos_Sal_Cap,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
  MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  SetAO Señal_Luminosa,2;

```

```

WaitDI P_Puerta,0;
Reset Paro_Resto;
SetAO Señal_Luminosa,1;
MoveL Pos_Sal_Cap,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
CASE "Cierre_Capo":
MoveL offs(Punt_Paro,0,0,20),v1500,fine,RB1000_WSC_70_700_200\WObj:=WO_In_Int;
MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI P_Puerta,0;
Reset Paro_Resto;
SetAO Señal_Luminosa,1;
MoveL offs(Punt_Paro,0,0,20),v1500,fine,RB1000_WSC_70_700_200\WObj:=WO_In_Int;
CASE "Pint_Capo_Sup":
MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI P_Puerta,0;
Reset Paro_Resto;
SetAO Señal_Luminosa,1;
CASE "Pint_Barra":
MoveL offs(Punt_Paro,0,0,750),v1500,fine,RB1000_WSC_70_700_200\WObj:=WO_In_Int;
MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI P_Puerta,0;
Reset Paro_Resto;
SetAO Señal_Luminosa,1;
MoveL offs(Punt_Paro,0,0,750),v1500,fine,RB1000_WSC_70_700_200\WObj:=WO_In_Int;
CASE "Pint_Techo":
MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI P_Puerta,0;
Reset Paro_Resto;
SetAO Señal_Luminosa,1;
CASE "Pint_Malet_Sup1":
MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI P_Puerta,0;
Reset Paro_Resto;
SetAO Señal_Luminosa,1;
CASE "Pint_Malet_Sup2":
MoveL Pos_Int5,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI P_Puerta,0;
Reset Paro_Resto;
SetAO Señal_Luminosa,1;
MoveL Pos_Int5,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
DEFAULT:
MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI P_Puerta,0;
Reset Paro_Resto;
SetAO Señal_Luminosa,1;
ENDTEST
MoveAbsJ jointPunt_Paro,v1500,fine,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Set Pos_Parada;
WaitDI OK_PosP,1;
RestoPath;
StartMove;
WaitTime 0.25;
Reset Stop_Conveyor;
Reset Pos_Parada;
SetAO Señal_Luminosa,3;
ENDTRAP

TRAP Limpieza_Atomicador
VAR robtarget Punt_Paro;
VAR jointtarget jointPunt_Paro;
VAR string N_PROC2;
StopMove\Quick;
Set Paro_Resto;
Set Stop_Conveyor;

```

```

SetAO Señal_Luminosa,1;
StorePath;
N_PROC2:=N_Proc;
Limpieza:="Limp2";
Punt_Paro:=CRobT(\Tool:=RB1000_WSC_70_700_200\WObj:=WO_In_Int);
jointPunt_Paro:=CJointT(\TaskRef:=T_ROB1Id);
TEST N_Proc
CASE "main":
  MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  Limpieza_Atom_Color;
  MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  SetAO Señal_Luminosa,2;
  WaitDI Limpieza_Atom,0;
  Reset Paro_Resto;
  SetAO Señal_Luminosa,1;
CASE "Principal":
  MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  Limpieza_Atom_Color;
  MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  SetAO Señal_Luminosa,2;
  WaitDI Limpieza_Atom,0;
  Reset Paro_Resto;
  SetAO Señal_Luminosa,1;
CASE "Limpieza_Atom_Color":
  SetAO Señal_Luminosa,2;
  WaitDI Limpieza_Atom,0;
  Reset Paro_Resto;
  SetAO Señal_Luminosa,1;
CASE "Abertura_Capo1":
  MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  Limpieza_Atom_Color;
  MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  SetAO Señal_Luminosa,2;
  WaitDI Limpieza_Atom,0;
  Reset Paro_Resto;
  SetAO Señal_Luminosa,1;
CASE "Abertura_Capo2":
  MoveL Abrir_Capo1,v2500,z50,tool0\WObj:=wobj_cnv1;
  MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  Limpieza_Atom_Color;
  MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  SetAO Señal_Luminosa,2;
  WaitDI Limpieza_Atom,0;
  Reset Paro_Resto;
  SetAO Señal_Luminosa,1;
  MoveL Abrir_Capo1,v2500,z50,tool0\WObj:=wobj_cnv1;
CASE "Abertura_Capo3":
  MoveL offs(Punt_Paro,150,0,-40),v1500,fine,RB1000_WSC_70_700_200\WObj:=WO_In_Int;
  MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  Limpieza_Atom_Color;
  MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  SetAO Señal_Luminosa,2;
  WaitDI Limpieza_Atom,0;
  Reset Paro_Resto;
  SetAO Señal_Luminosa,1;
  MoveL offs(Punt_Paro,150,0,-40),v1500,fine,RB1000_WSC_70_700_200\WObj:=WO_In_Int;
CASE "Pint_Capo_Inf":
  MoveL Pos_Sal_Cap,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
  MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  Limpieza_Atom_Color;
  MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  SetAO Señal_Luminosa,2;
  WaitDI Limpieza_Atom,0;
  Reset Paro_Resto;
  SetAO Señal_Luminosa,1;
  MoveL Pos_Sal_Cap,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
CASE "Cierre_Capo":
  MoveL offs(Punt_Paro,0,0,20),v1500,fine,RB1000_WSC_70_700_200\WObj:=WO_In_Int;
  MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  Limpieza_Atom_Color;
  MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;

```

```

SetAO Señal_Luminosa,2;
WaitDI Limpieza_Atom,0;
Reset Paro_Resto;
SetAO Señal_Luminosa,1;
MoveL offs(Punt_Paro,0,0,20),v1500,fine,RB1000_WSC_70_700_200\WObj:=WO_In_Int;
CASE "Pint_Capo_Sup":
MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atom_Color;
MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Limpieza_Atom,0;
Reset Paro_Resto;
SetAO Señal_Luminosa,1;
CASE "Pint_Barra":
MoveL offs(Punt_Paro,0,0,750),v1500,fine,RB1000_WSC_70_700_200\WObj:=WO_In_Int;
MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atom_Color;
MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Limpieza_Atom,0;
Reset Paro_Resto;
SetAO Señal_Luminosa,1;
MoveL offs(Punt_Paro,0,0,750),v1500,fine,RB1000_WSC_70_700_200\WObj:=WO_In_Int;
CASE "Pint_Techo":
MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atom_Color;
MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Limpieza_Atom,0;
Reset Paro_Resto;
SetAO Señal_Luminosa,1;
CASE "Pint_Malet_Sup1":
MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atom_Color;
MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Limpieza_Atom,0;
Reset Paro_Resto;
SetAO Señal_Luminosa,1;
CASE "Pint_Malet_Sup2":
MoveL Pos_Int5,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atom_Color;
MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Limpieza_Atom,0;
Reset Paro_Resto;
SetAO Señal_Luminosa,1;
MoveL Pos_Int5,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
DEFAULT:
MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atom_Color;
MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Limpieza_Atom,0;
Reset Paro_Resto;
SetAO Señal_Luminosa,1;
ENDTEST
MoveAbsJ jointPunt_Paro,v1500,fine,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
N_Proc:=N_PROC2;
Set Pos_Parada;
WaitDI OK_PosP,1;
RestoPath;
StartMove;
WaitTime 0.25;
Reset Stop_Conveyor;
Reset Pos_Parada;
SetAO Señal_Luminosa,3;
ENDTRAP
TRAP Parada

```

```

VAR robtarget Punt_Paro;
VAR jointtarget jointPunt_Paro;
SetAO Señal_Luminosa,1;
StopMove\Quick;
StorePath;
Punt_Paro:=CRobT(\Tool:=RB1000_WSC_70_700_200\WObj:=WO_In_Int);
jointPunt_Paro:=CJointT(\TaskRef:=T_ROB1Id);
TEST N_Proc
CASE "main":
  MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  SetAO Señal_Luminosa,2;
  WaitDI Paro_Propio,0;
  SetAO Señal_Luminosa,1;
CASE "Principal":
  MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  SetAO Señal_Luminosa,2;
  WaitDI Paro_Propio,0;
  SetAO Señal_Luminosa,1;
CASE "Limpieza_Atóm_Color":
  MoveL Limpieza_1,v1500,fine,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  SetAO Señal_Luminosa,2;
  WaitDI Paro_Propio,0;
  SetAO Señal_Luminosa,1;
  MoveL Limpieza_1,v1500,fine,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
CASE "Abertura_Capo1":
  MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  SetAO Señal_Luminosa,2;
  WaitDI Paro_Propio,0;
  SetAO Señal_Luminosa,1;
CASE "Abertura_Capo2":
  MoveL Abrir_Capo1,v2500,z50,tool0\WObj:=wobj_cnv1;
  MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  SetAO Señal_Luminosa,2;
  WaitDI Paro_Propio,0;
  SetAO Señal_Luminosa,1;
  MoveL Abrir_Capo1,v2500,z50,tool0\WObj:=wobj_cnv1;
CASE "Abertura_Capo3":
  MoveL offs(Punt_Paro,150,0,-40),v1500,fine,RB1000_WSC_70_700_200\WObj:=WO_In_Int;
  MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  SetAO Señal_Luminosa,2;
  WaitDI Paro_Propio,0;
  SetAO Señal_Luminosa,1;
  MoveL offs(Punt_Paro,150,0,-40),v1500,fine,RB1000_WSC_70_700_200\WObj:=WO_In_Int;
CASE "Pint_Capo_Inf":
  MoveL Pos_Sal_Cap,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
  MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  SetAO Señal_Luminosa,2;
  WaitDI Paro_Propio,0;
  SetAO Señal_Luminosa,1;
  MoveL Pos_Sal_Cap,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
CASE "Cierre_Capo":
  MoveL offs(Punt_Paro,0,0,20),v1500,fine,RB1000_WSC_70_700_200\WObj:=WO_In_Int;
  MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  SetAO Señal_Luminosa,2;
  WaitDI Paro_Propio,0;
  SetAO Señal_Luminosa,1;
  MoveL offs(Punt_Paro,0,0,20),v1500,fine,RB1000_WSC_70_700_200\WObj:=WO_In_Int;
CASE "Pint_Capo_Sup":
  MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  SetAO Señal_Luminosa,2;
  WaitDI Paro_Propio,0;
  SetAO Señal_Luminosa,1;
CASE "Pint_Barra":
  MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  SetAO Señal_Luminosa,2;
  WaitDI Paro_Propio,0;
  SetAO Señal_Luminosa,1;
CASE "Pint_Techo":
  MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  SetAO Señal_Luminosa,2;

```

```

WaitDI Paro_Propio,0;
SetAO Señal_Luminosa,1;
CASE "Pint_Malet_Sup1":
MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Paro_Propio,0;
SetAO Señal_Luminosa,1;
CASE "Pint_Malet_Sup2":
MoveL Pos_Int5,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Paro_Propio,0;
SetAO Señal_Luminosa,1;
MoveL Pos_Int5,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
DEFAULT:
SetAO Señal_Luminosa,2;
WaitDI Paro_Propio,0;
SetAO Señal_Luminosa,1;
ENDTEST
MoveAbsJ jointPunt_Paro,v1500,fine,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Set Pos_Parada;
WaitDI OK_PosP,1;
RestoPath;
StartMove;
WaitTime 0.25;
Reset Pos_Parada;
SetAO Señal_Luminosa,3;
ENDTRAP

TRAP Mantenimiento_Atomicador
VAR robtarget Punt_Paro;
VAR jointtarget jointPunt_Paro;
VAR string N_PROG2;
StopMove\Quick;
Set Paro_Resto;
Set Stop_Conveyor;
SetAO Señal_Luminosa,1;
StorePath;
N_PROG2:=N_Proc;
Limpieza:="Limp1";
Punt_Paro:=CRobT(\Tool:=RB1000_WSC_70_700_200\WObj:=WO_In_Int);
jointPunt_Paro:=CJointT(\TaskRef:=T_ROB1Id);
TEST N_Proc
CASE "main":
MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atomicador_Color;
MoveL Pos_Int_Limp,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Mantenimiento_Atomicador;
Reset Paro_Resto;
MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
CASE "Principal":
MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atomicador_Color;
MoveL Pos_Int_Limp,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Mantenimiento_Atomicador;
Reset Paro_Resto;
MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
CASE "Limpieza_Atomicador_Color":
MoveL Limpieza_1,v1500,fine,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atomicador_Color;
MoveL Pos_Int_Limp,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Mantenimiento_Atomicador;
Reset Paro_Resto;
MoveL Limpieza_1,v1500,fine,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
CASE "Abertura_Cap1":
MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atomicador_Color;
MoveL Pos_Int_Limp,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Mantenimiento_Atomicador;
Reset Paro_Resto;
MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;

```


CASE "Abertura_Capo2":
 MoveL Abrir_Capo1,v2500,z50,tool0\WObj:=wobj_cnv1;
 MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Limpieza_Atom_Color;
 MoveL Pos_Int_Limp,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Mantenimiento_Atom;
 Reset Paro_Resto;
 MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveL Abrir_Capo1,v2500,z50,tool0\WObj:=wobj_cnv1;

CASE "Abertura_Capo3":
 MoveL offs(Punt_Paro,150,0,-40),v1500,fine,RB1000_WSC_70_700_200\WObj:=WO_In_Int;
 MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Limpieza_Atom_Color;
 MoveL Pos_Int_Limp,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Mantenimiento_Atom;
 Reset Paro_Resto;
 MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveL offs(Punt_Paro,150,0,-40),v1500,fine,RB1000_WSC_70_700_200\WObj:=WO_In_Int;

CASE "Pint_Capo_Inf":
 MoveL Pos_Sal_Cap,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
 MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Limpieza_Atom_Color;
 MoveL Pos_Int_Limp,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Mantenimiento_Atom;
 Reset Paro_Resto;
 MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveL Pos_Sal_Cap,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;

CASE "Cierre_Capo":
 MoveL offs(Punt_Paro,0,0,20),v1500,fine,RB1000_WSC_70_700_200\WObj:=WO_In_Int;
 MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Limpieza_Atom_Color;
 MoveL Pos_Int_Limp,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Mantenimiento_Atom;
 Reset Paro_Resto;
 MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveL offs(Punt_Paro,0,0,20),v1500,fine,RB1000_WSC_70_700_200\WObj:=WO_In_Int;

CASE "Pint_Capo_Sup":
 MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Limpieza_Atom_Color;
 MoveL Pos_Int_Limp,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Mantenimiento_Atom;
 Reset Paro_Resto;
 MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;

CASE "Pint_Barra":
 MoveL offs(Punt_Paro,0,0,750),v1500,fine,RB1000_WSC_70_700_200\WObj:=WO_In_Int;
 MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Limpieza_Atom_Color;
 MoveL Pos_Int_Limp,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Mantenimiento_Atom;
 Reset Paro_Resto;
 MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveL offs(Punt_Paro,0,0,750),v1500,fine,RB1000_WSC_70_700_200\WObj:=WO_In_Int;

CASE "Pint_Techo":
 MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Limpieza_Atom_Color;
 MoveL Pos_Int_Limp,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Mantenimiento_Atom;
 Reset Paro_Resto;
 MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;

CASE "Pint_Malet_Sup1":
 MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Limpieza_Atom_Color;
 MoveL Pos_Int_Limp,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Mantenimiento_Atom;
 Reset Paro_Resto;
 MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;

CASE "Pint_Malet_Sup2":
 MoveL Pos_Int5,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
 MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Limpieza_Atom_Color;
 MoveL Pos_Int_Limp,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;

```

Mantenimiento_Atom;
Reset Paro_Resto;
MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
MoveL Pos_Int5,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
DEFAULT:
MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atom_Color;
MoveL Pos_Int_Limp,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Mantenimiento_Atom;
Reset Paro_Resto;
MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
ENDTEST
MoveAbsJ jointPunt_Paro,v1500,fine,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
N_Proc:=N_PROC2;
Set Pos_Parada;
WaitDI OK_PosP,1;
RestoPath;
StartMove;
WaitTime 0.25;
Reset Stop_Conveyor;
Reset Pos_Parada;
SetAO Señal_Luminosa,3;
ENDTRAP
ENDMODULE

```

8.1.4 Controlador virtual Cont Sup Der

```

MODULE Module1
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! Módulo: Module1
!
! Descripción: Procesos para la aplicación de la capa base sobre la carrocería empleando el robot superior derecho
!
! Autor: Arturo López López
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

!!!!!!!!!!!!!!!!!!!!!!!!!!!! DEFINICIÓN DE OBJETIVOS !!!!!!!!!!!!!!!!!!!!!!!

CONST jointtarget Pos_Inicial:=[[0,0,0,0,45,0],[6500,9E+09,9E+09,9E+09,9E+09,0]];
CONST jointtarget Pos_Int1:=[[-90.103274965,31.063177836,44.705695231,101.493168651,-64.26918409,-
15.569680883],[5500,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarg Paint_10:=[[3625,-49.014,1950],[0.75577563,0.00941491,-
0.654756415,0.002931545],[0,0,0,0],[3900,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarg Paint_20:=[[3798.74,-48.15,2264.77],[0.783616372,0.009278077,-
0.621166762,0.003339427],[0,0,0,0],[3900,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarg Paint_30:=[[3994.64,-49.62,2497.84],[0.857990958,0.008762819,-
0.513570133,0.004521891],[0,0,0,0],[3900,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarg Paint_40:=[[4206.02,-48.47,2721.14],[0.862439977,0.008723025,-
0.506063289,0.004598187],[0,0,0,0],[3900,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarg Paint_50:=[[4206.58,-200.91,2708.6],[0.8534766,0.008801946,-
0.521037866,0.004445248],[0,0,0,0],[3900,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarg Paint_60:=[[3997.92,-199.15,2494.25],[0.834772726,0.008951721,-
0.550506368,0.004135358],[0,0,0,0],[3900,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarg Paint_70:=[[3803.67,-197.12,2266.96],[0.783616372,0.009278077,-
0.621166762,0.003339427],[0,0,0,0],[3900,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarg Paint_80:=[[3620.26,-198.6,2026.67],[0.783616372,0.009278077,-
0.621166762,0.003339427],[0,0,0,0],[3900,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarg Paint_90:=[[3609.87,-347.55,2007.88],[0.783616372,0.009278077,-
0.621166762,0.003339427],[0,0,0,0],[3900,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarg Paint_100:=[[3811.57,-346.71,2266.24],[0.783616372,0.009278077,-
0.621166762,0.003339427],[0,0,0,0],[3900,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarg Paint_110:=[[4004.46,-348.61,2486.27],[0.783616372,0.009278077,-
0.621166762,0.003339427],[0,0,0,0],[3900,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarg Paint_120:=[[4207.68,-349.38,2686.74],[0.783616372,0.009278077,-
0.621166762,0.003339427],[0,0,0,0],[3900,9E+09,9E+09,9E+09,9E+09,0]];

```

LOCAL CONST robtarget Paint_130:=[[4193.89,-497.28,2638.7],[0.783616372,0.009278077,-0.621166762,0.003339427],[0,0,0],[3900,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_140:=[[4017.91,-499.57,2474.88],[0.783616372,0.009278077,-0.621166762,0.003339427],[0,0,0],[3900,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_150:=[[3812,-499.03,2254.66],[0.783616372,0.009278077,-0.621166762,0.003339427],[0,0,0],[3900,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_160:=[[3597.67,-496.86,1984.05],[0.75577563,0.00941491,-0.654756415,0.002931545],[0,0,0],[3900,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_170:=[[3582.5,-648.6,1952.18],[0.73159688,0.045125299,-0.67930303,-0.035736614],[0,0,0],[3900,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_180:=[[3580.18,-723.53,1941.3],[0.719629981,0.044494736,-0.691967695,-0.036518716],[0,0,0],[3900,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_190:=[[3728.9,-671.65,2129.49],[0.783616372,0.009278077,-0.621166762,0.003339427],[0,0,0],[3900,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_200:=[[3826.18,-670.88,2243.74],[0.783616372,0.009278077,-0.621166762,0.003339427],[0,0,0],[3900,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_210:=[[3932.94,-670.53,2350.26],[0.783616372,0.009278077,-0.621166762,0.003339427],[0,0,0],[3900,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_220:=[[4042.57,-667.98,2454.11],[0.783616372,0.009278077,-0.621166762,0.003339427],[0,0,0],[3900,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_230:=[[4161.67,-674.94,2549.25],[0.783616372,0.009278077,-0.621166762,0.003339427],[0,0,0],[3900,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_240:=[[4285.80132513,-558.502195022,2674.34732766],[0.783616372,0.009278077,-0.621166762,0.003339427],[0,0,0],[3900,9E+09,9E+09,9E+09,9E+09,0]];
 CONST robtarget Pos_Int2:=[[4288.184,-800,2766.904],[0.004916427,0.976980363,-0.195713251,-0.084743867],[-2,-1,-1,1],[3500,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_250:=[[3532.591288746,-49.368086098,1915.71059339],[0.003427052,-0.998257613,-0.002765011,0.058841713],[0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_260:=[[3883.94,-47.3,1878.85],[0.003427052,-0.998257613,-0.002765011,0.058841713],[0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_270:=[[4178.71,-47.46,1833.98],[0.003427052,-0.998257613,-0.002765011,0.058841713],[0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_280:=[[4473.25,-49,1770.47],[0.003427052,-0.998257613,-0.002765011,0.058841713],[0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_290:=[[4603.28,-48.2,1734.04],[0.003427052,-0.998257613,-0.002765011,0.058841713],[0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_300:=[[4581.04,-195.79,1731.17],[0.003427052,-0.998257613,-0.002765011,0.058841713],[0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_310:=[[4326.19,-196.99,1800.17],[0.003427052,-0.998257613,-0.002765011,0.058841713],[0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_320:=[[4033.55,-199.47,1855.86],[0.003427052,-0.998257613,-0.002765011,0.058841713],[0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_330:=[[3730.13,-198.16,1894.68],[0.003427052,-0.998257613,-0.002765011,0.058841713],[0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_340:=[[3682.11,-196.64,1899.23],[0.003427052,-0.998257613,-0.002765011,0.058841713],[0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_350:=[[3669.28,-345.49,1898.17],[0.003427052,-0.998257613,-0.002765011,0.058841713],[0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_360:=[[3883.32,-350.64,1873.48],[0.003427052,-0.998257613,-0.002765011,0.058841713],[0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_370:=[[4179.38,-344.23,1823.21],[0.003427052,-0.998257613,-0.002765011,0.058841713],[0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_380:=[[4469.28,-348.38,1749.72],[0.003427052,-0.998257613,-0.002765011,0.058841713],[0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_390:=[[4553.73,-347.1,1721.59],[0.003427052,-0.998257613,-0.002765011,0.058841713],[0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_400:=[[4508.37,-496.74,1708.61],[0.003427052,-0.998257613,-0.002765011,0.058841713],[0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_410:=[[4321.6,-499.78,1769.68],[0.003427052,-0.998257613,-0.002765011,0.058841713],[0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_420:=[[4032.18,-497.15,1839.54],[0.003427052,-0.998257613,-0.002765011,0.058841713],[0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_430:=[[3738.25,-497.87,1886.34],[0.003427052,-0.998257613,-0.002765011,0.058841713],[0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_440:=[[3620.69,-496.6,1898.92],[0.003427052,-0.998257613,-0.002765011,0.058841713],[0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_450:=[[3584.52,-646.26,1897.15],[0.003427052,-0.998257613,-0.002765011,0.058841713],[0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_460:=[[3880.42,-646.78,1851.49],[0.003427052,-0.998257613,-0.002765011,0.058841713],[0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_470:=[[4025.63,-648.72,1822.68],[0.003427052,-0.998257613,-0.002765011,0.058841713],[0,0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_480:=[[4174.61,-646.43,1790.47],[0.003427052,-0.998257613,-0.002765011,0.058841713],[0,0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_490:=[[4323.29,-612.37,1754.37],[0.003427052,-0.998257613,-0.002765011,0.058841713],[0,0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_500:=[[4465.91,-568.77,1712.07],[0.003427052,-0.998257613,-0.002765011,0.058841713],[0,0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_510:=[[4503.58,-555.86,1699.26],[0.003427052,-0.998257613,-0.002765011,0.058841713],[0,0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_520:=[[4462.94,-646.43,1676.91],[0.003427052,-0.998257613,-0.002765011,0.058841713],[0,0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_530:=[[4315.52,-677.91,1733.04],[0.003427052,-0.998257613,-0.002765011,0.058841713],[0,0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_540:=[[4169.88,-707.42,1768.84],[0.003427052,-0.998257613,-0.002765011,0.058841713],[0,0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_550:=[[4020.46,-746.01,1789.34],[0.003427052,-0.998257613,-0.002765011,0.058841713],[0,0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_560:=[[3876.53,-755.77,1813.97],[0.003427052,-0.998257613,-0.002765011,0.058841713],[0,0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_570:=[[3722.49,-760.68,1840.25],[0.003427052,-0.998257613,-0.002765011,0.058841713],[0,0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_580:=[[3432.551373666,-752.814310228,1903.054196485],[0.003427052,-0.998257613,-0.002765011,0.058841713],[0,0,0,0],[3500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_590:=[[3647.414556942,-686.314126602,1835.601176954],[0.212716674,0.975336672,0.015435147,-0.056848471],[0,0,0,0],[3200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_600:=[[3389.48,-682.96,1924.43],[0.463992801,0.871879529,-0.116178374,-0.105068326],[0,0,0,0],[3200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_610:=[[3241.92,-661.93,2001.41],[0.559329258,0.811511011,-0.159296797,-0.056790759],[0,0,0,0],[3200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_620:=[[3106.71,-632.46,2075.46],[0.64959301,0.750724782,-0.119099947,0.016013296],[0,0,0,0],[3200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_630:=[[2956.6,-597.42,2146.6],[0.658990013,0.732122221,-0.168593818,-0.036129513],[0,0,0,0],[3200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_640:=[[2866.21,-576.55,2187.31],[0.587835499,0.740265002,-0.292408597,-0.144756226],[0,0,0,0],[3200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_650:=[[2673.534436808,-536.425590132,2256.191424755],[0.145162171,0.620027111,-0.754918597,-0.156819125],[0,0,0,0],[3200,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_660:=[[2765.248675322,-704.207830237,2203.426820318],[0.059419547,0.641139149,-0.751126247,-0.145668353],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_670:=[[2815.68,-408.21,2229.19],[0.078824356,0.643541128,-0.748920663,-0.136964878],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_680:=[[2837.06,-261.21,2235.44],[0.101330589,0.64539887,-0.745801842,-0.130276716],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_690:=[[2852.01,-111.2,2237.74],[0.112136783,0.647656321,-0.744295584,-0.118282358],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_700:=[[2853.46,4.13,2239.28],[0.126182552,0.6506533,-0.742005461,-0.100777689],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_710:=[[2720.28,-0.98,2275.19],[0.064760979,0.654153449,-0.75083102,-0.064358846],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_720:=[[2718.66,-262.2,2267.45],[0.061463769,0.65129527,-0.750224632,-0.095914948],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_730:=[[2718.45,-543.05,2237.36],[0.042947926,0.646203688,-0.749725932,-0.135967996],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_740:=[[2568.91,-520.07,2269.78],[0.017112576,0.648678123,-0.754842497,-0.095585862],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_750:=[[2566.82,-265.12,2292.29],[0.028060589,0.652152074,-0.755272946,-0.058932609],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_760:=[[2568.21,1.63,2299.05],[0.049122688,0.653848592,-0.754173197,-0.035941189],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_770:=[[2415.66,-1.17,2312.95],[0.022548454,0.654121011,-0.755638898,-0.025042473],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_780:=[[2416.7,-261.98,2307.06],[0.010873814,0.653496989,-0.755778424,-0.040279269],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_790:=[[2415.39,-510.99,2287.42],[0.010770293,-0.651934754,0.755522723,0.063643481],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_800:=[[2267.72,-498.31,2297.17],[0.018965505,-0.6516916,0.756189259,0.055822695],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_810:=[[2265.42,-263.72,2315.49],[0.003477745,0.653168102,-0.756509576,-0.032443765],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_820:=[[2267.41,-1.42,2320.95],[0.015878192,0.653701372,-0.75636339,-0.018352694],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_830:=[[2114.69,-1.75,2325.22],[0.003832215,0.653792284,-0.75661975,-0.008216909],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_840:=[[2114.88,-263.32,2319.65],[0.00869694,-0.653555571,0.756538595,0.020948335],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_850:=[[2114.66,-485.79,2302.6],[0.033992023,-0.652484575,0.755695003,0.045094169],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_860:=[[1964.18,-478.43,2303.04],[0.04132801,-0.652054781,0.756075472,0.03829411],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_870:=[[1965.07,-263.92,2320.39],[0.016160658,-0.652966745,0.757046235,0.015945586],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_880:=[[1965.24,-2.82,2326.25],[0.002594992,-0.65314908,0.757220439,0.002598458],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_890:=[[1816.8,-1.61,2324.13],[0.015118963,-0.652981177,0.757183517,-0.007753716],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_900:=[[1814.89,-261.98,2317.9],[0.028673022,-0.653063955,0.756732164,0.006462223],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_910:=[[1814.07,-474.73,2299.4],[0.057933443,-0.652340171,0.755014453,0.032391257],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_920:=[[1665.87,-469.69,2292.8],[0.065712481,-0.651752687,0.754987505,0.029902723],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_930:=[[1664.02,-262.97,2311.7],[0.037063743,-0.652302358,0.757051968,0.000479994],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_940:=[[1665.83,-2.69,2318.49],[0.022553927,-0.652164727,0.757628353,-0.013106013],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_950:=[[1516.06,-2,2309.5],[0.029000698,-0.651959766,0.757469013,-0.018657884],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_960:=[[1514.66,-260.24,2302.26],[0.044666418,-0.652249676,0.756676462,-0.004000368],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_970:=[[1516.19,-466.6,2282.85],[0.072667415,-0.652168401,0.754157569,0.025341365],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_980:=[[1367.12,-460.02,2269.2],[0.087894576,-0.650581861,0.754179957,0.015178222],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_990:=[[1365.51,-261.73,2288.72],[0.058379292,-0.650528438,0.757120086,-0.013182783],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_1000:=[[1365.53,-1.33,2296.5],[0.041986968,-0.650016859,0.758214185,-0.028747655],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_1010:=[[1216.92,1.42,2278.95],[0.045056825,-0.649086236,0.757932778,-0.046849177],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_1020:=[[1218.66,-261.74,2270.62],[0.071218754,-0.650512773,0.755839895,-0.021611908],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_1030:=[[1216.101330775,-609.175735496,2234.36050256],[0.060508479,-0.701317919,0.709351986,-0.036216853],[0,0,0,0],[1500,9E+09,9E+09,9E+09,9E+09,0]];
 CONST robtarget Pos_Int3:=[[4500,-800,2766.904],[0.627211375,0.326505576,-0.627211375,-0.326505576],[0,0,0,0],[3900,9E+09,9E+09,9E+09,9E+09,0]];
 CONST robtarget Pos_Int4:=[[400,-608.398,1567.43],[0.176398953,-0.78183221,0.589095112,-0.102901671],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_1040:=[[1250.955,-686.816,2200],[0.007969103,0.673475854,-0.739004542,0.015461382],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_1050:=[[1152.4,-538.43,2216.07],[0.007969103,0.673475854,-0.739004542,0.015461382],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_1060:=[[1125.82,-481.39,2237.58],[0.007969103,0.673475854,-0.739004542,0.015461382],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_1070:=[[1118.31,-396.9,2242.61],[0.09423395,-0.786408063,0.609816689,-0.028388842],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_1080:=[[1101.46,-243.76,2254.01],[0.077937455,-0.786323384,0.611681836,-0.038296485],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_1090:=[[1097.8,-94.87,2259.01],[0.057836065,-0.784607984,0.613911165,-0.064485529],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_1100:=[[1092.57,-1.92,2259.18],[0.057966456,-0.784467858,0.613907375,-0.066089375],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_1110:=[[901.48,-0.35,2220.72],[0.073253463,-0.782626365,0.612380614,-0.084379416],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_1120:=[[900.73,-96.46,2219.39],[0.082048399,-0.783181566,0.611349575,-0.078398931],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_1130:=[[904.62,-251.84,2215.53],[0.096464687,-0.784534082,0.609340218,-0.062492694],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
 LOCAL CONST robtarget Paint_1140:=[[924.76,-392.92,2205.22],[0.131530672,-0.784548159,0.604592224,-0.040646157],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1150:=[[1004.19,-528.81,2187.57],[0.237554998,-0.752784861,0.613636895,-0.018230137],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1160:=[[1098.27,-578.49,2157.26],[0.335833428,-0.730805871,0.591395429,0.058224865],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1170:=[[1034.32,-554.47,2115.74],[0.140884006,-0.900395764,-0.037738575,-0.409896285],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1180:=[[978.38,-562.63,2060.52],[0.037800863,-0.938713234,0.006486818,-0.342558726],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1190:=[[894.12,-567.59,1989.26],[0.065026117,-0.946809003,-0.005403953,-0.315111271],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1200:=[[833.26,-569.34,1943.63],[0.065026117,-0.946809003,-0.005403953,-0.315111271],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1210:=[[851.11,-664.4,1919.96],[0.295218405,-0.904498253,0.195201962,-0.237960496],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1220:=[[746.77,-550.41,1904.76],[0.300176383,-0.815793928,0.437217802,-0.230683767],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1230:=[[681.15,-388.21,1892.88],[0.307379596,-0.73124961,0.554367228,-0.251930086],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1240:=[[646.31,-246.23,1883.89],[0.344642161,-0.60684629,0.676304984,-0.235734873],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1250:=[[630.96,-96.85,1882.25],[0.345631437,-0.593406254,0.676819725,-0.265184815],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1260:=[[628.02,-4.13,1881.51],[0.33995653,-0.584203659,0.679796523,-0.28462665],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1270:=[[586.78,0.19,1804.67],[0.387198464,-0.590503229,0.587154908,-0.395768114],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1280:=[[588.52,-97.87,1804.67],[0.424516454,-0.586312232,0.56018294,-0.402763978],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1290:=[[600.86,-247.79,1806.2],[0.448226616,-0.598538698,0.54178778,-0.383810277],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1300:=[[629.32,-399.55,1809.53],[0.449098296,-0.633709079,0.533418774,-0.334944676],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1310:=[[647.82,-452.17,1819.47],[0.431659513,-0.655752761,0.539269629,-0.304707479],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1320:=[[632.92,-486.06,1737.6],[0.509040752,-0.595367223,0.46420354,-0.413437366],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1330:=[[625.76,-397.38,1743.13],[0.454550513,-0.580117566,0.509448051,-0.444196043],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1340:=[[665.59,-247.41,1740.42],[0.444706021,-0.570458376,0.551772115,-0.41516422],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1350:=[[656.61,-97.41,1733.84],[0.462388505,-0.582012111,0.537058295,-0.398782096],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1360:=[[652.62,-10.46,1727.31],[0.463875151,-0.625348205,0.493387731,-0.387721567],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1370:=[[635.89,-4.22,1665.81],[0.484954915,-0.641665189,0.472684185,-0.360075236],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1380:=[[637.62,-99.9,1662.79],[0.442116914,-0.607978691,0.512976637,-0.41442673],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1390:=[[643.38,-243.79,1660.56],[0.418808017,-0.592839076,0.532045444,-0.435969403],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1400:=[[631.58,-329.45,1670.91],[0.436275089,-0.573546117,0.553918482,-0.416993062],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1410:=[[616.43,-396.51,1681.26],[0.45578018,-0.574831865,0.490854985,-0.469993764],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1420:=[[638.28,-535.15,1679.8],[0.542224974,-0.576660599,0.450472469,-0.412951796],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1430:=[[625.03,-497.53,1612.56],[0.542746189,-0.562794321,0.453188442,-0.428146427],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1440:=[[608.62,-397.79,1594.94],[0.537473214,-0.563416285,0.457622682,-0.429262291],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1450:=[[616.03,-315.32,1595.25],[0.436229374,-0.570357802,0.572634886,-0.395582101],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1460:=[[621.9,-251.15,1594.74],[0.445015862,-0.595832833,0.52787232,-0.410237653],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1470:=[[615.16,-97.54,1592.35],[0.470969498,-0.65174657,0.460624464,-0.375818097],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1480:=[[618.93,-7.75,1603.4],[0.432123586,-0.669140384,0.53695164,-0.277854801],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

LOCAL CONST robtarget Paint_1490:=[[586.28,-4.6,1539.97],[0.462604677,-0.698129199,0.38933759,-0.383443313],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];

```

LOCAL CONST robtarget Paint_1500:=[[587.86,-97.49,1538.46],[0.476616511,-0.659872228,0.449718688,-
0.367639015],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_1510:=[[593.88,-250.06,1541.36],[0.435607274,-0.625319364,0.489547239,-
0.423751692],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_1520:=[[606.71,-396.6,1544.15],[0.462418649,-0.633524439,0.469671405,-
0.405246283],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_1530:=[[637.508643576,-608.398380825,1567.430345144],[0.409956239,-
0.644343518,0.469850543,-0.442716364],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtarget Abertura_Maletero1:=[[215,-364.861,1680],[0.495370091,-
0.500294484,0.497745055,0.506521236],[0,0,0,0],[-230,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtarget Abertura_Maletero2:=[[311.691,-364.861,1680],[0.495369011,-
0.500291852,0.497746026,0.506523937],[0,0,0,0],[-230,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtarget Abertura_Maletero3:=[[311.690489014,-364.861,1700.792522053],[0.495369011,-
0.500291852,0.497746026,0.506523937],[0,0,0,0],[-230,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtarget Abertura_Maletero4:=[[100,-364.861,2320.808],[0.340908622,-
0.348009874,0.613949382,0.621076964],[0,0,0,0],[-230,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtarget Abertura_Maletero5:=[[287.5,-364.861,2923.772],[0.205510752,-
0.213931508,0.672515665,0.678027522],[0,0,0,0],[-230,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtarget Abertura_Maletero6:=[[50,-364.861,2923.771],[0.205510752,-
0.213931508,0.672515665,0.678027522],[0,0,0,0],[-230,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtarget Abertura_Maletero7:=[[49.999,-364.861,3200],[0.205510752,-
0.213931508,0.672515665,0.678027522],[0,0,0,0],[-230,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtarget Pos_Int5:=[[4000,-49.014,1950],[0.75577563,0.00941491,-
0.654756415,0.002931545],[0,0,0,0],[3900,9E+09,9E+09,9E+09,9E+09,0]];
CONST jointtarget Pos_Int6:=[[84.827752963,42.711440503,18.945225257,118.293617179,-55.513486045,-
60.774237763],[4400,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtarget Cierre_Maletero3:=[[100.5,-364.861,2355],[0.35705176,-0.36396149,0.604631314,0.611937128],[0,0,0,0],[-
230,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtarget Cierre_Maletero2:=[[313,-364.861,2958],[0.205504835,-0.213937377,0.672513798,0.678029316],[0,0,0,0],[-
230,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtarget Cierre_Maletero1:=[[315.499,-364.861,3025],[0.205504835,-
0.213937377,0.672513798,0.678029316],[0,0,0,0],[-230,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtarget Cierre_Maletero5:=[[100,-364.861,1800],[0.504133102,-0.508900868,0.48894069,0.497801892],[0,0,0,0],[-
230,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtarget Cierre_Maletero4:=[[287,-364.861,1735],[0.504133102,-0.508900868,0.48894069,0.497801892],[0,0,0,0],[-
230,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtarget Limpieza_1:=[[-500,-2350,1500],[0,1,0,0],[-1,1,-2,-1],[300,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtarget Limpieza_2:=[[-500,-2350,1360],[0,1,0,0],[-1,1,-2,-1],[300,9E+09,9E+09,9E+09,9E+09,0]];
CONST jointtarget Pos_Int7:=[[0,0,0,0,45,0],[4570,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtarget Pos_Sal_Cap:=[[4750,-349.38,2686.74],[0.353553391,0.612372436,-0.612372436,-
0.353553391],[0,0,0,0],[3900,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtarget Pos_Sal_Malet:=[[250,-608.398,2250],[0.353553391,-0.612372436,0.612372436,-
0.353553391],[0,0,0,0],[0,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtarget Pos_Sal_AMalet:=[[-500,139.233,2250],[0.180128727,-0.686530278,0.678742423,0.18851669],[-1,-2,1,0],[-
230,9E+09,9E+09,9E+09,9E+09,0]];
CONST jointtarget Mantenimiento_1:=[[0.000001113,93.270611268,2.054475225,102.819919237,-
31.050494174,77.180099432],[0,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtarget Mantenimiento_2:=[[-1000,-2150,1975],[0.707106781,0,-0.707106781,0],[-1,1,0,-
1],[0,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtarget Mantenimiento_3:=[[-1425,-2150,1975],[0.707106781,0,-0.707106781,0],[-1,1,0,-
1],[0,9E+09,9E+09,9E+09,9E+09,0]];
CONST jointtarget Pos_Mant_Int_1:=[[-60,-30,30,0,-45,0],[500,9E+09,9E+09,9E+09,9E+09,0]];
CONST jointtarget Pos_Mant_Int_2:=[[-45,50,20,45,-45,0],[500,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_1540:=[[-500,-1800,1450],[0,1,0,0],[-1,1,-2,-1],[300,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_1550:=[[-500,-1800,1185],[0,1,0,0],[-1,1,-2,-1],[300,9E+09,9E+09,9E+09,9E+09,0]];
LOCAL CONST robtarget Paint_1560:=[[-500,-1800,1450],[0,1,0,0],[-1,1,-2,-1],[300,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtarget Pos_Int_Limp:=[[279.256,-1080.862,3217.652],[0,1,0,0],[-1,0,-1,-1],[300,9E+09,9E+09,9E+09,9E+09,0]];

```

!!!!!!!!!!!!!!!!!!!! DEFINICIÓN DE VARIABLES !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

```

VAR num Color;
VAR intnum Limp_Atomo;
VAR intnum Paro_Robot;
VAR intnum Mant_Atomo;
VAR num Lectura_Color;
VAR iodev infile;
VAR num Contador;
VAR num Secuencia_Colores{1000};
VAR string date;
VAR string N_Proc;
VAR string Limpieza;

```

!!!!!!!!!!!!!!!!!!!!!! PROCESOS DE PINTADO !!!!!!!!!!!!!!!!!!!!!!!

```
PROC main()
  Reset Abrir_Maletero;
  Reset Cerrar_Maletero;
  Reset Continuar2;
  Reset Inic_Pint_Inf;
  Reset Stop_Conveyor;
  Reset Finalizado;
  Reset Cambio_Color;
  Reset Pos_Parada;
  Reset Paro_Resto;
  Reset Open_TMant;
  Reset Close_TMant;
  SetAO Señal_Luminosa,3;
  Lectura_Colores;
  ! Eliminar los avisos de trayectoria de fallo de esquina
  CornerPathWarning FALSE;

  ! Definicion de los parametros para la rutina TRAP de limpieza si se requiere la limpieza del atomizador
  IDelete Limp_Atom;
  CONNECT Limp_Atom WITH Limpieza_Atomizador;
  ISignalDI Limpieza_Atom,1,Limp_Atom;

  ! Definicion de los parametros para la rutina TRAP de paro del robot en caso de que otro de los robots lo solicite
  IDelete Paro_Robot;
  CONNECT Paro_Robot WITH Parada;
  ISignalDI Paro_Propio,1,Paro_Robot;

  ! Definicion de los parametros para la rutina TRAP de mantenimiento si se requiere realizar mantenimiento al atomizador
  IDelete Mant_Atom;
  CONNECT Mant_Atom WITH Mantenimiento_Atomizador;
  ISignalDI Mantenimiento,1,Mant_Atom;

  Color:=Secuencia_Colores{1};
  Set Cambio_Color;
  ActUnit CNV1;
  FOR i1 FROM 1 TO Contador-1 DO
    Principal;
    Reset Cambio_Color;
    N_Proc:="main";
    WaitRob\ZeroSpeed;
    IF Finalizado=1 AND Color<>Secuencia_Colores{i1+1} AND Secuencia_Colores{i1+1}<>0 THEN
      Limpieza:="Limp1";
      Limpieza_Atom_Color;
      Color:=Secuencia_Colores{i1+1};
      Set Cambio_Color;
      ! Purga de color para preparar el atomizador
      PaintL Paint_1540,v2000,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
      SetBrush Color\Z:=1300;
      PaintL Paint_1550,v800,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
      WaitTime 2;
      SetBrush 1\Z:=1300;
      PaintL Paint_1560,v800,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
      N_Proc:="main";
      MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
    ELSEIF Finalizado=1 AND Color<>Secuencia_Colores{i1+1} AND Secuencia_Colores{i1+1}=0 THEN
      Limpieza:="Limp1";
      Limpieza_Atom_Color;
      N_Proc:="main";
      MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
    ELSE
      Color:=Secuencia_Colores{i1+1};
      Set Cambio_Color;
      MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
    ENDIF
  ENDFOR
  DeactUnit CNV1;
  ! Stop;
ENDPROC
```



```

PROC Principal()
  Reset Continuar2;
  N_Proc:="Principal";
  MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  WaitWObj wobj_cnv1;
  Reset Finalizado;
  MoveAbsJ Pos_Int1,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  WaitDI Continuar,1;
  WaitTime 3.5;
  Pint_Capo_Inf;
  N_Proc:="Principal";
  MoveL Pos_Int3,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
  MoveLDO Pos_Int2,v2500,z5,RB1000_WSC_70_700_200\WObj:=wobj_cnv1,Continuar2,1;
  WaitDI Continuar3,1;
  Pint_Capo_Sup;
  Pint_Barra;
  Pint_Techo;
  Pint_Malet_Sup;
  N_Proc:="Principal";
  MoveL Pos_Int4,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
  WaitDI Continuar3,1;
  Abertura_Maletero;
  N_Proc:="Principal";
  MoveAbsJ Pos_Int6,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  WaitDI Continuar,1;
  Reset Inic_Pint_Inf;
  Cierre_Maletero;
  N_Proc:="Principal";
  Set Finalizado;
  MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  DropWObj wobj_cnv1;
ENDPROC

PROC Lectura_Colores()
  VAR string Num_CE;
  ! Lectura del archivo con la cadena de números referentes a los colores
  date:=CDate();
  Open date+"_Lista_Colores.txt",infile\Read;
  Lectura_Color:=ReadNum(infile\Delim:=" ");
  Contador:=1;
  Num_CE:=NumToStr(Contador,0);
  ! Comprobación de que el archivo no está vacío o contine codigos de color incorrectos.
  IF Lectura_Color>EOF_NUM THEN
    TPWrite "El archivo que contiene la Secuencia de colores está vacío/incorrecto, revíselo.";
    ErrWrite "Color error","El archivo que contiene la Secuencia de colores está vacío/incorrecto, revíselo.";
    Break;
    ! Stop;
  ELSEIF Lectura_Color>16 OR Lectura_Color<2 THEN
    TPWrite "El identificador de color "+Num_CE+" es incorrecto, revíselo.";
    ErrWrite "Color error","El identificador de color "+Num_CE+" es incorrecto, revíselo."\\RL2:="Identificador superior a 16 o inferior a 2.";
    Break;
  ENDIF
  ! Lectura de la secuencia de colores completa.
  Secuencia_Colores{Contador}:=Lectura_Color;
  WHILE Lectura_Color<EOF_NUM DO
    Contador:=Contador+1;
    Num_CE:=NumToStr(Contador,0);
    Lectura_Color:=ReadNum(infile\Delim:=" ");
    ! Comprobación de que el archivo no continen ningún codigo de color incorrecto.
    IF Lectura_Color<EOF_NUM THEN
      IF Lectura_Color>16 OR Lectura_Color<2 THEN
        TPWrite "El identificador de color "+Num_CE+" es incorrecto, revíselo.";
        ErrWrite "Color error","El identificador de color "+Num_CE+" es incorrecto, revíselo."\\RL2:="Identificador superior a 16 o inferior a 2.";
        Break;
      ENDIF
    ENDIF
    Secuencia_Colores{Contador}:=Lectura_Color;
  ENDWHILE

```

```
Secuencia_Colores{Contador}:=0;  
ENDPROC
```

```
PROC Limpieza_Atomo_Color()  
N_Proc:="Limpieza_Atomo_Color";  
MoveL Limpieza_1,v1500,fine, RB1000_WSC_70_700_0\WObj:=WO_In_Int;  
MoveL Limpieza_2,v100,fine, RB1000_WSC_70_700_0\WObj:=WO_In_Int;  
IF Limpieza="Limp1" THEN  
Set RUN_Limp;  
WaitTime 5;  
MoveL Limpieza_1,v100,fine, RB1000_WSC_70_700_0\WObj:=WO_In_Int;  
ELSEIF Limpieza="Limp2" THEN  
Set RUN_Limp;  
WaitTime 5;  
MoveL Limpieza_1,v100,fine, RB1000_WSC_70_700_0\WObj:=WO_In_Int;  
! Purga de color para preparar el atomizador  
PaintL Paint_1540,v2000,z50, RB1000_WSC_70_700_0\WObj:=WO_In_Int;  
SetBrush Color\Z:=1300;  
PaintL Paint_1550,v800,z50, RB1000_WSC_70_700_0\WObj:=WO_In_Int;  
WaitTime 2;  
SetBrush 1\Z:=1300;  
PaintL Paint_1560,v800,z50, RB1000_WSC_70_700_0\WObj:=WO_In_Int;  
ENDIF  
Reset RUN_Limp;  
ENDPROC
```

```
PROC Mantenimiento_Atomo()  
MoveAbsJ Pos_Mant_Int_1,v2500,z50, RB1000_WSC_70_700_0\WObj:=WO_In_Int;  
MoveAbsJ Pos_Mant_Int_2,v2500,z50, RB1000_WSC_70_700_0\WObj:=WO_In_Int;  
MoveAbsJ Mantenimiento_1,v2500,z50, RB1000_WSC_70_700_0\WObj:=WO_In_Int;  
MoveLDO Mantenimiento_2,v500,fine, RB1000_WSC_70_700_200\WObj:=WO_In_Int,Open_TMant,1;  
WaitDI Ok_TMant,1;  
VelSet 100,100;  
MoveL Mantenimiento_3,v100,fine, RB1000_WSC_70_700_0\WObj:=WO_In_Int;  
SetAO Señal_Luminosa,2;  
WaitDI Mantenimiento,0;  
SetAO Señal_Luminosa,1;  
Reset Open_TMant;  
MoveLDO Mantenimiento_2,v100,fine, RB1000_WSC_70_700_200\WObj:=WO_In_Int,Close_TMant,1;  
VelSet 100,MaxRobSpeed();  
MoveAbsJ Mantenimiento_1,v2500,z50, RB1000_WSC_70_700_0\WObj:=WO_In_Int;  
MoveAbsJ Pos_Mant_Int_2,v2500,z50, RB1000_WSC_70_700_0\WObj:=WO_In_Int;  
MoveAbsJ Pos_Mant_Int_1,v2500,z50, RB1000_WSC_70_700_0\WObj:=WO_In_Int;  
Reset Close_TMant;  
ENDPROC
```

```
PROC Pint_Capo_Inf()  
N_Proc:="Pint_Capo_Inf";  
MoveL Pos_Int5,v2500,z50, RB1000_WSC_70_700_200\WObj:=wobj_cnv1;  
PaintL Paint_10,v2000,z50, RB1000_WSC_70_700_200\WObj:=wobj_cnv1;  
SetBrush Color\Z:=2044.12;  
PaintL Paint_20,v1500,z50, RB1000_WSC_70_700_200\WObj:=wobj_cnv1;  
PaintL Paint_30,v1500,z50, RB1000_WSC_70_700_200\WObj:=wobj_cnv1;  
PaintL Paint_40,v1500,z50, RB1000_WSC_70_700_200\WObj:=wobj_cnv1;  
PaintL Paint_50,v1500,z50, RB1000_WSC_70_700_200\WObj:=wobj_cnv1;  
PaintL Paint_60,v1500,z50, RB1000_WSC_70_700_200\WObj:=wobj_cnv1;  
PaintL Paint_70,v1500,z50, RB1000_WSC_70_700_200\WObj:=wobj_cnv1;  
PaintL Paint_80,v1500,z50, RB1000_WSC_70_700_200\WObj:=wobj_cnv1;  
PaintL Paint_90,v1500,z50, RB1000_WSC_70_700_200\WObj:=wobj_cnv1;  
PaintL Paint_100,v1500,z50, RB1000_WSC_70_700_200\WObj:=wobj_cnv1;  
PaintL Paint_110,v1500,z50, RB1000_WSC_70_700_200\WObj:=wobj_cnv1;  
PaintL Paint_120,v1500,z50, RB1000_WSC_70_700_200\WObj:=wobj_cnv1;  
PaintL Paint_130,v1500,z50, RB1000_WSC_70_700_200\WObj:=wobj_cnv1;  
PaintL Paint_140,v1500,z50, RB1000_WSC_70_700_200\WObj:=wobj_cnv1;  
PaintL Paint_150,v1500,z50, RB1000_WSC_70_700_200\WObj:=wobj_cnv1;  
PaintL Paint_160,v1500,z50, RB1000_WSC_70_700_200\WObj:=wobj_cnv1;  
PaintL Paint_170,v1500,z50, RB1000_WSC_70_700_200\WObj:=wobj_cnv1;  
PaintL Paint_180,v1500,z50, RB1000_WSC_70_700_200\WObj:=wobj_cnv1;  
PaintL Paint_190,v1500,z50, RB1000_WSC_70_700_200\WObj:=wobj_cnv1;  
PaintL Paint_200,v1500,z50, RB1000_WSC_70_700_200\WObj:=wobj_cnv1;  
PaintL Paint_210,v1500,z50, RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
```

```
PaintL Paint_220,v1500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_230,v1500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
SetBrush 1\Z:=2585.51;
PaintL Paint_240,v1500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
ENDPROC
```

PROC Pint_Capo_Sup()

```
N_Proc:="Pint_Capo_Sup";
PaintL Paint_250,v2000,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
SetBrush Color\X:=3681.77;
PaintL Paint_260,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_270,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_280,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_290,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_300,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_310,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_320,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_330,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_340,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_350,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_360,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_370,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_380,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_390,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_400,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_410,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_420,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_430,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_440,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_450,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_460,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_470,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_480,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_490,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_500,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_510,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_520,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_530,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_540,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_550,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_560,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_570,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
SetBrush 1\X:=3579.1;
PaintL Paint_580,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
```

ENDPROC

PROC Pint_Barra()

```
N_Proc:="Pint_Barra";
PaintL Paint_590,v2000,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
SetBrush Color\X:=3505.6;
PaintL Paint_600,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_610,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_620,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_630,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_640,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
SetBrush 1\X:=2812.14;
PaintL Paint_650,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
```

ENDPROC

PROC Pint_Techo()

```
N_Proc:="Pint_Techo";
PaintL Paint_660,v2000,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
SetBrush Color\Y:=556.88;
PaintL Paint_670,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_680,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_690,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_700,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_710,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_720,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_730,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_740,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
```



```

PaintL Paint_1390,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_1400,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_1410,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_1420,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_1430,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_1440,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_1450,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_1460,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_1470,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_1480,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_1490,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_1500,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_1510,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
PaintL Paint_1520,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
SetBrush 1\Y:=-460.83;
PaintL Paint_1530,v800,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
ENDPROC

```

PROC Abertura_Maletero()

```

N_Proc:="Abertura_Maletero1";
MoveL Abertura_Maletero1,v2500,z50,tool0\WObj:=wobj_cnv1;
N_Proc:="Abertura_Maletero2";
MoveL Abertura_Maletero2,v2500,z1,tool0\WObj:=wobj_cnv1;
N_Proc:="Abertura_Maletero3";
MoveLDO Abertura_Maletero3,v50,z1,tool0\WObj:=wobj_cnv1,Abrir_Maletero,1;
N_Proc:="Abertura_Maletero4";
MoveC Abertura_Maletero4,Abertura_Maletero5,v300,z1,tool0\WObj:=wobj_cnv1;
MoveLDO Abertura_Maletero6,v2500,z50,tool0\WObj:=wobj_cnv1,Inic_Pint_Inf,1;
Reset Abrir_Maletero;
MoveL Abertura_Maletero7,v2500,z50,tool0\WObj:=wobj_cnv1;
ENDPROC

```

PROC Cierre_Maletero()

```

N_Proc:="Cierre_Maletero1";
MoveL Cierre_Maletero1,v2500,z50,tool0\WObj:=wobj_cnv1;
MoveLDO Cierre_Maletero2,v50,z1,tool0\WObj:=wobj_cnv1,Cerrar_Maletero,1;
N_Proc:="Cierre_Maletero2";
MoveC Cierre_Maletero3,Cierre_Maletero4,v300,z1,tool0\WObj:=wobj_cnv1;
Reset Cerrar_Maletero;
MoveL Cierre_Maletero5,v2500,z50,tool0\WObj:=wobj_cnv1;
ENDPROC

```

!!!!!!!!!!!!!!!!!!!!!!!!!!!! RUTINAS TRAP !!!!!!!!!!!!!!!!!!!!!!!!!!!!!

TRAP Limpieza_Atomicador

```

VAR robtarget Punt_Paro;
VAR jointtarget jointPunt_Paro;
VAR string N_PROG2;
StopMove\Quick;
Set Paro_Resto;
Set Stop_Conveyor;
SetAO Señal_Luminosa,1;
StorePath;
N_PROG2:=N_Proc;
Limpieza:="Limp2";
Punt_Paro:=CRobT(\Tool:=RB1000_WSC_70_700_200\WObj:=WO_In_Int);
jointPunt_Paro:=CJointT(\TaskRef:=T_ROB1Id);
TEST N_Proc
CASE "main":
MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atomicolor;
MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Limpieza_Atomicolor,0;
Reset Paro_Resto;
SetAO Señal_Luminosa,1;
CASE "Principal":
MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atomicolor;
MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;

```

```

SetAO Señal_Luminosa,2;
WaitDI Limpieza_Atom,0;
Reset Paro_Resto;
SetAO Señal_Luminosa,1;
CASE "Limpieza_Atom_Color":
SetAO Señal_Luminosa,2;
WaitDI Limpieza_Atom,0;
Reset Paro_Resto;
SetAO Señal_Luminosa,1;
CASE "Pint_Capo_Int":
MoveL Pos_Sal_Cap,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atom_Color;
MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Limpieza_Atom,0;
Reset Paro_Resto;
SetAO Señal_Luminosa,1;
MoveL Pos_Sal_Cap,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
CASE "Pint_Capo_Sup":
MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atom_Color;
MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Limpieza_Atom,0;
Reset Paro_Resto;
SetAO Señal_Luminosa,1;
CASE "Pint_Barra":
MoveL offs(Punt_Paro,0,0,750),v1500,fine,RB1000_WSC_70_700_200\WObj:=WO_In_Int;
MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atom_Color;
MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Limpieza_Atom,0;
Reset Paro_Resto;
SetAO Señal_Luminosa,1;
MoveL offs(Punt_Paro,0,0,750),v1500,fine,RB1000_WSC_70_700_200\WObj:=WO_In_Int;
CASE "Pint_Techo":
MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atom_Color;
MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Limpieza_Atom,0;
Reset Paro_Resto;
SetAO Señal_Luminosa,1;
CASE "Pint_Malet_Sup1":
MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atom_Color;
MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Limpieza_Atom,0;
Reset Paro_Resto;
SetAO Señal_Luminosa,1;
CASE "Pint_Malet_Sup2":
MoveL Pos_Sal_Malet,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atom_Color;
MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Limpieza_Atom,0;
Reset Paro_Resto;
SetAO Señal_Luminosa,1;
MoveL Pos_Sal_Malet,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
CASE "Abertura_Maletero1":
MoveL Pos_Sal_AMalet,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atom_Color;
MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Limpieza_Atom,0;
Reset Paro_Resto;

```

```

SetAO Señal_Luminosa,1;
MoveL Pos_Sal_AMalet,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
CASE "Abertura_Maletero2":
MoveL Abertura_Maletero1,v2500,z50,tool0\WObj:=wobj_cnv1;
MoveL Pos_Sal_AMalet,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atom_Color;
MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Limpieza_Atom,0;
Reset Paro_Resto;
SetAO Señal_Luminosa,1;
MoveL Pos_Sal_AMalet,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Abertura_Maletero1,v2500,z50,tool0\WObj:=wobj_cnv1;
CASE "Abertura_Maletero3":
MoveL Abertura_Maletero2,v2500,z1,tool0\WObj:=wobj_cnv1;
MoveL Abertura_Maletero1,v2500,z50,tool0\WObj:=wobj_cnv1;
MoveL Pos_Sal_AMalet,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atom_Color;
MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Limpieza_Atom,0;
Reset Paro_Resto;
SetAO Señal_Luminosa,1;
MoveL Pos_Sal_AMalet,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Abertura_Maletero1,v2500,z50,tool0\WObj:=wobj_cnv1;
MoveL Abertura_Maletero2,v2500,z1,tool0\WObj:=wobj_cnv1;
CASE "Abertura_Maletero4":
MoveL offs(Punt_Paro,-150,0,-20),v1500,fine,RB1000_WSC_70_700_200\WObj:=WO_In_Int;
MoveL Pos_Sal_AMalet,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atom_Color;
MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Limpieza_Atom,0;
Reset Paro_Resto;
SetAO Señal_Luminosa,1;
MoveL Pos_Sal_AMalet,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL offs(Punt_Paro,-150,0,-20),v1500,fine,RB1000_WSC_70_700_200\WObj:=WO_In_Int;
CASE "Cierre_Maletero1":
MoveL offs(Punt_Paro,0,0,100),v1500,fine,RB1000_WSC_70_700_200\WObj:=WO_In_Int;
MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atom_Color;
MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Limpieza_Atom,0;
Reset Paro_Resto;
SetAO Señal_Luminosa,1;
MoveL offs(Punt_Paro,0,0,100),v1500,fine,RB1000_WSC_70_700_200\WObj:=WO_In_Int;
CASE "Cierre_Maletero2":
MoveL offs(Punt_Paro,0,0,100),v1500,fine,RB1000_WSC_70_700_200\WObj:=WO_In_Int;
MoveL Abertura_Maletero6,v2500,z50,tool0\WObj:=wobj_cnv1;
MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atom_Color;
MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Limpieza_Atom,0;
Reset Paro_Resto;
SetAO Señal_Luminosa,1;
MoveL Abertura_Maletero6,v2500,z50,tool0\WObj:=wobj_cnv1;
MoveL offs(Punt_Paro,0,0,100),v1500,fine,RB1000_WSC_70_700_200\WObj:=WO_In_Int;
DEFAULT:
MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atom_Color;
MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Limpieza_Atom,0;
Reset Paro_Resto;
SetAO Señal_Luminosa,1;
ENDTEST

```

```

MoveAbsJ jointPunt_Paro,v1500,fine,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
N_Proc:=N_PROC2;
Set Pos_Parada;
WaitDI OK_PosP,1;
RestoPath;
StartMove;
WaitTime 0.25;
Reset Stop_Conveyor;
Reset Pos_Parada;
SetAO Señal_Luminosa,3;
ENDTRAP

```

TRAP Parada

```

VAR robtarget Punt_Paro;
VAR jointtarget jointPunt_Paro;
SetAO Señal_Luminosa,1;
StopMove\Quick;
StorePath;
Punt_Paro:=CRobT(\Tool:=RB1000_WSC_70_700_200\WObj:=WO_In_Int);
jointPunt_Paro:=CJointT(\TaskRef:=T_ROB1Id);
TEST N_Proc
CASE "main":
    MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
    SetAO Señal_Luminosa,2;
    WaitDI Paro_Propio,0;
    SetAO Señal_Luminosa,1;
CASE "Principal":
    MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
    SetAO Señal_Luminosa,2;
    WaitDI Paro_Propio,0;
    SetAO Señal_Luminosa,1;
CASE "Limpieza_Atóm_Color":
    MoveL Limpieza_1,v1500,fine,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
    MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
    SetAO Señal_Luminosa,2;
    WaitDI Paro_Propio,0;
    SetAO Señal_Luminosa,1;
    MoveL Limpieza_1,v1500,fine,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
CASE "Pint_Capo_Inf":
    MoveL Pos_Sal_Cap,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
    MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
    SetAO Señal_Luminosa,2;
    WaitDI Paro_Propio,0;
    SetAO Señal_Luminosa,1;
    MoveL Pos_Sal_Cap,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
CASE "Pint_Capo_Sup":
    MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
    SetAO Señal_Luminosa,2;
    WaitDI Paro_Propio,0;
    SetAO Señal_Luminosa,1;
CASE "Pint_Barra":
    MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
    SetAO Señal_Luminosa,2;
    WaitDI Paro_Propio,0;
    SetAO Señal_Luminosa,1;
CASE "Pint_Techo":
    MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
    SetAO Señal_Luminosa,2;
    WaitDI Paro_Propio,0;
    SetAO Señal_Luminosa,1;
CASE "Pint_Malet_Sup1":
    MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
    SetAO Señal_Luminosa,2;
    WaitDI Paro_Propio,0;
    SetAO Señal_Luminosa,1;
CASE "Pint_Malet_Sup2":
    MoveL Pos_Sal_Malet,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
    MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
    SetAO Señal_Luminosa,2;
    WaitDI Paro_Propio,0;
    SetAO Señal_Luminosa,1;

```



```

MoveL Pos_Sal_Malet,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
CASE "Abertura_Maletero1":
MoveL Pos_Sal_AMalet,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Paro_Propio,0;
SetAO Señal_Luminosa,1;
MoveL Pos_Sal_AMalet,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
CASE "Abertura_Maletero2":
MoveL Abertura_Maletero1,v2500,z50,tool0\WObj:=wobj_cnv1;
MoveL Pos_Sal_AMalet,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Paro_Propio,0;
SetAO Señal_Luminosa,1;
MoveL Pos_Sal_AMalet,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Abertura_Maletero1,v2500,z50,tool0\WObj:=wobj_cnv1;
CASE "Abertura_Maletero3":
MoveL Abertura_Maletero2,v2500,z1,tool0\WObj:=wobj_cnv1;
MoveL Abertura_Maletero1,v2500,z50,tool0\WObj:=wobj_cnv1;
MoveL Pos_Sal_AMalet,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Paro_Propio,0;
SetAO Señal_Luminosa,1;
MoveL Pos_Sal_AMalet,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL Abertura_Maletero1,v2500,z50,tool0\WObj:=wobj_cnv1;
MoveL Abertura_Maletero2,v2500,z1,tool0\WObj:=wobj_cnv1;
CASE "Abertura_Maletero4":
MoveL offs(Punt_Paro,-150,0,-20),v1500,fine,RB1000_WSC_70_700_200\WObj:=WO_In_Int;
MoveL Pos_Sal_AMalet,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Paro_Propio,0;
SetAO Señal_Luminosa,1;
MoveL Pos_Sal_AMalet,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
MoveL offs(Punt_Paro,-150,0,-20),v1500,fine,RB1000_WSC_70_700_200\WObj:=WO_In_Int;
CASE "Cierre_Maletero1":
MoveL offs(Punt_Paro,0,0,100),v1500,fine,RB1000_WSC_70_700_200\WObj:=WO_In_Int;
MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Paro_Propio,0;
SetAO Señal_Luminosa,1;
MoveL offs(Punt_Paro,0,0,100),v1500,fine,RB1000_WSC_70_700_200\WObj:=WO_In_Int;
CASE "Cierre_Maletero2":
MoveL offs(Punt_Paro,0,0,100),v1500,fine,RB1000_WSC_70_700_200\WObj:=WO_In_Int;
MoveL Abertura_Maletero6,v2500,z50,tool0\WObj:=wobj_cnv1;
MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Paro_Propio,0;
SetAO Señal_Luminosa,1;
MoveL Abertura_Maletero6,v2500,z50,tool0\WObj:=wobj_cnv1;
MoveL offs(Punt_Paro,0,0,100),v1500,fine,RB1000_WSC_70_700_200\WObj:=WO_In_Int;
DEFAULT:
SetAO Señal_Luminosa,2;
WaitDI Paro_Propio,0;
SetAO Señal_Luminosa,1;
ENDTEST
MoveAbsJ jointPunt_Paro,v1500,fine,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Set Pos_Parada;
WaitDI OK_PosP,1;
RestoPath;
StartMove;
WaitTime 0.25;
Reset Pos_Parada;
SetAO Señal_Luminosa,3;
ENDTRAP

TRAP Mantenimiento_Atomicador
VAR robtarget Punt_Paro;
VAR jointtarget jointPunt_Paro;

```

```

VAR string N_PROC2;
StopMove\Quick;
Set Paro_Resto;
Set Stop_Conveyor;
SetAO Señal_Luminosa,1;
StorePath;
N_PROC2:=N_Proc;
Limpieza:="Limp1";
Punt_Paro:=CRobT(\Tool:=RB1000_WSC_70_700_200\WObj:=WO_In_Int);
jointPunt_Paro:=CJointT(\TaskRef:=T_ROB1Id);
TEST N_Proc
CASE "main":
  MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  Limpieza_Atom_Color;
  MoveL Pos_Int_Limp,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  Mantenimiento_Atom;
  Reset Paro_Resto;
  MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
CASE "Principal":
  MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  Limpieza_Atom_Color;
  MoveL Pos_Int_Limp,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  Mantenimiento_Atom;
  Reset Paro_Resto;
  MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
CASE "Limpieza_Atom_Color":
  MoveL Limpieza_1,v1500,fine,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  Limpieza_Atom_Color;
  MoveL Pos_Int_Limp,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  Mantenimiento_Atom;
  Reset Paro_Resto;
  MoveL Limpieza_1,v1500,fine,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
CASE "Pint_Capo_Inf":
  MoveL Pos_Sal_Cap,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
  MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  Limpieza_Atom_Color;
  MoveL Pos_Int_Limp,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  Mantenimiento_Atom;
  Reset Paro_Resto;
  MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  MoveL Pos_Sal_Cap,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
CASE "Pint_Capo_Sup":
  MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  Limpieza_Atom_Color;
  MoveL Pos_Int_Limp,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  Mantenimiento_Atom;
  Reset Paro_Resto;
  MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
CASE "Pint_Barra":
  MoveL offs(Punt_Paro,0,0,750),v1500,fine,RB1000_WSC_70_700_200\WObj:=WO_In_Int;
  MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  Limpieza_Atom_Color;
  MoveL Pos_Int_Limp,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  Mantenimiento_Atom;
  Reset Paro_Resto;
  MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  MoveL offs(Punt_Paro,0,0,750),v1500,fine,RB1000_WSC_70_700_200\WObj:=WO_In_Int;
CASE "Pint_Techo":
  MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  Limpieza_Atom_Color;
  MoveL Pos_Int_Limp,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  Mantenimiento_Atom;
  Reset Paro_Resto;
  MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
CASE "Pint_Malet_Sup1":
  MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  Limpieza_Atom_Color;
  MoveL Pos_Int_Limp,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
  Mantenimiento_Atom;
  Reset Paro_Resto;
  MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;

```

CASE "Pint_Malet_Sup2":
 MoveL Pos_Sal_Malet,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
 MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Limpieza_Atom_Color;
 MoveL Pos_Int_Limp,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Mantenimiento_Atom;
 Reset Paro_Resto;
 MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveL Pos_Sal_Malet,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;

CASE "Abertura_Maletero1":
 MoveL Pos_Sal_AMalet,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
 MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Limpieza_Atom_Color;
 MoveL Pos_Int_Limp,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Mantenimiento_Atom;
 Reset Paro_Resto;
 MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveL Pos_Sal_AMalet,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;

CASE "Abertura_Maletero2":
 MoveL Abertura_Maletero1,v2500,z50,tool0\WObj:=wobj_cnv1;
 MoveL Pos_Sal_AMalet,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
 MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Limpieza_Atom_Color;
 MoveL Pos_Int_Limp,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Mantenimiento_Atom;
 Reset Paro_Resto;
 MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveL Pos_Sal_AMalet,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
 MoveL Abertura_Maletero1,v2500,z50,tool0\WObj:=wobj_cnv1;

CASE "Abertura_Maletero3":
 MoveL Abertura_Maletero2,v2500,z1,tool0\WObj:=wobj_cnv1;
 MoveL Abertura_Maletero1,v2500,z50,tool0\WObj:=wobj_cnv1;
 MoveL Pos_Sal_AMalet,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
 MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Limpieza_Atom_Color;
 MoveL Pos_Int_Limp,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Mantenimiento_Atom;
 Reset Paro_Resto;
 MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveL Pos_Sal_AMalet,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
 MoveL Abertura_Maletero1,v2500,z50,tool0\WObj:=wobj_cnv1;
 MoveL Abertura_Maletero2,v2500,z1,tool0\WObj:=wobj_cnv1;

CASE "Abertura_Maletero4":
 MoveL offs(Punt_Paro,-150,0,-20),v1500,fine,RB1000_WSC_70_700_200\WObj:=WO_In_Int;
 MoveL Pos_Sal_AMalet,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
 MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Limpieza_Atom_Color;
 MoveL Pos_Int_Limp,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Mantenimiento_Atom;
 Reset Paro_Resto;
 MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveL Pos_Sal_AMalet,v2500,z50,RB1000_WSC_70_700_200\WObj:=wobj_cnv1;
 MoveL offs(Punt_Paro,-150,0,-20),v1500,fine,RB1000_WSC_70_700_200\WObj:=WO_In_Int;

CASE "Cierre_Maletero1":
 MoveL offs(Punt_Paro,0,0,100),v1500,fine,RB1000_WSC_70_700_200\WObj:=WO_In_Int;
 MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Limpieza_Atom_Color;
 MoveL Pos_Int_Limp,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Mantenimiento_Atom;
 Reset Paro_Resto;
 MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 MoveL offs(Punt_Paro,0,0,100),v1500,fine,RB1000_WSC_70_700_200\WObj:=WO_In_Int;

CASE "Cierre_Maletero2":
 MoveL offs(Punt_Paro,0,0,100),v1500,fine,RB1000_WSC_70_700_200\WObj:=WO_In_Int;
 MoveL Abertura_Maletero6,v2500,z50,tool0\WObj:=wobj_cnv1;
 MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Limpieza_Atom_Color;
 MoveL Pos_Int_Limp,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
 Mantenimiento_Atom;
 Reset Paro_Resto;
 MoveAbsJ Pos_Int7,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;

```

MoveL Abertura_Maletero6,v2500,z50,tool0\WObj:=wobj_cnv1;
MoveL offs(Punt_Paro,0,0,100),v1500,fine,RB1000_WSC_70_700_200\WObj:=WO_In_Int;
DEFAULT:
MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Limpieza_Atom_Color;
MoveL Pos_Int_Limp,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
Mantenimiento_Atom;
Reset Paro_Resto;
MoveAbsJ Pos_Inicial,v2500,z50,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
ENDTEST
MoveAbsJ jointPunt_Paro,v1500,fine,RB1000_WSC_70_700_0\WObj:=WO_In_Int;
N_Proc:=N_PROC2;
Set Pos_Parada;
WaitDI OK_PosP,1;
RestoPath;
StartMove;
WaitTime 0.25;
Reset Stop_Conveyor;
Reset Pos_Parada;
SetAO Señal_Luminosa,3;
ENDTRAP
ENDMODULE

```

8.1.5 Controlador virtual Cont Puertas Izq

```

MODULE Module1
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
! Módulo: Module1
!
! Descripción: Procesos para la apertura y cierre de las puertas del lado izquierdo de la carrocería empleando el robot de
asistencia izquierdo
!
! Autor: Arturo López López
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

!!!!!!!!!!!!!!!!!!!!!!!!!!!! DEFINICIÓN DE OBJETIVOS !!!!!!!!!!!!!!!!!!!!!

CONST jointtarget Pos_Inicial:=[[ -84.999989843,4.999999428,5.000000354,0,0,0],[7500,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtargtarget Abrir_PI_Del1:=[[2768.033337856,1206.05828095,1928.874581213],[0,-0.169599228,0.985513116,0],[1,0,0,0],[3491.483647647,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtargtarget Abrir_PI_Del2:=[[2763.534933052,739.410057806,1928.874581213],[0,-0.490191806,0.87161459,0],[1,-1,0,0,0],[3107.51099205,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtargtarget Abrir_PI_Del3:=[[2763.535,739.41,1814.927],[0,-0.492423562,0.870355695,0],[0,0,0,0],[3107.51099205,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtargtarget Abrir_PI_Del4:=[[2750.794,834.736,1814.927],[0,-0.422618264,0.906307786,0],[0,0,0,0],[3212.23074717,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtargtarget Abrir_PI_Del5:=[[2751.114,928.804,1814.927],[0,-0.342020147,0.939692619,0],[0,0,0,0],[3316.950082779,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtargtarget Abrir_PI_Del6:=[[2760.47,1003.792,1814.927],[0,-0.284015347,0.958819734,0],[0,0,0,0],[3386.763252859,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtargtarget Abrir_PI_Del7:=[[2780.979,1087.226,1814.927],[0,-0.224951056,0.974370064,0],[0,0,0,0],[3456.576422938,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtargtarget Abrir_PI_Del8:=[[2817.764,1180.84,1814.927],[0,-0.207911693,0.9781476,0],[0,0,0,0],[3526.389593018,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtargtarget Abrir_PI_Del9:=[[2860.329,1257.416,1814.927],[0,-0.182235528,0.983254907,0],[0,0,0,0],[3596.202763098,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtargtarget Abrir_PI_Del10:=[[2909.242,1324.007,1814.927],[0,-0.147809413,0.989015863,0],[0,0,0,0],[3666.015933178,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtargtarget Abrir_PI_Del11:=[[2963.391,1382.591,1814.927],[0,-0.121869345,0.992546151,0],[0,0,0,0],[3735.829103258,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtargtarget Abrir_PI_Del12:=[[2983.818,1401.352,1814.927],[0,-0.139173103,0.990268068,0],[0,0,0,0],[3757,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtargtarget Abrir_PI_Del14:=[[2976.654,1394.84,1814.927],[0,-0.147809413,0.989015863,0],[0,0,0,0],[3745,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtargtarget Abrir_PI_Del12_2:=[[5421.817949635,1401.351998299,1814.927050447],[0,-0.133942721,0.990989075,0],[1,0,0,0],[6195,9E+09,9E+09,9E+09,9E+09,0]];

```

```

CONST robtarget Abrir_PI_Tras1:=[[1887.496046658,1103.977226955,1985.6],[0,-0.297012388,0.954873626,0],[-
1,0,0,0],[2530.438247012,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtarget Abrir_PI_Tras2:=[[1891.175034736,704.866022581,1985.6],[0,-0.477539313,0.878610383,0],[-
1,0,0,0],[2240,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtarget Abrir_PI_Tras3:=[[1891.175,704.866,1870.669],[0,-
0.477158758,0.878817114,0],[0,0,0,0],[2240,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtarget Abrir_PI_Tras4:=[[1869.498,771.117,1870.669],[0,-
0.406736642,0.913545458,0],[0,0,0,0],[2325,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtarget Abrir_PI_Tras5:=[[1862.785,802.04,1870.669],[0,-
0.390731128,0.920504854,0],[0,0,0,0],[2345,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtarget Abrir_PI_Tras6:=[[1856.391,853.363,1870.669],[0,-
0.358367949,0.933580427,0],[0,0,0,0],[2385,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtarget Abrir_PI_Tras7:=[[1855.413,901.771,1870.669],[0,-
0.317304656,0.948323655,0],[0,0,0,0],[2425,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtarget Abrir_PI_Tras8:=[[1859.467,951.855,1870.669],[0,-
0.284015344,0.958819735,0],[0,0,0,0],[2465,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtarget Abrir_PI_Tras9:=[[1864.875,983.818,1870.669],[0,-
0.241921894,0.970295727,0],[0,0,0,0],[2495,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtarget Abrir_PI_Tras10:=[[1874.319,1022.767,1870.669],[0,-
0.258819044,0.965925827,0],[0,0,0,0],[2515,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtarget Abrir_PI_Tras11:=[[1881.823,1046.029,1870.669],[0,-
0.250380003,0.968147641,0],[0,0,0,0],[2535,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtarget Abrir_PI_Tras12:=[[1894.791,1079.307,1870.669],[0,-
0.233445363,0.972369921,0],[0,0,0,0],[2565,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtarget Abrir_PI_Tras13:=[[1905.347,1102.171,1870.669],[0,-
0.224951053,0.974370065,0],[0,0,0,0],[2585,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtarget Abrir_PI_Tras14:=[[1920.863,1130.743,1870.669],[0,-
0.20791169,0.978147601,0],[0,0,0,0],[2615,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtarget Abrir_PI_Tras15:=[[1939.044,1159.428,1870.669],[0,-
0.190808995,0.981627184,0],[0,0,0,0],[2645,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtarget Abrir_PI_Tras16:=[[1953.903,1180.035,1870.669],[0,-
0.199367934,0.979924705,0],[0,0,0,0],[2665,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtarget Abrir_PI_Del12_1:=[[4070.058107115,1401.351998299,1814.927050447],[0,-0.133942721,0.990989075,0],[-
1,0,0,0],[4843.24015748,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtarget Abrir_PI_Del12_2_1:=[[5186.817949635,1401.351998299,1814.927050447],[0,-
0.133942721,0.990989075,0],[-1,0,0,0],[5960,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtarget Abrir_PI_Tras17_4:=[[5852.578780619,1180.034863135,1870.669120526],[0,-0.199616195,0.979874163,0],[-
1,0,0,0],[6563.675403595,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtarget Abrir_PI_Del12_3_1:=[[4966.93628063,1401.351989874,1814.927124953],[0,-0.133934622,0.99099017,0],[-
1,0,0,0],[5740.120260254,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtarget Abrir_PI_Del12_3:=[[5130.024752982,1401.351989874,1814.927124953],[0,-0.133934622,0.99099017,0],[-
1,0,0,0],[5903.208732605,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtarget Abrir_PI_Tras17_5:=[[5338.90270104,1180.034945082,1870.669105625],[0,-0.199613465,0.979874719,0],[-
1,0,0,0],[6050,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtarget Abrir_PI_Tras17_7:=[[5398.903015272,1180.034729119,1870.669075823],[0,-0.199614523,0.979874503,0],[-
1,0,0,0],[6110,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtarget Abrir_PI_Del12_6:=[[5146.816041666,1401.351912553,1814.927050447],[0,-0.133934666,0.990990164,0],[-
1,0,0,0],[5920,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtarget Abrir_PI_Tras1_2:=[[1887.496269224,1103.977235606,1650.6],[0,-0.297012669,0.954873539,0],[-
1,0,0,0],[2530.438247012,9E+09,9E+09,9E+09,9E+09,0]];
CONST jointtarget Pos_Int_1:=[[-84.999989843,4.999999428,5.000000354,0,0,0],[4000,9E+09,9E+09,9E+09,9E+09,0]];
CONST jointtarget Pos_Inicial_UP:=[[-84.999989843,4.999999428,340,0,0,0],[7500,9E+09,9E+09,9E+09,9E+09,0]];
CONST jointtarget Pos_Mant:=[[-84.999989843,-30,325,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtarget Pos_Mant_Int:=[[203.686946762,1524.164077699,1970.6],[0,-0.087155826,0.996194691,0],[-
1,0,0,0],[1000,9E+09,9E+09,9E+09,9E+09,0]];

```

!!!!!!!!!!!!!!!!!!!! DEFINICIÓN DE VARIABLES !!!!!!!!!!!!!!!!!!!!!

```

VAR intnum Paro_Robot;
VAR string N_Proc;
VAR num Contador;

```

!!!!!!!!!!!!!!!!!!!! PROCESOS DE ASISTENCIA !!!!!!!!!!!!!!!!!!!!!

```

PROC main()
Reset Abrir_Puerta_Del;
Reset Cerrar_Puerta_Del;
Reset Abrir_Puerta_Tras;
Reset Cerrar_Puerta_Tras;
Reset Int_Pint_P;
Reset Stop_Conveyor;

```

```

Reset Pos_Parada;
Reset Paro_Resto;
Reset Open_TMant;
Reset Close_TMant;
SetAO Señal_Luminosa,3;
! Eliminar los avisos de trayectoria de fallo de esquina
CornerPathWarning FALSE;

```

! Definición de los parametros para la rutina TRAP de paro del robot en caso de que otro de los robots lo solicite

```

IDelete Paro_Robot;
CONNECT Paro_Robot WITH Parada;
ISignalDI Paro_Propio,1,Paro_Robot;
N_Proc:="main";
ActUnit CNV1;
FOR i1 FROM 1 TO Contador-1 DO
  Principal;
  WaitDI Finalizado,1;
  IF Mantenimiento=1 THEN
    Set Paro_Resto;
    Set Stop_Conveyor;
    SetAO Señal_Luminosa,1;
    Mantenimiento_Gripper;
    N_Proc:="main";
    MoveAbsJ Pos_Inicial,v2500,z50,IRB5350_Gripper\WObj:=WO_Ini_Int;
    Reset Paro_Resto;
    Reset Close_TMant;
    Set Pos_Parada;
    WaitDI OK_PosP,1;
    WaitTime 0.25;
    Reset Stop_Conveyor;
    Reset Pos_Parada;
    SetAO Señal_Luminosa,3;
  ENDIF
ENDFOR
DeactUnit CNV1;
! Stop;
ENDPROC

```

```

PROC Principal()
  N_Proc:="Principal";
  Reset Int_Pint_P;
  MoveAbsJ Pos_Inicial,v2500,z50,IRB5350_Gripper\WObj:=WO_Ini_Int;
  WaitWObj wobj_cnv1;
  WaitDI Abrir,1;
  Abertura_Cierre_PI_Del;
  N_Proc:="Principal";
  WaitDI Abrir,1;
  Abertura_Cierre_PI_Tras;
  N_Proc:="Principal";
  WaitDI Avance,1;
  MoveAbsJ Pos_Int_1,v2500,z50,IRB5350_Gripper\WObj:=WO_Ini_Int;
  N_Proc:="Principal2";
  WaitDI Volver,1;
  N_Proc:="Principal";
  MoveAbsJ Pos_Inicial,v2500,z50,IRB5350_Gripper\WObj:=WO_Ini_Int;
  DropWObj wobj_cnv1;
ENDPROC

```

```

PROC Mantenimiento_Gripper()
  MoveLDO Pos_Mant_Int,v2500,z50,IRB5350_Gripper\WObj:=WO_Ini_Int,Open_TMant,1;
  WaitDI Ok_TMant,1;
  VelSet 100,100;
  MoveAbsJ Pos_Mant,v100,fine,IRB5350_Gripper\WObj:=WO_Ini_Int;
  SetAO Señal_Luminosa,2;
  WaitDI Mantenimiento,0;
  SetAO Señal_Luminosa,1;
  Reset Open_TMant;
  MoveLDO Pos_Mant_Int,v100,z50,IRB5350_Gripper\WObj:=WO_Ini_Int,Close_TMant,1;
  VelSet 100,MaxRobSpeed();
ENDPROC

```

```

PROC Abertura_Cierre_PI_Del()
  N_Proc:="Abertura_Cierre_PI_Del";
  MoveL Abrir_PI_Del1,v2500,z50,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PI_Del2,v2500,z1,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveLDO Abrir_PI_Del3,v150,z1,IRB5350_Gripper\WObj:=wobj_cnv1,Abrir_Puerta_Del,1;
  MoveLDO Abrir_PI_Del4,v300,z5,IRB5350_Gripper\WObj:=wobj_cnv1,Abrir_Puerta_Del,0;
  MoveL Abrir_PI_Del5,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PI_Del6,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PI_Del7,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PI_Del8,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PI_Del9,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PI_Del10,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PI_Del11,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveLDO Abrir_PI_Del12,v300,z1,IRB5350_Gripper\WObj:=wobj_cnv1,Int_Pint_P,1;
  MoveLDO Abrir_PI_Del12_3_1,v40,fine,IRB5350_Gripper\WObj:=WO_Ini_Int,Int_Pint_P,0;
  MoveL Abrir_PI_Del12_6,v40,fine,IRB5350_Gripper\WObj:=WO_Ini_Int;
  MoveLDO Abrir_PI_Del14,v300,z1,IRB5350_Gripper\WObj:=wobj_cnv1,Cerrar_Puerta_Del,1;
  MoveLDO Abrir_PI_Del11,v300,z5,IRB5350_Gripper\WObj:=wobj_cnv1,Cerrar_Puerta_Del,0;
  MoveL Abrir_PI_Del10,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PI_Del9,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PI_Del8,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PI_Del7,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PI_Del6,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PI_Del5,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PI_Del4,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PI_Del3,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PI_Del2,v2500,z5,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PI_Del1,v2500,z50,IRB5350_Gripper\WObj:=wobj_cnv1;
ENDPROC

```

```

PROC Abertura_Cierre_PI_Tras()
  N_Proc:="Abertura_Cierre_PI_Tras";
  MoveL Abrir_PI_Tras1,v2500,z50,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PI_Tras2,v2500,z1,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveLDO Abrir_PI_Tras3,v150,z1,IRB5350_Gripper\WObj:=wobj_cnv1,Abrir_Puerta_Tras,1;
  MoveLDO Abrir_PI_Tras4,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1,Abrir_Puerta_Tras,0;
  MoveL Abrir_PI_Tras5,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PI_Tras6,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PI_Tras7,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PI_Tras8,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PI_Tras9,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PI_Tras10,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PI_Tras11,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PI_Tras12,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PI_Tras13,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PI_Tras14,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PI_Tras15,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveLDO Abrir_PI_Tras16,v300,z1,IRB5350_Gripper\WObj:=wobj_cnv1,Int_Pint_P,1;
  MoveLDO Abrir_PI_Tras17_5,v40,fine,IRB5350_Gripper\WObj:=WO_Ini_Int,Int_Pint_P,0;
  MoveL Abrir_PI_Tras17_7,v40,fine,IRB5350_Gripper\WObj:=WO_Ini_Int;
  MoveLDO Abrir_PI_Tras15,v300,z1,IRB5350_Gripper\WObj:=wobj_cnv1,Cerrar_Puerta_Tras,1;
  MoveLDO Abrir_PI_Tras14,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1,Cerrar_Puerta_Tras,0;
  MoveL Abrir_PI_Tras13,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PI_Tras12,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PI_Tras11,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PI_Tras10,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PI_Tras9,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PI_Tras8,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PI_Tras7,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PI_Tras6,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PI_Tras5,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PI_Tras4,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PI_Tras3,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PI_Tras2,v2500,z5,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PI_Tras1,v2500,z50,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PI_Tras1_2,v2500,z50,IRB5350_Gripper\WObj:=wobj_cnv1;
ENDPROC

```

!!!!!!!!!!!!!!!!!!!!!!!!!!!! RUTINAS TRAP !!!!!!!!!!!!!!!!!!!!!!!!!!!!!

TRAP Parada


```

CONST robtarget Abrir_PTras17_4:=[[5852.578780619,-1180.034863135,1870.669120526],[0,0.199616195,0.979874163,0],[-
1,0,0,0],[6563.675403595,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtarget Abrir_PDel_12_9:=[[5919.252732244,-
1401.351998299,1814.927050447],[0,0.133942721,0.990989075,0],[0,0,0,-1],[6692.434782609,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtarget Abrir_PDel_12_2_2:=[[5556.817643856,-
1401.351931666,1814.927124953],[0,0.1339414,0.990989254,0],[0,0,0,-1],[6330,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtarget Abrir_PTras17_5:=[[6529.620370256,-
1180.034890242,1870.669075823],[0,0.140955468,0.990015937,0],[0,0,0,0],[7240.717391304,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtarget Abrir_PTras17_2_1:=[[5963.904110672,-
1180.034738033,1870.669105625],[0,0.199619115,0.979873568,0],[0,0,0,-1],[6675,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtarget Abrir_PDel_12_2_4:=[[5126.816950208,-
1401.351908369,1814.927169657],[0,0.133938476,0.990989649,0],[0,0,0,-1],[5900,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtarget Abrir_PDel_12_2_6:=[[5346.816980628,-
1401.35189205,1814.927050447],[0,0.133938594,0.990989633,0],[0,0,0,-1],[6120,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtarget Abrir_PTras17_2_2:=[[5448.906275673,-
1180.034725382,1870.669105625],[0,0.199628153,0.979871727,0],[0,0,0,-1],[6160,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtarget Abrir_PTras17_2_4:=[[5538.90598722,-
1180.034634096,1870.669075823],[0,0.199626839,0.979871994,0],[0,0,0,-1],[6250,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtarget Abrir_PTras1_2:=[[1887.496269224,-1103.977235606,1650.6],[0,0.297012669,0.954873539,0],[0,0,0,-
1],[2530.438247012,9E+09,9E+09,9E+09,9E+09,0]];
CONST jointtarget Pos_Int_1:=[[85,-5,5,0,0,0],[4000,9E+09,9E+09,9E+09,9E+09,0]];
CONST jointtarget Pos_Inicial_UP:=[[85,-5,340,0,0,0],[7500,9E+09,9E+09,9E+09,9E+09,0]];
CONST robtarget Pos_Mant_Int:=[[203.68694676,-1524.164077714,1970.6],[0,0.087155826,0.996194691,0],[0,0,0,-
1],[1000,9E+09,9E+09,9E+09,9E+09,0]];
CONST jointtarget Pos_Mant:=[[84.999989843,30,325,0,0,0],[-425,9E+09,9E+09,9E+09,9E+09,0]];

```

!!!!!!!!!!!!!!!!!!!!!! DEFINICIÓN DE VARIABLES !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

```

VAR intnum Paro_Robot;
VAR string N_Proc;
VAR num Contador;

```

!!!!!!!!!!!!!!!!!!!!!! PROCESOS DE ASISTENCIA !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

PROC main()

```

Reset Abrir_Puerta_Del;
Reset Cerrar_Puerta_Del;
Reset Abrir_Puerta_Tras;
Reset Cerrar_Puerta_Tras;
Reset Int_Pint_P;
Reset Stop_Conveyor;
Reset Pos_Parada;
Reset Paro_Resto;
Reset Open_TMant;
Reset Close_TMant;
SetAO Señal_Luminosa,3;

```

```

! Eliminar los avisos de trayectoria de fallo de esquina
CornerPathWarning FALSE;

```

! Definición de los parametros para la rutina TRAP de paro del robot en caso de que otro de los robots lo solicite

```

IDelete Paro_Robot;
CONNECT Paro_Robot WITH Parada;
ISignalDI Paro_Propio,1,Paro_Robot;
N_Proc:="main";
ActUnit CNV1;
FOR i1 FROM 1 TO Contador-1 DO
  Principal;
  WaitDI Finalizado,1;
  IF Mantenimiento=1 THEN
    Set Paro_Resto;
    Set Stop_Conveyor;
    SetAO Señal_Luminosa,1;
    Mantenimiento_Gripper;
    N_Proc:="main";
    MoveAbsJ Pos_Inicial,v2500,z50,IRB5350_Gripper\WObj:=WO_In_Int;
    Reset Paro_Resto;
    Reset Close_TMant;
    Set Pos_Parada;
    WaitDI OK_PosP,1;
    WaitTime 0.25;
    Reset Stop_Conveyor;
  
```

```

Reset Pos_Parada;
SetAO Señal_Luminosa,3;
ENDIF
ENDFOR
DeactUnit CNV1;
! Stop;
ENDPROC

PROC Principal()
N_Proc:="Principal";
Reset Int_Pint_P;
MoveAbsJ Pos_Inicial,v2500,z50,IRB5350_Gripper\WObj:=WO_In_Int;
WaitWObj wobj_cnv1;
WaitDI Abrir,1;
Abertura_Cierre_PD_Del;
N_Proc:="Principal";
WaitDI Abrir,1;
Abertura_Cierre_PD_Tras;
N_Proc:="Principal";
WaitDI Avance,1;
MoveAbsJ Pos_Int_1,v2500,z50,IRB5350_Gripper\WObj:=WO_In_Int;
N_Proc:="Principal2";
WaitDI Volver,1;
N_Proc:="Principal";
MoveAbsJ Pos_Inicial,v2500,z50,IRB5350_Gripper\WObj:=WO_In_Int;
DropWObj wobj_cnv1;
ENDPROC

PROC Mantenimiento_Gripper()
MoveLDO Pos_Mant_Int,v2500,z50,IRB5350_Gripper\WObj:=WO_In_Int,Open_TMant,1;
WaitDI Ok_TMant,1;
VelSet 100,100;
MoveAbsJ Pos_Mant,v100,fine,IRB5350_Gripper\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Mantenimiento,0;
SetAO Señal_Luminosa,1;
Reset Open_TMant;
MoveLDO Pos_Mant_Int,v100,z50,IRB5350_Gripper\WObj:=WO_In_Int,Close_TMant,1;
VelSet 100,MaxRobSpeed();
ENDPROC

PROC Abertura_Cierre_PD_Del()
N_Proc:="Abertura_Cierre_PD_Del";
MoveL Abrir_PDel_1,v2500,z50,IRB5350_Gripper\WObj:=wobj_cnv1;
MoveL Abrir_PDel_2,v2500,z1,IRB5350_Gripper\WObj:=wobj_cnv1;
MoveLDO Abrir_PDel_3,v150,z1,IRB5350_Gripper\WObj:=wobj_cnv1,Abrir_Puerta_Del,1;
MoveLDO Abrir_PDel_4,v300,z5,IRB5350_Gripper\WObj:=wobj_cnv1,Abrir_Puerta_Del,0;
MoveL Abrir_PDel_5,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
MoveL Abrir_PDel_6,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
MoveL Abrir_PDel_7,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
MoveL Abrir_PDel_8,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
MoveL Abrir_PDel_9,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
MoveL Abrir_PDel_10,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
MoveL Abrir_PDel_11,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
MoveLDO Abrir_PDel_12,v300,z1,IRB5350_Gripper\WObj:=wobj_cnv1,Int_Pint_P,1;
MoveLDO Abrir_PDel_12_2_4,v40,fine,IRB5350_Gripper\WObj:=WO_In_Int,Int_Pint_P,0;
MoveL Abrir_PDel_12_2_6,v40,fine,IRB5350_Gripper\WObj:=WO_In_Int;
MoveLDO Abrir_PDel_14,v300,z1,IRB5350_Gripper\WObj:=wobj_cnv1,Cerrar_Puerta_Del,1;
MoveLDO Abrir_PDel_11,v300,z5,IRB5350_Gripper\WObj:=wobj_cnv1,Cerrar_Puerta_Del,0;
MoveL Abrir_PDel_10,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
MoveL Abrir_PDel_9,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
MoveL Abrir_PDel_8,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
MoveL Abrir_PDel_7,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
MoveL Abrir_PDel_6,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
MoveL Abrir_PDel_5,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
MoveL Abrir_PDel_4,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
MoveL Abrir_PDel_3,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
MoveL Abrir_PDel_2,v2500,z5,IRB5350_Gripper\WObj:=wobj_cnv1;
MoveL Abrir_PDel_1,v2500,z50,IRB5350_Gripper\WObj:=wobj_cnv1;
ENDPROC

```

```

PROC Abertura_Cierre_PD_Tras()
  N_Proc:="Abertura_Cierre_PD_Tras";
  MoveL Abrir_PTras1,v2500,z50,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PTras2,v2500,z1,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveLDO Abrir_PTras3,v150,z1,IRB5350_Gripper\WObj:=wobj_cnv1,Abrir_Puerta_Tras,1;
  MoveLDO Abrir_PTras4,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1,Abrir_Puerta_Tras,0;
  MoveL Abrir_PTras5,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PTras6,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PTras7,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PTras8,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PTras9,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PTras10,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PTras11,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PTras12,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PTras13,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PTras14,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PTras15,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveLDO Abrir_PTras16,v300,z1,IRB5350_Gripper\WObj:=wobj_cnv1,Int_Pint_P,1;
  MoveLDO Abrir_PTras17_2_2,v40,fine,IRB5350_Gripper\WObj:=WO_In_Int,Int_Pint_P,0;
  MoveL Abrir_PTras17_2_4,v40,fine,IRB5350_Gripper\WObj:=WO_In_Int;
  MoveLDO Abrir_PTras15,v300,z1,IRB5350_Gripper\WObj:=wobj_cnv1,Cerrar_Puerta_Tras,1;
  MoveLDO Abrir_PTras14,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1,Cerrar_Puerta_Tras,0;
  MoveL Abrir_PTras13,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PTras12,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PTras11,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PTras10,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PTras9,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PTras8,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PTras7,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PTras6,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PTras5,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PTras4,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PTras3,v300,z15,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PTras2,v2500,z5,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PTras1,v2500,z50,IRB5350_Gripper\WObj:=wobj_cnv1;
  MoveL Abrir_PTras1_2,v2500,z50,IRB5350_Gripper\WObj:=wobj_cnv1;

```

ENDPROC

!!!!!!!!!!!!!!!!!!!!!!!!!!!! RUTINAS TRAP !!!!!!!!!!!!!!!!!!!!!!!

TRAP Parada

```

  VAR jointtarget jointPunt_Paro;
  VAR num ZOffset;
  VAR jointtarget jointPunt_Paro_ZOffset;
  SetAO Señal_Luminosa,1;
  StopMove\Quick;
  StorePath;
  jointPunt_Paro:=CJointT(\TaskRef:=T_ROB1Id);
  ZOffset:=345-jointPunt_Paro.robax.rax_3;
  jointPunt_Paro_ZOffset:=jointPunt_Paro;
  jointPunt_Paro_ZOffset.robax.rax_3:=jointPunt_Paro.robax.rax_3+ZOffset;
  TEST N_Proc
  CASE "main":
    MoveAbsJ Pos_Inicial,v2500,z50,IRB5350_Gripper\WObj:=WO_In_Int;
    SetAO Señal_Luminosa,2;
    WaitDI Paro_Propio,0;
    SetAO Señal_Luminosa,1;
  CASE "Principal":
    MoveAbsJ Pos_Inicial,v2500,z50,IRB5350_Gripper\WObj:=WO_In_Int;
    SetAO Señal_Luminosa,2;
    WaitDI Paro_Propio,0;
    SetAO Señal_Luminosa,1;
  CASE "Principal2":
    SetAO Señal_Luminosa,2;
    WaitDI Paro_Propio,0;
    SetAO Señal_Luminosa,1;
  CASE "Abertura_Cierre_PD_Del":
    MoveAbsJ jointPunt_Paro_ZOffset,v1500,fine,IRB5350_Gripper\WObj:=WO_In_Int;
    MoveAbsJ Pos_Inicial_UP,v2500,z50,IRB5350_Gripper\WObj:=WO_In_Int;
    MoveAbsJ Pos_Inicial,v2500,z50,IRB5350_Gripper\WObj:=WO_In_Int;
    SetAO Señal_Luminosa,2;

```

```

WaitDI Paro_Propio,0;
SetAO Señal_Luminosa,1;
MoveAbsJ Pos_Inicial_UP,v2500,z50,IRB5350_Gripper\WObj:=WO_In_Int;
MoveAbsJ jointPunt_Paro_ZOffset,v1500,fine,IRB5350_Gripper\WObj:=WO_In_Int;
CASE "Abertura_Cierre_PD_Tras":
MoveAbsJ jointPunt_Paro_ZOffset,v1500,fine,IRB5350_Gripper\WObj:=WO_In_Int;
MoveAbsJ Pos_Inicial_UP,v2500,z50,IRB5350_Gripper\WObj:=WO_In_Int;
MoveAbsJ Pos_Inicial,v2500,z50,IRB5350_Gripper\WObj:=WO_In_Int;
SetAO Señal_Luminosa,2;
WaitDI Paro_Propio,0;
SetAO Señal_Luminosa,1;
MoveAbsJ Pos_Inicial_UP,v2500,z50,IRB5350_Gripper\WObj:=WO_In_Int;
MoveAbsJ jointPunt_Paro_ZOffset,v1500,fine,IRB5350_Gripper\WObj:=WO_In_Int;
DEFAULT:
SetAO Señal_Luminosa,2;
WaitDI Paro_Propio,0;
SetAO Señal_Luminosa,1;
ENDTEST
MoveAbsJ jointPunt_Paro,v1500,fine,IRB5350_Gripper\WObj:=WO_In_Int;
Set Pos_Parada;
WaitDI OK_PosP,1;
RestoPath;
StartMove;
WaitTime 0.25;
Reset Pos_Parada;
SetAO Señal_Luminosa,3;
ENDTRAP

ENDMODULE

```



8.2 Fichas técnicas de los robots de la estación



IRB 5500 FlexPainter

A new way of exterior painting



The IRB 5500 FlexPainter takes painting closer to perfection by integrating the paint application equipment. Combined with its large work area and high acceleration and painting speed, the result is the most efficient and flexible paint robot solution for basically any application.

Paint savings

Our compact and light-weight paint application components enable us to put vital paint regulation equipment, like the pumps, as close as 15 cm from the wrist. This reduces paint and solvent waste during color change significantly.

We have integrated the process equipment in the IRB 5500 FlexPainter in addition to the fully integrated process control (hardware and software). The IRC5P is controlling both the paint process and the robot motion so you can enjoy substantial savings.

Powered by IPS

The “push-out” function integrated in the IPS system is one specific feature that enables a reduction of paint even further. The basic architecture of IPS is built on combining process control and motion control as one, this simplified the system set up and enables for real savings and process perfection.

Built for painting

Standard solutions accommodate color change valves for up to 32* colors with circulation, integrated in the process arm of the robot. Also two pumps, driven by integrated servo motors, 64 pilot valves, atomizer control with dual shape air and closed loop regulation, closed loop regulation of bell speed and high voltage control – all fully integrated. Solutions for both solvent- and water-borne paint are available. Please note that more is available on special request.

Less atomizers, higher flow

ABB’s high flow RB1000 atomizer family is specifically designed for the high acceleration and speed of the IRB 5500 FlexPainter. This unique combination significantly reduces the number of robots needed in a spray booth and provides superior performance and high finish quality.

Modular paint components

To integrate the process into the IRB 5500 FlexPainter, ABB has developed a series of light-weight, compact and modular paint application components. This includes color change valves, 2K mixers, air- and paint regulators and pumps. All components are designed for maximum flow.

Paint robot control system

IRC5P is a modern control system, specifically designed for the paint shop. With the IPS (Integrated Process System), the userfriendly Ex certified FlexPaint Pendant and RobView 5 it is a combination of standardized functions for the paint installation and to fit specific needs. This package includes standard applications for defining User screens, Program editing and Version control, and many more. RobView 5 can also be a components in a larger Cell control HMI, like ABB FlexUI or other.



01



02



03

01 IRB 5500-22 process robot/elevated mounted

02 IRB 5500-22 process robot/floor mounted

03 IRB 5500-23 process robot on clean-wall rail

Specification	
Number of axes	6
Payload on wrist	13 kg
Protection	IP66 (wrist IP54)
Ex approval	Explosion protected Ex i/Ex p/Ex c for installation in hazardous area Zone 1 & Zone 21 (Europe) and Division I, Class I & II.
Mounting	Wall, floor, tilted, inverted, clean-wall rail

Technical information

Electrical Connections

Mains voltage	200 - 600VAC, 3-phase, 50/60 Hz
Energy consumption	According to international standards

Physical

Dimensions	
Robot footprint	500 x 680 mm
Robot controller	1450 x 725 x 710 mm
Weight	
Robot unit	600 kg
Robot controller	180 kg

Environment

Ambient temperature	
Robot unit	0 °C to +40 °C*
Robot controller	+48 °C maximum
Relative humidity	
	95 % maximum

*Recommended max ambient temp <30°

Technical information

PC Tools

RobView 5	Paint cell supervision and operation (included)
ShopFloor Editor	Off-line path- and process tuning using 3D graphics
RobotStudio® Paint	Full 3D simulation and programming of the paint cell

Interface

Backup	USB connection and Ethernet
I/O boards	Analog, digital, encoder and process I/O boards available
Fieldbus support	Interbus-S, ProfiBus, Profinet, CC Link, DeviceNet and Ethernet IP available
Network	Ethernet FTP/NFS

Performance

	Static repeatability (mm)	Wrist work envelope
	0.15	±140°

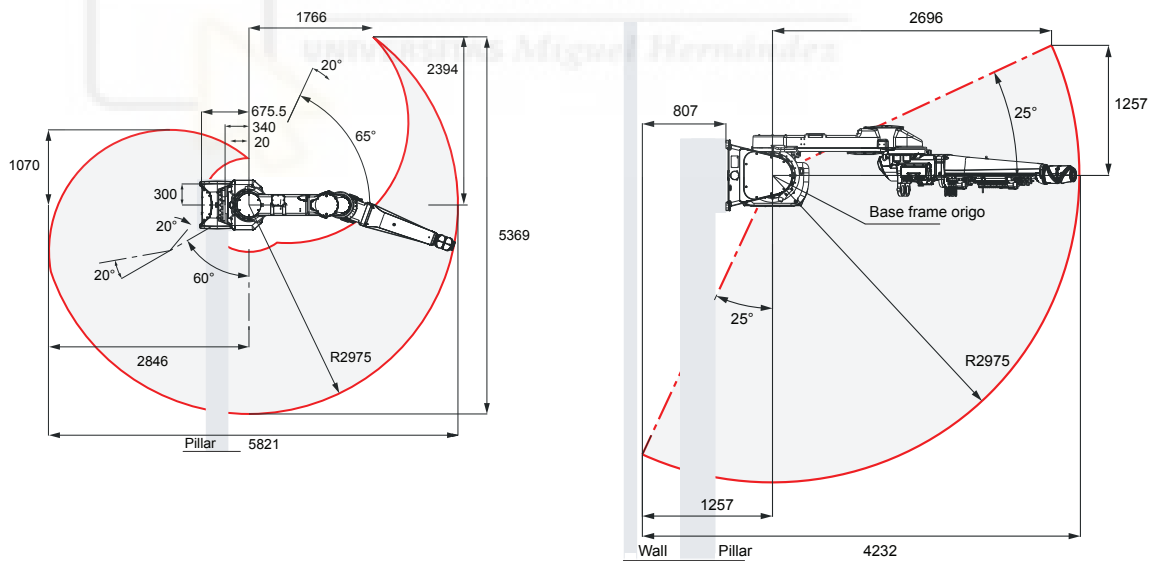
Specification

	IRB 5500-22	IRB 5500-22 Floor mounted	IRB 5500-23 Clean wall rail
Robot foot-print (mm)	500 x 680	581 x 718	500 x 680
Weight robot unit (kg)	600	610	600
Weight trolley (kg)	N/A	N/A	330
Rail system	N/A	N/A	Rail is available in steps of 0.5 m from 2-15 m for 1 robot on rail and 4-8 m for 2 robots on rail as standard. Longer rail can be delivered on request.
Cable chain wide (mm)	N/A	N/A	400
Accuracy (mm)	0.15	0.15	Robot: 0.15 (on TCPO) Trolley: 0,3 Velocity rail axis: 1,4 m/sec

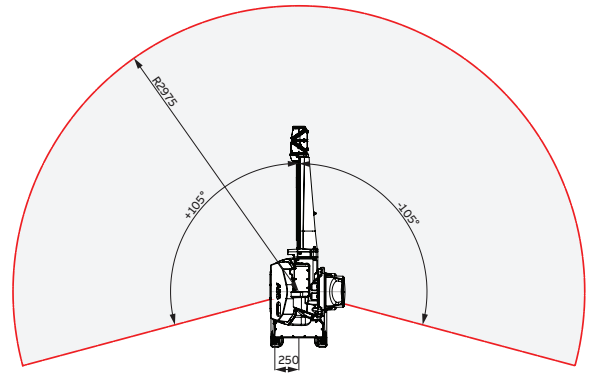
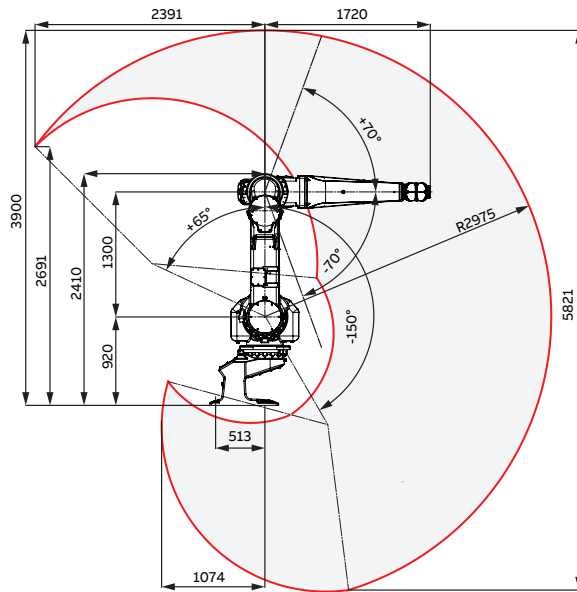
IRB 5500 FlexPainter

Axis movement	Working range	Axis max speed
Axis 1	See work envelope drawings	100°/s
Axis 2	See work envelope drawings	100°/s
Axis 3	See work envelope drawings	100°/s
Axis 4 wrist	See work envelope drawings	465°/s
Axis 5 bend	See work envelope drawings	350°/s
Axis 6 rotation	See work envelope drawings	535°/s

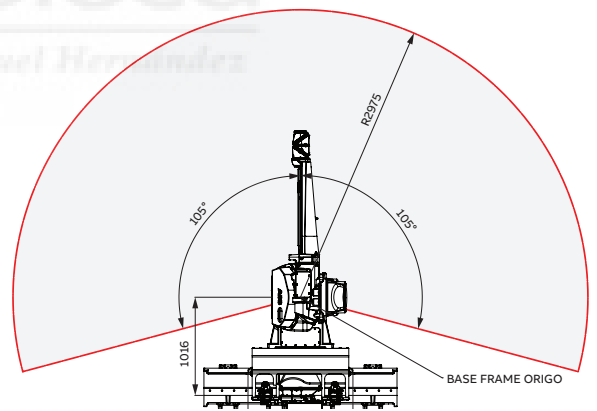
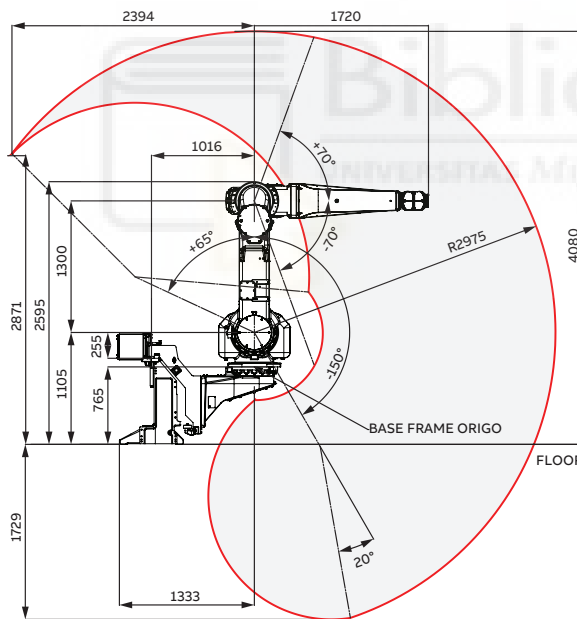
IRB 5500 FlexPainter, working range



IRB 5500-22 Floor mounted working range



IRB 5500-23 Clean wall rail working range



Elevated Rail for the IRB 5500-25



The Elevated Rail for the IRB 5500 reduces cycle times by as much as 10% on Stop & Go automotive paint lines.



Description

The Elevated Rail for the IRB 5500 system is one of the most advanced paint solutions available on the market. Ideally suited for both interior and exterior automotive painting as well as other paint applications, the system expands the flexibility of the IRB 5500 FlexPainter, which is the most versatile working envelope of any paint robot.

Designed for Stop & Go automotive paint lines, the Elevated Rail for the IRB 5500 increases the flexibility on these types of lines and allows for easy repositioning of the robot, guaranteeing an effective backup solution. It is able to accommodate most car body sizes which has the added advantage of potentially reducing the number of robots on a paint line. This added flexibility and the ability to place the robot in an optimal position directly contributes to reducing cycle times by up to 10%.

Reliability

Based on more than 30 years of rail experience, the Elevated Rail for the IRB 5500 is the fifth generation of ABB rail systems for automotive paint shops.

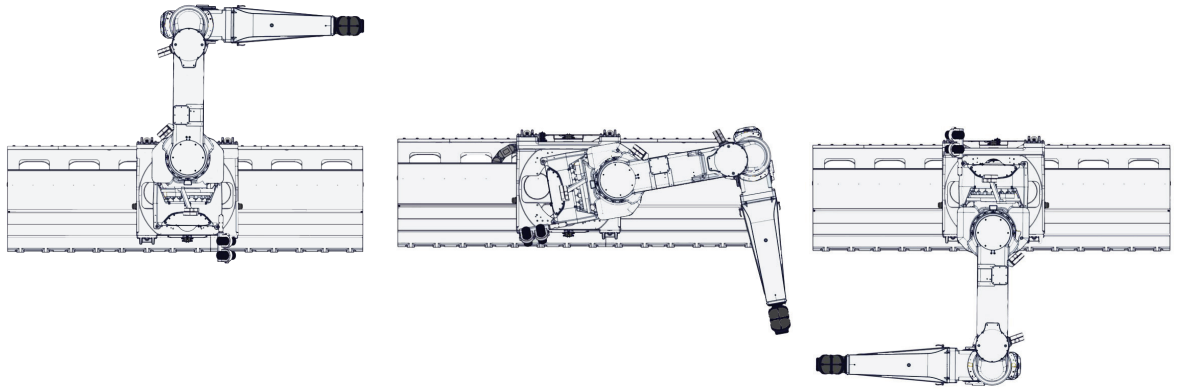
The Elevated Rail for the IRB 5500, eliminates the need for additional engineering and other costs associated with installing an IRB 5500 on a third party rail. The integrated ABB dynamic modelling for all axes lowers energy consumption, increases accuracy and provides a fully synchronized and balanced paint robot motion.

The elevated rail sections (rail elements) are designed to be a self-carrying structure without the need for a beam support under the rail.

To maintain accuracy and prevent deflection only two pillars with up to 4.7 meters (center-to-center) are employed with one or two robots on the rail.

RobotStudio®

The Elevated Rail for the IRB 5500 is fully supported for offline programming and 100% accurate cycle checks and path accuracy for all components (including the rail).



Features and benefits

- Extends unique features of IRB 5500, such as fully integrated process and motion- control and the special integration into the paint booth.
- Supports up to four robots on a single rail.
- Two pillar design accommodates most car body length configurations
- Maximum flexibility for use with a variety of applications.
- Total height (dog house) 1123 mm.
- Numerous manipulator mounting positions on a single trolley design – tilted, upright, inverted.
- Protected cable chain – 300 mm cable chain integrated into the rail elements.
- Centralized lubrication - track rails and gear rack, and trolley rollers.
- Automatic calibration sequence on axis 7 - sensor to detect the calibration position; no calibration tooling needed.

Technical data

Environment

Payload	13 kg (at 350 mm COG) - same as standard IRB 5500
Reach	2975 mm (at TCP 0) Same as IRB 5500 + from 2 m to 9 m as standard, up to 15m rail, or more, on request. Capable of supporting from 1-4 robots on the same rail (1 & 2 is covered as standard)
Accuracy	Robot: 0.15 mm (on TCP) Trolley: 0,3 mm. ABB dynamic modeling for all axes. Velocity axis 7 (rail axis): 1,4 m/sec
Deflection of rail	< 1 mm (with maximum speed and acceleration of manipulator in vertical direction)
Weight	Manipulator 600 kg. Trolley 370 kg, Rail element 380 kg/meter
Mounting position	Floor or Elevated. Robot: tilted, upright, inverted
Temperature	0°C - 40°C degree
IP Protection	IP 67 as standard (Ex zone 1, Class 1 Div 1)
Ex approval	Explosion protected Exi/Exp for installation in hazardous area Zone 1 & Zone 21 (Europe) and Division I, Class I & II.

Data and dimensions may be changed without notice.

IRB 5350, Door Opener Robot

Compact – Flexible – High performance



The IRB 5350 door opener robot is a compact and precise robot assistant for automotive interior painting, both for stop-and-go and moving-line solutions. A specially designed door opener tool with integrated sensors for search and force feedback makes this an efficient and important part of the interior paint zone.

Optimized design for various applications

The IRB 5350 robot provides two options for different interior painting solutions: the three axes stop-and-go version and the four axes moving-line version. This capable and efficient interior painting solution can support booth width from 4.5 to 6 meters, booth length from 3 to 10 meters and conveyor speeds ranging from 5 to 10 m/min.

Compact design and flexible installation

With the compact arm/foot/rail system, the IRB 5350 can accomplish the following:

- easily integrated into a narrow booth
- increased flexibility for the paint robot by using a dedicated rail system
- common solutions to be used on both sides of the booth, shift the working range for axis 2 for a left or right version
- the rail system support floor mounting.

High performance and reliability

For the last several decades, ABB has been peerless in robotic motion control technology. With the QuickMove™/TrueMove™ technology, the IRB 5350 robot is able to utilize swift acceleration and smart sensor tooling to complete a door opening cycle within 3 seconds (A typical cycle includes approaching, searching, gripping, opening, closing, releasing etc.)

The IRB 5350 handles a gripper tool of up to 7kg to grip, open and close different kinds of car doors using its advanced sensors, built into the tooling for detecting the door. Wellknown for its high protection standards, ABB's IRB 5350 has an IP66 rating. The rail axis has IP66 protection as standard.

Easy control and programming

With clear functionality, ABB offers a manageable solution to any interior painting challenge.

- Use the IRC5P robot controller to command both the IRB 5350 door opener robot along with the ABB's paint robots, common spare parts and interface.
- The EX-certified Teach Pendant can be used inside the paint booth for program modifications and testing.
- Offline programming of the entire interior zone is possible with use of ABB's innovative RobotStudio.

Global service and support

ABB customers can take advantage of the company's service organization; with more than 40 years of experience in the paint application area. ABB has support offices in 53 countries.



Specification	
Number of axes	3 axes/4 axes when rail-mounted
Robot mounting	Floor and rail mounted
Payload on tower	7 kg
Opening and closing force	Max 150N, Force is Perpendicular to the door blade.
Ingress protection degree	IP66
Robot unit ambient temperature	+0°C to +40°C*
Relative humidity, non-condensing	95% maximum
Ex classification	II 2 G Ex ib px IIB T4 Gb II 2 D Ex ib pd IIIC T65°C FM Class I, II. Div.1, Group C, D, G 135°C
Robot controller ambient temperature	+48°C maximum

* Recommended max ambient temp <30°C

Performance (according to ISO 9283)

Position repeatability (RP)	0.02 mm
Path repeatability (RT)	0.13 mm

Physical

Robot footprint	410 x 430 mm (standard foot) 465 mm x (3 – 10 m) length* (rail)
Robot height	1256 mm
Robot weight (3 axes)	215 kg
Robot weight (4 axes)	316 kg
Rail weight	124 kg/m
Robot Controller (H x W x D)	1450 x 725 x 710 mm
Robot controller weight	180 – 200 kg

*Longer rail lengths on request

Flexible rail system

The rails are designed for overspray protection and are available in a 1 m module that combines to the desired length, up to the 10 m (standard). It can be mounted on floor. One or two door opener robots can be mounted on one rail.



Movement			
Axis motion	Working range		Max speed
	3 axes	4 axes	
1 – Inner arm	+150° to -150°	+88° to -88°	167°/s
2 – Outer arm	+175° to -125° (left)	+175° to -125° (left)	180°/s
	+125° to -175° (right)	+125° to -175° (right)	180°/s
3 – Tower	350 mm	350 mm	780 mm/s
4 – Rail axis*	/	**	1920 mm/s

*Optional axis for rail motion

**Determined by the rail length

Controller interfaces

Backup	USB connection and Ether net
I/O boards	Analog, digital, relay, 120VAC, encoder and process I/O boards available
Fieldbus support	Interbus-S, ProfiBus, Profinet, CC Link, DeviceNet and Ether net IP available
Network	Ethernet FTP/NFS

PC tools

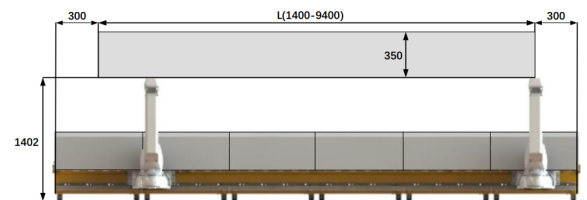
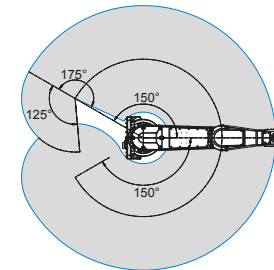
RobView 5 (included)	Paint cell supervision and operation
ShopFloor Editor	Off-line editing and process tuning
RobotStudio Paint	3D off-line simulation and programming

Electrical connection

Mains voltage	200 – 600VAC, 3-phase, 50/60 Hz
Power consumption	Stand by <300 W, production <800 W
Electrical safety	According to international standards
Emission	EMC/EMI shielded

Information may be changed or updated without notice

Work envelope, left version



EXTERNAL USE



PRODUCT PRESENTATION, 2018

RB1000i Series Presentation

ABB Ability™ Connected Atomizer

Carrie Zou – Global Product Manager - Paint



Content

- First in class in terms of performance
- Target customer & markets
- Customer merit
- Data sheet



First in class in terms of performance

Key features and benefits

- Modular concept – easy to maintain
- Better pattern control for higher T.E.
- Longer life time of parts
- Low contamination
- Reduced energy consumption
- Easy to maintain
- Less weight
- MTTR improvement



First in class in terms of performance

Smart thanks to the multiple sensors

- RFID for all critical parts checked at application
 - Individual follow-up of components
 - The right part in the right condition with the right parameters for the perfect application
- Vibration sensors
 - Gyroscope
 - Temperature
- Modular embedded sensor



New technology

Sensor and RFID

- **IoT (Internet of Things) - the first function in the coating industry**
 - RFID (Radio Frequency ID) equipped
 - Vibration and gyroscope sensors equipped
- **Air motor shaft lock function**



First in class in terms of performance

Improvement on performance

- RFID
 - Bell cup
 - SA nozzle
 - Air motor
- Vibration, gyro sensor
- Air motor shaft lock



Differentiated value proposition

ABB Ability™

Improvement on performance

- RFID
 - Bell cup
 - SA nozzle
 - Air motor
- **Vibration, gyro sensor**
- Air motor shaft lock



Target customer& markets

Paint solution and paint type

		RB1000i		
		SSD	SAD	S2K
Paint solution				
Automobile	Exterior	✓	✓	✓
	Interior	✓	✓	✓
	Bumper	✓	✓	✓
	Part	✓	✓	✓
General industry	Large	✓	✓	✓
	Small	✓	✓	✓
Paint type				
Primer paint		✓	✓	✓
Base paint		✓	✓	-
Clear paint		✓	✓	✓
Solventborne	1K	✓	✓	-
	2K	✓	-	✓
Waterborne*	1K	✓	✓	-
	2K	✓	-	✓

*Applicable to non-electrostatic type only

RB1000i

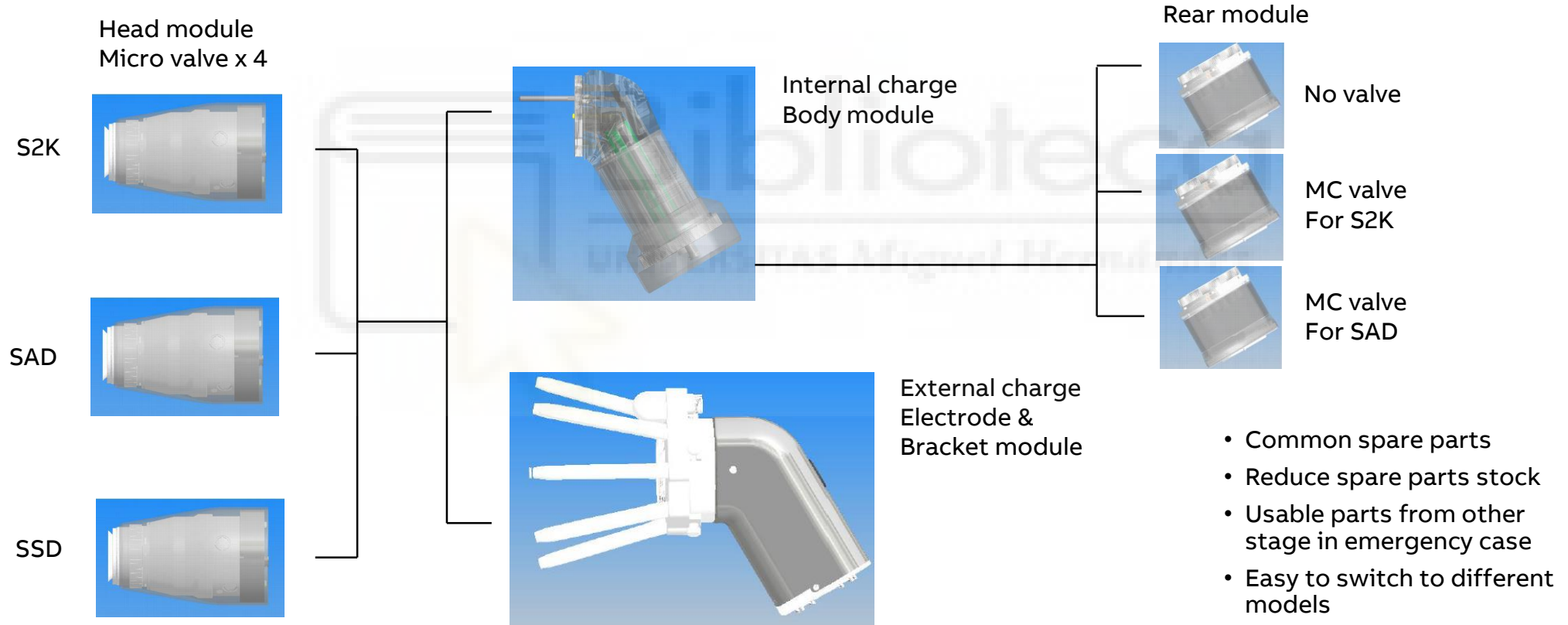


Multiple sensors



Customer merit

Modular design concept



Customer Merit

Improved conventional atomizer performance + IoT function

Atomizer

- Modular concept, common spare parts
- Light weight 8.7kg
- BOC (Bell Outside Cleaning)
- SFC (SA nozzle Face Cleaning)
- Super pattern control
- 6 bar air pressure motor
- Paint waste 6.7cc inside atomizer
- Bell cleaning 25cc / 1.5sec
- Built-in FO cable
- 1-step Bell cup removal
- Robust and long life Air Motor
- Quick valve replacement
- Zone 1 certification

ABB Ability

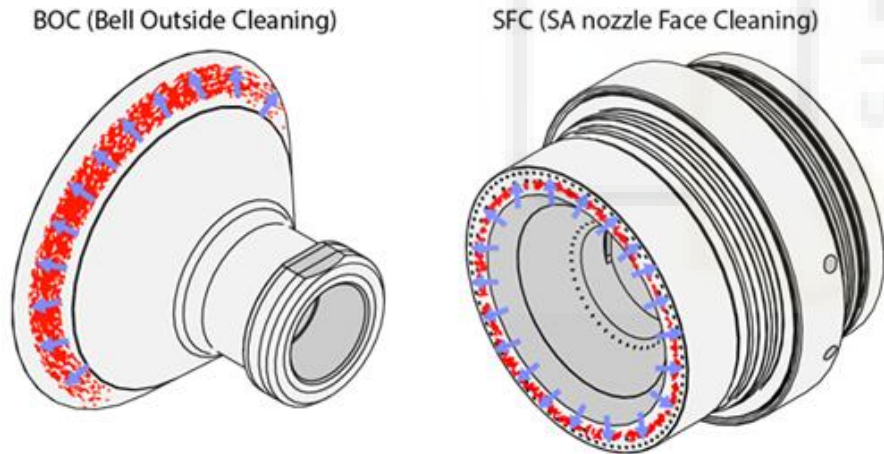
- Identification of parts number
- Before production;
- Confirmation of right bell cup, air motor, SA at right place;
- Genuine parts identification;
- Parts Life Management;
- Predictive maintenance;
- Smart diagnostic;

Customer Merit

BOC - Bell Outside Cleaning & SFC - SA nozzle Face Cleaning

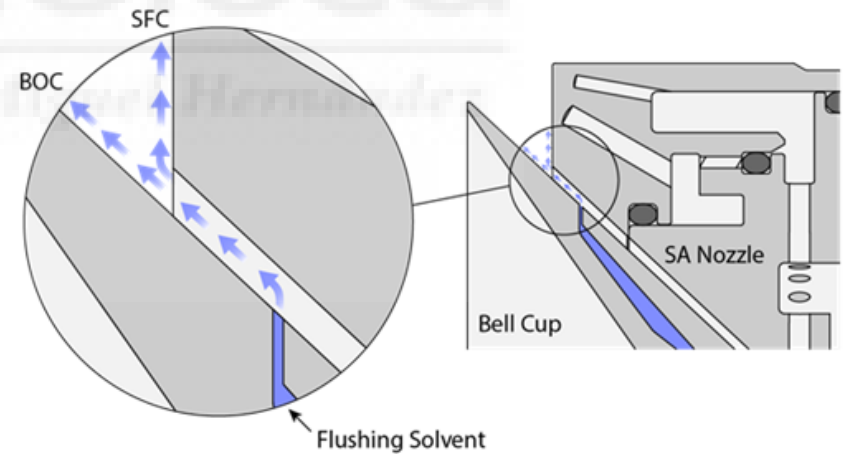
Items

- BOC & SFC



Merit

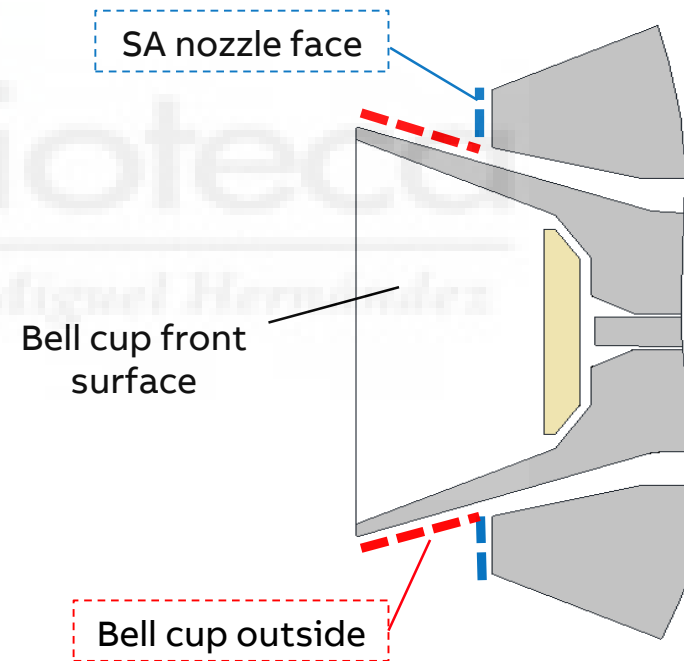
- Keep clean, keep good quality
- Less maintenance cycle



Customer Merit

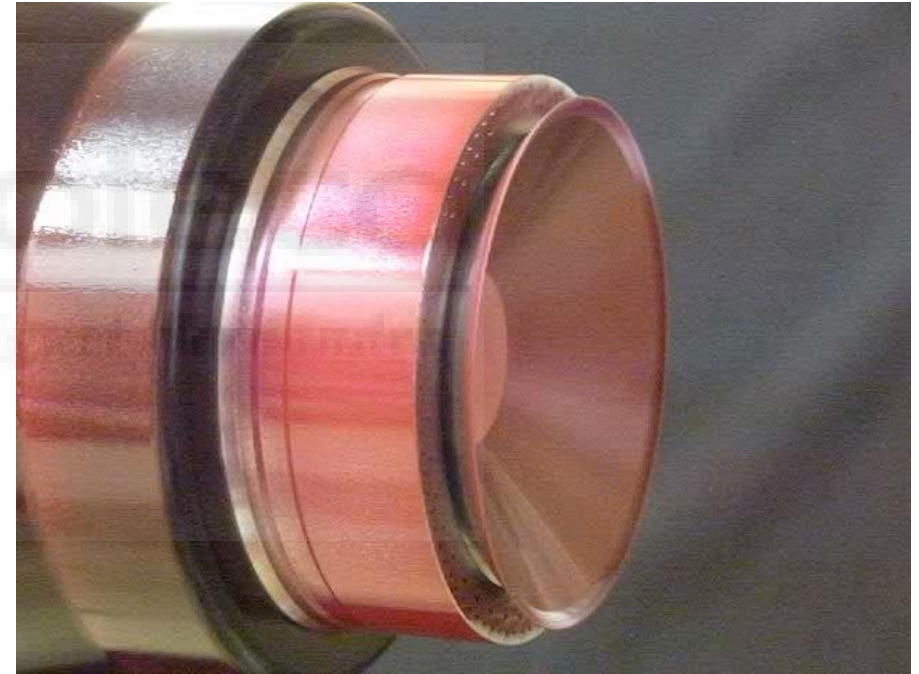
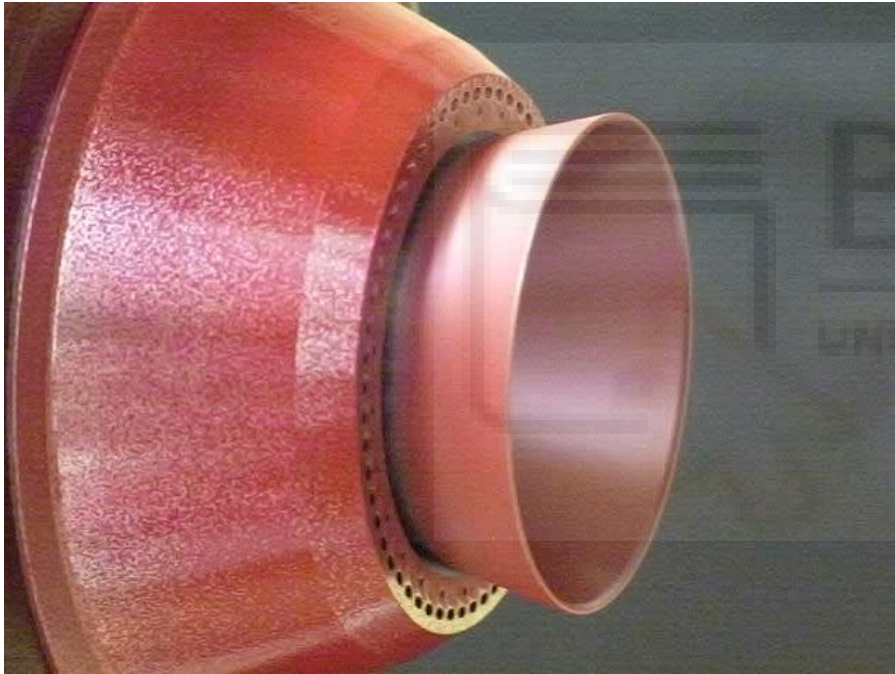
BOC - Bell Outside Cleaning & SFC - SA nozzle Face Cleaning

- Clean the following 3 parts simultaneously
 - Bell cup front surface
 - Bell cup outside (BOC; Bell Outside Cleaning)
 - SA nozzle face (SFC; SA nozzle Face Cleaning)
- Contaminate the bell cup intentionally and demonstrate the cleaning



Customer Merit

BOC - Bell Outside Cleaning & SFC - SA nozzle Face Cleaning



Cleaning conditions: Flushing solvent flow rate: 1000 cc/min, Flushing time: 1.0 sec, Usage: approx. 17 cc

Customer Merit

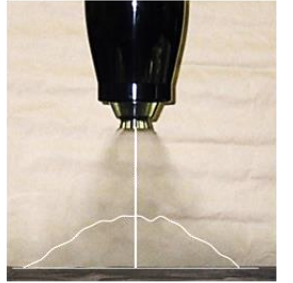
Merit

- Less over spray, less paint waste
- Reduce paint usage
- Less booth cleaning
- Less contamination

- Spray both interior & exterior car body available by D40mm

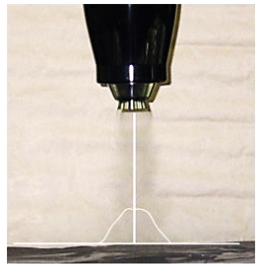
Items

- Super Pattern Control: Large pattern
- 40mm Bell cup
 - Effective pattern size: 400 mm
 - 70mm Bell cup
 - Effective pattern size: 500 mm



Items

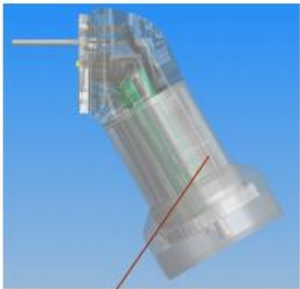
- Super Pattern Control: Small pattern
- 40mm Bell cup
 - Effective pattern size: 60 mm
 - 70mm Bell cup
 - Effective pattern size: 250 mm



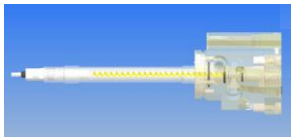
Customer Merit

Items

- Built-in FO cable & Mixer



FO cable



Mixer

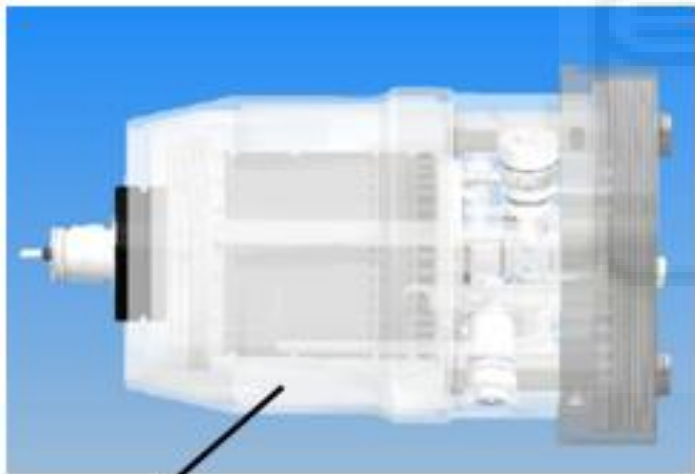
Merit

- Less damage risk of Fiber cable by solvent
- Easy rotation adjustment
- Less paint stuck risk, shorter mixing area
- Less cleaning time and flushing solvent consumption
- Easy replace mixer

Customer Merit

Items

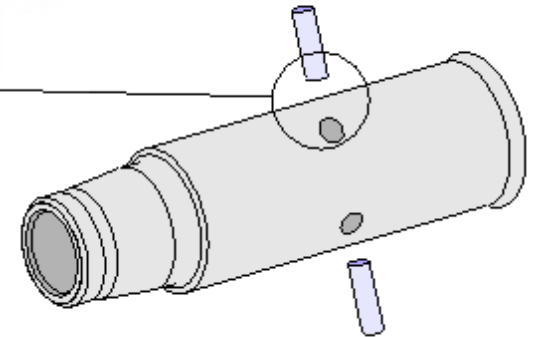
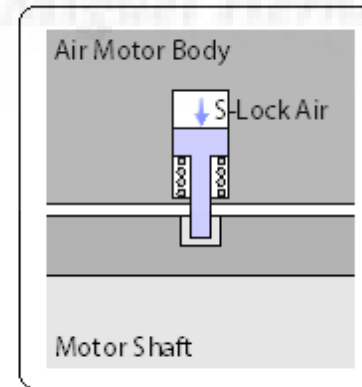
- 1-step Bell cup removal



Shaft Locking Cylinder

Merit

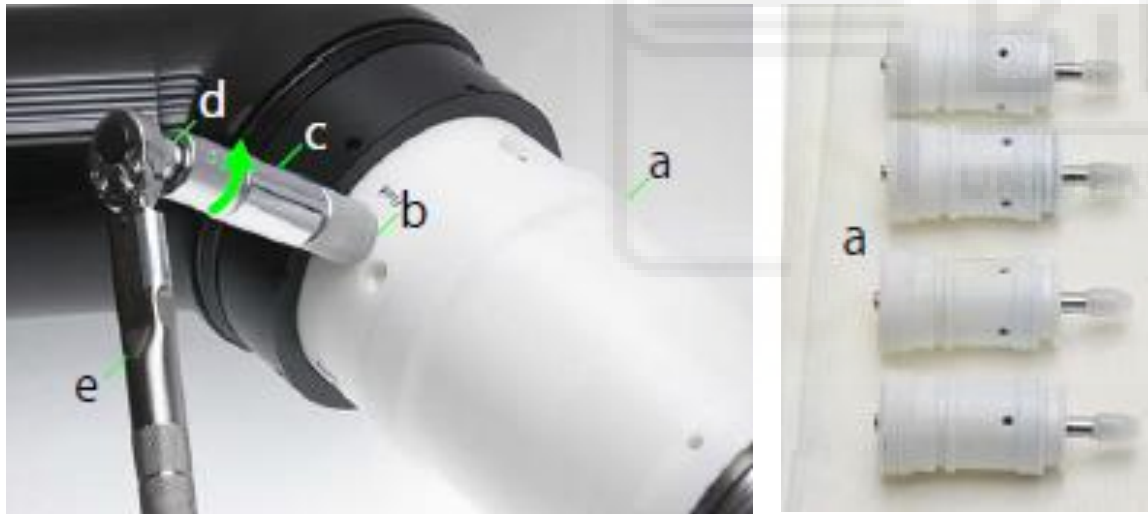
- Quick removal bell cup (1 min.)
- No tool required to remove bell cup
- Safety by interlock, no risk touch rotated bell cup



Customer Merit

Items

- Quick valve & Mixer replacement



Merit

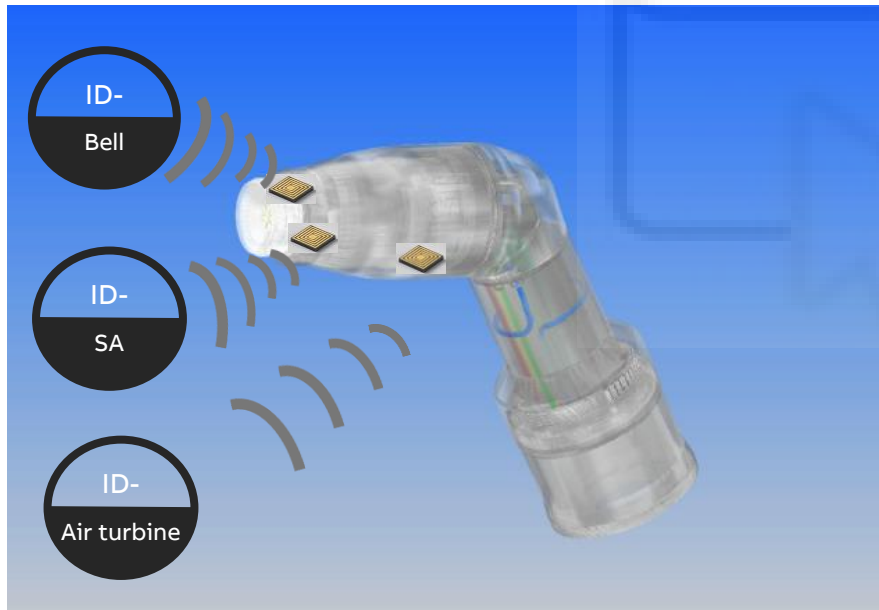
- 2 minutes valve replacement
- Mixer replace without disassemble air motor
- Quick recovery
- Easy maintenance



Customer Merit

Items

- IoT
 - RFID (Radio Frequency Identification)



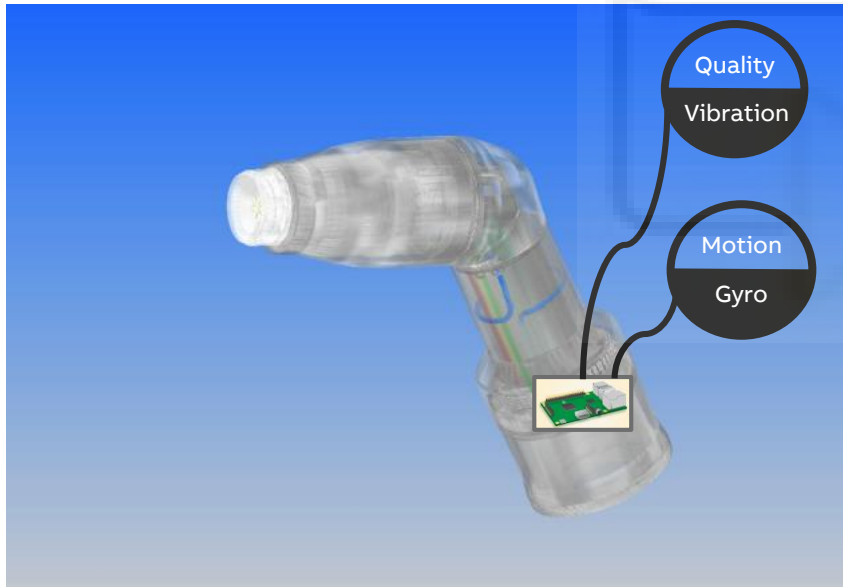
Merit

- Easy; parts management
- Alarm; incorrect parts assembly
- RFID Parts
 - Bell cup
 - SA nozzle
 - Air turbine

Customer Merit

Items

- IoT
 - Sensor (option)



Merit

- Gyro
- Vibration
 - Detect; unbalance of bell cup
 - Detect; damaged air motor
- Detect; gyro speed (considering air motor damage)

Data Sheet

Specification

Bell cup			φ70(BOC), φ40(BOC)
Shaping air nozzle	Function		Super pattern control function SA nozzle face cleaning (SFC) function
	Effective spray pattern width	φ70	250~500mm
		φ40	60~400mm
High voltage	Charging type		Internal charge
	Voltage applied		Max. -90 kV
	HV current		Max. 150 μA
Rotation speed	Set rotation speed		Max. 60,000 rpm
Weight ^{*01}			Approx. 8.8~9.0 kg
Total air consumption ^{*02}		φ70	695~1565 NI/min
Flow rate ^{*03*04}	Paint	φ70	100~1000 cc/min
Sensor ^{*05}			Vibration sensor
			Gyroscope
			Temperature sensor
RFID tag ^{*06}			Bell cup
			SA nozzle
			Air bearing motor

*01 The weight differs depending on the specification

*02 These are the values when the flow rate is 400 cc/min. For other conditions or details, please contact ABB.

*03 The maximum flow rate differs depending on paint viscosity or built-in paint tube

*04 These are general values and do not necessarily guarantee quality.

*05 The sensor is optional

*06 A reader of the RFID tags is optional

RB1000i – connected atomizer for ABB Ability

Will help you secure high paint quality and uptime





ABB