Universidad Miguel Hernández de Elche

MÁSTER UNIVERSITARIO EN ROBÓTICA

"Programación de una estación robótica en RobotStudio y comunicación OPC con PLC S7-1200"

Trabajo de Fin de Máster
2020-2021

Autor: Juan Carlos Rives Sempere
Tutor: Carlos Pérez Vidal

AGRADECIMIENTOS

En primer lugar, quisiera agradecer a mi familia su comprensión por el tiempo robado para la realización de este máster. Sin su apoyo no hubiera sido posible.

A todos los profesores del Máster de Robótica por su constante apoyo e implicación y en especial a mi tutor Carlos Pérez Vidal por las facilidades y la ayuda prestadas para la realización del presente TFM.

RESUMEN

En el presente Trabajo Fin de Máster se lleva a cabo la comunicación entre una estación robótica ABB y un PLC-HMI con el objetivo de poder controlar y adaptar la tarea ejecutada por el robot dependiendo de las condiciones y necesidades de producción.
En primer lugar, se realiza la fase de adquisición de conocimientos y habilidades para poder utilizar el software de programación de robots RobotStudio de ABB haciendo uso de los manuales y tutoriales que se encuentran disponibles en la red.
Una vez adquiridos los conocimientos y habilidades necesarias estableceremos la comunicación entre ambos sistemas (ABB y PLC). La comunicación a establecer será mediante OPC. Para ello se realizan las conexiones entre ambos sistemas con el servidor OPC y posteriormente se enlazarán ambas conexiones para controlar la tarea realizada por el robot.
Una vez que tenemos comunicación entre ambos sistemas se diseña el HMI para que el operario pueda modificar los parámetros del robot y adaptarlos a la producción.
Por último, se programa en RAPID el robot para que dependiendo de la información recibida del HMI este realice una tarea u otra.

Palabras clave: ABB, OPC, RAPID, RobotStudio, RobotWare, SIEMENS, controlador, estación robotizada, proceso industrial, PLC, HMI.

ABSTRACT

In this Master's Final Project, communication is carried out between an ABB robotic station and a PLC-HMI with the aim of being able to control and adapt the task executed by the robot depending on the conditions and production needs.

First, the knowledge and skills acquisition phase is carried out to be able to use ABB's RobotStudio robot programming software using the manuals and tutorials that are available on the net.

Once the necessary knowledge and skills are acquired, communication is established between both systems (ABB and PLC) and this is done through OPC. Specifically, the connections between both systems are made with the OPC server and later both connections will be linked to control the task carried out by the robot.

Once we have communication between both systems, the HMI is designed so that the operator can modify the robot parameters and adapt them to production.

Finally, the robot is programmed in RAPID so that, depending on the information received from the HMI, it performs one task or another.

Keywords: ABB, OPC, RAPID, RobotStudio, RobotWare, SIEMENS, controller, robotic station, industrial process, PLC, HMI.

INDICE DE FIGURAS

1 Introducción, motivación, objetivo

1.1 Introducción

Actualmente la mayoría de las industrias ya incorporan en sus procesos de producción algún tipo (por simple que sea) de automatización. Estas automatizaciones y la aplicación de robots industriales son de gran ayuda para mejorar la productividad, seguridad, calidad e incluso coste de los productos finales fabricados.
Inicialmente la robótica industrial se implementó fundamentalmente en el sector automovilístico que fue pionero en el uso de estas tecnologías. Básicamente los procesos robotizados se usaban para realizar tareas repetitivas o trabajos que requerían grandes esfuerzos físicos. De esta manera se liberaba al operario de las tareas más costosas mejorando así la productividad de las líneas de montaje.
A lo largo de las últimas décadas los procesos robotizados han evolucionado de manera exponencial, al compás de la evolución en otras materias como la informática o la electrónica.
Cada vez se utilizan robots en una gama más amplia de tareas industriales, como pueden ser:

- *Movimiento de piezas.*
- *Tareas pick and place.*
- *Soldadura.*
- *etc.*

Muchos de estos robots están integrados en el propio proceso de fabricación y es por eso que cada vez necesiten de más opciones de configuración para poder interactuar con multitud de elementos distintos dentro del proceso de fabricación (sensores, cámaras, etc.) para mejorar la calidad y productividad de los procesos.
Este amplio abanico de posibilidades de interacción con el entorno hace que se tengan que centralizar las comunicaciones entre diferentes dispositivos y lograr un sistema sencillo y ligero en su instalación a la hora de integrarlo en el proceso. Además, se hace imprescindible dejarlo abierto a posibles modificaciones o ampliaciones, tanto del proceso en el que se integra como del propio sistema de robotización.

1.2 Motivación

La principal motivación para la realización de este proyecto en particular y del Máster en Robótica en general es aprender acerca del mundo de la robótica, tanto por inquietud personal como también en búsqueda de evolucionar positivamente a nivel profesional.

También es una motivación el poder aplicar los diferentes conocimientos adquiridos durante el Máster en Robótica, y poder realizar un trabajo que pudiera ser aplicado en una

línea de producción. Debido a que la aplicación de la robótica en los procesos industriales cada vez es más habitual y necesaria, es muy motivador poder profundizar en este campo.

Con este proyecto pretendo aprender y ganar experiencia en el mundo de la robótica, perfeccionar los conocimientos adquiridos en la programación de robots y ser capaz de comunicar el robot con otros dispositivos (sensores, visión artificial, etc.) para mejorar la productividad de este.

1.3 Objetivos

El presente Trabajo Fin de Máster tiene como principal objetivo adquirir los conocimientos y habilidades necesarios para diseñar y programar una estación de trabajo mediante el software de simulación y programación de robots industriales RobotStudio.

Una vez que se tienen los suficientes conocimientos para diseñar una estación de trabajo se pretende ampliar el campo de aplicación de la estación. Para ello, se realizando la configuración necesaria para establecer una comunicación OPC o PROFINET con un PLC que es el encargado de controlar la línea de producción.

Comunicar la estación de trabajo mediante cualquiera de estos protocolos supone una amplia mejora de esta a la hora de introducir nuevos elementos de control en la línea de producción.

También dotaremos de una colaboración directa entre el operario que controla la línea de producción, mediante un HMI, y la tarea a realizar por el robot. Pudiendo el operario modificar y adaptar la tarea a realizar por la estación robótica dependiendo de las condiciones que en cada momento determinado se necesiten.

1.4 Estructura TFM

El presente Trabajo Fin de Máster está distribuido en siete capítulos, a continuación, se explicará brevemente cada uno de ellos:

Capítulo 1: Introducción. En este capítulo se realiza una breve introducción de la situación actual de la robótica industrial, se expone la motivación para realizar este TFM, se definen los objetivos del presente TFM y se define la estructura del TFM para una mejor visión general de sus contenidos.
Capítulo 2: Introducción a la Robótica. Se introduce el mundo de la robótica, incluyendo un poco de historia y tipos; se habla de robótica industrial y su incorporación a las líneas de producción.

Capítulo 3: Metodología. Se realiza una introducción al programa RobotStudio ABB, después de un pequeño aprendizaje y creación de una estación en un entorno de Simulación. Posteriormente se hace un aprendizaje de la programación RAPID. Y por último se comentan los métodos de comunicación disponibles entre el robot y el entorno.
Capítulo 4: Resultados. Se explicarán los dos tipos de comunicaciones sobre los que se basa el TFM, realizando la configuración y posterior puesta en marcha de las mismas.
A su vez se realiza una programación RAPID incluyendo señales de la comunicación para el control del flujo del programa realizado por el robot.
Capítulo 5: Conclusiones y futuros proyectos. Se exponen las conclusiones del proyecto. Se proponen mejoras o nuevas líneas de investigación para futuros trabajos.
Capítulo 6: Bibliografía. Se indican las referencias bibliográficas utilizadas en la realización de este trabajo.
Capítulo 7: Anexos. En este apartado se añade información complementaria referente a los dispositivos utilizados en este TFM.

2 Estado del arte

2.1 ¿Qué es la robótica?

La robótica [1] [2] en la industria es una rama de la ingeniería que incorpora múltiples disciplinas para diseñar, construir, programar y utilizar los equipos robóticos. Así mismo, se refiere al uso de sistemas de control, computadoras y tecnología de la información para el manejo de diversos procesos y maquinaria en una industria.
El objetivo final es reemplazar el trabajo manual y aumentar la eficiencia, la velocidad y el rendimiento general de los procesos productivos.
El término «robot» viene de la palabra checa «robota», generalmente traducida como «trabajo artificial». Esto describe bastante bien a la mayoría de los robots, diseñados para trabajos de fabricación pesados y repetitivos. De igual manera, manejan tareas que son difíciles, peligrosas o aburridas para los seres humanos.

2.2 Historia de la robótica

Durante siglos el ser humano ha construido máquinas que imitan las partes del cuerpo humano. Los antiguos egipcios unieron brazos mecánicos a las estatuas de sus dioses. Los griegos construyeron estatuas que operaban con sistemas hidráulicos, los cuales se utilizaban para fascinar a los adoradores de los templos.

El inicio de la robótica [3][4] actual puede fijarse en la industria textil del siglo XVIII, cuando Joseph Jacquard inventa en 1801 una máquina textil programable mediante tarjetas perforadas. La revolución industrial impulsó el desarrollo de estos agentes mecánicos, entre los cuales destacaron el torno mecánico motorizado de Babbitt (1892) y el mecanismo programable para pintar con spray de Pollard y Roselund (1939). Además de esto, durante los siglos XVII y XVIII fueron construidos en Europa muñecos mecánicos muy ingeniosos que tenían algunas características de los robots. Jacques de Vauncansos construyó varios músicos de tamaño humano a mediados del siglo XVIII. Esencialmente se trataba de robots mecánicos diseñados para un propósito específico: la diversión. En 1805, Henri Maillardert construyó una muñeca mecánica que era capaz de hacer dibujos. Una serie de levas se utilizaban como "*el programa"* para el dispositivo en el proceso de escribir y dibujar. Estas creaciones mecánicas de forma humana deben considerarse como inversiones aisladas que reflejan el genio de hombres que se anticiparon a su época.

2.3 Cómo funciona la Robótica Industrial

La robótica industrial es una gran industria y evoluciona a una gran velocidad [5]. La tecnología ha cambiado notablemente en las últimas décadas, así como el volumen y la variedad de los usos.
No hace mucho tiempo, los robots estaban presentes únicamente en nuestras pantallas de televisión, en las salas de cine y en juguetes para niños. Casi no tenían ningún papel en nuestra sociedad.
Sin embargo, las cosas han cambiado mucho en los últimos años. Hoy en día, los robots se encargan de numerosas tareas que antes realizaban los humanos, como ensamblar cosas (coches, electrodomésticos, aparatos), realizar cirugías, pintar piezas etc.
Existe una gran variedad de aplicaciones en la automatización industrial que la robótica industrial está preparada para realizar.
Por lo general, los robots se diseñan o integran con una tarea específica y se adaptan para satisfacer las necesidades únicas de esa tarea.

2.4 ¿Cuál fue el primer robot industrial?

Desde que el primer robot industrial (Fig. 1), Unimate [6] [7], se instaló en una planta de General Motors en los años 50, la automatización industrial se ha asociado con grandes empresas que realizan operaciones enormes que implican líneas de producción masivas.

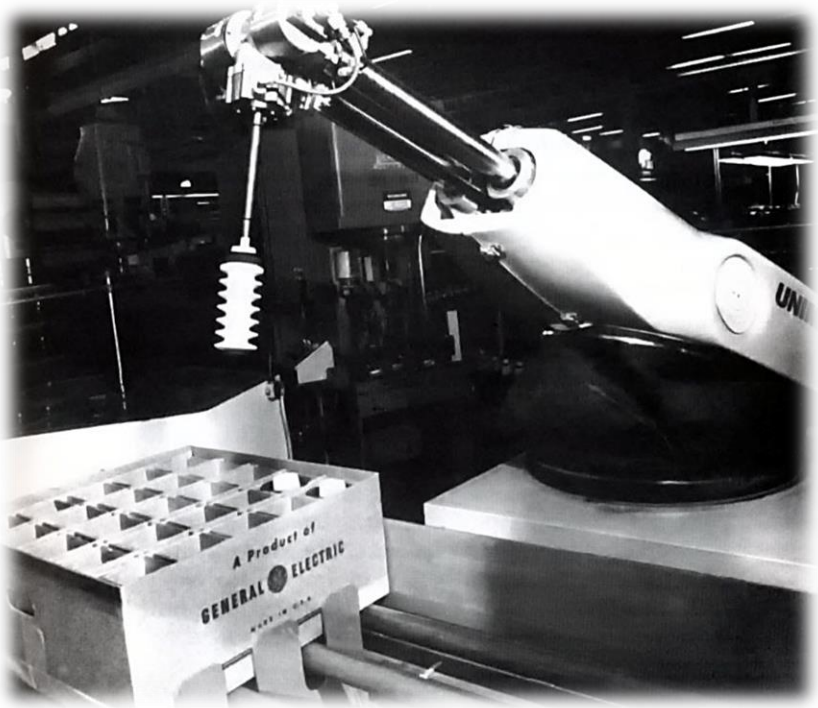

Figura 1 Unimate

El tamaño, la forma y la dinámica de los robots industriales reflejaron durante mucho tiempo esta realidad. Los primeros robots eran grandes, ruidosos, peligrosos, y requerían sus propias jaulas para mantenerlos bien alejados de las cargas de trabajo en las empresas. En la última década se ha producido una transformación radical de la automatización industrial, que es tanto un cambio tecnológico como una consecuencia del cambio de la economía, que depende cada vez más de las tiradas pequeñas, el transporte rápido y las operaciones ágiles.
Una nueva generación de robots refleja estos cambios. Son rápidamente desplegables, hábiles para las tareas, más pequeños que sus torpes antepasados, y pueden trabajar junto a los operarios fuera de las jaulas.

2.5 Componentes de un robot industrial

El diseño de un robot industrial se basa en tres pilares fundamentales: la mecánica, la electrónica y la informática.

- La mecánica, que representa la parte que realmente se ve del robot. Esta parte es importante para la precisión, la velocidad y la carga útil del robot.
- La electrónica, que permite que el sistema de control dirija todos los motores y obtenga información del entorno a partir de los sensores.
- La informática que hace al robot «inteligente» a través de la colaboración con el entorno del robot y el usuario.

Hasta ahora los robots han sido utilizados para tareas muy específicas. Con mejores ordenadores e inteligencia artificial, podrán evolucionar hacia un nuevo tipo de inteligencia. Los robots pronto serán capaces de interactuar plenamente con su entorno.

2.6 Como funciona un robot industrial

El robot de fabricación más común es el brazo robótico industrial. Un brazo robótico típico está compuesto por siete segmentos de metal, unidos por seis articulaciones. El sistema informático controla el robot a través de motores rotatorios individuales conectados a cada articulación (algunos brazos más grandes utilizan la hidráulica o la neumática).
A diferencia de los motores normales, los motores paso a paso se mueven en incrementos exactos. Esto permite al software mover el brazo con mucha precisión, repitiendo exactamente el mismo movimiento una y otra vez. El robot utiliza sensores de movimiento para asegurarse de que se mueve en la cantidad justa.

El trabajo de nuestro brazo es mover nuestra mano de un lugar a otro. De manera similar, el trabajo del brazo robótico es mover un extremo efector de un lugar a otro. Podemos

equipar los brazos robóticos con todo tipo de efectores finales, que se adaptan a una aplicación particular. Un efector final común es una versión simplificada de la mano, que puede agarrar y transportar diferentes objetos.
Las manos robóticas, normalmente, tienen sensores de presión incorporados que le dicen al ordenador cómo de fuerte está agarrando el robot un objeto en particular. Esto evita que al robot se le caiga o se le rompa lo que esté llevando. Otros efectos finales incluyen sopletes, taladros, aerosoles de pintura u otros.
Los robots industriales están diseñados para hacer exactamente lo mismo, en un ambiente controlado, una y otra vez.

Para enseñar a un robot a hacer su trabajo, el programador guía el brazo a través de los movimientos usando un controlador de mano. El robot almacena la secuencia exacta de movimientos en su memoria, y lo hace una y otra vez cada vez que una nueva unidad baja por la línea de montaje.

La mayoría de los brazos robóticos trabajan en líneas de fabricación de automóviles. Los robots pueden hacer este trabajo mucho más eficientemente que los seres humanos porque son muy precisos y no sufren la fatiga propia de los seres vivos. Siempre perforan en el mismo lugar y siempre aprietan los tornillos con la misma fuerza, sin importar cuántas horas hayan trabajado.

Los robots de fabricación también son muy importantes en la industria informática. Se necesita una mano increíblemente precisa para armar un microchip diminuto.

2.7 Para qué sirve la robótica industrial

Los robots son máquinas programables capaces de realizar una serie de acciones de forma autónoma o semiautónoma. Interactúan con el mundo físico a través de sensores y actuadores. Al ser reprogramables, son más flexibles que las máquinas de una sola función.

Dentro de la automatización industrial, los robots se utilizan como una forma flexible de automatizar una tarea o proceso físico. Los robots de colaboración están diseñados para llevar a cabo la tarea de la misma manera que lo haría un humano. Los robots industriales más tradicionales tienden a llevar a cabo la tarea más eficientemente que un humano.
Las aplicaciones de la robótica son infinitas. A menudo, los robots se diseñan o integran con una tarea específica y se adaptan para satisfacer las necesidades únicas de esa tarea.
Algunas formas comunes de automatización de la robótica industrial incluyen:

- Soldadura por arco (coches)
- Operaciones de montaje
- Pintura y pulverización
- Corte por láser
- Placas de circuito electrónicos
- Embalaje
- Inspección del producto
- Medición y verificación por láser del ensamblaje de las piezas
- Robots móviles (por ejemplo, en áreas sensibles)

Gracias a los robots, todas esas operaciones se procesan con gran resistencia, velocidad y precisión.

2.8 Clasificación de la robótica industrial

2.8.1 Robots de primera generación

Los robots de primera generación son dispositivos que actúan como "esclavo" mecánico de un hombre (Fig. 2), quien provee mediante su intervención directa el control de los órganos de movimiento.

Esta transmisión tiene lugar mediante servomecanismos actuados por las extremidades superiores del hombre, caso típico manipulación de materiales radiactivos, obtención de muestras submarinas, etc.

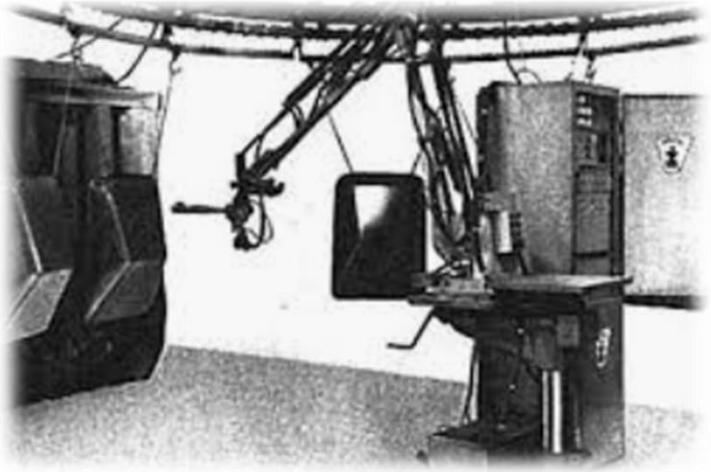

Figura 2 Robot primera generación

2.8.2 Robots de segunda generación

Este otro tipo de dispositivos actúan automáticamente sin intervención humana frente a posiciones fijas, en las que el trabajo ha sido preparado y ubicado de modo adecuado ejecutando movimientos repetitivos en el tiempo, que obedecen a lógicas combinatorias, secuenciales, programadores paso a paso, neumáticos o Controladores Lógicos Programables. Un aspecto muy importante está constituido por la facilidad de rápida reprogramación que convierte a estos Robots en unidades "versátiles" cuyo campo de aplicación no sólo se encuentra en la manipulación de materiales sino en todos los procesos de manufactura, como, por ejemplo: en el estampado en frío y en caliente asistiendo a las máquinas-herramientas para la carga y descarga de piezas; en la inyección de termoplásticos y metales no ferrosos, en los procesos de soldadura a punto y continúa, en tareas de pintado, etc.

2.8.3 Robots de tercera generación

Son dispositivos que, habiendo sido construidos para alcanzar determinados objetivos, serán capaces de elegir la mejor forma de hacerlo teniendo en cuenta el ambiente que los circunda. Para obtener estos resultados es necesario que el robot posea algunas condiciones que posibiliten su interacción con el ambiente y los objetos. Las mínimas aptitudes requeridas son: capacidad de reconocer un elemento determinado en el espacio y la capacidad de adoptar propias trayectorias para conseguir el objetivo deseado. Los métodos de identificación empleados hacen referencia a la imagen óptica por ser esta el lenguaje humano en la observación de los objetos, Sin embargo, no puede asegurarse que la que sea natural para el hombre constituya la mejor solución de observación para el robot.

2.9 Tipos de robots industriales

Los robots que operan en la industria pueden clasificarse según diferentes criterios como: el tipo de movimiento (grados de libertad), la aplicación a ejecutar (proceso de fabricación), la arquitectura (en serie o en paralelo) y la marca. También existe un nuevo calificativo para los robots industriales, y es que puedan ser colaborativos o no.
A continuación, se describen algunas de las formas industriales más comunes de automatización robótica [8].

2.9.1 Robot Cartesiano

Un robot cartesiano (Fig. 3) se mueve a lo largo de 3 ejes (X, Y, Z) con una trayectoria lineal y la herramienta de efecto final siempre mantiene la misma orientación. Esta solución es una de las más fáciles y baratas.
Por ejemplo, los robots cartesianos están bien diseñados para la impresión en 3D. Sin embargo, debido a la falta de gestión de la orientación, los casos de uso son de alguna manera limitados.

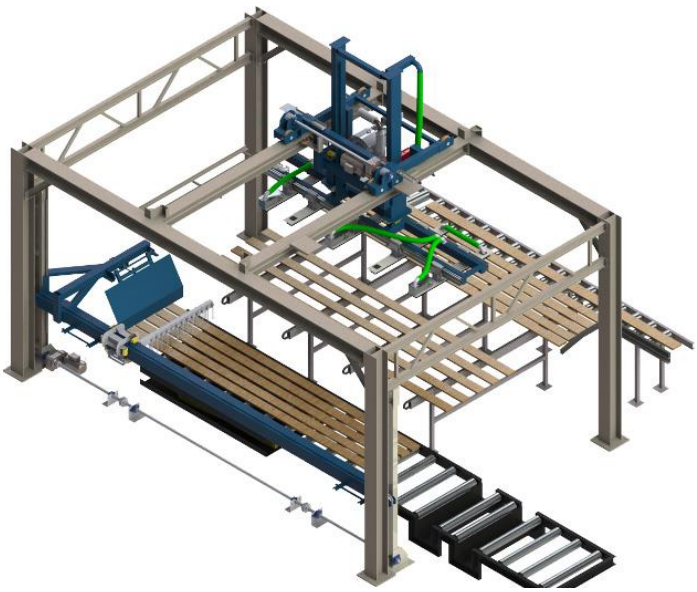

Figura 3 Robot Cartesiano

2.9.2 Robots con múltiples grados de libertad

Un robot multi-DOF (Fig. 4) tiene más articulaciones y más ejes. Una configuración bien conocida es un robot con 3 ejes para moverse y 3 ejes para orientar el efector final (6 grados de libertad). El robot puede alcanzar cualquier punto con cualquier orientación.
Esta solución es mejor para operaciones multitarea y complejas, aunque el precio es mayor. Las posibilidades de uso de los robots polares son infinitas.

Figura 4 Robots con múltiples grados de libertad

2.9.3 Robot SCARA

Un robot SCARA (Fig. 5) es una mezcla de los dos robots de arriba. Puede moverse a lo largo de 3 ejes (X, Y, Z) pero tiene un eje más para orientar el efector final en una dirección. Este tipo de robot es más específico y funciona bien para tareas de «recoger y colocar».

Figura 5 Robots SCARA

2.9.4 Robot redundante

Los robots redundantes (Fig.6) también pueden posicionar completamente su herramienta en una posición determinada. Pero mientras que los robots de 6 ejes sólo pueden tener una postura para una posición dada de la herramienta, los robots redundantes pueden acomodar una posición dada de la herramienta bajo diferentes posturas.
Es como el brazo humano que puede sostener un mango fijo mientras mueve las articulaciones del hombro y el codo.

Figura 6 Robot Redundante

2.10 Beneficios de la robótica industrial

La robótica industrial ha transformado la industria manufacturera por una razón: tiene muchos beneficios. Su primer y más importante beneficio es su eficiencia. Completan las tareas más rápidamente que las labores manuales, y su tiempo de funcionamiento es significativamente mayor.

La combinación de velocidad de ejecución y tiempo de funcionamiento sin sufrir fatiga conduce a un mayor rendimiento con menores costes operativos.

Además, los robots, cuando se programan adecuadamente, son por naturaleza altamente replicables. Esto mejora la producción de forma drástica, aumentando la calidad general del producto y reduciendo los residuos.

Los brazos robóticos industriales suelen ofrecer un gran retorno de la inversión (ROI) a pesar de los altos costes iniciales. Los beneficios de productividad derivados de la eficiencia, la consistencia y la reducción de los costes operativos se suman rápidamente, lo cual forma parte de lo que ha hecho que los robots industriales sean tan populares entre los fabricantes durante la última década.

Los robots ofrecen muchos beneficios en el balance final, independientemente del tipo de robot industrial que se esté implementando. Mientras un robot esté programado adecuadamente y se adapte a las necesidades únicas de una determinada aplicación, es casi seguro que superará el trabajo manual.

El mercado de los robots de fabricación es grande y crece rápidamente. La tecnología utilizada cambia con la misma rapidez. Puede ser difícil mantenerse al ritmo de los rápidos cambios en la industria de la robótica, pero comprender los tipos de robots industriales y los beneficios que proporcionan es un gran comienzo.

3 Metodología

3.1 Estructura estación robótica industrial

Cuando se habla del uso de una aplicación robótica en el entorno industrial su campo de aplicación es muy extenso, como, por ejemplo:

- *Aplicación de movimiento de una pieza.*
- *Aplicación de pick and place.*
- *Aplicación de soldadura.*
- *etc.*

Lo que tienen en común todas estas aplicaciones, por muy simples o complejas que puedan ser, son los elementos básicos (Fig. 7) de los que se compone un sistema robótico. Estos elementos básicos sin los cuales no existiría la aplicación son:

- *Brazo Robótico*
- *Controladora.*
- *Consola Programación Portátil.*
- *PC Programación.*

Figura 7 Elementos básicos de un sistema robótico

La principal función de cada uno de estos elementos es:

Brazo Robótico, es la cadena cinemática encargada de realizar los movimientos para posicionar la herramienta de trabajo en el punto deseado.

Está compuesta por eslabones, articulaciones y motores, los cuales mediante su control realizan los movimientos deseados.

***Controladora**, es la encargada de la ejecución del programa diseñado por el usuario para que el robot realice la tarea deseada.*

Está compuesta por una unidad de potencia para el control de los motores, una unidad de control que es la encargada de ejecutar el programa de usuario y una interface E/S para su conexión con el entorno.

***FlexPendant**, es una consola de programación portátil, muy útil para operaciones de campo y pequeñas modificaciones. Esta unidad dispone de casi todas las opciones disponibles en el software de programación.*

***Software Programación**, RobotStudio; este software permite la programación necesaria para que la aplicación realice la tarea asignada.*

En este software se puede virtualizar la aplicación a realizar y de esta manera poder hacer modificaciones y ajustes en la programación sin que afecte a la producción (si el robot está en funcionamiento) o realizar estudios de cómo se comportaría una nueva aplicación antes de ser montada en campo.

3.2 Introducción a RobotStudio

Para la realización de este TFM hemos utilizado el software de programación de ABB RobotStudio [9] [10], que es un software de elevado coste de adquisición. Por ello, para la realización de este trabajo, utilizaremos la última versión del software, RobotStuido 2021.2 mediante una licencia en red proporcionada por la universidad.
También tendríamos la opción de usar una versión de prueba de 30 días, pero al disponer la licencia en red de la universidad nos decantamos por esa opción.

3.2.1. Interface RobotStudio

Al arrancar el software nos aparece el menú de inicio de este (Fig. 8), en el que nos ofrece varias opciones para crear estaciones nuevas.

Por un lado, tenemos la opción de crear una estación vacía. Esta es la opción más larga, pero a la vez es la más útil para familiarizarse con el software.

Otra opción es la de crear una solución con estación vacía; es muy similar a la anterior.

La última opción que nos ofrece el software es la de crear una estación con un controlador virtual. Esta opción ya nos ahorraría todos los pasos que tendríamos que hacer con la estación vacía, pero de esta manera no nos familiarizaríamos tanto con el software. Esta última opción también nos permite crear una estación mediante una copia de seguridad realizada anteriormente.

Como he comentado antes, para iniciar la familiarización con el entorno RobotStudio realizamos una nueva estación vacía. De esta manera trabajamos la inserción de los demás

elementos de la estación de trabajo de manera manual, siendo esta la manera más didáctica.

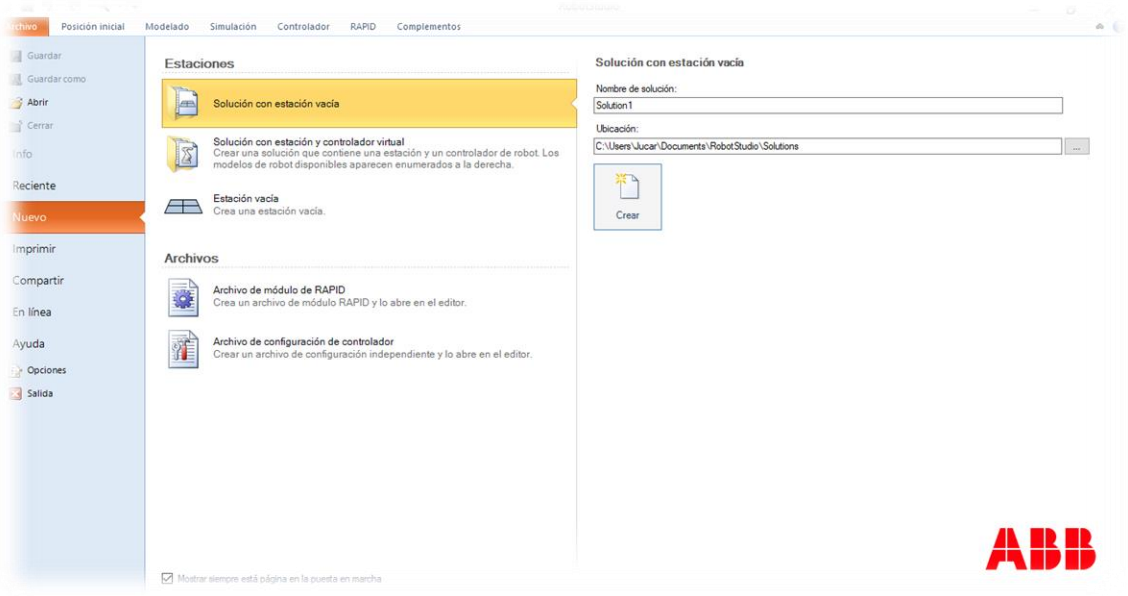

Figura 8 Menú inicio software RobotStudio

3.2.2 Creación de una estación de trabajo

Para familiarizarnos con el entorno de programación RobotStudio [11] comenzaremos con los primeros pasos a la hora de crear una estación y su posterior programación.

Una vez que tenemos la estación creada vemos como el entorno de trabajo es muy similar a casi todos los programas de la actualidad, por lo que el manejo entre pestañas, menús y demás opciones es muy similar a los programas a los que estamos habituados, por no decir igual.

En la pestaña inicial (Fig. 9) es donde tenemos los principales elementos de diseño de nuestra estación y los elementos para la creación de trayectorias a realizar por el robot.

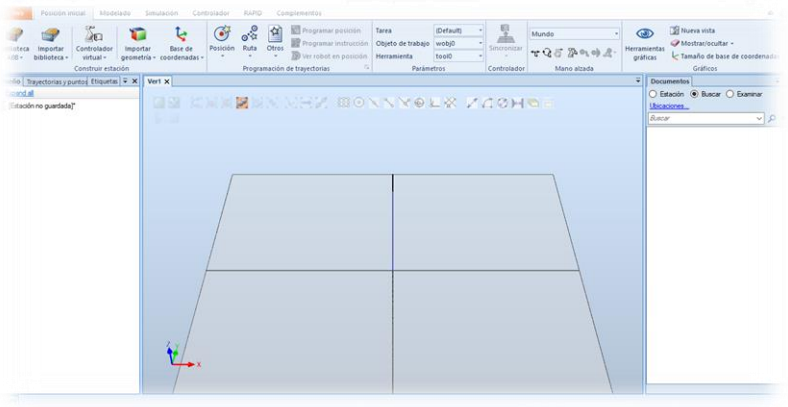

Figura 9 Pestaña inicial software RobotStudio

En la pestaña modelado (Fig. 10) será donde tengamos herramientas de diseño gráfico para crear y modificar objetos diseñados por nosotros mismos. Es una herramienta de diseño gráfico muy limitada, si tenemos que crear objetos muy complejos debemos recurrir a otros software de diseño más avanzados.

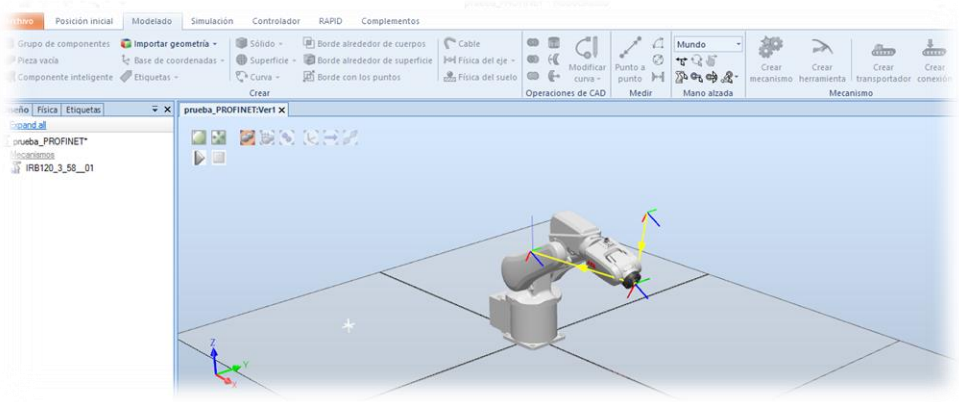

Figura 10 Pestaña modelado software RobotStudio.

En la pestaña simulación (Fig. 11) será donde configuremos y pongamos en marcha la simulación de la estación creada. También podremos manejar las diversas opciones de las que disponemos a la hora de realizar la simulación de la estación, como puede ser el de grabar la simulación entre otras muchas opciones.

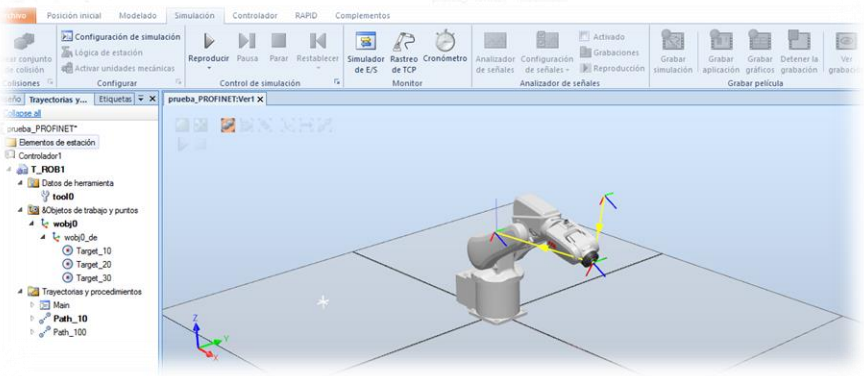

Figura 11 Pestaña simulación software RobotStudio

En la pestaña controlador (Fig. 12) será en la que realizaremos todas las modificaciones referentes al controlador IRC5 del ROBOT, como pueden ser la configuración de las comunicaciones, añadir opciones, reiniciar controladora, realizar copia de seguridad, etc.

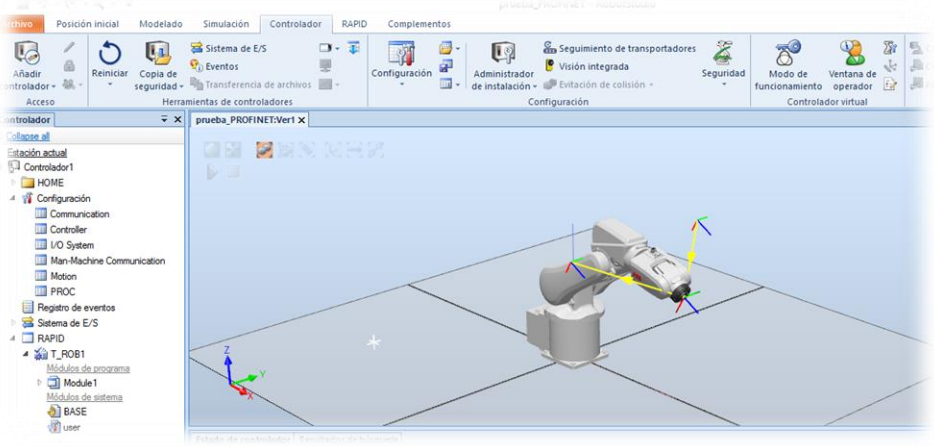

Figura 12 Pestaña controladora software RobotStudio

En la pestaña RAPID (Fig. 13) será donde tengamos ubicado el código de programación del ROBOT. En ella podremos realizar modificaciones del programa, modificar posiciones, iniciar simulación, etc.

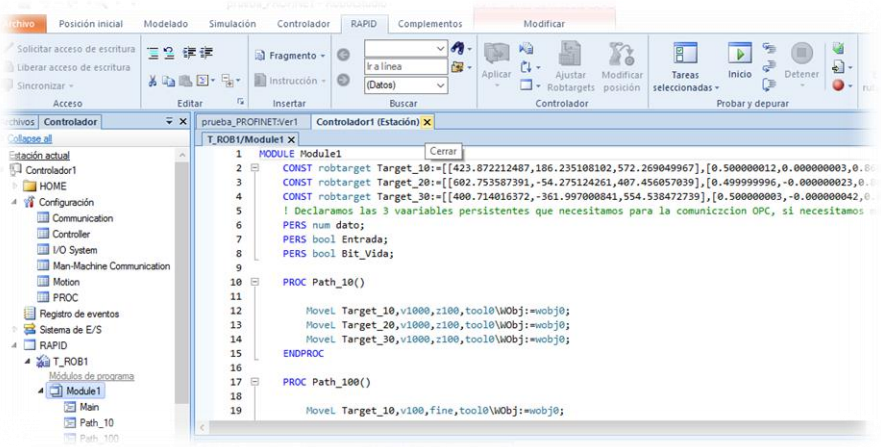

Figura 13 Pestaña RAPID software RobotStudio

Por último, en la pestaña complementos (Fig. 14) será donde podremos ver los paquetes instalados en el ROBOT, versión de ROBOTWARE instalada y desde esta pestaña podremos instalar nuevos paquetes y apps.

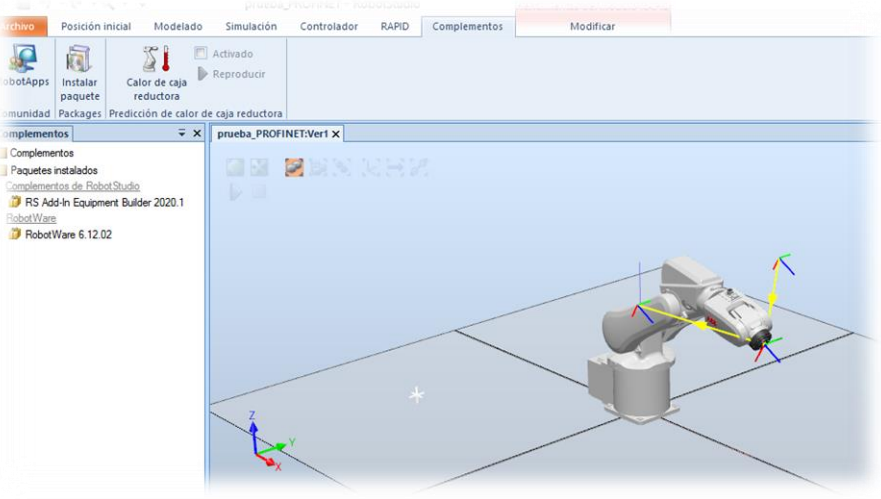

Figura 14 Pestaña complementos software RobotStudio

3.2.2.1 Insertar un robot en una estación de trabajo

Una vez vistas las diferentes pestañas disponibles en RobotStudio y una vez creada la estación vacía, el primer paso será el de insertar un robot (Fig. 15) a esa estación para iniciar la configuración de la estación deseada. Para ello seguimos los pasos que nos aparecen cuando seleccionamos la opción de “Biblioteca ABB” en el menú “posición inicial”.

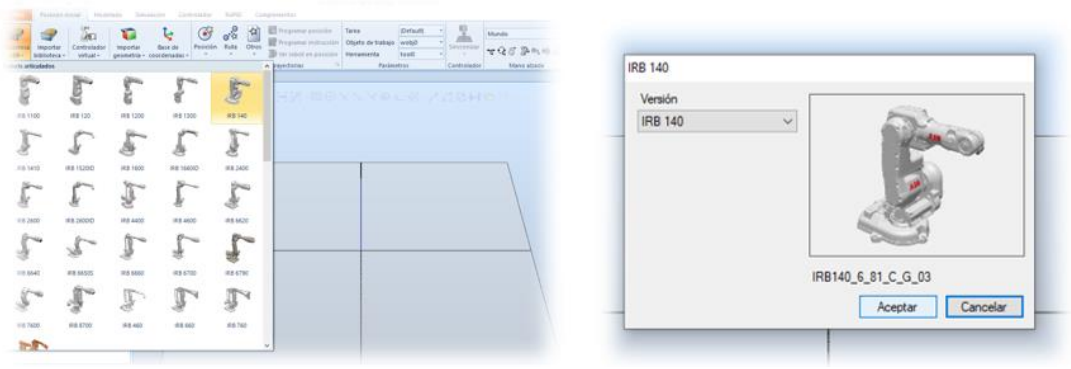

Figura 15 Insertar robot en estación vacía

Una vez que tenemos el robot insertado en la estación todavía no podemos hacer mucho con él ya que no tenemos insertado el controlador de este.

En este estado podríamos mover la ubicación del robot sobre la estación (Fig. 16) para colocarlo en su ubicación definitiva. Para ello podemos moverlo de varias formas.

La primera seria desde la "pestaña inicial" seleccionando la opción de "mano alzada". De esta manera movemos manualmente todo el ROBOT o solamente los eslabones seleccionados.

La otra forma de moverlo es desde el "árbol de diseño", seleccionando el robot insertado y mediante botón derecho de ratón acceder a "posición"; de esta manera se abre un desplegable donde podemos introducir la nueva posición del ROBOT.

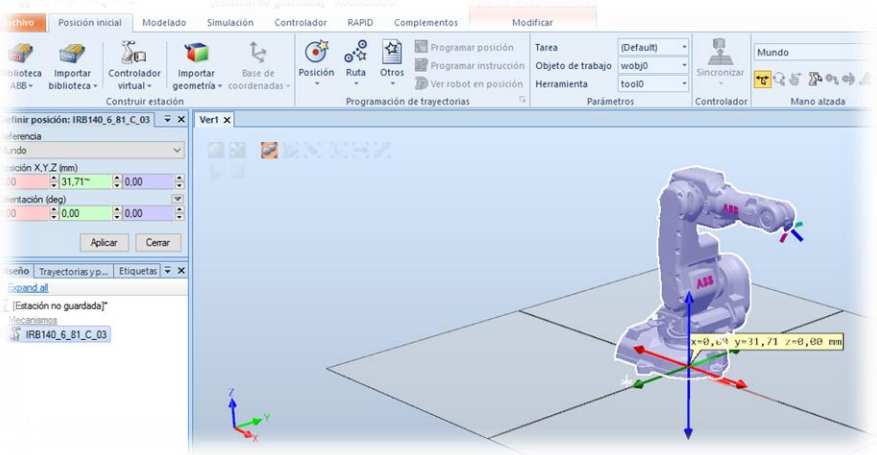

Figura 16 Movimiento ROBOT manualmente sobre estación vacía

3.2.2.2 Insertar una controladora en una estación de trabajo

Una vez introducido el robot y colocado en su posición definitiva en el entorno de trabajo, el siguiente paso sería el de insertar el controlador del robot en la estación de trabajo.

Para introducir el controlador, desde la “pestaña inicial” seleccionaremos la opción de “controlador virtual” (Fig.17) y seguiremos los pasos indicados para crear el nombre del controlador, seleccionar su ubicación y seleccionar la versión de ROBOTWARE a instalar.

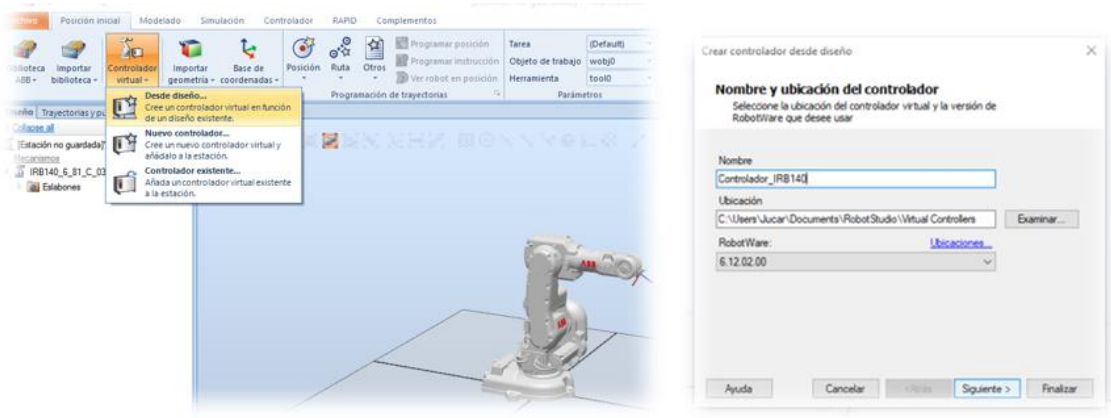

Figura 17 Creación controlador sobre estación vacía

Una vez realizados los pasos anteriores, nos aparece el menú (Fig. 18) para seleccionar las opciones del controlador. En este caso para poder probar las conexiones OPC y PROFINET, que veremos más adelante, tenemos que seleccionarlas en el menú desplegable.

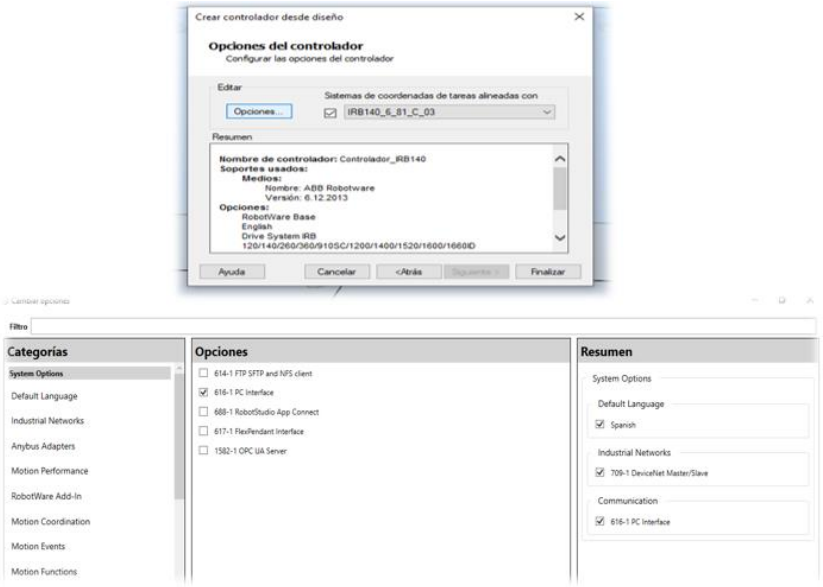

Figura 18 Selección opciones controlador

Una vez seleccionadas las opciones y al darle “finalizar” se creará el controlador y se reiniciará para insertarlo en la estación de trabajo. Esos pasos los podemos ver en la parte inferior derecha de la pantalla (Fig. 19). La ventana pasará de rojo a amarillo y si todo es correcto se pondrá verde; será en este momento cuando el ROBOT y la controladora estén correctamente integrados en la estación y podamos realizar movimientos del robot a través de la controladora.

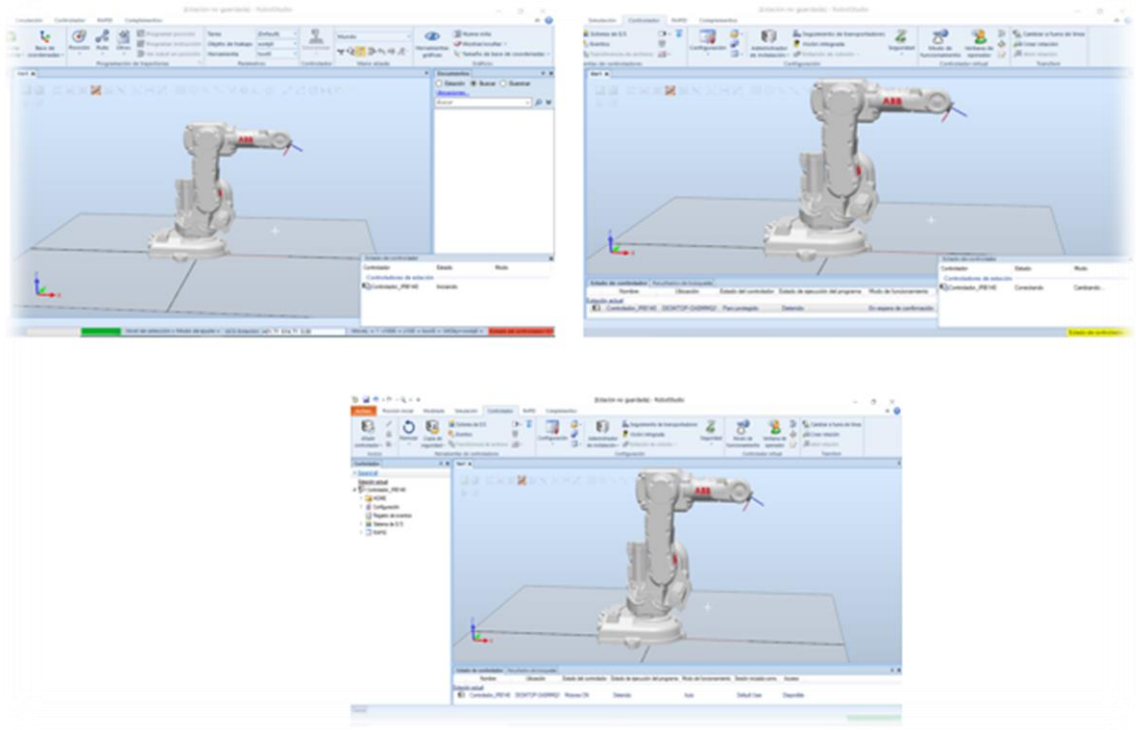

Figura 19 Reinicio controlador

Una vez insertado el controlador en la estación de trabajo y reiniciada la misma, podemos ver como en el menú “Mano Alzada” se han habilitado varias opciones más. Esto también es un indicador de que el controlador y el ROBOT están correctamente conectados y en funcionamiento.

Estas opciones habilitadas (Fig.20) nos permitirán mover el ROBOT mediante el controlador por lo que, si seleccionamos la muñeca esférica del ROBOT y lo movemos, veremos cómo se mueven todos los ejes del ROBOT, controlados por la controladora.

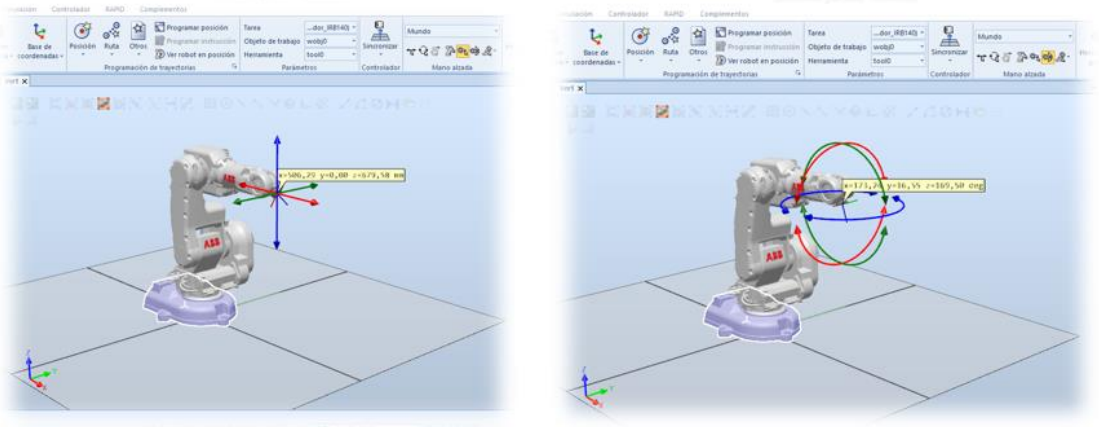

Figura 20 Habilitación opciones menú mano alzada

3.2.3 Creación de la herramienta del robot

Una vez que tenemos la controladora insertada y en marcha, el siguiente paso será añadir una herramienta al ROBOT.

Para añadir una herramienta tenemos 2 opciones: por un lado, podemos crear una manualmente mediante el modelado, y por otro lado podemos seleccionar una del catálogo disponible en el programa.

3.2.3.1 Creación de la herramienta manualmente

Para crear una herramienta manualmente lo haremos desde la pestaña “Modelado” (Fig. 21) y seleccionaremos un “Sólido” y en concreto un cono.

Figura 21 Creación de objeto en estación

Una vez seleccionado el objeto a insertar, se desplegará un menú a la izquierda (Fig. 22) de la pantalla en el que tendremos que insertar las dimensiones del objeto a crear. Una vez introducidas las medidas del objeto a crear, este nos aparecerá en la estación de trabajo.

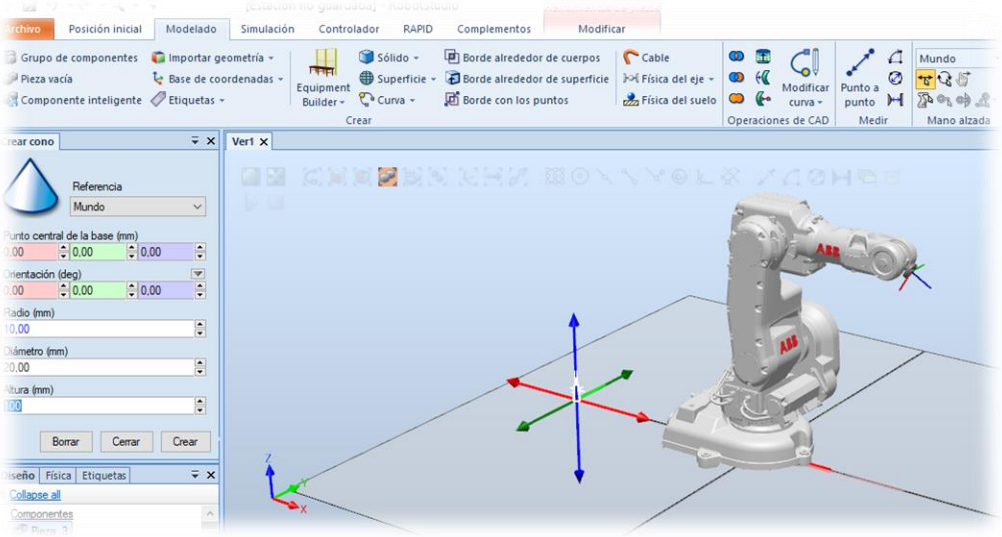

Figura 22 Dimensiones nuevo objeto en estación

Cuando tenemos creado el nuevo objeto en la estación de trabajo, el siguiente paso sería colocarlo en la muñeca esférica de nuestro ROBOT para utilizarlo como herramienta.

Para colocarlo en la muñeca esférica del ROBOT podemos hacerlo arrastrando el objeto creado hasta el ROBOT de la estación. Lo haremos desde el árbol situado a la izquierda

de la pantalla (Fig. 23) o en ese mismo árbol seleccionando el objeto creado y con el botón derecho seleccionar “Conectar a” y en el desplegable que se abre deberíamos seleccionar el ROBOT de nuestra estación.

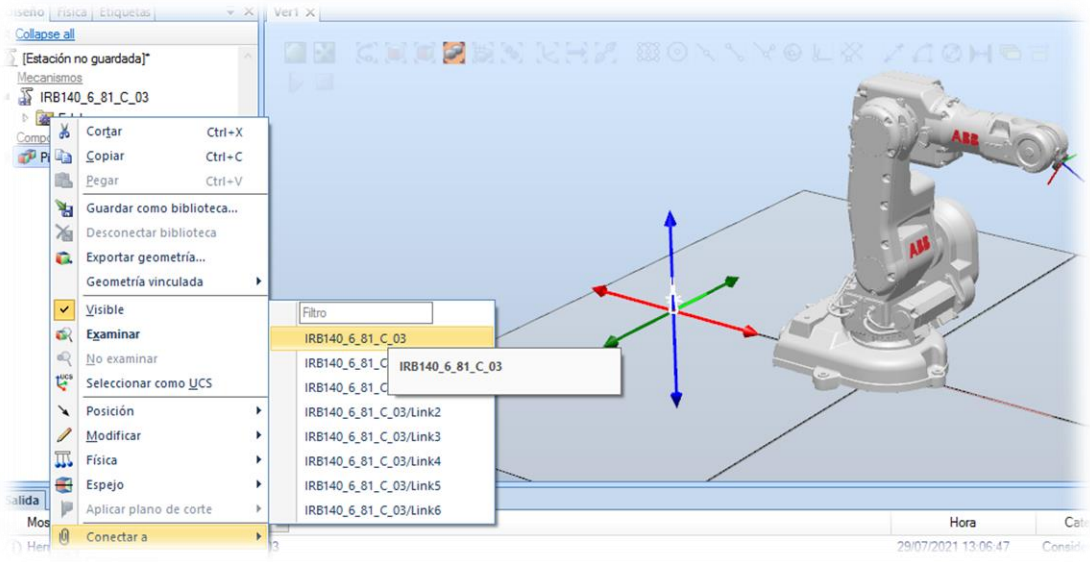

Figura 23 Insertar objeto en ROBOT

Una vez que tenemos el objeto creado podremos insertarlo en la muñeca esférica del ROBOT, pero este objeto todavía no es una herramienta y si lo insertamos en el ROBOT podremos ver como el TCP (Punto Central de la Herramienta) continúa siendo la muñeca esférica de nuestro ROBOT (Fig.24) y, por lo tanto, si realizamos movimientos del robot estos utilizarán un TPC que no es el correcto.

Para poder usar este objeto como herramienta lo que tenemos que hacer es definir este objeto como herramienta para de esta manera poder insertarlo en el ROBOT como una herramienta y que los movimientos del ROBOT utilicen el TCP creado por esta nueva herramienta.

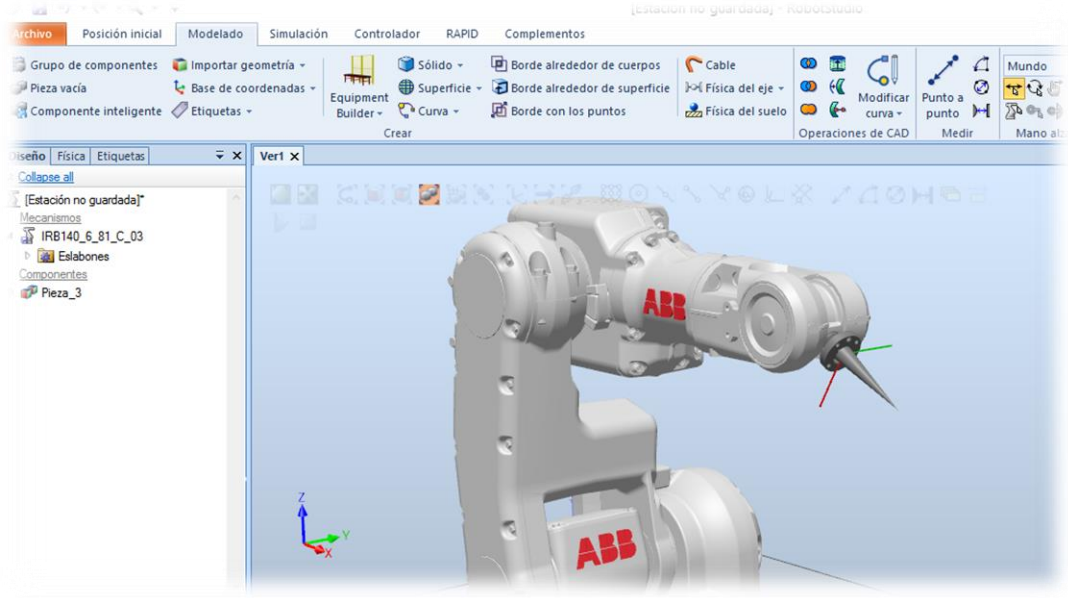

Figura 24 Objeto insertado en ROBOT, pero no como herramienta

Para convertir el objeto creado a herramienta de nuestro ROBOT tendremos que hacerlo manualmente. Para ello en la pestaña “Modelado” (Fig. 25) seleccionaremos la opción de “Crear Herramienta” y realizaremos los 2 pasos de los que consta esta opción.

En el primer paso le daremos un nombre a la nueva herramienta y seleccionaremos sobre qué objeto queremos crearla. También deberíamos introducir los datos que pide el menú desplegable, si los conocemos, para que el sistema los tenga en cuenta a la hora de crear las trayectorias.

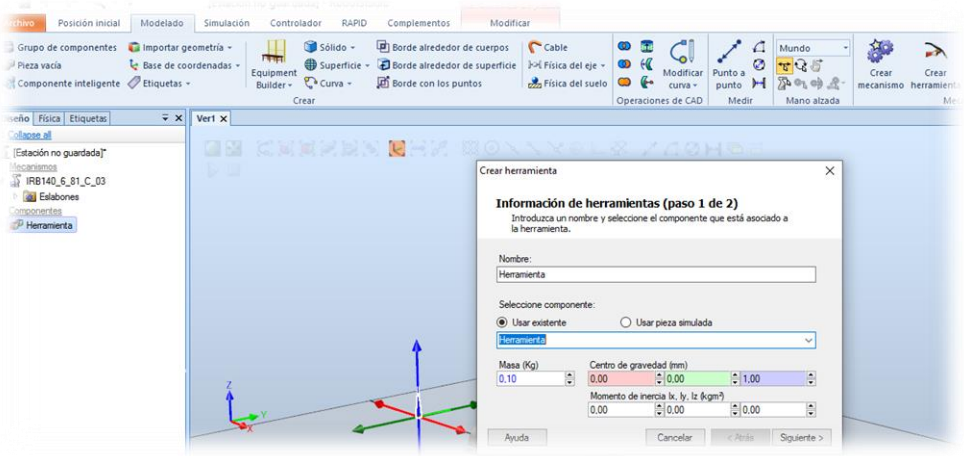

Figura 25 Creación de herramienta sobre objeto

En el segundo paso (Fig. 26) lo que tenemos que hacer es definir el nuevo TCP del robot, incluyendo ahora la herramienta creada, de modo que al finalizar la configuración el robot tome ese nuevo TCP como referencia y todos los movimientos que haga sean sobre él.

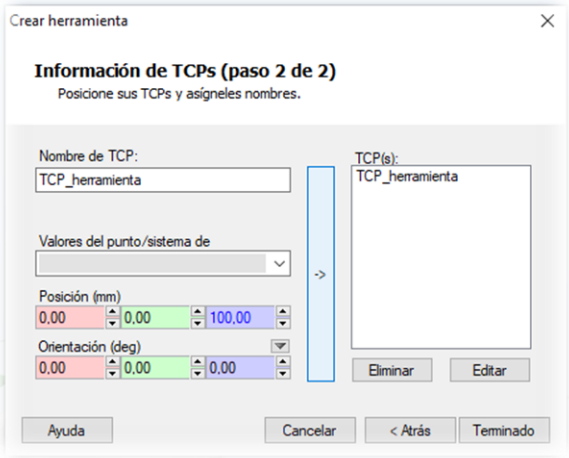

Figura 26 Definición nuevo TCP sobre herramienta creada

Una vez finalizada la creación de la herramienta y establecido el nuevo TCP, al insertarla en el ROBOT (Fig. 27) ya la tenemos disponible para reposicionar el ROBOT mediante ese nuevo TCP.

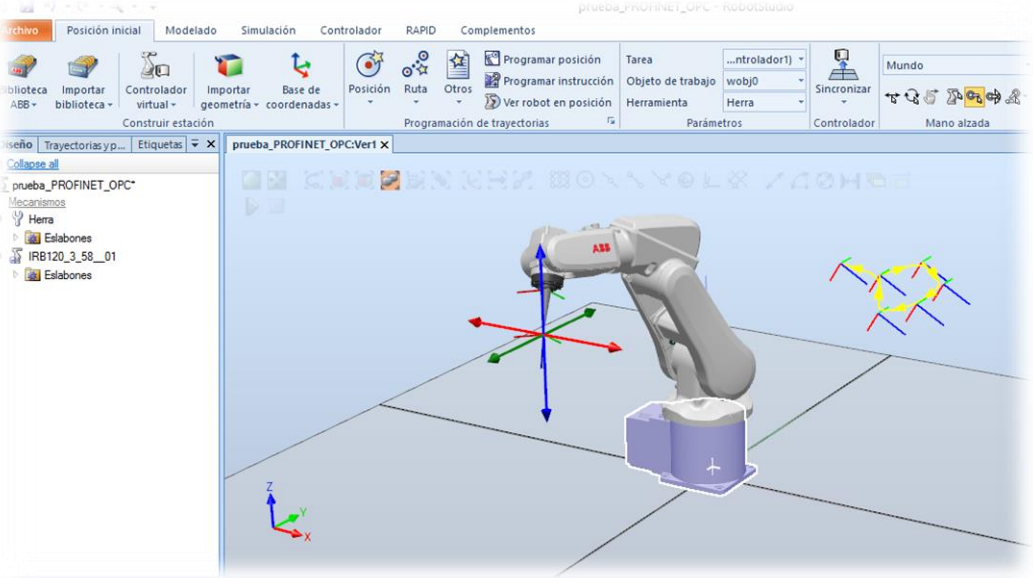

Figura 27 Definición nuevo TCP sobre herramienta creada

3.2.3.2 Creación de la herramienta usando biblioteca de RobotStudio

La otra forma de insertar una herramienta en el ROBOT es utilizar las herramientas disponibles en la biblioteca de RobotStudio (Fig.28). Para ello dentro de la pestaña "Posición inicial" seleccionaremos en el desplegable la opción de "importar biblioteca" y una vez dentro de este menú seleccionar "equipamiento" y a su vez seleccionar la herramienta que se adapte a la aplicación a realizar.

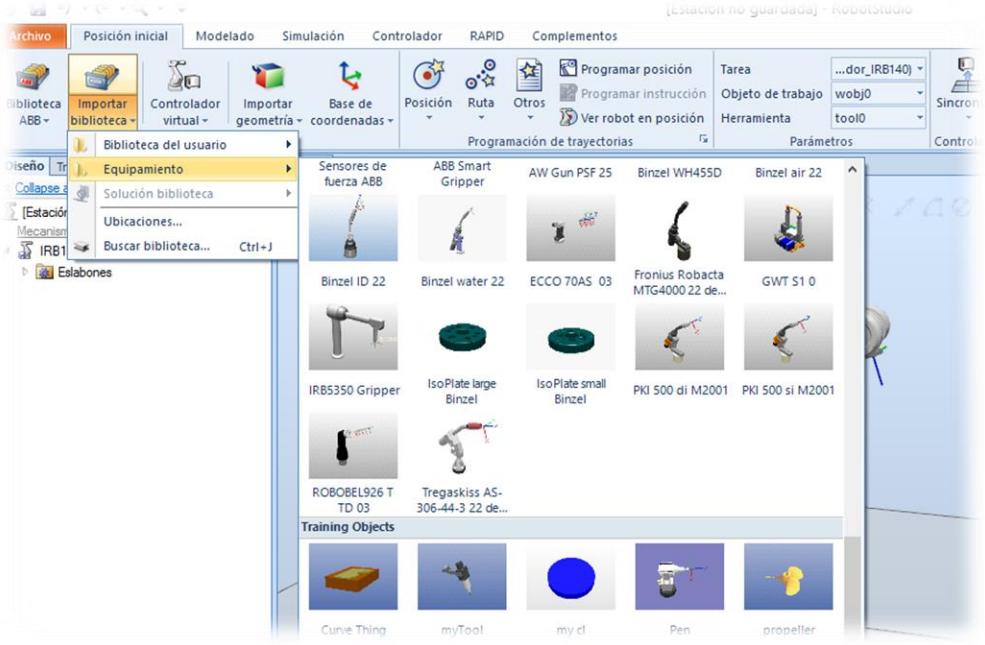

Figura 28 Selección herramienta desde biblioteca RobotStudio

Una vez que tenemos la herramienta seleccionada e insertada en la estación de trabajo, el procedimiento para insertarla en el ROBOT cualquiera de los dos que hemos seguido para insertar la herramienta creada manualmente. Una vez insertada (Fig. 29) ya podremos hacer uso de la herramienta en el ROBOT.

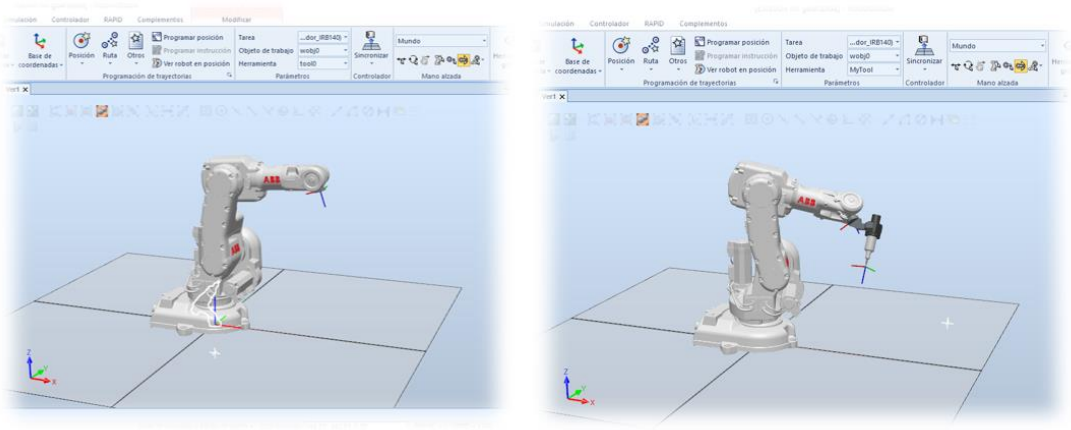

Figura 29 Herramienta de biblioteca insertada ROBOT

3.2.4 Creación de Workobjects

Una cosa que debemos tener en cuenta a la hora de crear trayectorias es saber sobre que Workobject se van a crear estas trayectorias, ya que dependiendo de sobre que Workobject esté creada será necesario reposicionar esos puntos si el objeto sobre el que está referenciado se mueve.

Para evitar tener que reposicionar los puntos creados, crearemos un punto inicial (HOME) referenciados sobre el Workobject 0, que es el objeto de trabajo principal y que tiene como referencia el eje de coordenadas de la base del robot.

Posteriormente crearemos un Workobject de usuario para referenciar los demás puntos sobre los que trazaremos la trayectoria deseada. De esta manera si reposicionamos el objeto sobre el que se han creado esos puntos, estos se reposicionaran también, no siendo necesario reposicionarlos manualmente como sucedería si estuviesen referenciados al Workobject 0.

3.2.5 Creación de trayectorias

Una vez creada la estación de trabajo, insertado el controlador y colocado una herramienta el siguiente paso será el de marcar unas posiciones manualmente para posteriormente crear una ruta sobre esos puntos.

A continuación, veremos la forma para crear trayectorias moviendo el robot y marcando unos puntos en la estación de trabajo y otra forma que será la creación de trayectorias en sobre objetos insertados en la estación de trabajo.

3.2.5.1 Marcando puntos en estación trabajo

Para la creación de trayectorias sobre unos puntos marcados en la estación de trabajo, mediante la opción “mano alzada” movemos robot a la posición donde queremos crear el punto y seleccionamos “Programar posición”. De esta manera se quedará marcado el punto en la posición seleccionada. Al marcar el punto nos mostrará un aviso (Fig. 30) indicándonos sobre qué herramienta y objeto de trabajo lo estamos creando. Esto hay que tenerlo en cuenta en el momento que queramos definir un punto.
Si deseamos crear más puntos procederíamos de la misma manera.
Como vemos en el menú de la izquierda de la imagen, cada vez que marcamos un punto se nos añade al Objeto de Trabajo seleccionado y una vez que tenemos todos los puntos (Fig. 31) deseados el siguiente paso será el de crear una trayectoria.

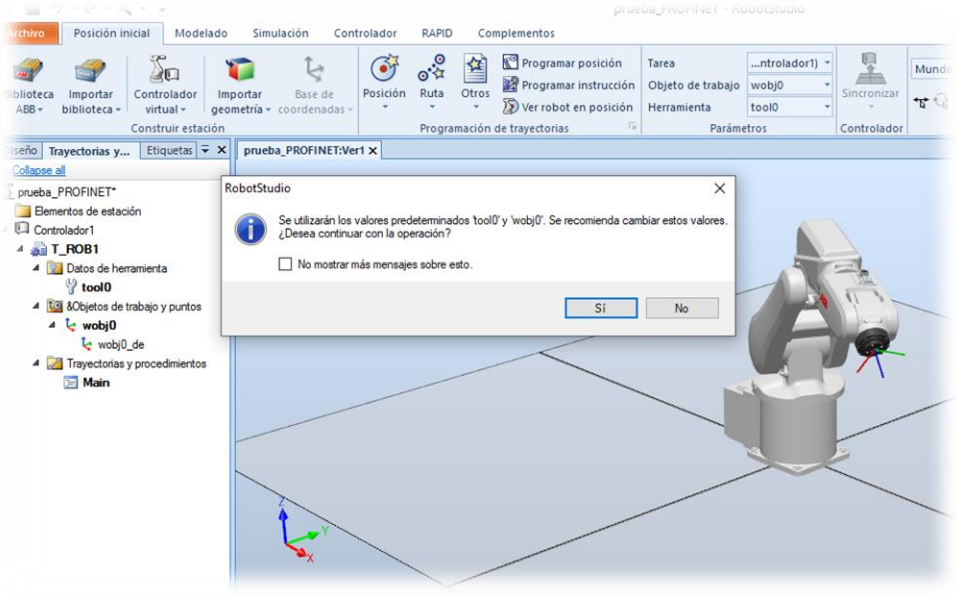

Figura 30 Aviso a la hora de crear un punto en estación trabajo

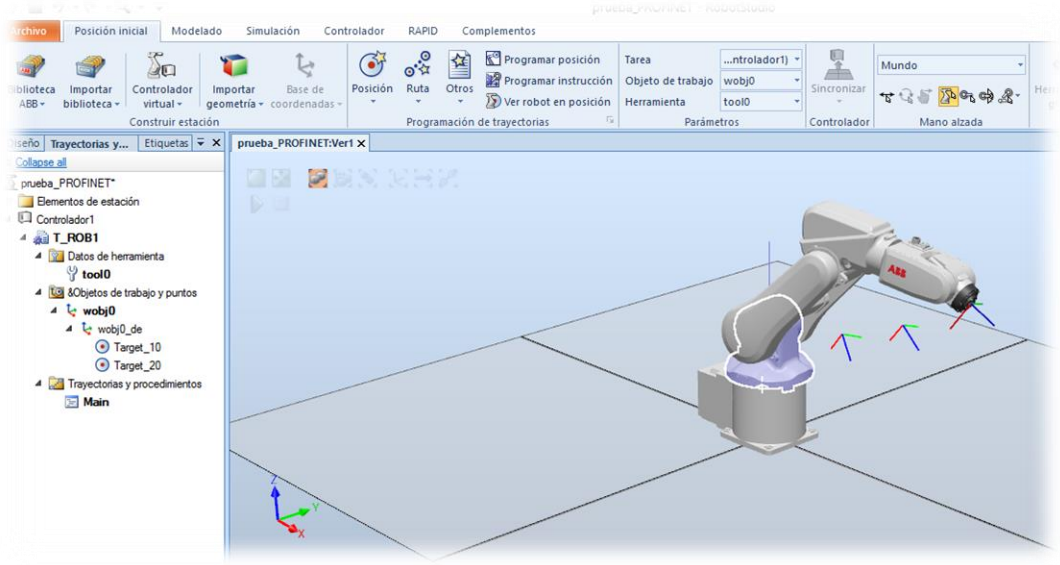

Figura 31 Puntos creados en estación trabajo

Para crear una trayectoria, si en la pestaña “Posición inicial” seleccionamos la opción “ruta” (Fig. 32) veremos como en el árbol de trayectorias se ha añadido una nueva trayectoria a continuación de “Main”.
Una vez creada esa nueva trayectoria, que actualmente está vacía, solo queda añadir los puntos creados anteriormente y el orden de paso por cada punto. Esto lo podemos hacer arrastrando (Fig. 33) los puntos directamente a la trayectoria creada.
De esta manera ya tendremos la trayectoria creada y ya podríamos simularla en el robot.

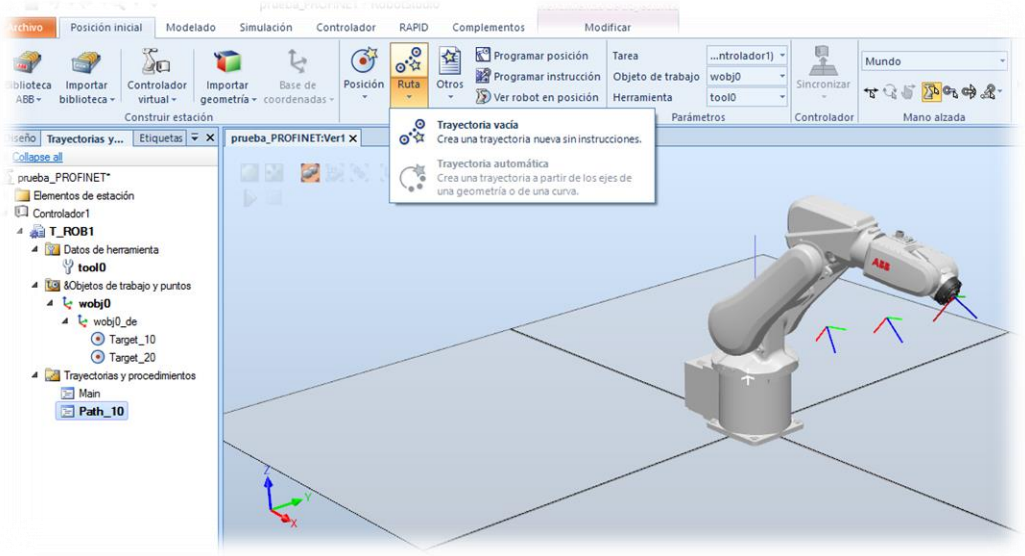

Figura 32 Creación nueva trayectoria vacía

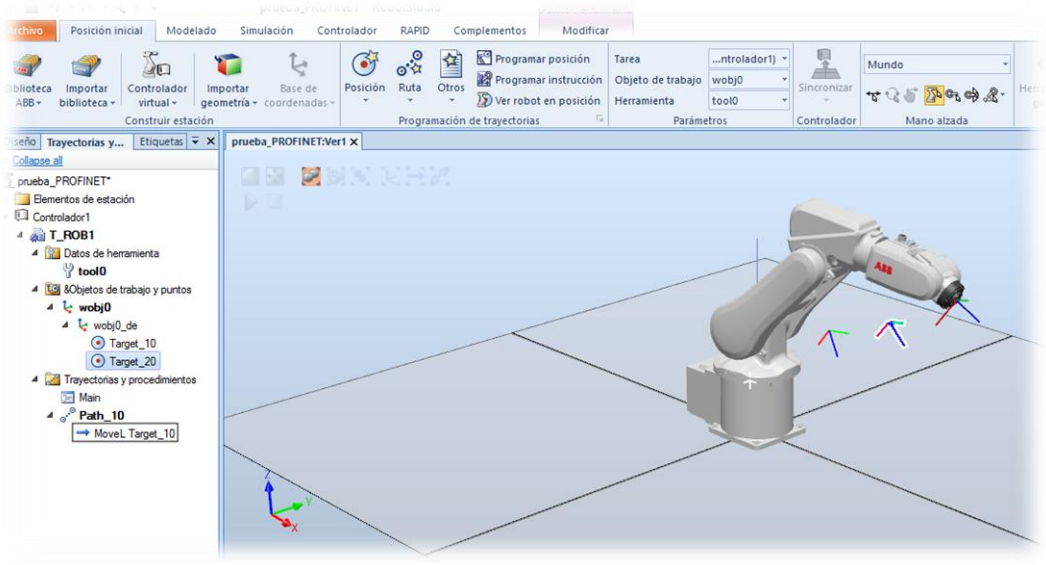

Figura 33 Añadir puntos a nueva trayectoria

Una vez que tenemos todos los puntos en la trayectoria creada automáticamente, se ha generado el código RAPID necesario para que el ROBOT realice ese movimiento programado. Este código generado lo podemos ver si accedemos a la pestaña “RAPID” y desplegamos el árbol de la derecha (Fig. 34). En el código podemos distinguir 3 partes.

La primera es la inicial donde están declarados los puntos generados y su posición en la estación de trabajo; esta parte solamente se ejecutará al iniciar la simulación.

La segunda parte es la función “Path” y es donde son programados los movimientos de la trayectoria y su configuración.

Y la última es la función “Main” y será sobre esta donde estará corriendo el programa una vez que se ejecute el simulador. En ella podemos hacer tantas llamadas como queramos y en el orden que queramos de todas las trayectorias disponibles en la estación de trabajo.

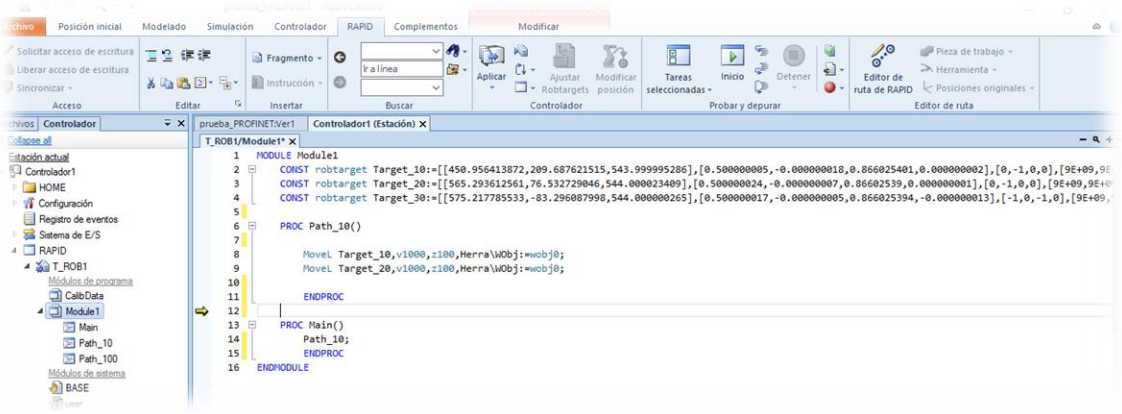

Figura 34 Código RAPID generado en la trayectoria

Como he comentado anteriormente, cuando creamos una trayectoria entre puntos, el programa genera automáticamente el código RAPID necesario para su funcionamiento. Pero este código se genera dependiendo de la configuración que tengamos por defecto en el programa (Fig. 35), la cual se puede ver en la parte inferior de la pantalla. En ella vemos los diferentes tipos de movimientos que tenemos:

Tipo de movimiento del ROBOT entre los puntos de la trayectoria: como, por ejemplo:

Movimiento lineal “MoveL”

Movimiento articular “MoveJ”

Movimiento circular “MoveC”

Velocidad del movimiento: en este podremos seleccionar la velocidad a la que el ROBOT ejecutará el movimiento.

“Zone”: este hace referencia a la precisión a la que el ROBOT buscará el paso por los puntos creados. El valor hace referencia a la distancia a la que pasa el TCP sobre el punto marcado.

Herramienta: esta opción indica sobre que herramienta se hace en movimiento.

La última opción que ofrece este menú hace referencia a sobre qué objeto de trabajo se genera el movimiento.

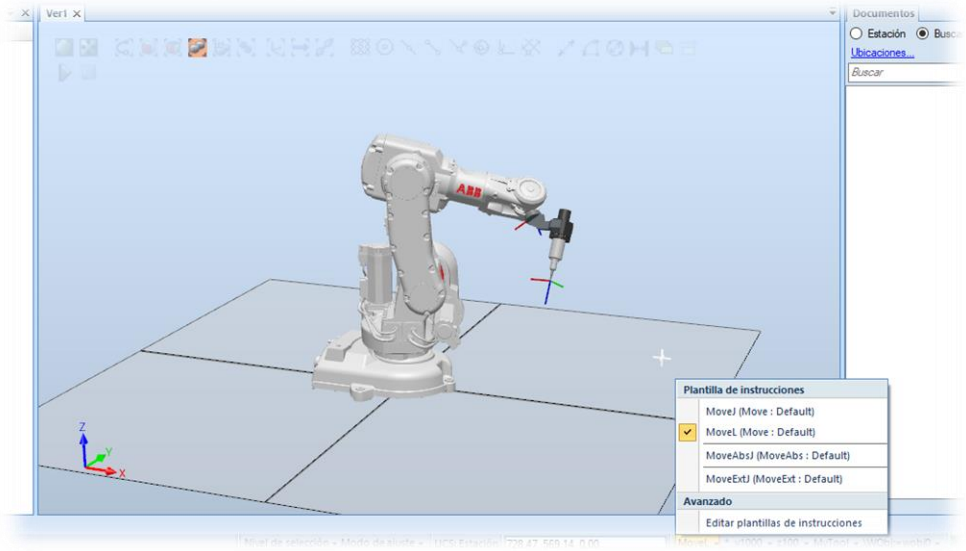

Figura 35 Configuración generación código RAPID

3.2.5.2 Insertar objetos en espacio de trabajo.

Otra forma de crear una trayectoria es definiendo los puntos sobre un objeto insertado en el espacio de trabajo.

En este caso insertaremos un nuevo objeto sobre el que marcaremos los puntos sobre los cuales posteriormente trazaremos la trayectoria.

Sobre el objeto insertado en el espacio de trabajo definiremos un nuevo Workobject sobre el que estarán referenciados estos puntos. De esta manera si cambiamos de ubicación u orientación ese objeto los puntos se reposicionarán automáticamente, evitando tener que hacerlo manualmente uno a uno.

Para la creación del objeto de trabajo lo haremos desde la pestaña “Modelado” y seleccionaremos la “creación de un tetraedro” (Fig. 36); una vez seleccionada esta opción se abrirá un menú para insertar las dimensiones de este y al finalizar la creación se insertará en el espacio de trabajo el objeto creado.

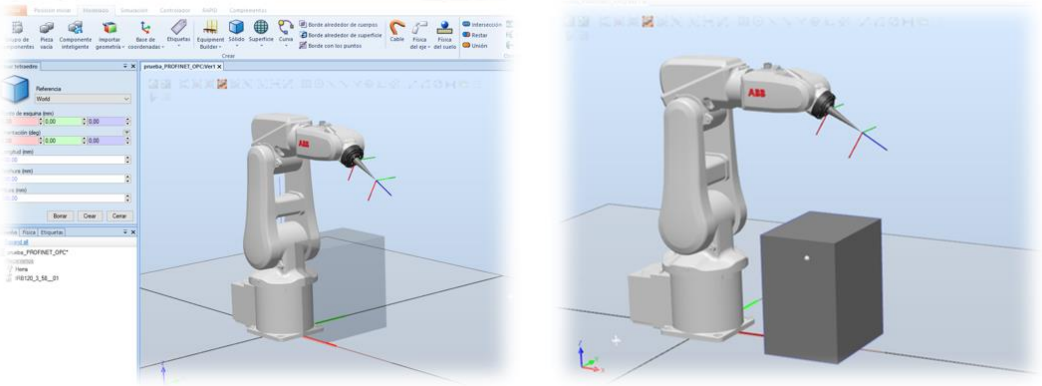

Figura 36 Insertar tetraedro en espacio trabajo

Una vez que tenemos el nuevo objeto creado en el espacio de trabajo lo que haremos es crear un nuevo Workobject para ese objeto. Desde la pestaña “Posición inicial” seleccionaremos la opción “otros” (Fig. 37) y dentro de las opciones que aparecen disponibles seleccionaremos “Crear objeto de trabajo”. A continuación, en el menú que aparece a la izquierda de la pantalla seleccionaremos el nombre del Workobject y en la opción “sistema de coordenadas de usuario” seleccionaremos “sistema de coordenadas” y nos aparecerá otro menú (Fig. 38) en el que marcaremos la opción “3 puntos”.

En este menú nos estará pidiendo que marquemos 3 puntos sobre el nuevo objeto para establecer la referencia de ese nuevo Workobject; seguimos las indicaciones del menú y una vez finalizado el proceso se creará el nuevo sistema de referencia del nuevo Workobject creado.

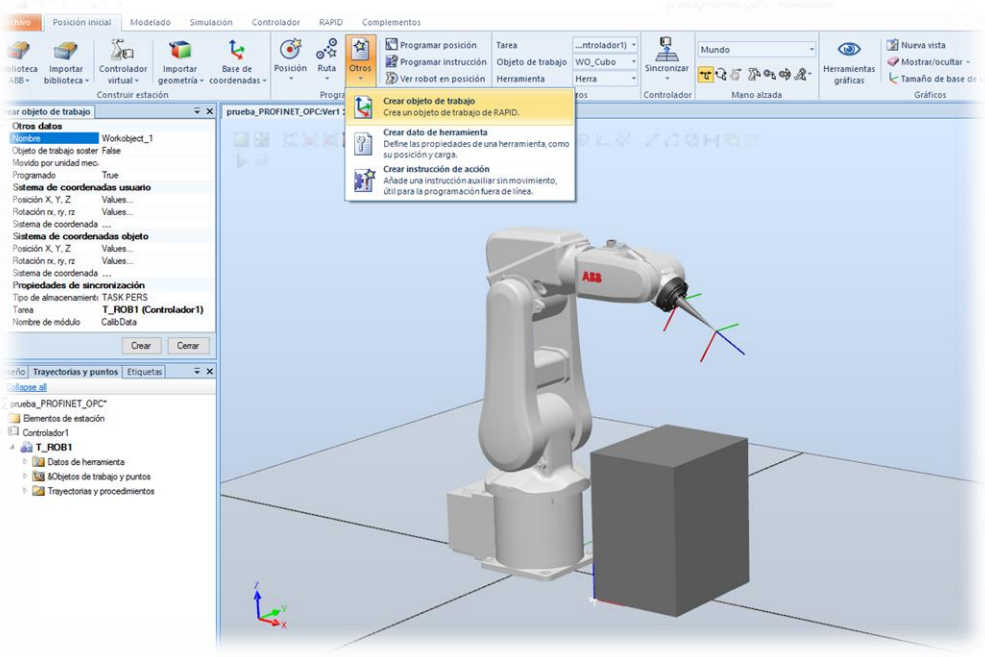

Figura 37 Creación Nuevo Workobject en espacio trabajo

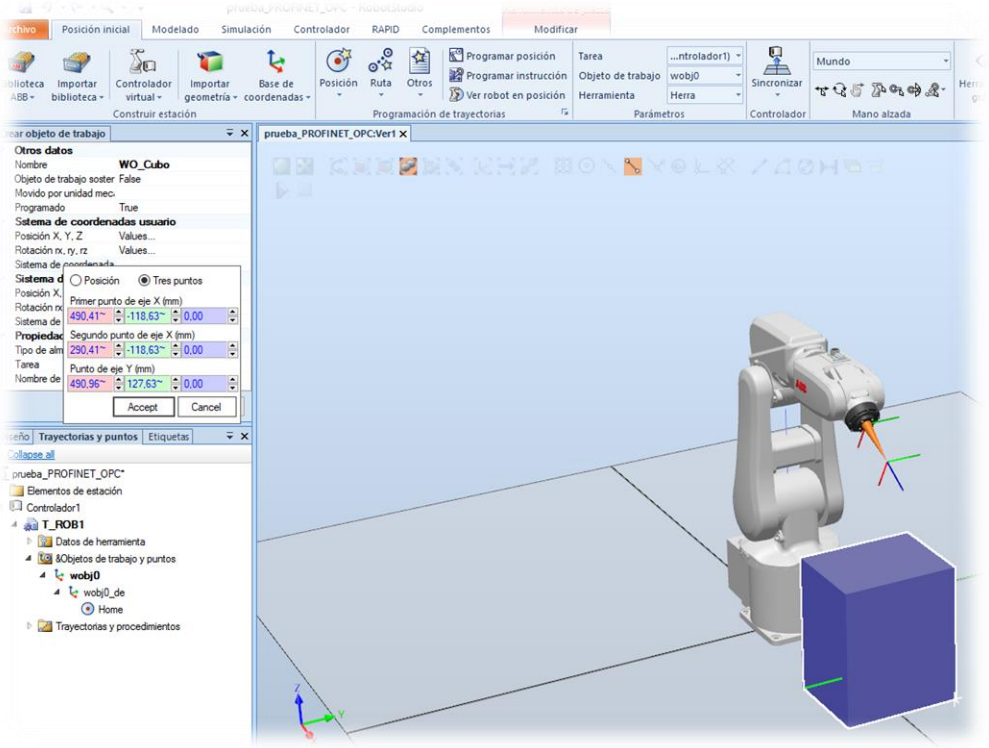

Figura 38 Marcadas referencias nuevo Workobject

Una vez creado el nuevo Workobject, sobre el objeto creado nos aparece el sistema de referencia de este, que puede o no coincidir con el sistema de referencia del mundo (en la esquina inferior izquierda de la pantalla). Si ambos sistemas no coinciden tendríamos que girarlos para hacerlos coincidir y tener así todos los sistemas de referencia en la misma orientación.

Para girar este sistema de referencia, en el árbol de trayectorias (Fig. 39) seleccionaremos el Workobject y con el botón derecho de ratón seleccionamos “girar”. Nos aparece un menú para realizar el giro y en el mismo podremos indicar los grados que lo queremos girar y sobre qué eje queremos realizar el giro.

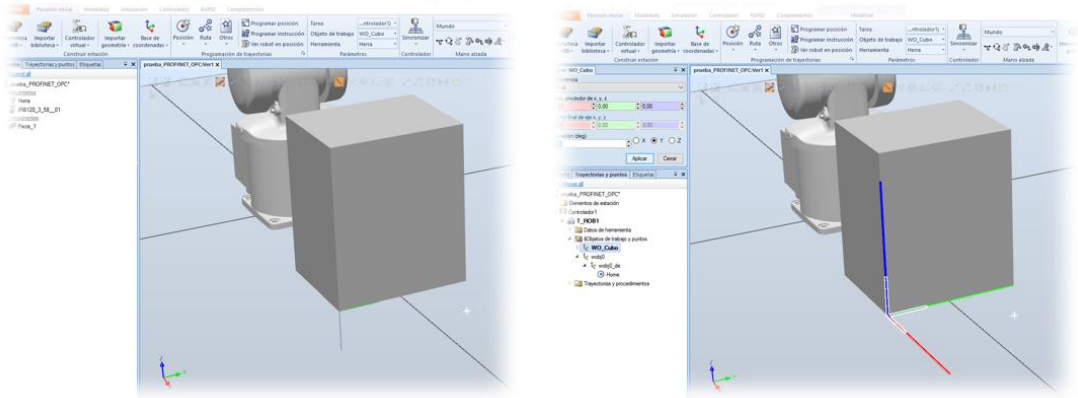

Figura 39 Girar sistema referencia nuevo Workobject

3.2.5.3 Creación de posiciones sobre la superficie del cubo

Una vez creado el tetraedro y definido un nuevo Workobject sobre el mismo, el siguiente paso es el de seleccionar varios puntos sobre el objeto para posteriormente definir una trayectoria.

La definición de los puntos la haremos de manera automática. Para ello en el menú "Posición inicial" seleccionaremos la opción de "posición" y dentro de las opciones que aparecen seleccionamos "Crear punto" (Fig. 40). Una vez seleccionado nos aparecerá en la parte izquierda de la pantalla un menú para indicar las posiciones de los puntos a crear.

Para indicar automáticamente la posición de cada punto seleccionaremos cualquiera de los 3 campos de posición del punto y mediante la herramienta de "ajustar a final" en la pantalla de diseño iremos marcando los puntos a crear.

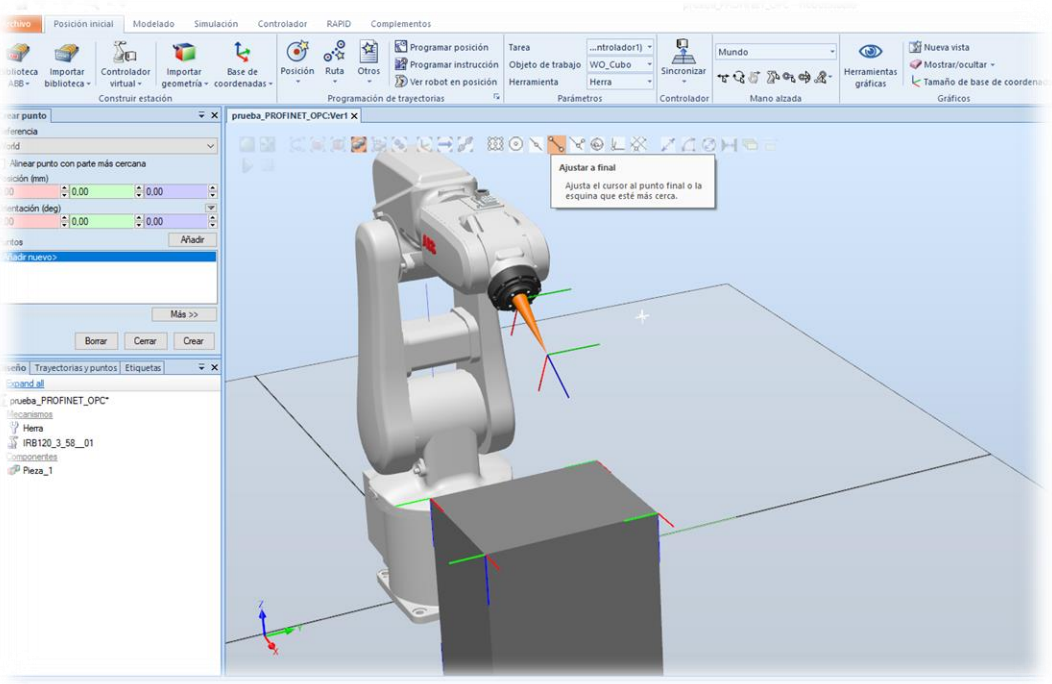

Figura 40 Selección puntos sobre objeto

Cuando ya tenemos marcados todos los puntos que queremos crear, al darle al botón "crear" estos puntos aparecerán insertados (Fig. 41) en el árbol de trayectorias dentro del Workobject correspondiente.

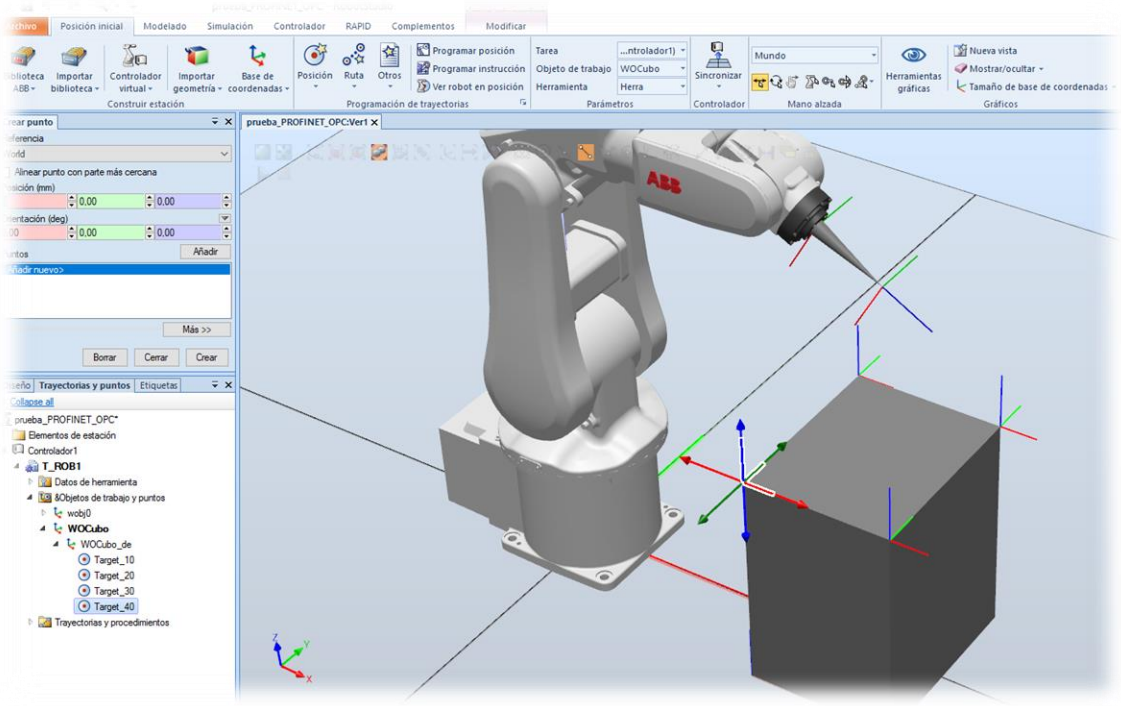

Figura 41 Nuevos puntos creados en Workobject

3.2.5.4 Ver herramienta en posición

Una vez creados los puntos deseados, el siguiente paso a realizar será el de ver cómo queda el robot posicionado sobre esos puntos por si, en caso de tener que reposicionarlo, pueda llegar a todos los puntos sin problemas.

Para ver como queda la herramienta en las posiciones creadas lo que haremos es seleccionar uno de los puntos creados (Fi.g 42) y, mediante el boton derecho de ratón, marcar la opcion “Ver herramienta en posicion” y automaticamente aparecerá sobre la estacion de trabajo como quedaría la herramienta en el punto seleccionado.

Si al marcar esta opcion vemos que el robot no puede posicionarse con la configuracion actual, sobre ese punto tendríamos que moverlo para que el robot pueda alzanzar dicho punto.

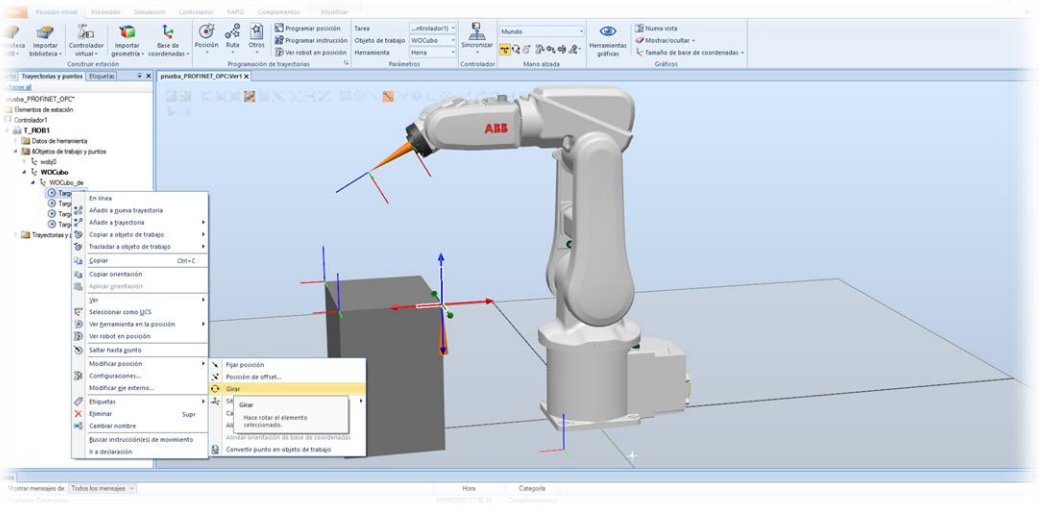

Figura 42 Herramienta en posición sobre punto creado

Para mover la orientación del punto para que el robot pueda posicionarse sobre el mismo sin problemas, tendremos que seleccionar el punto a modificar y con el botón derecho de ratón marcar opción de “girar” y en el menú que aparece (Fig. 43) indicar los grados y sobre qué eje realizaremos el giro.
Una vez que tenemos el punto reorientado y vemos que el robot puede posicionarse en él sin problemas, comprobaremos los demás puntos y si tenemos que modificarlos podremos hacerlo de varias formas.
La primera es volver a realizar estos pasos para todos los puntos creados y la segunda en mediante la opción de “copiar orientación” y “aplicar orientación” (Fig. 44) que tenemos en el menú de cada punto.

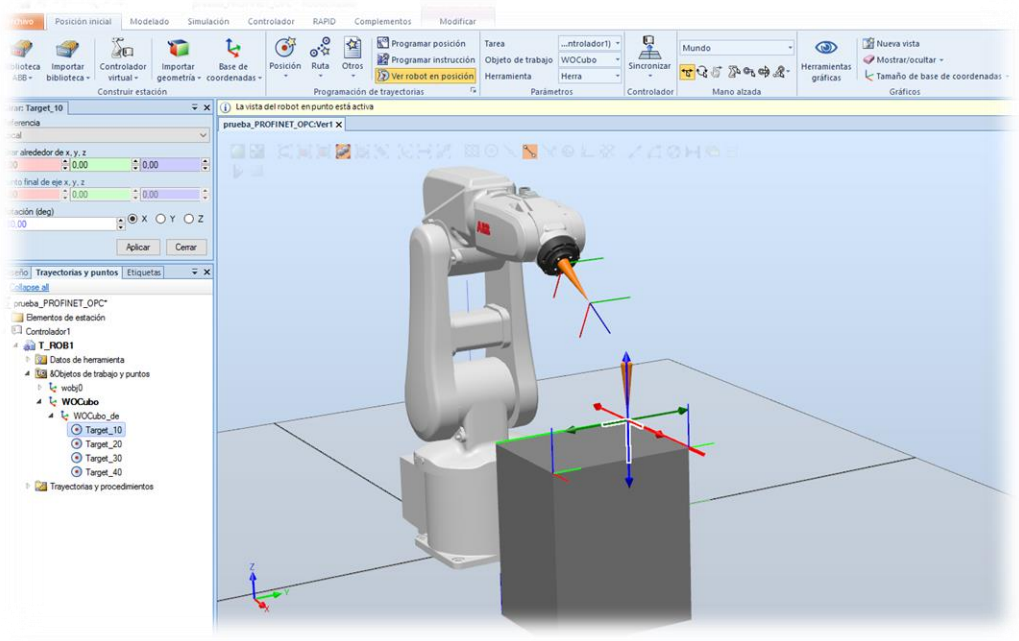

Figura 43 Girar punto creado para posicionar robot

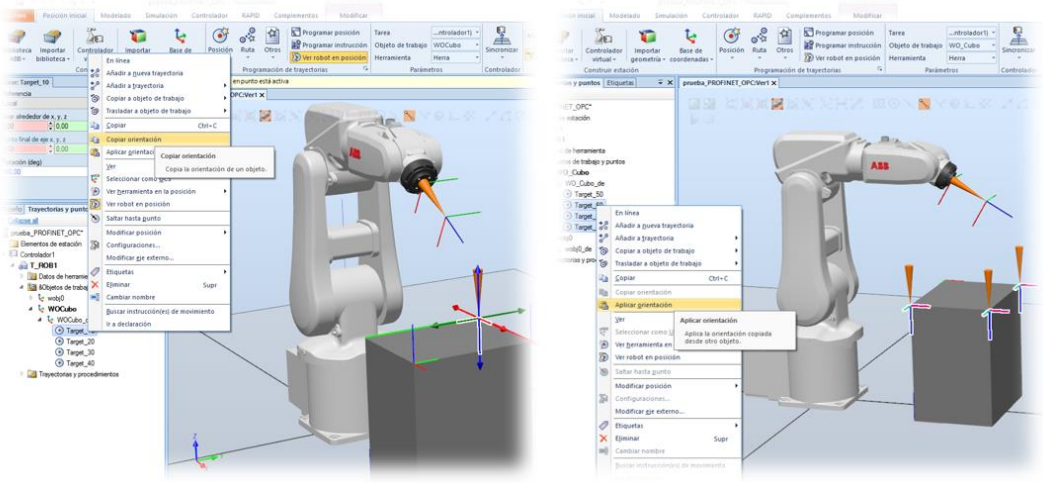

Figura 44 Copiar orientación puntos creado para posicionar robot

Una vez que tenemos creados y reorientados todos los puntos creados, el ultimo paso sería ver como queda el robot posicionado en los diferentes puntos antes de crear la trayectoria deseada.

Para ver el robot sobre los puntos creados (Fig. 45) procederemos de la misma manera que cuando hemos visto el posicionado de la herramienta sobre los puntos creados. Si una vez comprobados todos los puntos el posicionado del robot es correcto procederemos a la creacion de la ruta con los movimeintos deseados.

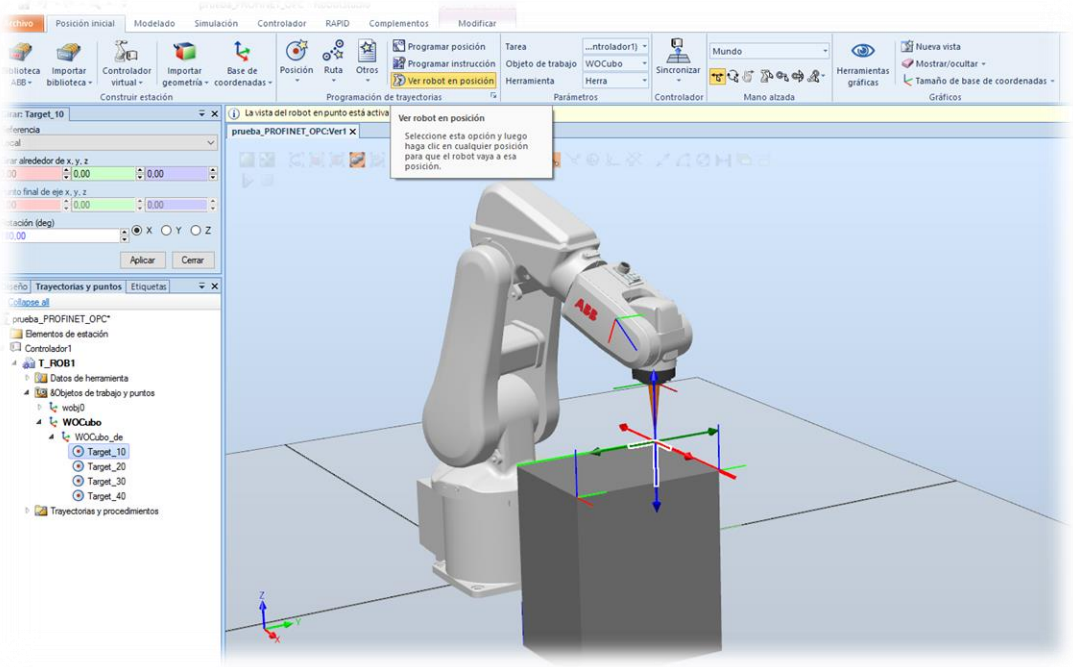

Figura 45 Robot en posición sobre puntos

Para configurar los movimientos tendremos en cuenta qué tipo de movimiento es por si lo hacemos con las instrucciones MoveL o MoveJ y, dependiendo el tipo de movimiento, también le ajustaremos la aproximación al punto final y la velocidad a la hora de realizar el movimiento.

3.2.6 Flexpendant

FlexPendant [30] es la consola manual utilizada a nivel de campo por el operador para controlar el robot. Permite realizar casi las mismas tareas que se pueden hacer mediante Robotstudio, como son: carga y modificación de programas, actuar sobre entradas/salida, calibración, movimiento de ejes…
Está compuesto por una pantalla táctil, un joystick, botones de seguridad y botones de acceso rápido. RobotStudio permite simular de forma virtual el FlexPendant del robot.
Para acceder a la versión virtual de RobotStudio tenemos que acceder a la pestaña "controlador" (Fig. 46) y seleccionar la opción de "flexpendant".

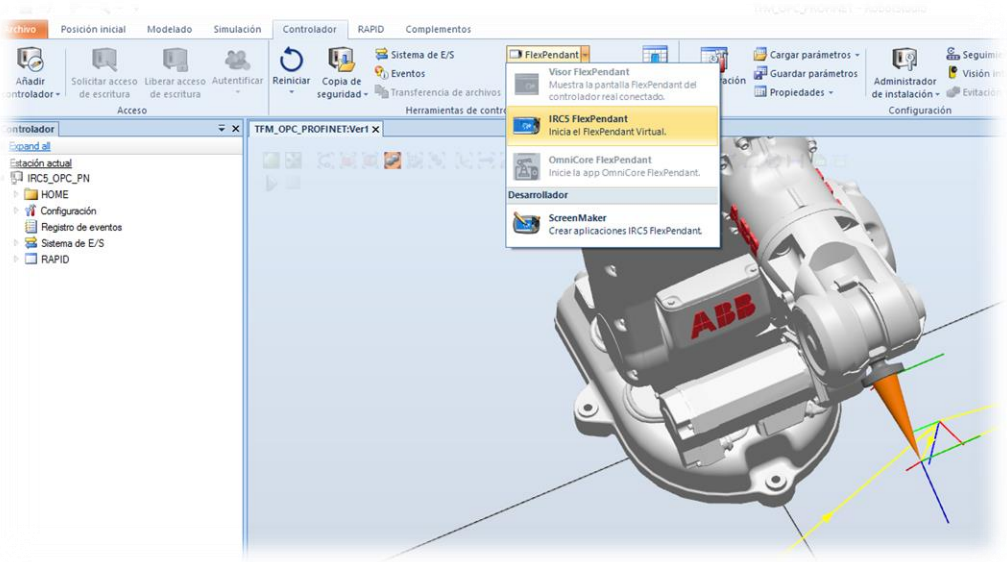

Figura 46 Acceso virtual a FlexPendant

Una vez ejecutada la versión virtual en el menú principal (Fig. 47) del Flexpendant, tenemos todos los submenús disponibles sobre los que podremos hacer diferentes cosas en el robot.

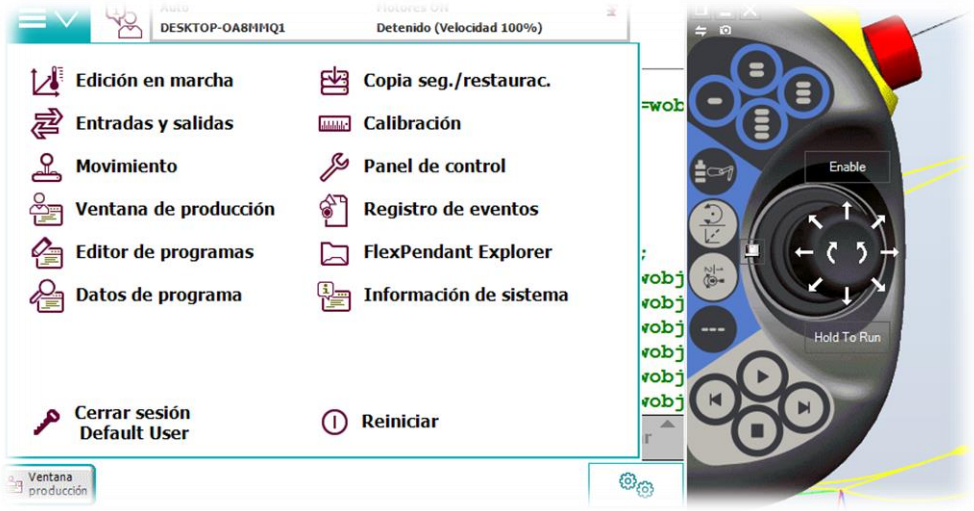

Figura 47 Menú principal FlexPendant

• Edición en marcha “HotEdit”: HotEdit (Fig. 48) se usa para ajustar las posiciones programadas. Puede hacerse en todos los modos de funcionamiento, e incluso mientras el programa se está ejecutando. Es posible ajustar tanto las coordenadas como la orientación. HotEdit sólo puede usarse con las posiciones con nombre del tipo “robtarget”.

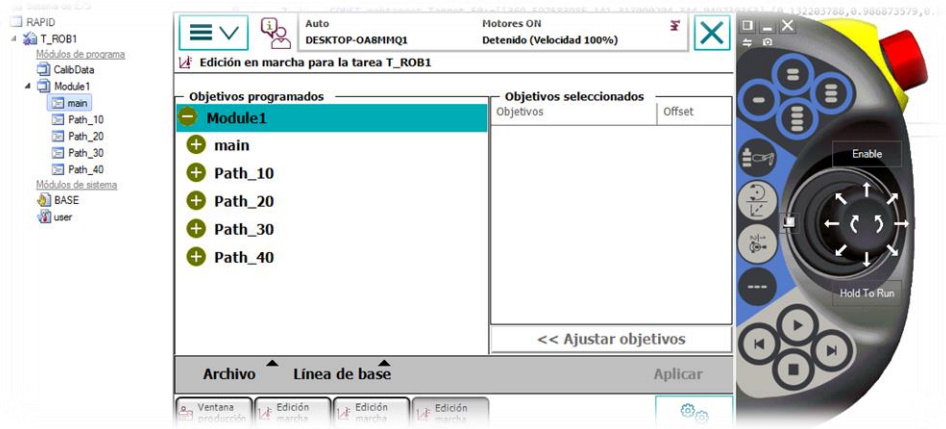

Figura 48 Edición en marcha

• Entradas y salidas **“Inputs and Outputs”**: Las entradas y salidas (Fig. 49), E/S, son señales utilizadas en el sistema de robot. Las señales se configuran utilizando parámetros del sistema.

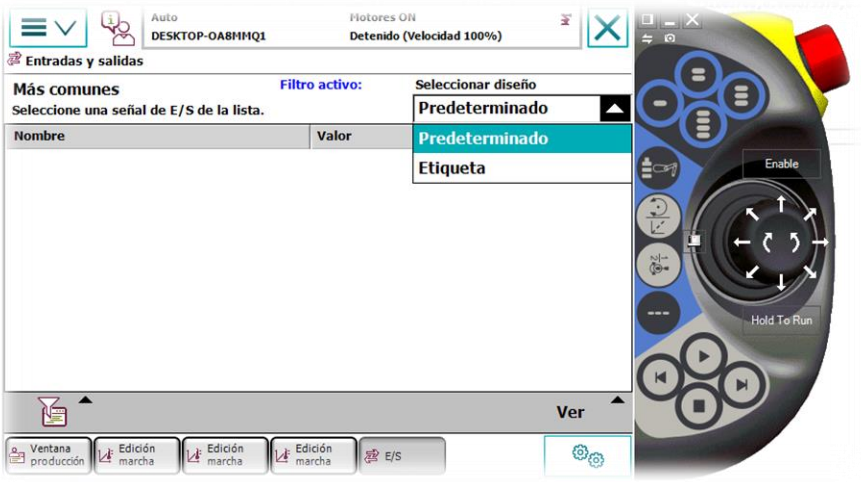

Figura 49 Entradas y salidas

• Movimiento (Fig. 50): En este menú encontramos las distintas funciones de movimiento disponibles y nos indicará sobre qué ejes o motores realizaremos los movimientos. También nos permite saber la posición del robot en todo momento.

Figura 50 Menú movimiento

• Ventana de producción “Production Window”: La ventana de producción (Fig. 51) se utiliza para el código del programa mientras éste se está ejecutando. Permite operar con el código Rapid del programa durante la ejecución.

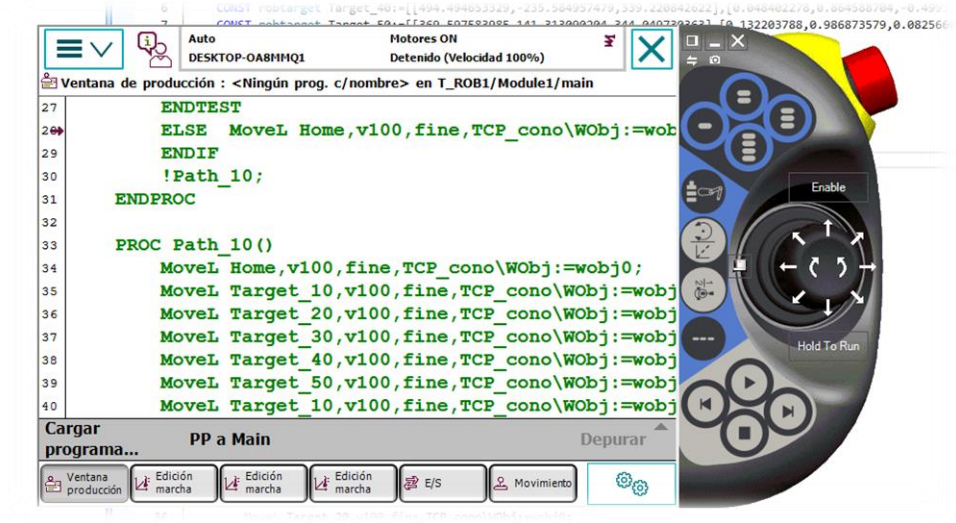

Figura 51 Ventana de producción

• Editor de programas “Program Editor”: El Editor de programas (Fig. 52) es donde se crean o modifican los programas. Puede abrir más de una ventana del Editor de programas, lo cual puede resultar útil cuando se tiene instalada la opción Multitasking. El botón Editor de programas de la barra de tareas muestra el nombre de la tarea.

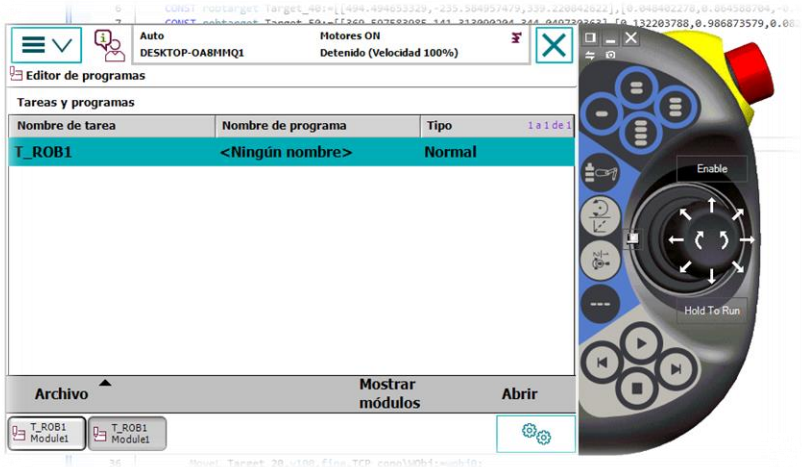

Figura 52 Editor de programas

• Datos de programa: La vista Datos de programa (Fig. 53) contiene funciones de visualización y utilización de tipos de datos e instancias. “Datos de programa” puede abrir más de una ventana, algo que puede resultar útil cuando se trabaja con muchas instancias o tipos de datos.

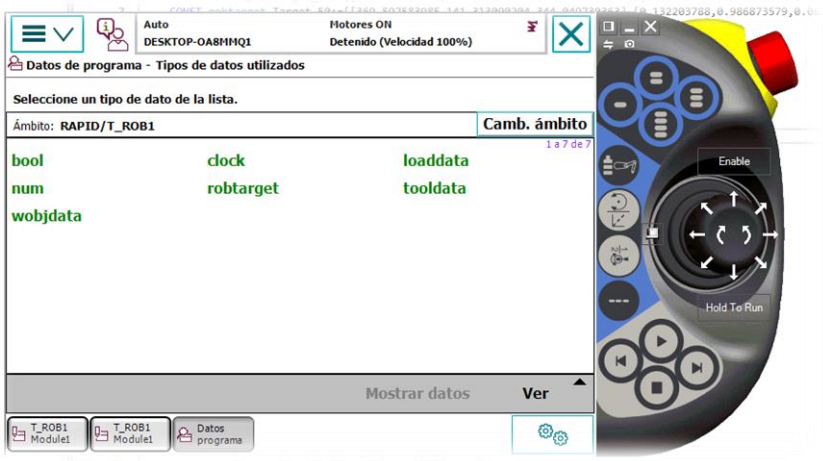

Figura 53 Datos de programa

• Copia de seguridad y restauración “Backup and Restore”: El menú “Copia de seguridad y restauración” (Fig. 54) se usa para realizar copias de seguridad y restaurar el sistema.

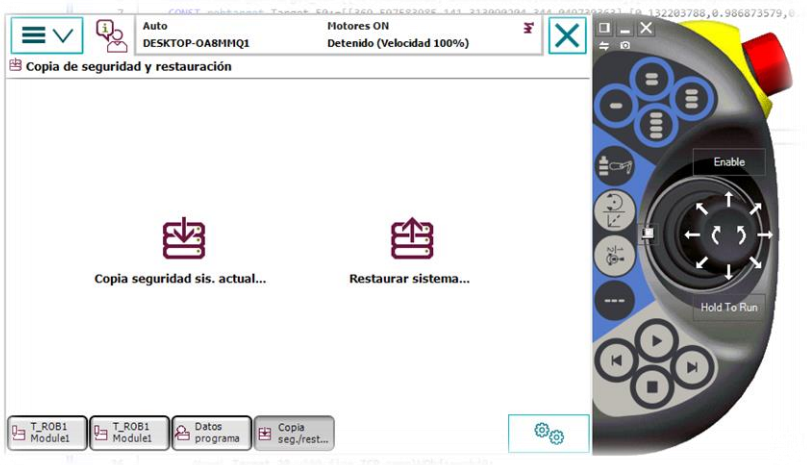

Figura 54 Copia de seguridad y restauración

• Calibración “Calibración”: El menú “Calibración” (Fig. 55) se utiliza para calibrar las unidades mecánicas del sistema de robot. La calibración puede ser realizada con la opción “Calibration Pendulum”.

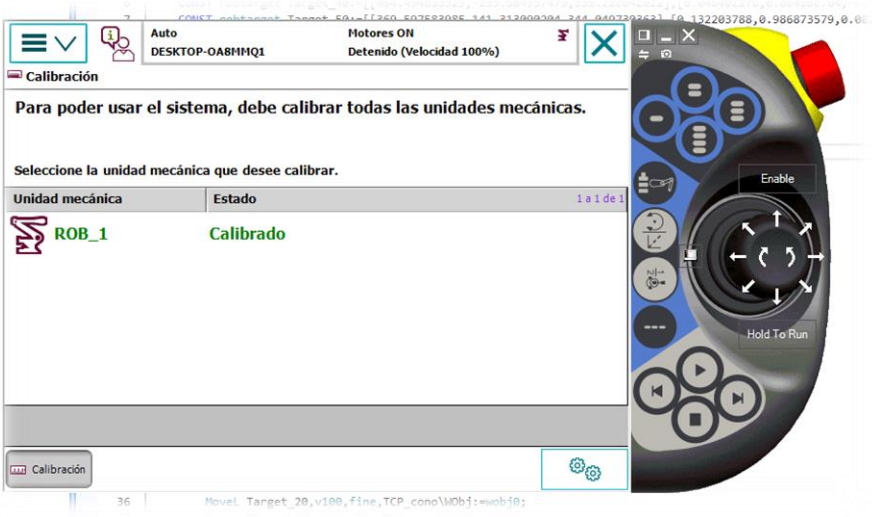

Figura 55 Calibración

• Panel de control **"Control Panel"**: El panel de control (Fig.56) contiene funciones que permiten personalizar el sistema de robot y el FlexPendant.

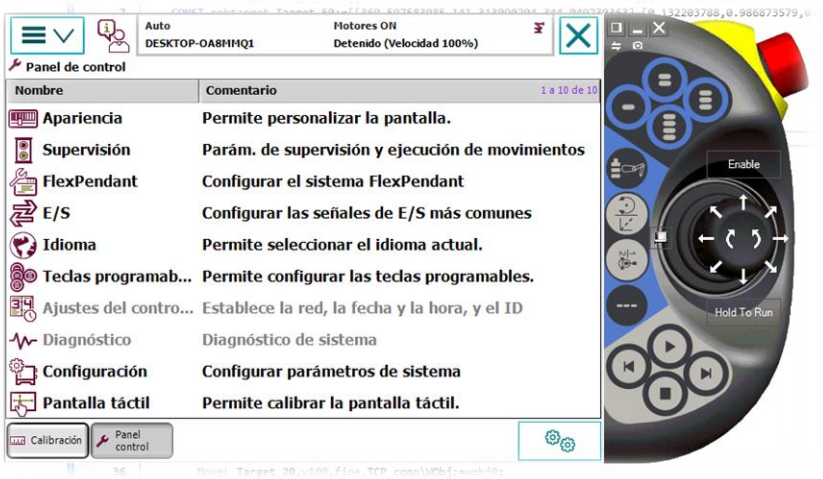

Figura 56 Panel de control

• Registro de eventos "Event Log": Con frecuencia los sistemas de robot funcionan sin la intervención de ninguna persona. La función de registro (Fig. 57) es una forma de almacenar información acerca de los eventos que han tenido lugar, como información de referencia futura y para facilitar la solución de problemas.

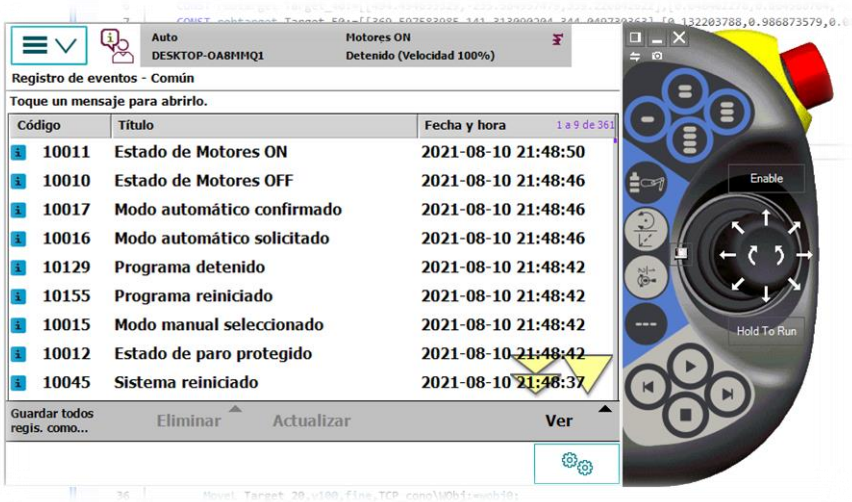

Figura 57 Registro eventos

• FlexPendant Explorer: Este *Explorer* (Fig.58) es un administrador de archivos, similar al Explorador de Windows, que permite ver el sistema de archivos del controlador. Se pueden eliminar o trasladar archivos o carpetas, o cambiar su nombre.

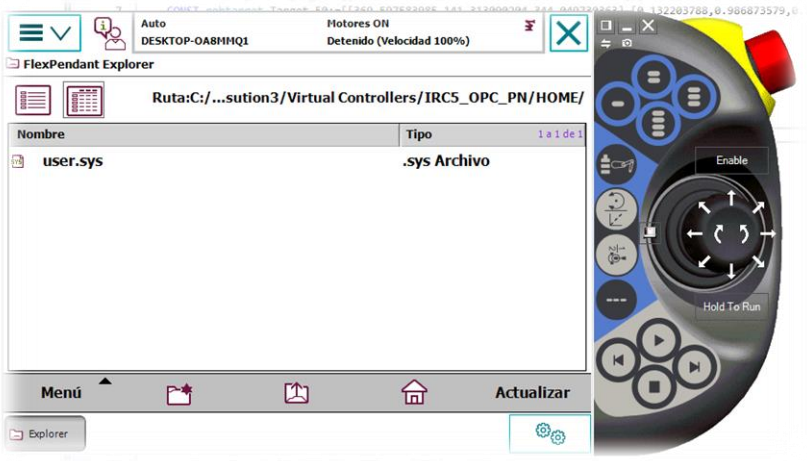

Figura 58 FlexPendant Explorer

• Información del sistema “System Info”: En Información del sistema (Fig. 59) se muestra información acerca del controlador y el sistema cargado. Aquí se puede encontrar la versión de RobotWare y las opciones en uso actualmente, las claves actuales de los módulos de control y Drive Modules, las conexiones de red, etc.

Figura 59 Información del sistema

3.2.7 Lógica de estación

A la hora de crear una estación robótica y dotarla de elementos externos que no son del robot, elementos del entorno de trabajo, se hace necesario actuar sobre ellos para un correcto funcionamiento de la estación robótica.
Esto se puede hacer mediante la lógica de la estación (Fig. 60) ya que es la responsable de interconectar las señales entre los distintos dispositivos para su correcto funcionamiento.

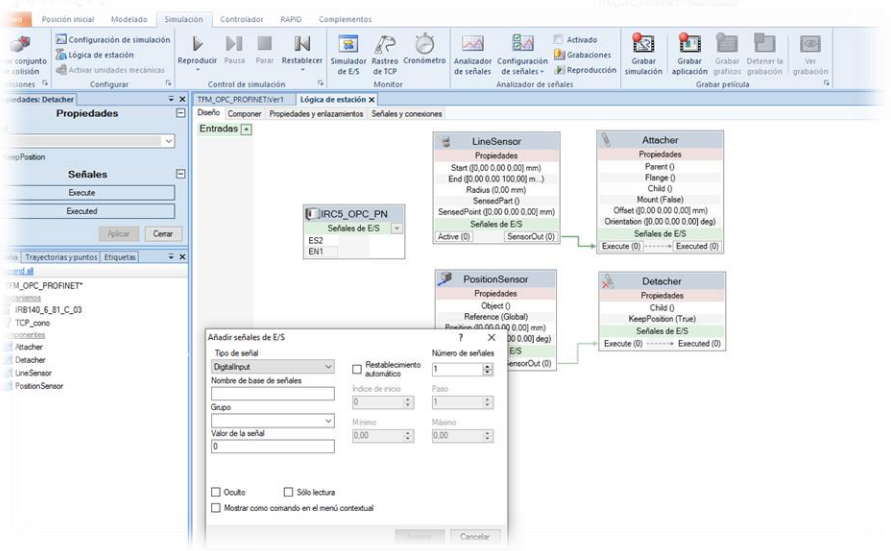

Figura 60 Lógica de estación

Cuando creamos una lógica de estación podemos insertar multitud de condiciones para ejecutar el entorno de trabajo de la estación creada. Para ellos desde la ventana de lógica de estación y con el botón derecho nos aparecerá un menú (Fig. 61) en el que podemos elegir numerosos elementos a insertar en la estación.

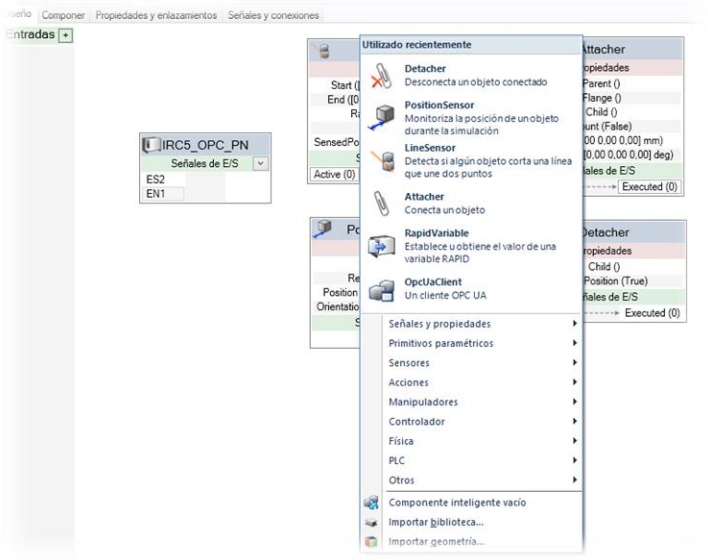

Figura 61 Menú lógica estación

Entre las diferentes opciones encontramos:

- Señales y propiedades (Fig. 62), aquí podremos realizar múltiples lógicas de combinación y operaciones aritméticas entre las señales conectadas.

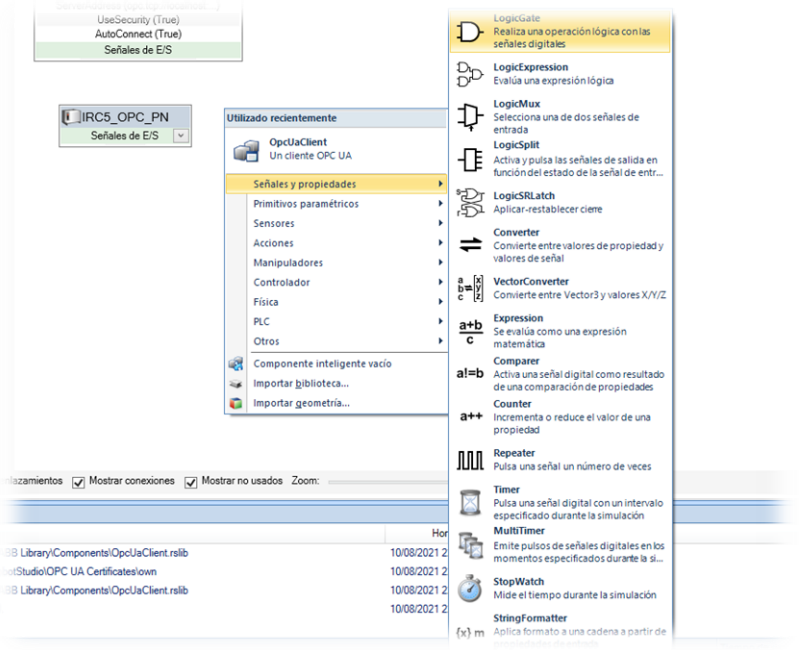

Figura 62 Señales y propiedades lógica estación

- Sensores (Fig. 63), aquí podremos insertar en la estación diversos tipos de sensores y detectores que condicionaran el funcionamiento de la estación creada.

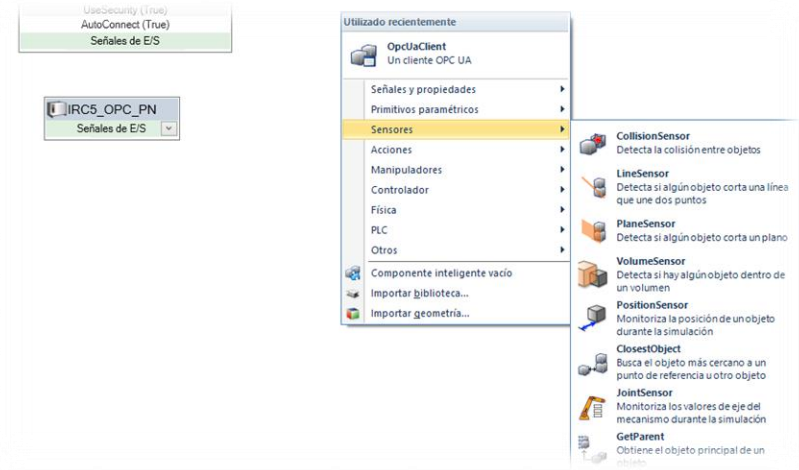

Figura 63 Sensores lógica estación

- Acciones (Fig. 64), aquí definiremos acciones a realizar por el robot o el entorno dependiendo de las condiciones programadas.

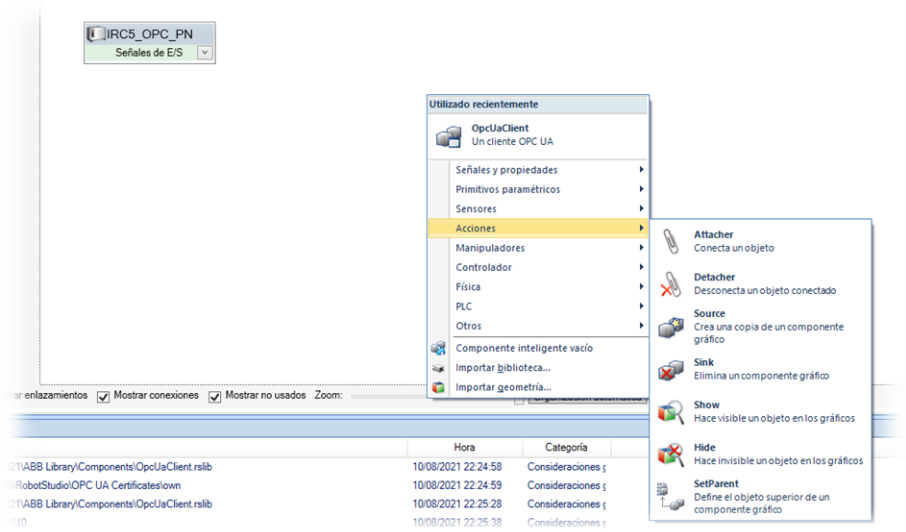

Figura 64 Acciones lógica estación

- Manipuladores (Fig. 65), se pueden realizar movimientos y giros sobre los objetos de la estación de trabajo.

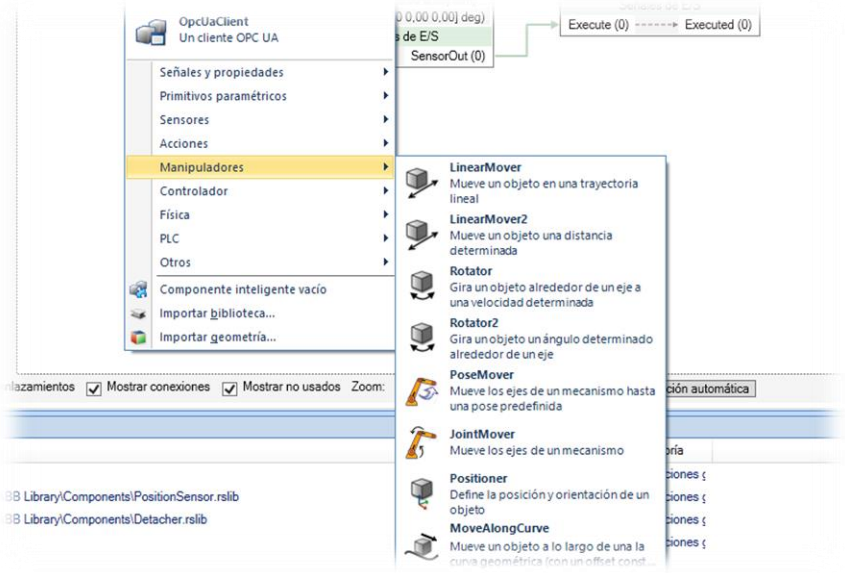

Figura 65 Manipuladores lógica estación

- Controlador (Fig. 66), permite obtener el valor de una variable para realizar operaciones con ella, combinándola con otras opciones.

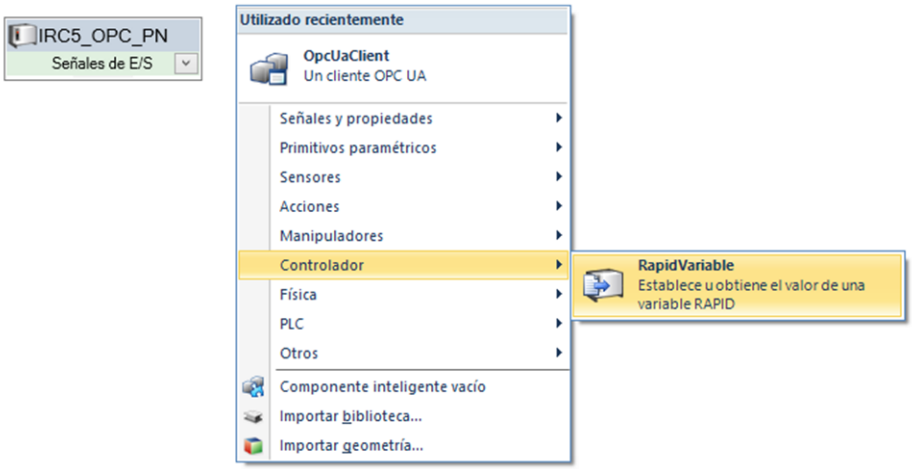

Figura 66 Acciones lógica estación

- Otros (Fig. 67), este menú permite insertar otros elementos para controlar la estación creada.

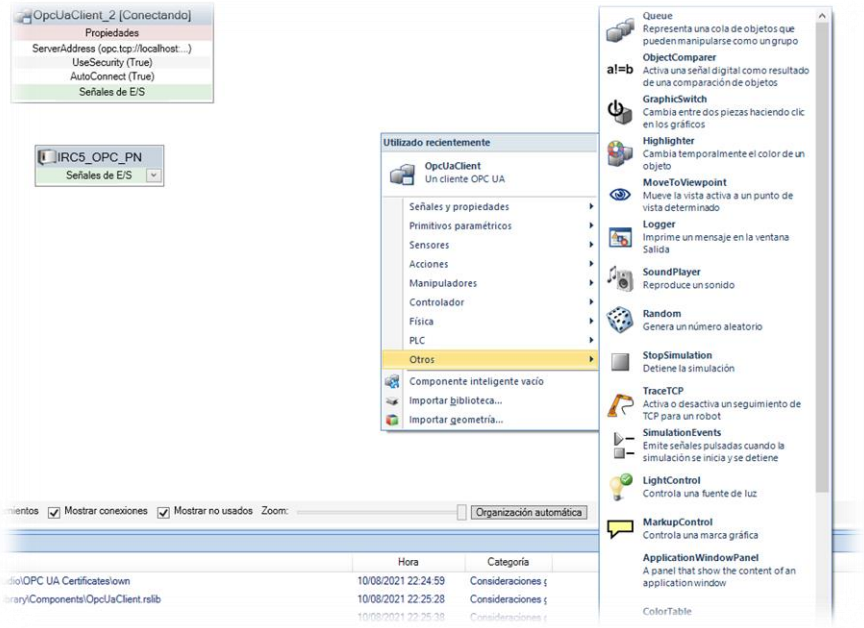

Figura 67 Otras acciones lógica estación

En particular durante el desarrollo de este TFM no se realiza ninguna lógica de estación ya que esta función se realizará mediante el PLC conectado (OPC o PROFINET) y este será el encargado de controlar toda la estación y condicionar la tarea realizada por el robot dependiendo del estado de la línea de producción.

3.3 Introducción a RAPID

El lenguaje de programación RAPID (**R**obotics **A**pplication **P**rogramming **I**nteractive **D**ialogue) [12],[13],[14] es el lenguaje desarrollado por ABB para la programación de sus Robots bajo el software RobotStudio.

El trabajo entre RAPID y RobotStudio es bidireccional, es decir, siempre que hagamos un cambio en nuestra estación y queramos reflejarlo en RAPID o viceversa habremos de sincronizar. De ahí que si pinchamos sobre el icono "sincronizar" encontraremos las dos soluciones: sincronizar con RAPID y sincronizar con la estación.

Una aplicación RAPID consta de un programa (Fig. 68) y una serie de módulos del sistema.

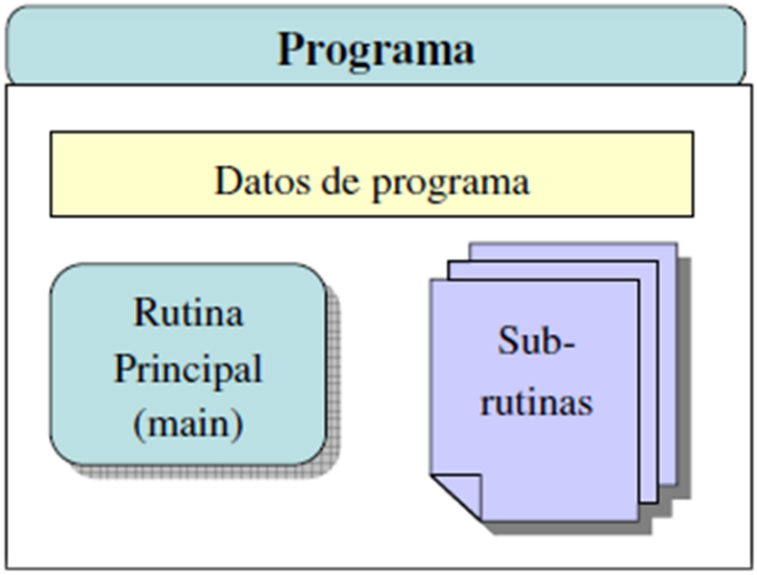

Figura 68 Formato programa RAPID

La estructura de un programa RAPID (Fig. 69) es muy similar a otros lenguajes de programación, en el que inicialmente tendremos:

-Declaración de variables (numéricas, constantes, booleanas, etc.)

-Función principal MAIN, que será sobre la que se iniciará y desde la que llamaremos a las demás funciones creadas.

-Funciones de Programa, serán funciones definidas por nosotros y mediante las que realizaremos la programación de las diferentes trayectorias. Sirven para estructurar el programa y poder realizar una ejecución más controlada.

Básicamente la secuencia de ejecución de un programa RAPID es:

En primer lugar, se ejecutará una sola vez la creación de variables del sistema y posteriormente se ejecutará de manera cíclica la función principal MAIN que será la encargada de llamar a las funciones secundarias dependiendo de las condiciones programadas dentro de la función MAIN.

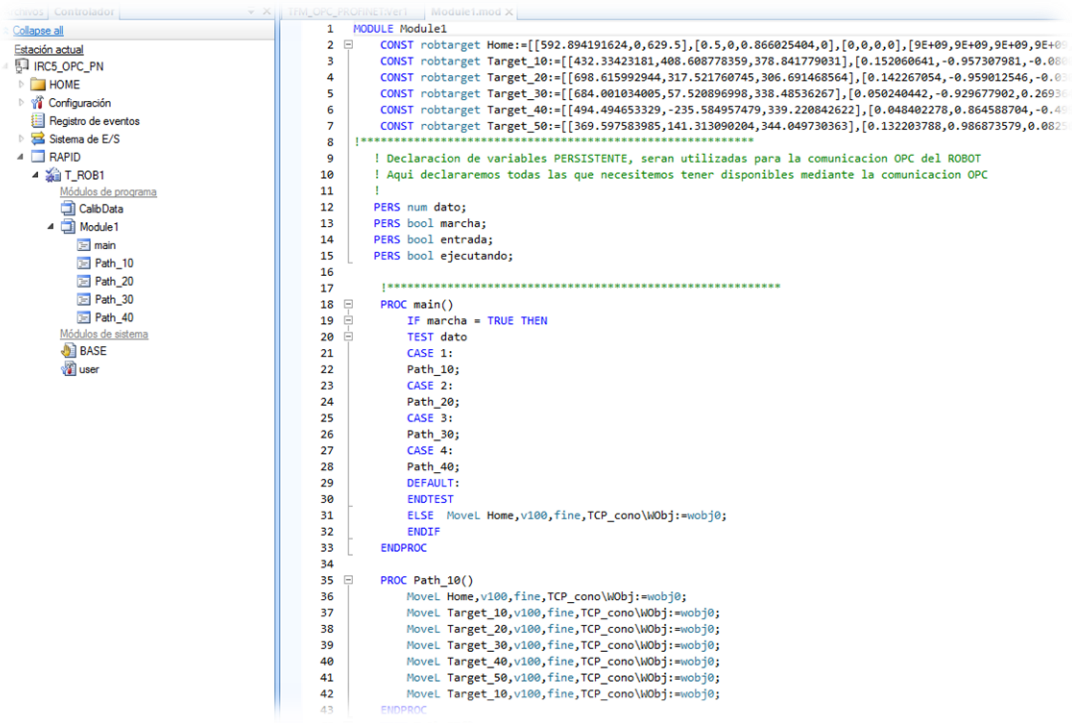

Figura 69 Estructura de programa RAPID

3.3.1 Elementos básicos de RAPID

En RAPID cuando realizamos un programa tenemos unos elementos básicos [15], que a su vez tienen unas condiciones de uso dentro de la programación.

Como principales elementos básicos tenemos:

-**Identificadores**, que nos permitirán darle nombre a módulos, datos, etiquetas, rutinas, etc. Estos identificadores también tienen unas condiciones para su uso, como son: primer carácter es una letra, longitud máxima de 16 caracteres y diferencia entre mayúsculas y minúsculas.

-**Palabras Reservadas**, existen diversas palabras reservadas (propias del sistema) que no podremos utilizarlas como identificadores

-Comentarios, sirven para facilitar la comprensión del programa, ocupan una línea entera comenzando con el símbolo “!”.
-Valores de cadena, secuencia de caracteres entre comillas.

3.3.2 Tipos de datos en RAPID

Los datos a manejar pueden ser definidos como:

Constantes (CONS): representen datos de un valor fijo a los que no se puede reasignar un nuevo valor.
Variables (VAR): son datos a los que se les puede asignar un nuevo valor durante la ejecución del programa.
Persistentes (PERS): se trata de variables en las que cada vez que se cambia su valor durante la ejecución del programa, también se cambia el valor de su inicialización.

3.3.3 Instrucciones RAPID

Durante la programación tanto de la función MAIN como de las funciones creadas para las diferentes trayectorias en RAPID se pueden utilizar diversas instrucciones [14] para control de las mismas funciones o para actuar sobre el entorno de trabajo.
Algunas de esas instrucciones de control del programa son la actuación sobre entradas y salidas del sistema o las instrucciones para control de ejecución del programa.

3.3.3.1 Utilización de entradas y salidas

Con el uso de este tipo de instrucciones principalmente se actuará sobre las entradas y salidas del sistema, bien para leer el valor de una entrada o para activar una salida.
La instrucción que hace referencia a la lectura de entradas del sistema es "WaitDI". Esta instrucción esperará a que la entrada indicada tenga el valor prefijado para realizar el paso a la siguiente línea de programa.

```
WaitDI Entrada1, 1;
```

En este caso la ejecución del programa se detendrá en esa línea esperando a que el valor de "entrada1" tenga valor "1" para continuar a la siguiente línea de programa.
Otra forma de actuar sobre el valor de una entrada es mediante una condición para ejecutar un código de programa u otro.

```
IF Entrada1 = TRUE THEN …
```

En este caso el programa ejecutará una parte del programa condicionado al valor de la entrada a comprobar, si el valor de esa comprobación no se cumple no se ejecutará.
En lo referente a las salidas del sistema, la función será de activar el valor de la salida o resetear ese valor si estaba fijado previamente.
Para activar el valor de las salidas se utiliza:
- Set: activa el valor digital a 1.

- Reset: resetea el valor digital a 0.
- SetDO: fija una salida digital a un valor simbólico (activado o desactivado).

SetDO do1, 1 ! Activación =1 Desactivación = 0

- SetAO: fija el valor de una salida analógica.

3.3.3.2 Control de ejecución del programa

Este tipo de instrucciones son similares a los utilizados en otros lenguajes de programación y sirven para estructurar la ejecución del programa.
Las instrucciones más utilizadas son:

- IF … THEN: esta instrucción ejecutará una serie de instrucciones si se cumple una determinada condición de la variable evaluada.
- FOR: esta instrucción repite una sección del programa un determinado número de veces indicado.
- WHILE: esta instrucción repetirá una sección del programa mientras se cumpla la condición evaluada.
- TEST/CASE: esta instrucción ejecuta diferentes instrucciones en función del valor del dato comprobado.
- GOTO: esta instrucción hace salto incondicional a un punto del programa.

3.3.3.3 instrucciones de movimiento.

Para mover el robot (Fig. 70) hay tres instrucciones:

- Si se tiene que mover el robot hacia un punto usando coordenadas articulares. Cuando no tiene que seguir ninguna trayectoria determinada.
  -**MoveJ** Punto, Velocidad, Zona, Herramienta

- Si se tiene que mover el robot hacia un punto usando la línea recta.
  -**MoveL** Punto, Velocidad, Zona, Herramienta

- Si se tiene que mover el extremo del robot hacia el punto de destino pasando por el punto del circulo trazando un arco de circunferencia.
  -**MoveC** Punto_Circulo, Punto_Destino, Velocidad, Zona, Herramienta;

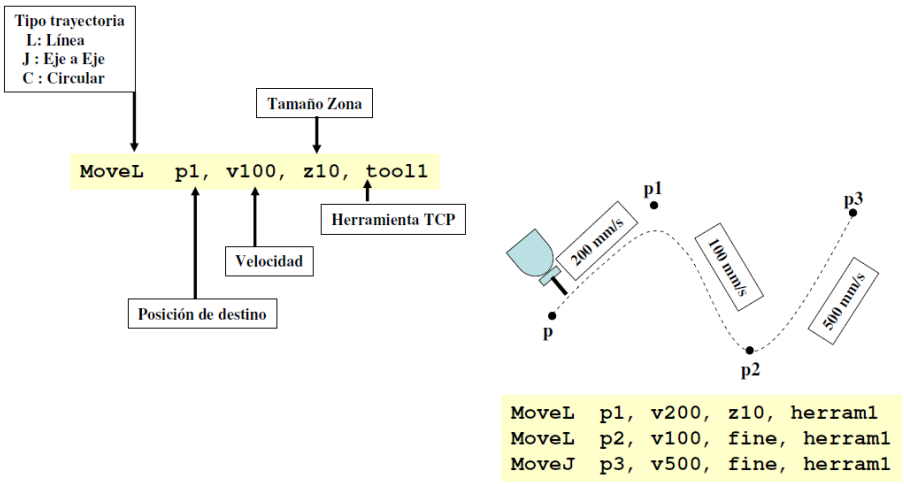

Figura 70 Tipos de movimiento de robot

Otra opción de movimiento del robot es mediante el posicionamiento (Fig. 71) desde un punto conocido en la estación de trabajo.

Para ello la línea de programa es similar a cualquiera de los movimientos vistos antes, pero añadiendo la función "Offs", que sirve para desplazar el robot en los 3 ejes la distancia indicada durante la programación del movimiento.

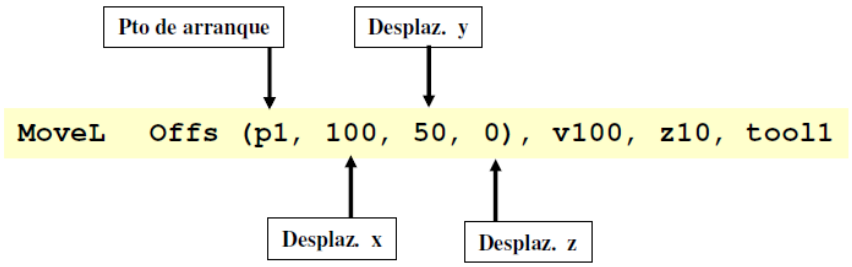

Figura 71 Posicionamiento de robot

3.4 Comunicación OPC

3.4.1 Historia OPC

El estándar OPC (inicialmente OLE for Proccess Control, pero renombrado Open Platform Communications) [16]surgió a mediados de los años 90 como una tecnología para la comunicación del software SCADA y otros HMI con los dispositivos de control en el ámbito de la industria. Consiste en realidad en un conjunto de especificaciones que resuelven diferentes tareas imprescindibles en este ámbito, para lo cual hacen uso de tecnologías estándar del mercado.

3.4.2 ¿Que es OPC?

OPC [17] [18] [19] es una tecnología de comunicación con una arquitectura de cliente y servidor. Una aplicación actúa de servidor proporcionando datos y otra actúa como cliente leyéndolos o manipulándolos.
OPC es, con mucha diferencia, la tecnología de comunicación industrial estándar. Ello permite el intercambio de información entre múltiples dispositivos y aplicaciones de control sin restricciones o límites impuestos por los fabricantes. Un servidor OPC puede estar comunicándose continuamente con los PLCs de campo, RTUs, estaciones HMI u otras aplicaciones. Aunque el hardware y el software provengan de diferentes marcas comerciales, el cumplimiento del estándar OPC posibilita la comunicación continua en tiempo real.
Por ello, OPC ha permitido una mejor cooperación entre proveedores y usuarios, ayudando a construir soluciones completamente transversales, dando a los consumidores más poder de elección entre diferentes aplicaciones industriales. La interoperabilidad, las soluciones modulares y la libertad de elección han sido los grandes motivadores para que los usuarios de todo el mundo – y por tanto los proveedores – hayan incorporado OPC a sus entornos industriales.
Si OPC es un estándar abierto es porque su interoperabilidad viene dada por la creación, mantenimiento y mejora de unas especificaciones estándar realizadas por un grupo de trabajo multidisciplinar y que no se decanta por ninguna marca en particular. La primera especificación fue resultado de la colaboración de un conjunto de fabricantes en colaboración con Microsoft. De ello resultó una tecnología basada en el DCOM de los sistemas operativos de Microsoft. La especificación detallaba un set de objetos, interfaces y métodos independientes para asegurar la interoperabilidad. La tecnología DCOM proporcionaba el framework para el desarrollo de las soluciones software de comunicación. Esa es la llamada especificación OPC Data Access.
Los softwares que tienen la capacidad de adquirir datos de los dispositivos de campo y servirlos en OPC son los llamados Servidores OPC o OPC Servers. El gran beneficio que

aportan es poder desvincular los sistemas de explotación de datos superiores de la casuística concreta de campo. Con ello, se consigue toda la información del sistema, aunque se tengan controladores, HMIs, RTUs, … de diferentes fabricantes que utilicen diferentes protocolos.
En su inicio, una aplicación OPC Server únicamente traducía un solo protocolo de un fabricante – Modbus, por ejemplo – a OPC DA. Actualmente, los OPC Servers líderes del mercado son capaces, desde una única aplicación, adquirir información de centenares de protocolos diferentes y servirlos por OPC DA y OPC UA

3.4.3 Arquitectura OPC

Como Arquitectura OPC (Fig. 72) nos referimos a la infraestructura de comunicaciones que incluye uno o varios Clientes OPC y Servidores OPC comunicándose entre sí con el fin de intercambiar información entre los dispositivos y/o aplicaciones.

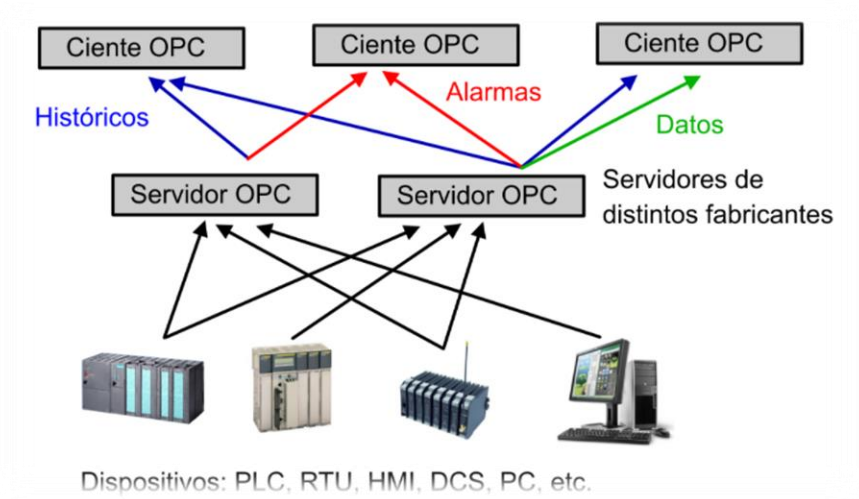

Figura 72 Arquitectura OPC

-**Servidor OPC**: es una herramienta de software que cuenta con al menos una de las especificaciones definidas por la OPC Foundation, y que permite establecer una interfaz de comunicación estándar entre las fuentes de datos utilizando sus protocolos nativos (PLCs, módulos I/O, controladores, reguladores, etc.) y las aplicaciones que contienen Clientes OPC (SCADAs, HMIs, generadores de informes, graficadores, etc.).

-**Cliente OPC**: es un módulo de software que cuenta con al menos una de las especificaciones definidas por la OPC Foundation, y que permite a la aplicación desde donde se está ejecutando (SCADA, HMI, graficadores, historiadores, etc.), establecer comunicación bidireccional con servidores OPC compatibles (un Cliente OPC puede

comunicarse con varios servidores OPC de forma simultánea). Conceptualmente, los Clientes OPC hacen las veces de usuario final de los datos, iniciando y controlando la comunicación bidireccional con Servidores OPC, de acuerdo con las peticiones realizadas desde la aplicación en la que se encuentran embebidos.

Con OPC, la integración de sistemas (Fig. 73) en un entorno heterogéneo se tornará simple.

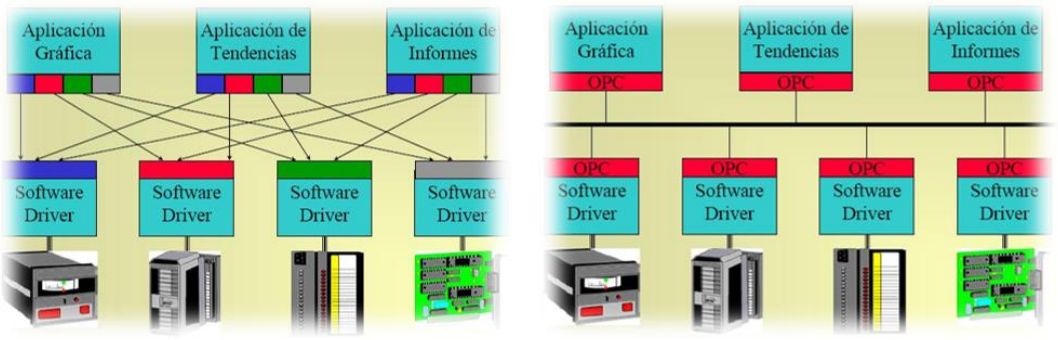

Figura 73 Integración sistemas mediante OPC

3.4.4 Acceso de Datos OPC

El acceso a los datos de intercambio relacionados (Fig. 74) por OPC se puede estructurar de la siguiente manera:

-El servidor (server) Mantiene información sobre el servidor y sirve como container para objetos del grupo OPC

-El grupo (group) Mantiene información sobre sí mismo. Provee mecanismos para contener/organizar lógicamente ítems

-El elemento (ítem) Representan conexiones a fuentes de datos dentro de un servidor

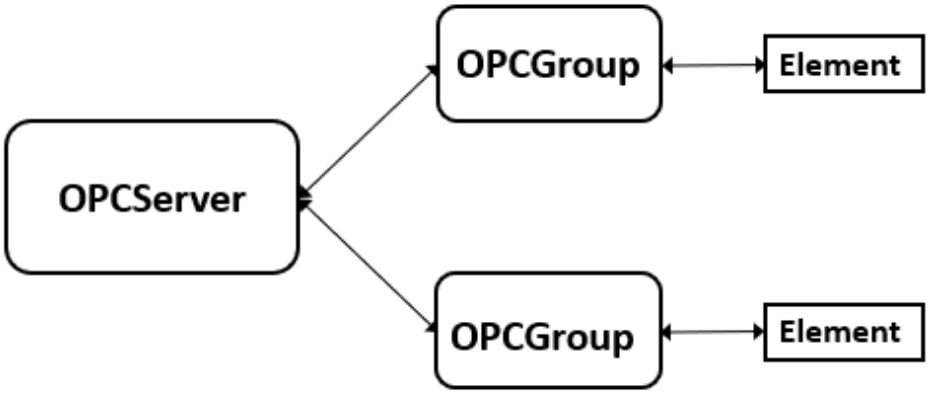

Figura 74 Estructura acceso datos OPC

3.5 Comunicación PROFINET

3.5.1 Historia PROFINET

La automatización en fábrica ya tiene una larga trayectoria en lo que concierne a comunicaciones; hace 20 años, el único requisito del bus de campo consistía en transferir de modo fiable datos de procesos y controlar con precisión dispositivos. En el transcurso de los años, los requisitos se endurecían con el aumento del número de capas en los procesos, y algunos buses de campo no eran capaces de cumplir estas exigencias. Las empresas líderes en el desarrollo de PROFIBUS se han asegurado de que el estándar avanzara en concordancia con el progreso.

Hoy en día, los sistemas de producción requieren una red que también permita intercambiar datos entre diferentes procesos y máquinas. Esto ha contribuido a que los protocolos tiendan a adoptar ideas de otras áreas, tales como telecomunicaciones y redes, es decir, la migración al protocolo TCP/IP.

Sin embargo, Ethernet, de acuerdo con el estándar IEEE 802.3, no es apta para la comunicación horizontal necesaria para la producción porque permite bloquear el tráfico de red incluso en el caso de que otro dispositivo tenga que enviar un mensaje urgente. Por eso nació Ethernet Industrial [20], una tecnología enfocada hacia las fábricas, producción, procesamiento y control, especialmente diseñada para el entorno industrial. Ofrece comunicación en tiempo real y, de este modo, garantiza que cada dispositivo conectado a red pueda recibir y enviar datos en cualquier momento.

Comercializado por primera vez en 2005, PROFINET [20],[21],[22],[23] (Fig. 75) no es solamente PROFIBUS para Ethernet, sino que ofrece mucho más. PROFINET utiliza

conceptos similares a PROFIBUS, pero con un uso más eficiente de ancho de banda en la red, siguiendo un modelo de comunicación entre proveedor y consumidor, en vez de la solución maestro-esclavo convencional. Además, PROFINET ofrece más nodos (casi un número ilimitado), más datos (un marco PROFINET tiene un máximo de E/S de 1.440 bytes), más velocidad (100 Mbits/s, Ethernet dúplex completo) y, hacia mediados del año que viene, cumplirá el estándar IEEE 802.11 para comunicaciones inalámbricas.

Figura 75 Logos PROFIBUS y PROFINET

3.5.2 ¿Qué es PROFINET?

PROFINET es el estándar abierto de Ethernet Industrial de la asociación PROFIBUS Internacional (PI) según IEC 61784-2 y uno de los estándares de comunicación más utilizados en redes de automatización.

PROFINET está basado en Ethernet Industrial, TCP/IP y algunos estándares de comunicación pertenecientes al mundo TI. Entre sus características destaca que es Ethernet en tiempo real, donde los dispositivos que se comunican por el bus de campo acuerdan cooperar en el procesamiento de solicitudes que se realizan dentro del bus.

Partiendo de una conectividad básica, como es el cable Ethernet, y unas tramas de comunicaciones establecidas que correspondería a los niveles 1 y 2 del modelo OSI, PROFINET va incorporando nuevas funcionalidades denominadas “perfiles” de utilidad como ProfiSafe o ProfiEnergy, mediante una interpretación específica para cada caso de los datos transmitidos, modificando el nivel 7 (de aplicación). En el caso de Profisafe, se transmiten datos de seguridad (safety), y en el caso de ProfiEnergy, datos y comandos para el ahorro y control energético.

Con PROFINET es posible conectar dispositivos, sistemas y celdas (conjuntos de dispositivos aislados entre sí), mejorando tanto la velocidad como la seguridad de sus comunicaciones, reduciendo costes y optimizando la producción. Por sus características, PROFINET permite la compatibilidad con comunicaciones Ethernet más propias de entornos TI, aprovechando todas las características de éstas, salvo la diferencia de velocidad que posee una comunicación Ethernet situada en una red corporativa frente al rendimiento en tiempo real que necesita una red industrial.

3.5.3 Servicios comunicación PROFINET

Para la adquisición y transferencia de datos la comunicación PROFINET (Fig. 76) utiliza 3 servicios de comunicación [24]:

-Standard TCP/IP: Este servicio se utiliza para funciones no deterministas, como parametrización, transmisiones de vídeo/audio y transferencia de datos a sistemas TI de nivel superior.

-Real Time: Las capas TCP/IP no son utilizadas para dar un rendimiento determinista a las aplicaciones de automatización, funcionando con unos tiempos de retardo en el rango 1-10ms. Este hecho representa una solución basada en software adecuada para aplicaciones típicas de E/S, incluyendo control de movimiento y requisitos de alto rendimiento.

-Isochronous Real Time: La priorización de señal y la conmutación programada proporcionan una sincronización de alta precisión para aplicaciones como el control de movimiento. Las velocidades de ciclo en rangos de milisegundos son posibles, con jitter (variabilidad temporal durante el envío de señales digitales) en el rango de microsegundos.

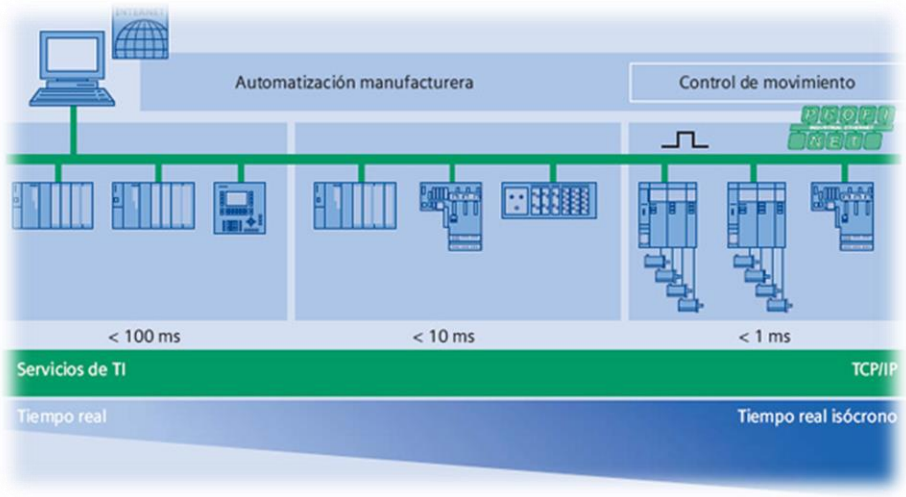

Figura 76 Comunicación PROFINET

Existen varios protocolos definidos dentro del contexto PROFINET. Una lista de estos protocolos junto con su uso concreto es la siguiente:

-PROFINET/CBA: Protocolo asociado a las aplicaciones de automatización distribuida en entornos industriales.

-PROFINET/DCP: Descubrimiento y configuración básica. Es un protocolo basado en la capa de enlace, utilizado para configurar nombres de dispositivos y direcciones IP. Se restringe a una red y se usa principalmente en aplicaciones pequeñas o medianas que no disponen de un servidor DHCP.
-PROFINET/IO: A veces llamado PROFINET-RT (RealTime), es utilizado para comunicaciones con periferias descentralizadas.
-PROFINET/MRP: Protocolo utilizado para la redundancia de medios. Utiliza los principios básicos para la reestructuración de las redes en caso de sufrir un fallo cuando la red posee una topología en anillo. Este tipo de protocolo es utilizado en redes en las que la disponibilidad ha de ser máxima.
-PROFINET/MRRT: Su objetivo es dar soluciones a la redundancia de medios para PROFINET/RT.
-PROFINET/PTCP: Protocolo de Control de Precisión de Tiempo basado en la capa de enlace, para sincronizar señales de reloj/tiempo en varios PLC.
-PROFINET/RT: Transferencia de datos en tiempo real.
-PROFINET/IRT: Transferencia de datos isócrono en tiempo real

4 Resultados

Para llevar a cabo la integración de la conexión OPC entre el Robot y el PLC, en primer lugar, crearé la conexión individual del servidor OPC de cada uno de los equipos (el controlador y el PLC) para posteriormente y tras comprobar su correcto funcionamiento, realizar la interconexión entre ambos.

4.1 Conexión OPC-PLC S7-1200

En este apartado vamos a explicar cómo se realiza la comunicación OPC entre el PLC S7 1200 [25] de SIEMENS y el servidor OPC.
Para configurar el servidor OPC del PLC S7 1200 utilizaremos el software KEPServerEX [26].

https://Pwww.kepserverexopc.com/descarga/

NOTA: El software KEPServerEX necesita una licencia de elevado coste económico por lo que para este trabajo utilizaremos su versión de prueba. Esta versión de prueba nos permite un uso completo del software, pero con limitaciones del tiempo de uso. El software puede ser usado durante 2 horas continuas y si superásemos este tiempo deberíamos reiniciar el servidor. En nuestro caso podemos realizar nuestras conexiones y pruebas en un tiempo inferior a 2 horas por lo que será suficiente con la versión de prueba del software.

4.1.1 Creación del canal de comunicación

Una vez instalado y ejecutado el software [27], lo primero que debemos hacer es añadirle un nuevo canal de comunicación (Fig. 77). Para ello en el árbol de la derecha seleccionamos la opción “nuevo canal” y seguimos los pasos que se nos indican para establecer la conexión con el PLC S7 1200.

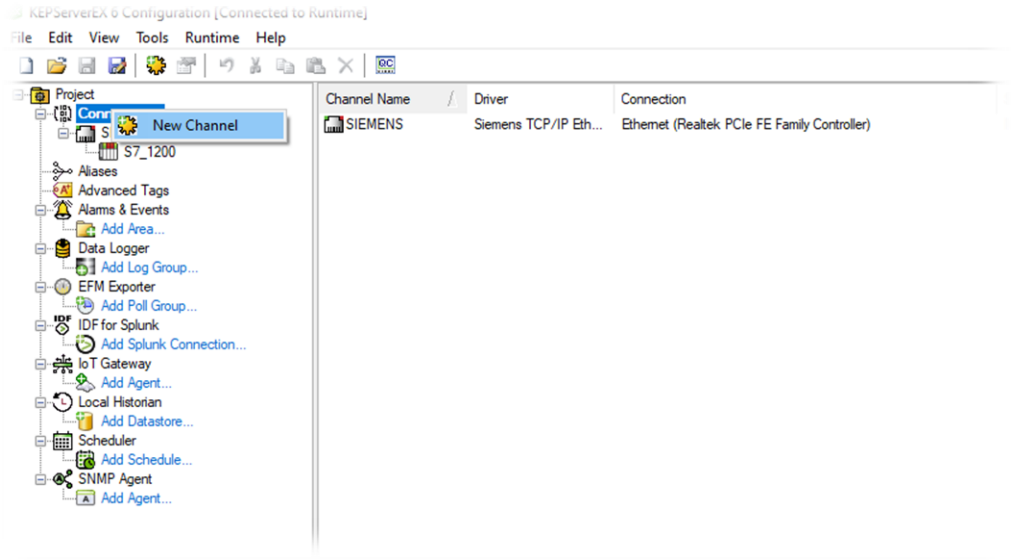

Figura 77 Crear canal de comunicación

Cuando ya se haya seleccionado la opción de crear un nuevo canal, el siguiente paso será indicar con qué Familia y método de comunicación realizaremos la conexión (Fig. 78). En este caso será “Siemens TCP/IP Ethernet”.

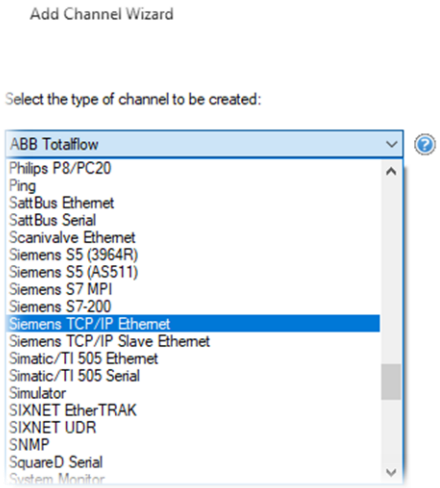

Figura 78 Selección tipo de comunicación

El programa ahora nos pedirá que seleccionemos el adaptador mediante el cual se realizará la conexión (Fig. 79). En este caso la conexión la realizaremos mediante la tarjeta PCI del ordenador a la que hemos asignado una IP del rango configurado en el PLC.

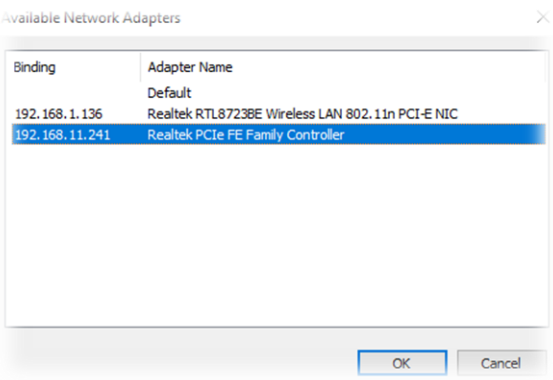

Figura 79 Selección adaptador del canal comunicación

Una vez finalizados estos pasos ya tenemos creado el nuevo canal de comunicación entre el servidor OPC y el PLC S7 1200 de SIEMENS.

4.1.2 Añadir un dispositivo al canal de comunicación

Ahora lo siguiente que tenemos que hacer es añadir un dispositivo a ese canal de comunicación creado.
Para la comunicación con PLC el dispositivo añadido será la CPU. Para ello seleccionamos el canal creado y seleccionamos la opción de "añadir dispositivo" y seguimos los pasos que nos marca el asistente:

En primer lugar, le tendremos que asignar un nombre a dicho dispositivo (Fig. 80). Este paso es esencial para poder diferenciarlos rápidamente si tenemos diferentes dispositivos en el canal.

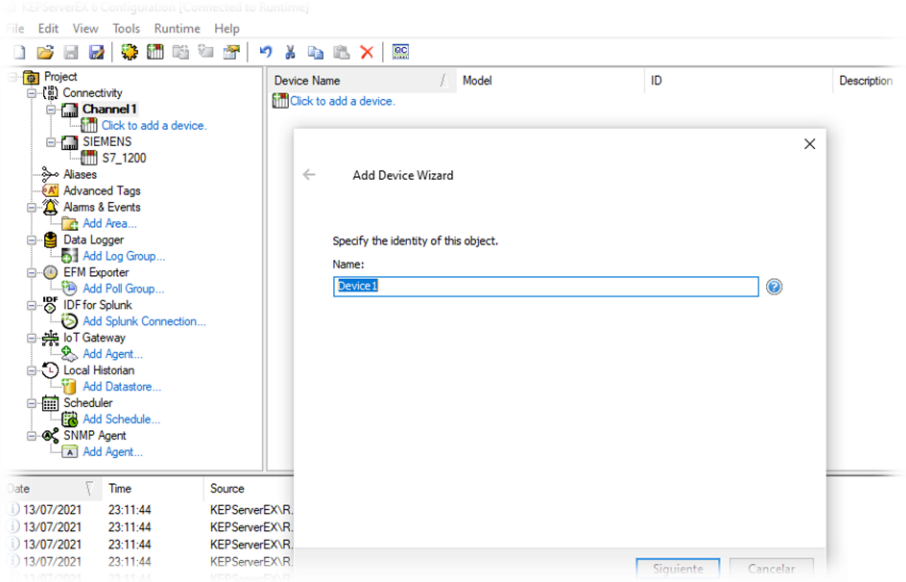

Figura 80 Asignación de nombre a nuevo dispositivo creado

Una vez asignado el nombre al nuevo dispositivo le tenemos que especificar a qué familia de CPU pertenece (Fig. 81) el PLC sobre el que estamos haciendo la conexión.

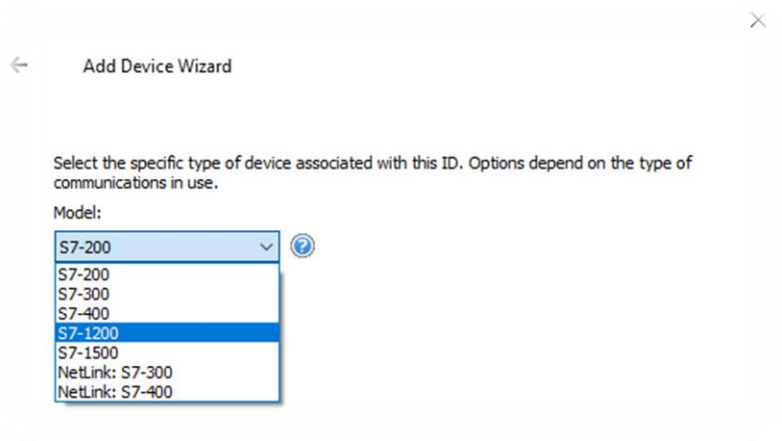

Figura 81 Selección familia CPU

Por último, indicaremos cual es la dirección IP del dispositivo. En este caso será la dirección IP asignada al PLC durante su configuración (Fig. 82)

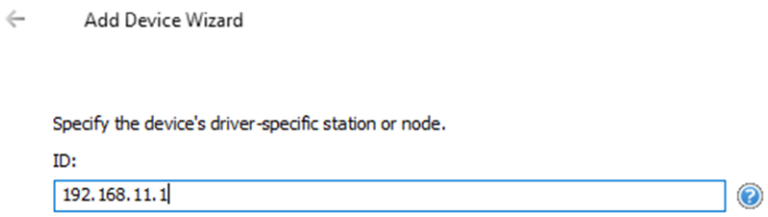

Figura 82 Especificación IP de la CPU en la red

Finalizados todos estos pasos ya tendremos configurada la conexión del servidor OPC con el PLC S7 1200.

4.1.3 Añadir las señales necesarias

Una vez que tenemos configurado el servidor OPC con la conexión al PLC, el siguiente paso será añadir las señales que se utilizarán para el intercambio entre ambos.
Para la creación de estas señales seguiremos unos pasos similares a los seguidos para la creación del nuevo canal de comunicación y la posterior asignación del dispositivo al canal.

Seleccionaremos el dispositivo sobre el que añadir las señales y pulsando botón derecho del ratón nos aparecerá una ventana con opciones. Elegimos la opción de "crear una nueva señal de intercambio" (Fig. 83).

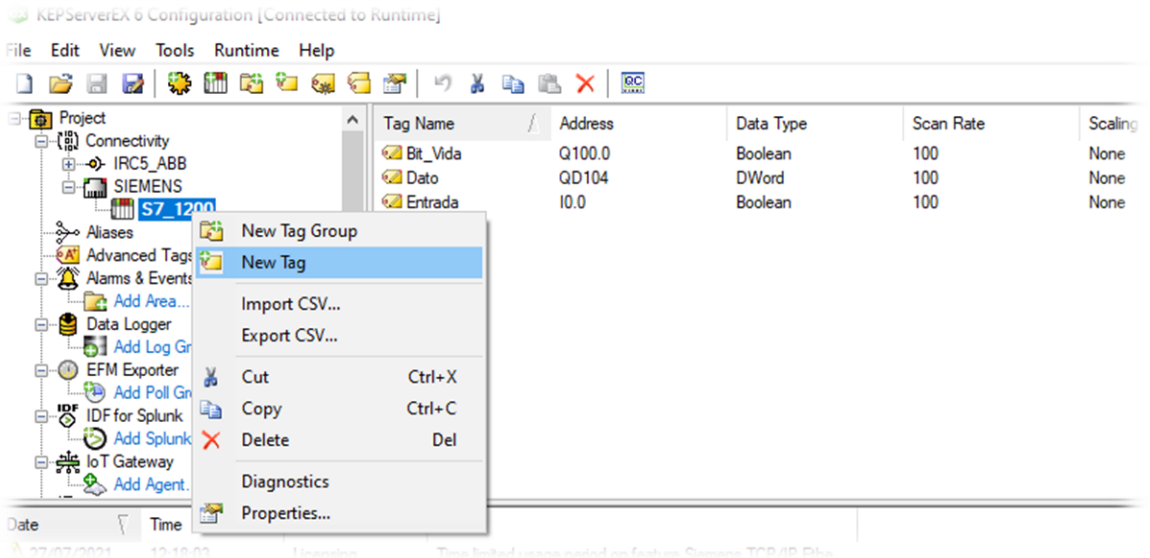

Figura 83 Creación de señales de intercambio OPC

Cuando ya tengamos creada la nueva señal tendremos que configurarla (Fig. 84) correctamente para adaptarla a nuestras necesidades.

Primeramente le daremos un nombre a la señal para su correcta identificación.
A continuación asignaremos la dirección de la señal deseada asignada en el PLC.
Por último seleccionaremos el tipo de dato de dicha señal.

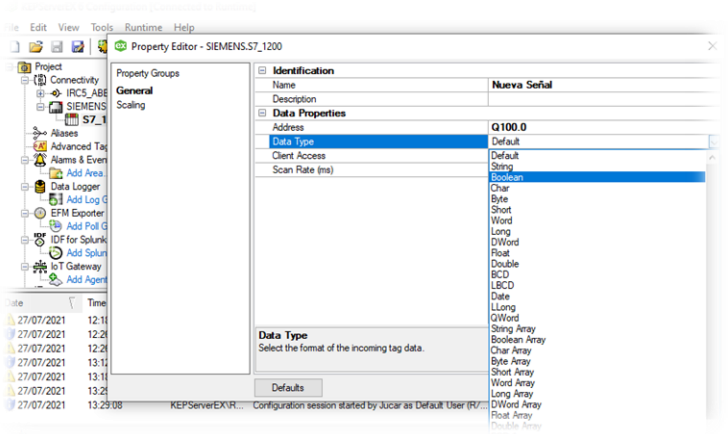

Figura 84 Configuración de señales de intercambio OPC

Una vez realizados todos estos pasos ya tendremos configurada la señal deseada. Para añadir nuevas señales sólo hay que volver a realizar estos mismos pasos hasta que tengamos configuradas todas las señales que queremos utilizar en la comunicación OPC entre el PLC y el Controlador IRC5 para posteriormente utilizarlas en la programación RAPID y el control del ROBOT.

Finalizada la configuración de las señales necesarias, la configuración de la conexión OPC del PLC quedaría de la siguiente forma (Fig. 85):

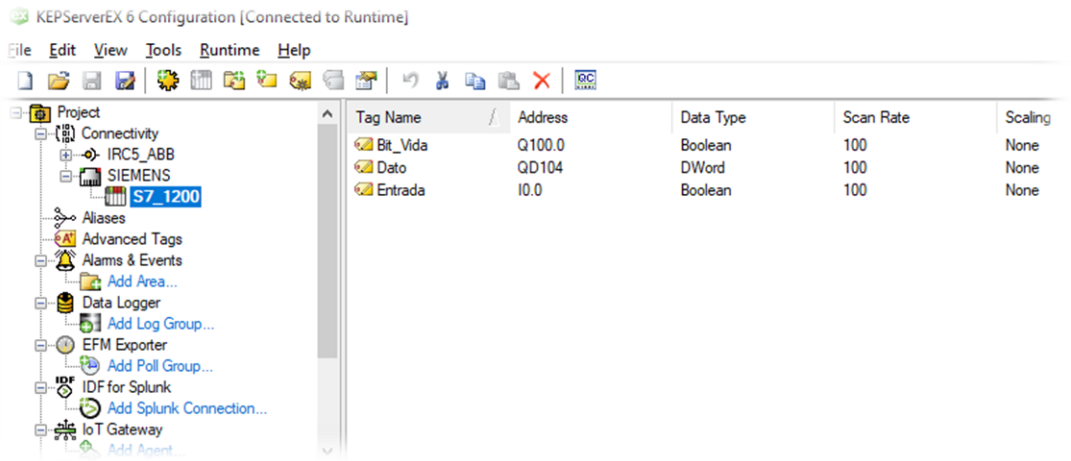

Figura 85 Configuración y señales asignadas a comunicación OPC

4.1.4 Comprobación de la comunicación OPC–PLC

Antes de seguir avanzando realizaré una comprobación de la configuración realizada para comprobar que la comunicación es correcta y funciona adecuadamente. Para ello se realiza un pequeño programa en el PLC para que mediante la activación de varias entradas podamos ver cómo se actualizan los valores en el servidor OPC.

Para la comprobación de la conexión OPC - PLC procederemos de la siguiente manera:

- *Se ejecuta el OPC Quick Client desde KEPServerEX y al seleccionar la conexión con el PLC nos aparecerán las 3 señales declaradas para el intercambio OPC.*
- *Si la señal de conexión es buena (Fig. 86) y está todo correcto nos aparecerá marcado como "GOOD" en la columna "Quality". En caso contrario habremos de revisar todo el proceso anteriormente explicado de configuración de señales hasta que obtengamos la configuración correcta.*
- *Si todo es correcto, en la columna "Value" podremos ver el valor de la señal. Si ahora modificásemos el valor de la señal en el PLC podríamos ver como también se modificaría en la conexión OPC.*

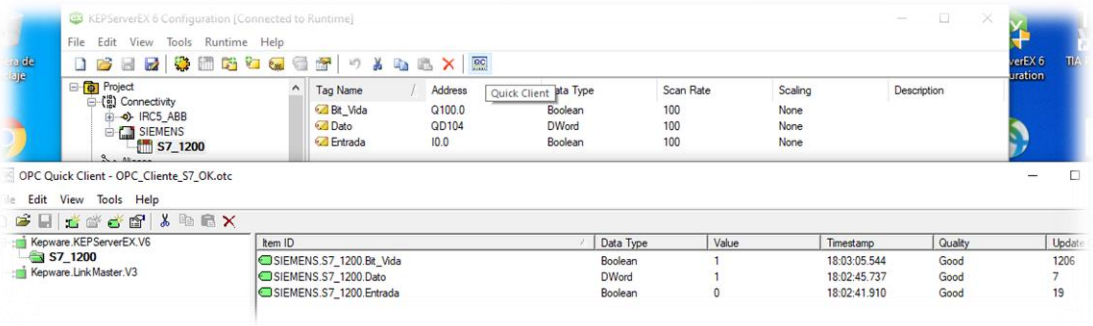

Figura 86 Comprobacion conexión OPC PLC

4.2 Conexión OPC-IRC5 de ABB

Una vez comprobada la correcta configuración y funcionamiento de la conexión entre el servidor OPC y el PLC, el siguiente paso es el de crear la conexión OPC server con el controlador IRC5 de ABB.

Para la configuración del servidor OPC del controlador IRC5 se utilizará el software ABB IRC5 OPC [28], el cual es de libre distribución y está disponible en la web de ABB.

https://new.abb.com/products/robotics/es/robotstudio/descargas

Con el nuevo software descargado e instalado crearemos la conexión entre servidor OPC y controlador IRC5 de la siguiente manera:

4.2.1 Creación alias en server OPC ABB

Ejecutaremos el software y seleccionaremos la opción de crear un nuevo alias (Fig. 87). En este paso y si tenemos la controladora IRC5 instalada en el ordenador, hay que seleccionar la opción “Scan” para la creación del servidor.

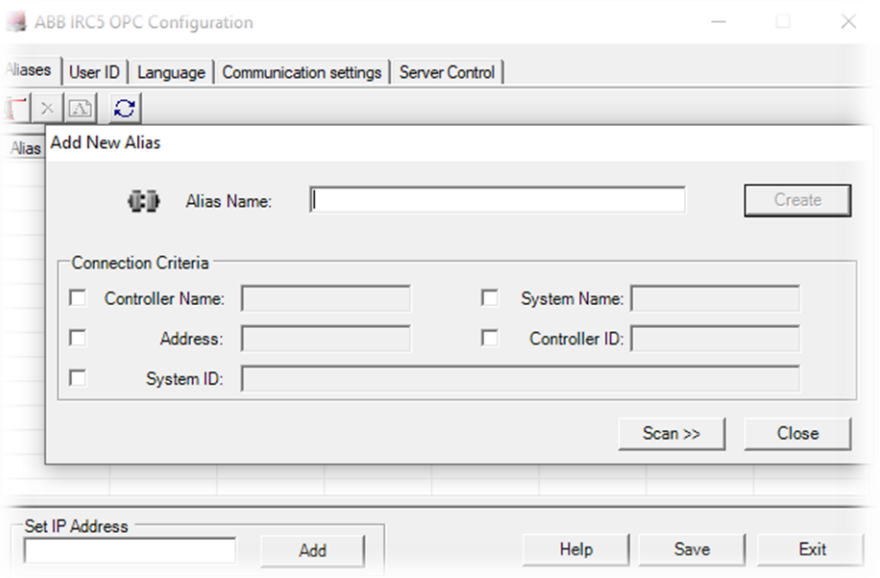

Figura 87 Creación conexión server OPC ABB con IRC5

Al seleccionar la opción “Scan”, si la controladora está correctamente instalada, automáticamente nos mostrará los datos de esta (Fig. 88). Una vez localizada la controladora la seleccionamos y le asignamos un nombre para su posterior identificación.

Queda así creada la conexión y ya la tendremos lista para su uso (Fig. 89). Pero como vemos en la imagen todavía no está conectada a la controladora. En las columnas “IO” y “RAPID” aparece un signo de interrogación.

Una vez creado el servicio, para establecer la conexión hay que activar el icono de refresco y si todo es correcto se establecerá la conexión (Fig. 90), quedando iniciado el servicio OPC con el controlador IRC5.

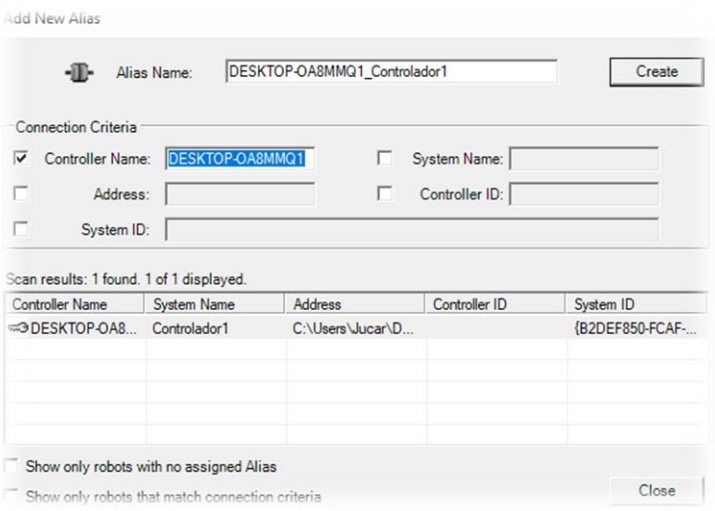

Figura 88 Escaneo automático de las controladoras

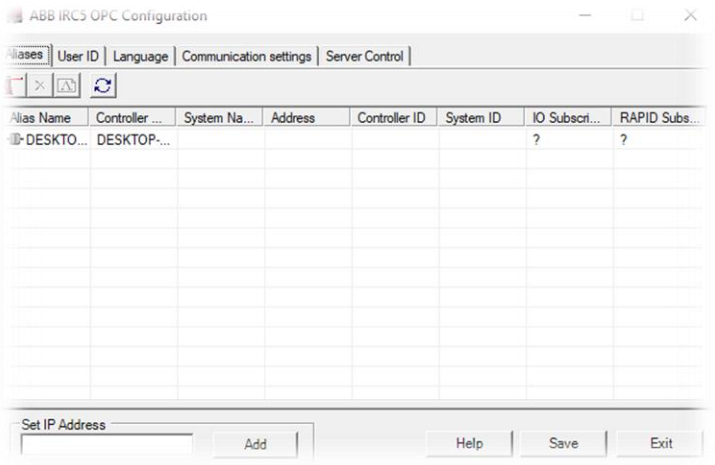

Figura 89 Conexión OPC server-IRC5 creada

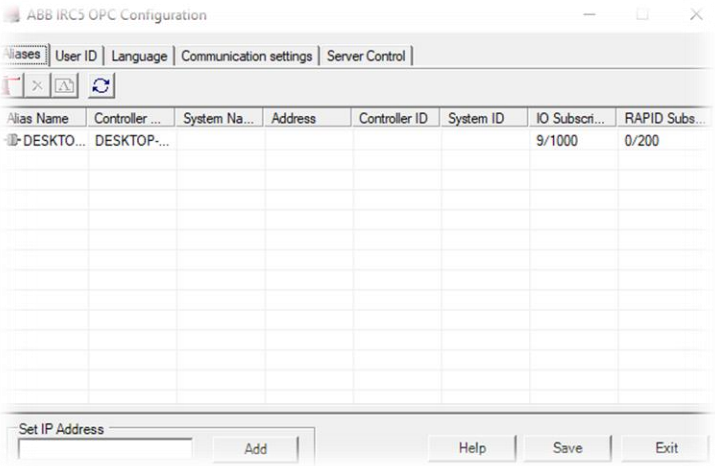

Figura 90 Conexión OPC server-IRC5 establecida

4.2.2. Añadir la conexión OPC-IRC5 al software KEPserverEX

Una vez configurado el server OPC de IRC5 tendremos que añadirlo como nueva conexión en el software KEPServerEX (Fig. 91) al igual que hemos realizado anteriormente para la creación del canal de comunicación con el PLC. Posteriormente habrá que añadirle un nuevo dispositivo, en este caso OPC DA.
Una vez realizados estos pasos ya tenemos realizada la conexión con el servidor OPC de la controladora IRC5 y estaría listo para trabajar con ella.

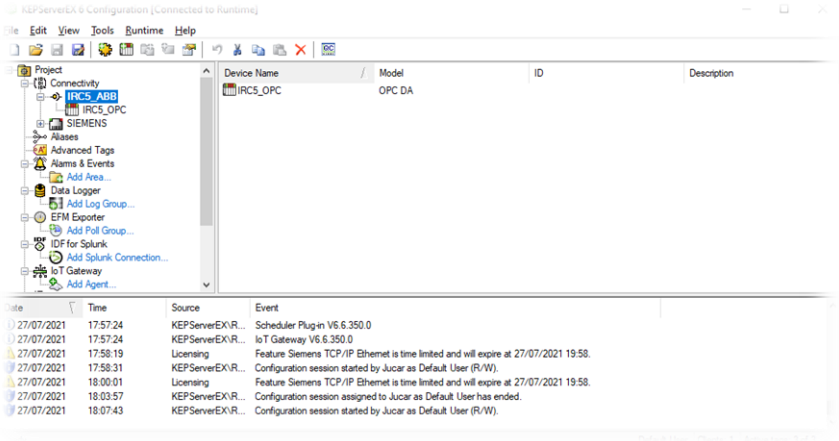

Figura 91 Creación conexión OPC controlador IRC5

A diferencia de la configuración en la conexión del servidor OPC con el PLC, en este caso no es necesario añadir las señales de intercambio ya que el controlador podrá acceder a todas las que tengamos declaradas en RAPID como variables persistentes (Fig. 92). Esto quiere decir que solamente tendremos que declarar todas las señales en RAPID de manera persistente para poder hacer uso de ellas mediante OPC (Fig. 93).

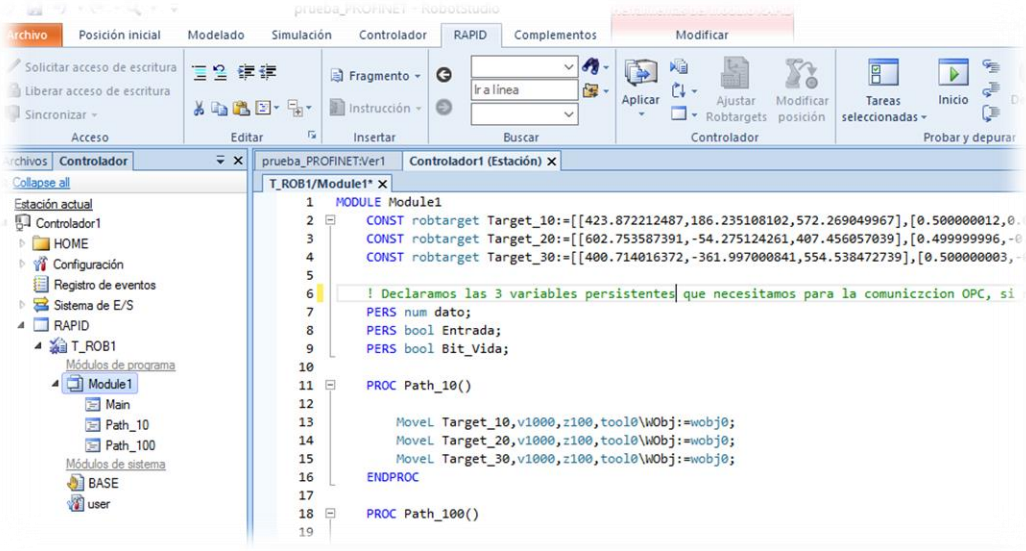

Figura 92 Declaración variables persistentes RAPID

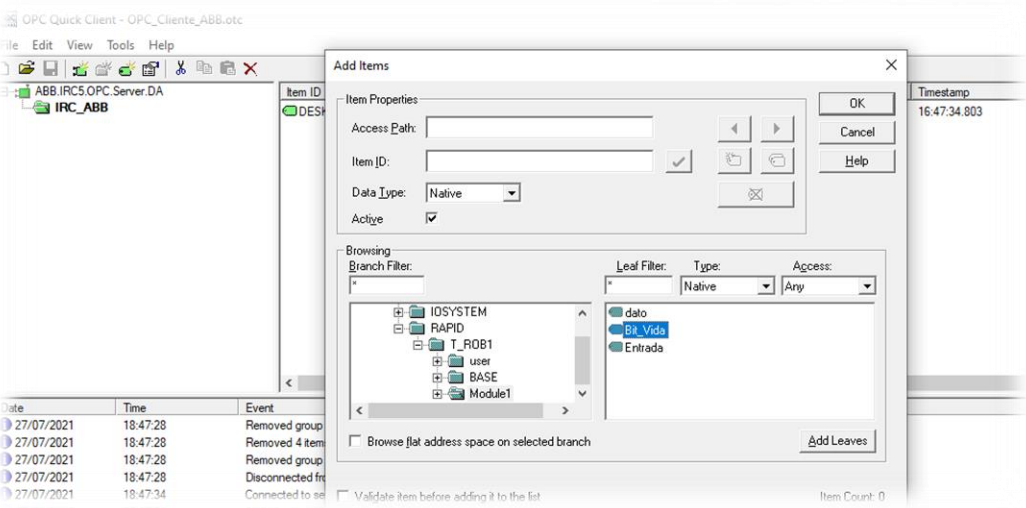

Figura 93 Variables disponibles en OPC

Además, cuando establecemos un canal de comunicación OPC mediante el server de ABB IRC5 también podemos acceder mediante OPC a diferentes variables internas de la controladora para intercambiar información con los diversos dispositivos conectados (Fig. 94). A continuación, se pueden ver algunas de estas señales disponibles.

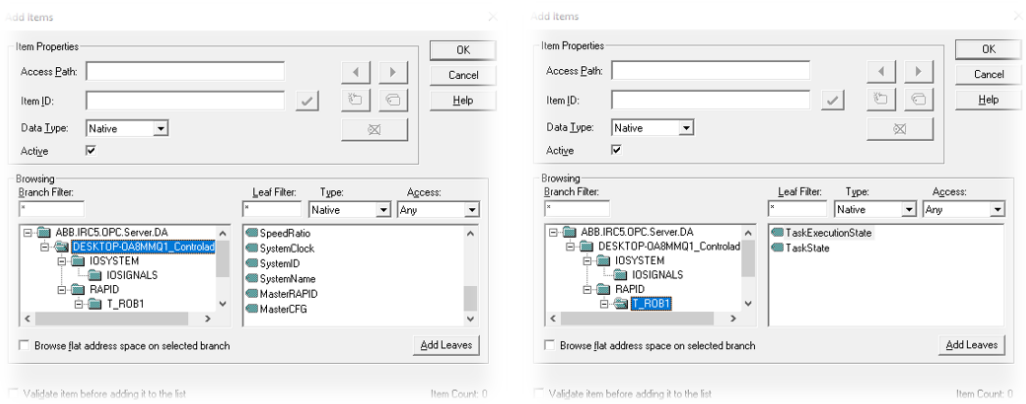

Figura 94 Señales internas controlador IRC5 disponibles OPC

4.2.3 Comprobación de la comunicación OPC–IRC5 ABB

Al igual que hemos hecho al crear la conexión OCP con el PLC, para comprobar que el servidor OPC comunica correctamente con el controlador IRC5 realizaremos un pequeño programa en RAPID. En este programa incluiremos varias condiciones para su ejecución, las cuales serán las señales de intercambio mediante el servicio OPC.
En este caso al ser estas señales de entrada al ROBOT, para ver su comportamiento en la programación realizada en RAPID tendremos que modificar los valores manualmente y ver cómo afectan estas señales a la trayectoria programada en el ROBOT.

Para la comprobación de la conexión procederemos de la siguiente manera:

- *Se ejecuta el OPC Quick Client desde KEPServerEX y al seleccionar la conexión con el IRC_ABB nos aparecerán las 3 señales declaradas para el intercambio OPC.*
- *Si la señal de conexión es buena (Fig. 95) y está todo correcto nos aparecerá marcado como "GOOD" en la columna "Quality". En caso contrario habremos de revisar todo el proceso anteriormente explicado de configuración de señales hasta que obtengamos la configuración correcta.*
- *Si todo es correcto, en la columna "Value" podremos ver el valor de la señal. Si ahora modificásemos el valor de la señal manualmente podríamos ver como en el programa RAPID creado también nos modificaría ese valor (Fig. 96). Para modificar manualmente el valor de una señal tendremos que seleccionarla y mediante el botón derecho del ratón seleccionamos en el menú desplegable la opción "escritura síncrona" (Fig. 97). De esta manera nos abrirá una nueva ventana (Fig. 98) en la que podremos indicar el nuevo valor de dicha variable.*

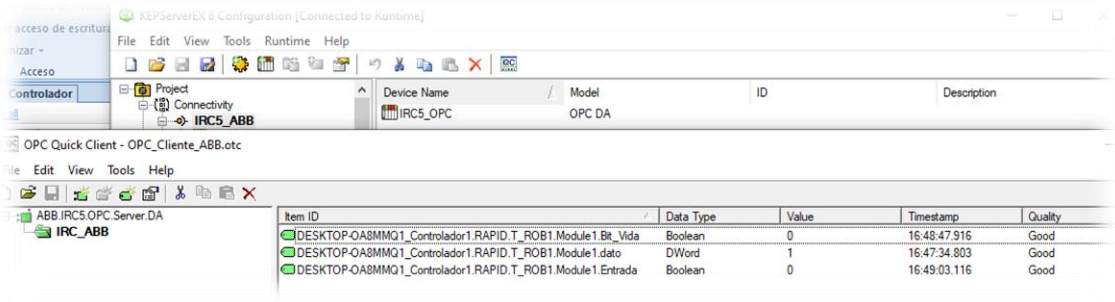

Figura 95 Señales comunicación OPC con IRC5

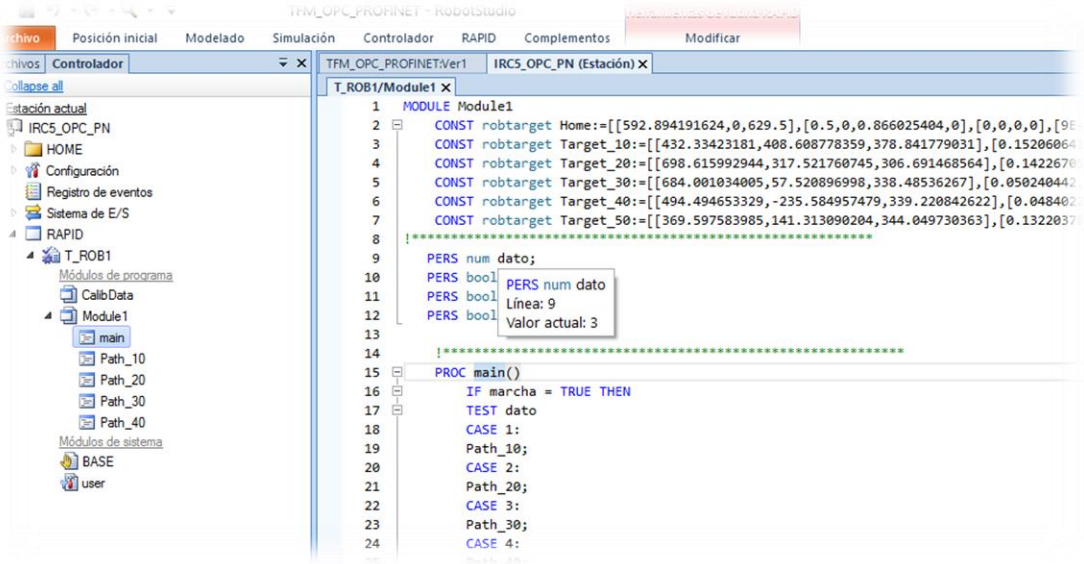

Figura 96 Actualizacion dato modificado mediante OPC

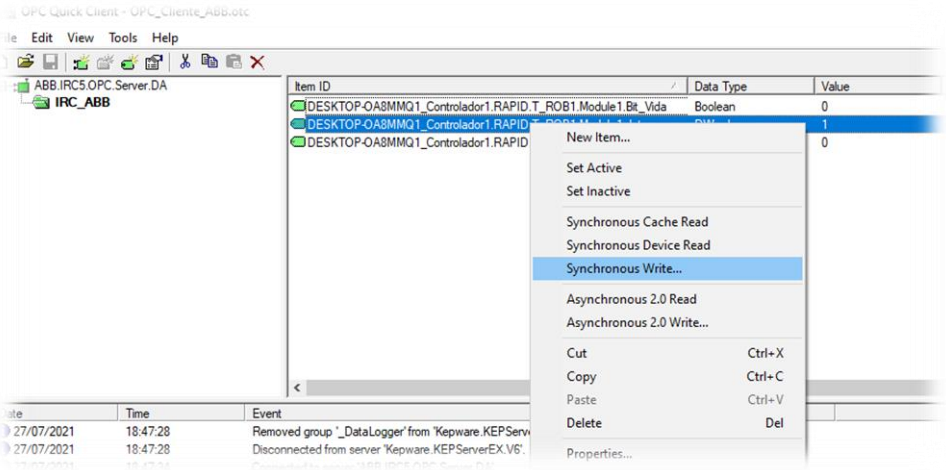

Figura 97 Selección de variable a modificar

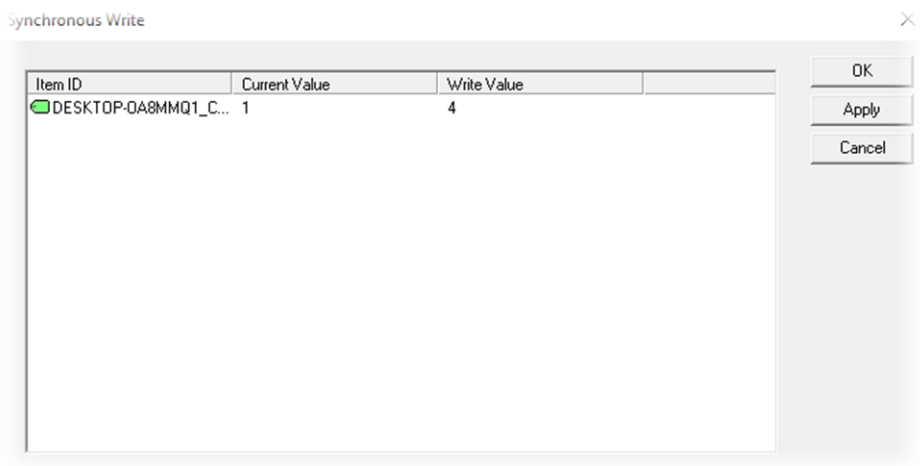

Figura 98 Nuevo valor aplicado a la variable seleccionada

Por último, una vez comprobado que todas las señales se actualizan en RAPID mediante OPC, creamos un pequeño programa RAPID condicionado por esas señales. Podemos ver como durante la simulación del ROBOT mediante RobotStudio la trayectoria realizada por el mismo varía dependiendo del valor de la señal de intercambio.

4.3 Conectar variables con LinkMaster

Una vez que tenemos creadas las dos conexiones OPC y comprobado su correcto funcionamiento de manera individual, el siguiente paso será el de enlazar todas las variables de manera que las salidas del PLC lleguen como entradas al controlador IRC5 y las salidas del controlador IRC5 lleguen como entradas el PLC.

Para ello utilizaremos el software LinkMaster [29], que también pertenece al paquete de KEPServerEX y por el mismo motivo que al usar KEPServerEX, utilizaremos la versión de prueba limitada a 2 horas máximo de funcionamiento continuo.

4.3.1 Creación grupo LinkMaster

Para la creación del link de todas las señales ejecutamos el programa LinkMaster y una vez arrancado en la pantalla principal (Fig. 99) seleccionamos la opción de crear un nuevo grupo.

Una vez creado el nuevo grupo (que contendrá todas las señales a intercambiar), el siguiente paso será el de crear todas esas señales que intercambiaremos entre PLC-ABB. Para ello seleccionamos la opción de crear nuevo link (Fig. 100) dentro del grupo creado.

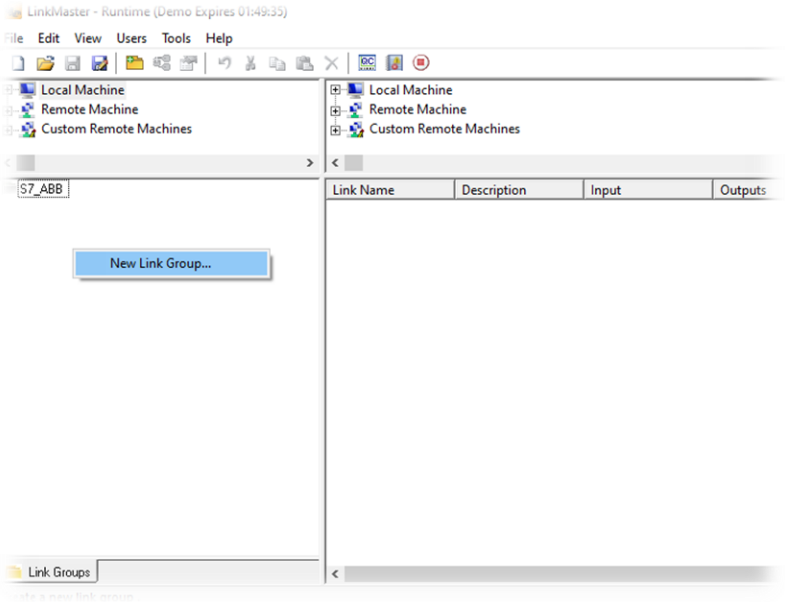

Figura 99 Creación link de grupo

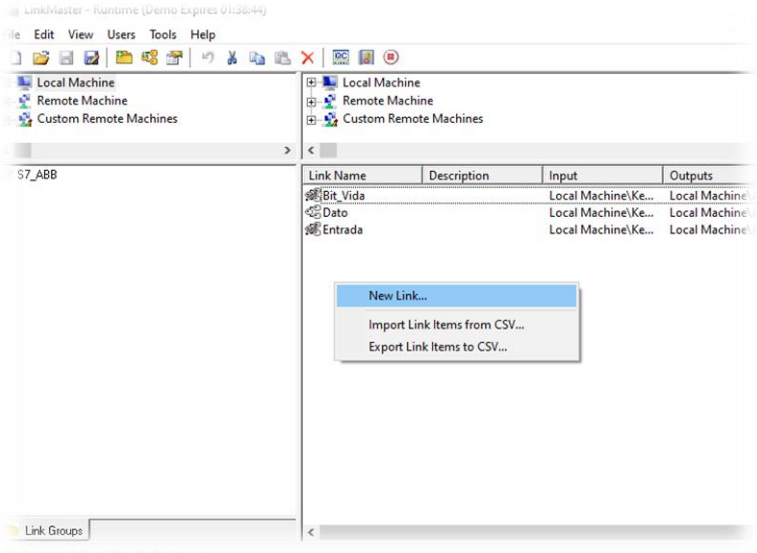

Figura 100 Creación señales del grupo

4.3.2 Creación señales en grupo

Una vez creado el grupo y seleccionada la opción de crear un nuevo link es cuando tendremos que enlazar las señales a compartir. En primer lugar, le daremos un nombre a ese nuevo link (Fig. 101) y posteriormente seleccionaremos cual será la entrada (Fig. 102)

de ese link (de donde viene la señal). Por último, seleccionaremos cual será la señal de salida (Fig. 103) del link (a dónde va esa señal).

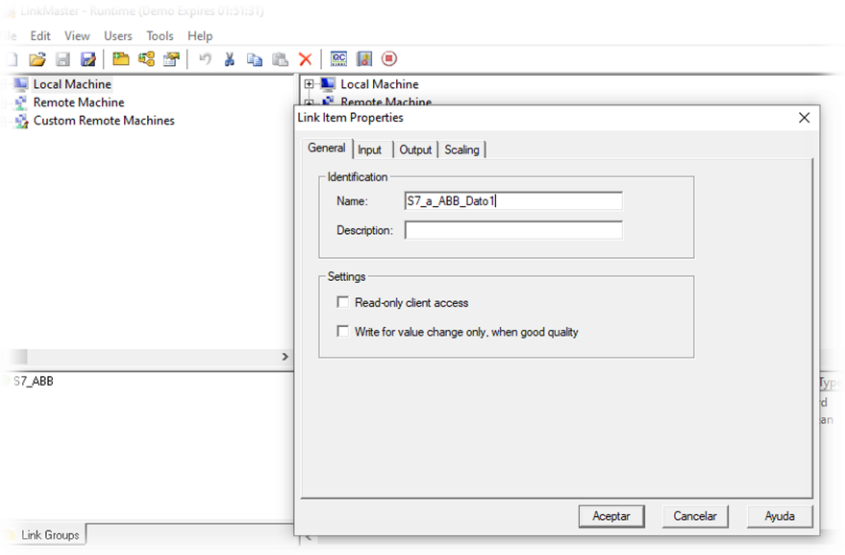

Figura 101 Selección de nombre link

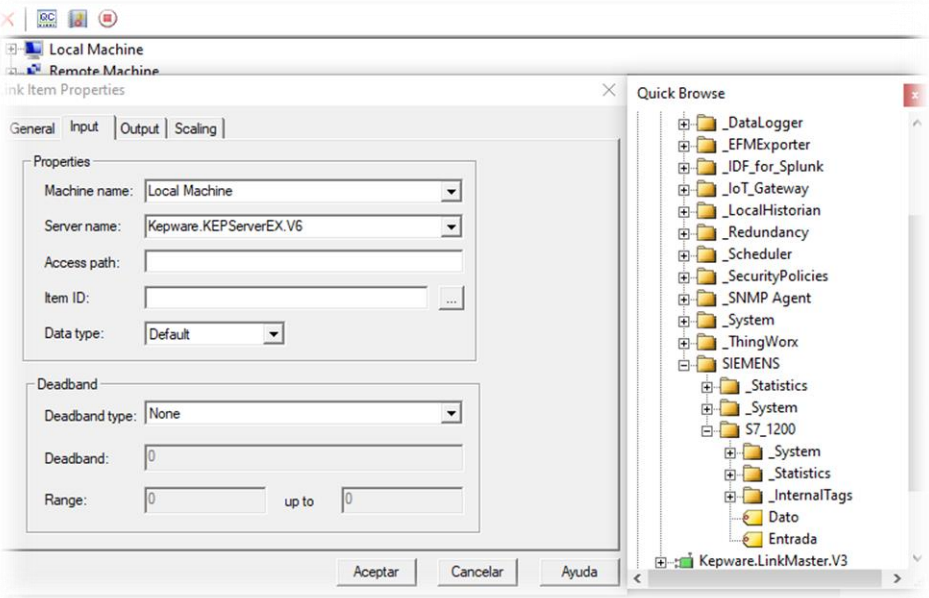

Figura 102 Selección entrada link

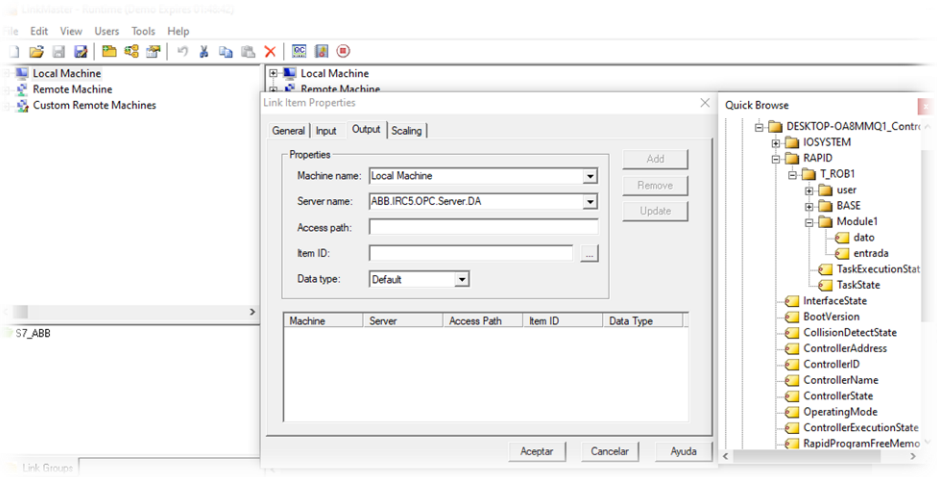

Figura 103 Selección salida link

Estos pasos los repetiremos para todas las señales que tengamos que enlazar, quedando el grupo final configurado para el intercambio de señales entre PLC y controladora IRC5 (Fig. 104).

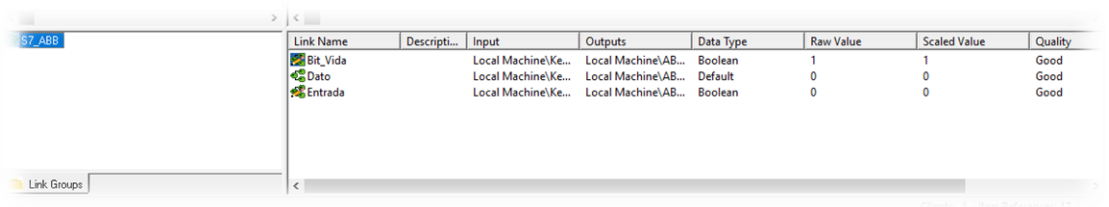

Figura 104 Grupo de señales intercambio creado

Una vez tenemos finalizada la configuración de todas las señales necesarias ya tendremos el sistema listo para compartir todas las señales y poder controlar y condicionar los movimientos del ROBOT mediante estas señales y su programación en RAPID.

4.4 Comunicación ABB-PLC mediante PROFINET

Una vez visto y comprobado el correcto funcionamiento de la comunicación entre ABB-PLC mediante OPC, realizaremos las configuraciones y pruebas para establecer esa comunicación mediante PROFINET.

4.4.1 Configuración PROFINET en PLC

4.4.1.1 Instalación de los GSD en TIA PORTAL

Si es la primera vez que trabajamos en TIA PORTAL mediante comunicación PROFINET con controladora de robot ABB IRC5, para poder integrar la controladora IRC5 en el proyecto de TIA PORTAL tendremos que instalar en el entorno de programación los GSDML correspondientes a la controladora IRC5 utilizada.

En el caso de ABB estos archivos GSDML están ubicados en la carpeta donde tenemos instalado el software RobotWare utilizado en nuestra controladora. Para poder localizarlos en las carpetas de instalación de RobotWare tenemos la opción de buscarlos manualmente o desde el propio software de RobotStudio (Fig. 105). Debemos copiar la ruta de la carpeta y para ello accedemos a la pestaña de "complementos" y seleccionamos la carpeta de la versión de RobotWare correspondiente. Mediante el botón derecho del ratón seleccionamos la opción de "abrir carpeta de paquete". Una vez abierta (Fig.106) copiaremos la ruta de dicha carpeta para agregarla en TIA PORTAL.

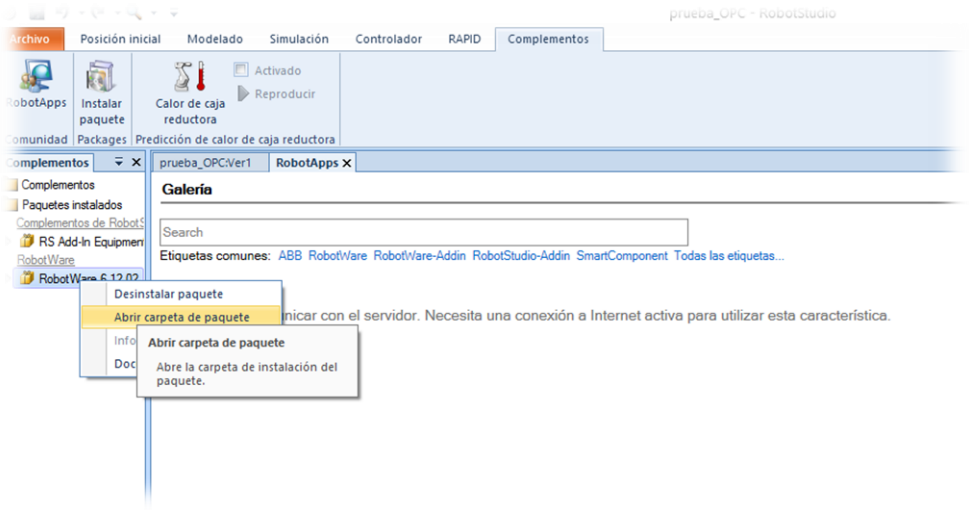

Figura 105 Selección ruta GSDML desde RobotStuido

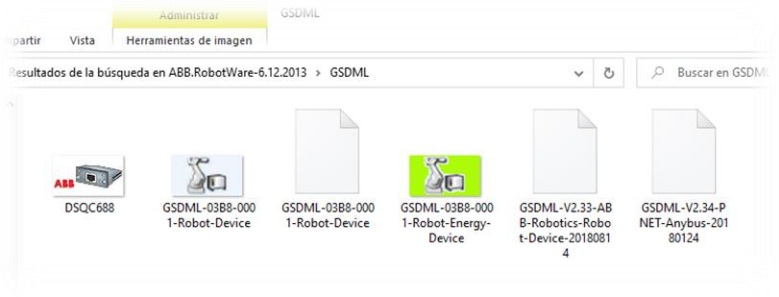

Figura 106 Contenido carpeta GSDML

Una vez que tenemos la ruta donde se encuentran los GSD necesarios para integrarlos en el proyecto de TIA PORTAL, el siguiente paso será instalarlos para poder hacer uso de ellos al confeccionar el proyecto.

Para su instalación arrancamos TIA PORTAL (Fig. 107) y accedemos a la pestaña "Opciones" y seleccionamos "administrar archivos de descripción de dispositivos" y en la opción "ruta de origen" indicaremos la ruta anteriormente copiada. Una vez reconocido por el programa los GSDML, como no los tenemos instalados nos dará la opción de instalarlos en el programa.

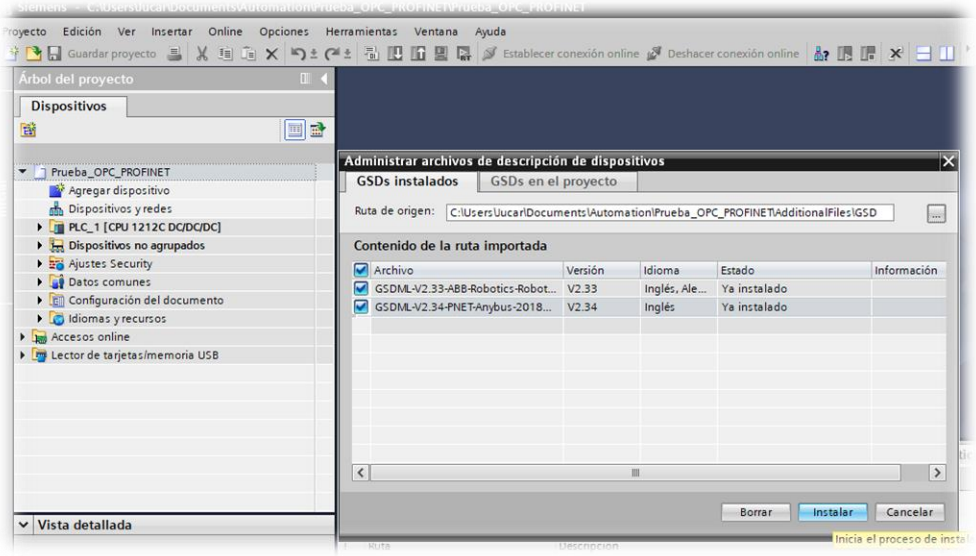

Figura 107 Instalación GSDML en TIA PORTAL

Después de instalado el GSD en TIA PORTAL ya podremos usar la tarjeta de comunicación PROFINET en controladora IRC5 de ABB en el proyecto de TIA PORTAL.

4.4.1.2 Implementar tarjeta PROFINET ABB en TIA PORTAL

El siguiente paso será integrar la conexión PROFINET correspondiente a la controladora IRC5 en la configuración Hardware del proyecto creado en TIA PORTAL.
Para ello procederemos de la siguiente forma:

En primer lugar, insertamos la tarjeta de comunicación PROFINET instalada en controladora de Robot IRC5 (Fig. 108).

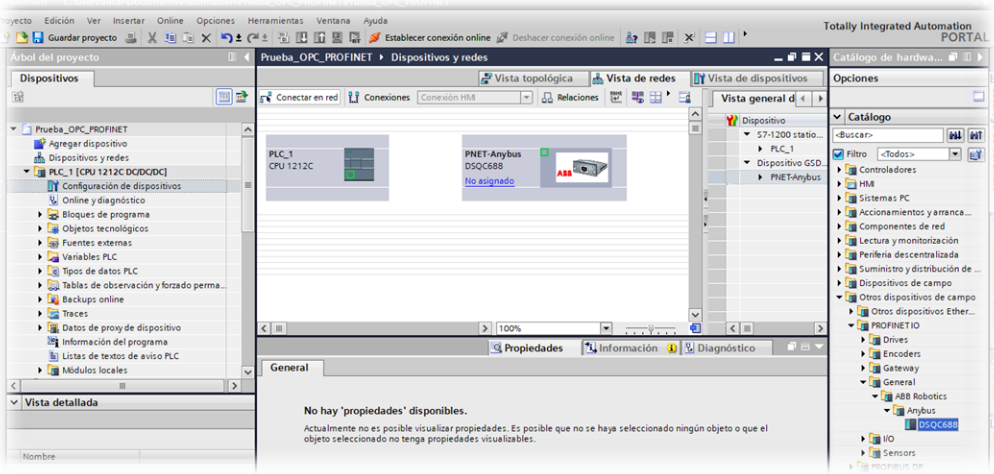

Figura 108 Insertar tarjeta comunicación en proyecto

Una vez tenemos la tarjeta insertada en el proyecto, el siguiente paso es el de conectarla a la red existente en el proyecto para que ambos dispositivos estén conectados en la misma red (Fig. 109).

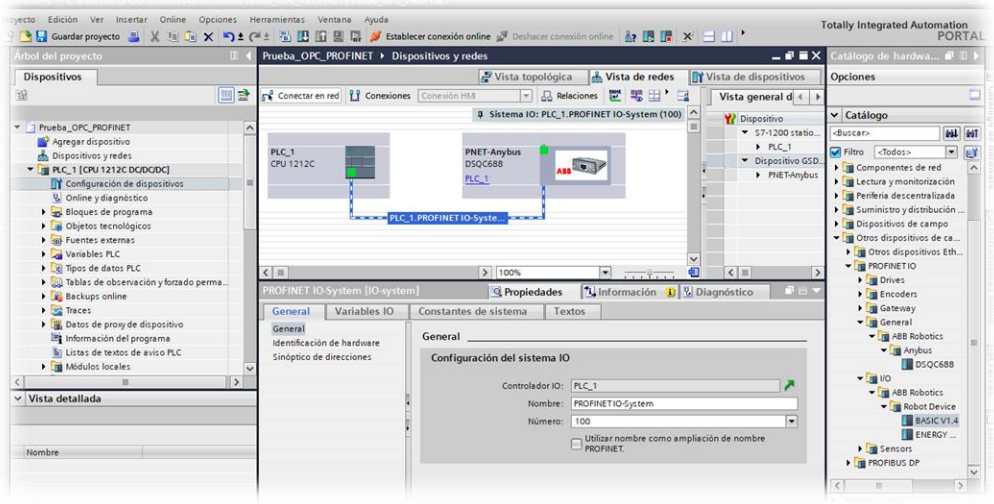

Figura 109 Conexión del PLC-Tarjeta ABB

A continuación, asignaremos la dirección IP y el nombre del equipo (Fig. 110) en la red PROFINET creada entre PLC, HMI y Controladora IRC5 para que de esta manera se pueda establecer una correcta comunicación entre los dispositivos.

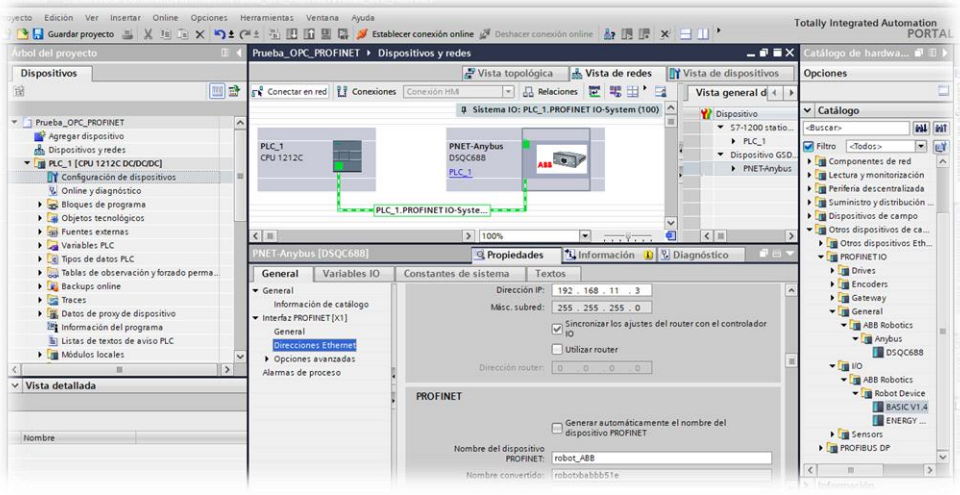

Figura 110 Asignación IP y Nombre tarjeta ABB

Finalizados estos pasos ya tendremos correctamente insertada la tarjeta de comunicación PROFINET de la controladora IRC5 en la red creada por el PLC y de esta manera ya se puede establecer una comunicación entre el Robot y el PLC S7 1200.

4.4.1.3 Asignación Bytes de comunicación

Una vez que tenemos configurado todo el hardware en el PLC y asignados los nombres y direcciones IP, lo que tendremos que hacer es dimensionar los bytes de comunicación entre el robot y el PLC.

Para ello añadiremos al hardware creado los módulos correspondientes para tener los Bytes necesarios para la transferencia de información entre PLC y robot.

Para crear los bytes de comunicación accederemos al dispositivo creado (tarjeta de comunicación IRC5, DSQC688) en el proyecto de PLC y añadiremos los módulos de salida y entrada necesarios. El tamaño de estos módulos dependerá de la cantidad de datos que tengamos que transferir entre los dispositivos. En nuestro caso hemos seleccionado 32 byte de intercambio de datos (Fig. 111) y le asignaremos una dirección de comunicación (Fig. 112). En este caso le he asignado la dirección comenzando por el byte 100. Esta dirección será la que posteriormente utilizaremos en el programa de TIA PORTAL para intercambiar dichos datos con la controladora IRC5.

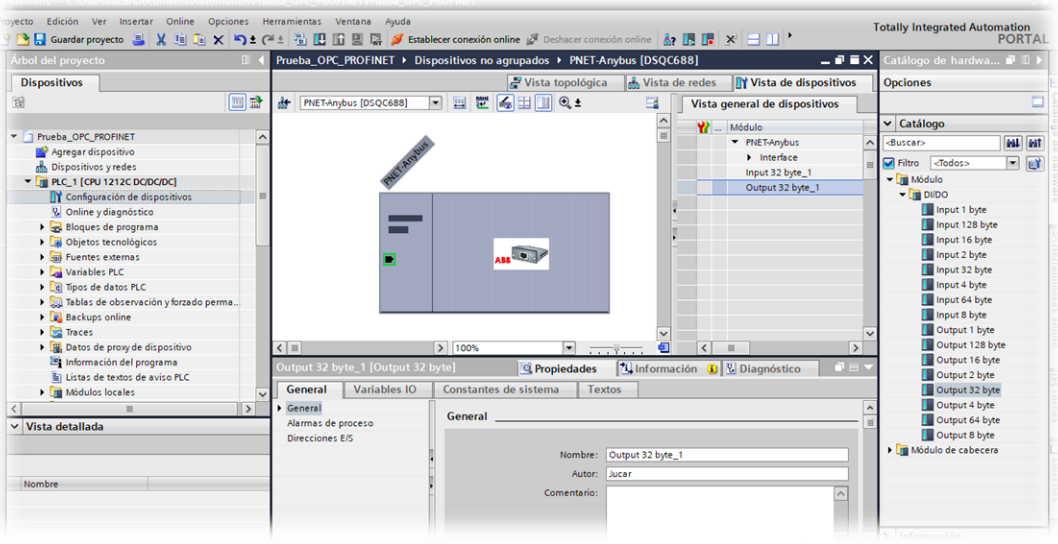

Figura 111 Creación módulos entrada y salida de comunicación.

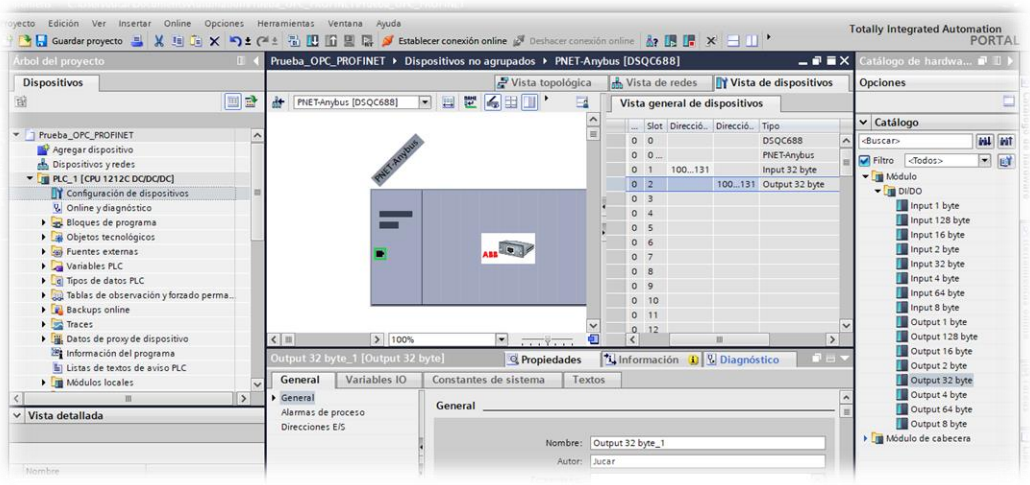

Figura 112 Asignación dirección módulos entrada y salida de comunicación

4.4.1.4 Cargar hardware a PLC

Una vez finalizada toda la configuración del módulo de comunicación le cargamos todo el Hardware al PLC para dejarlo operativo y así poder establecer la comunicación con la controladora IRC5.

Después de todo este proceso tendremos cargado el programa en el PLC y nos conectaremos Online (Fig. 113). Vemos como tenemos toda la configuración cargada y el PLC en marcha, pero éste nos indica que tenemos un error en un dispositivo. Esto es debido a que no tenemos físicamente la tarjeta de comunicación (DSQC688) en la controladora y por este motivo el PLC no es capaz de verla en la red PROFINET creada (Fig. 114).

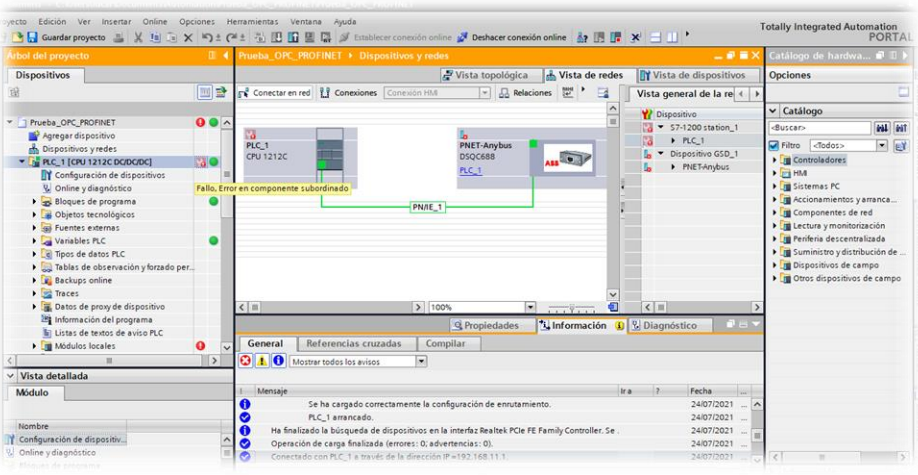

Figura 113 Conexión Online con PLC con nuevo Hardware

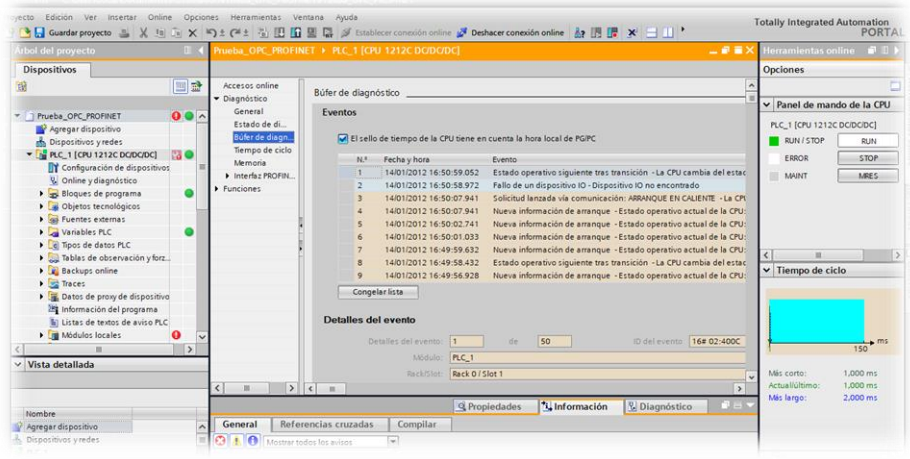

Figura 114 Mensaje error PLC al no encontrar dispositivo IO

En este caso al simular en ROBOTSTUDIO la comunicación PROFINET y no tener físicamente la tarjeta NO podemos dejar operativo este tipo de comunicación ya que la red PROFINET no detecta la tarjeta de comunicación del robot y no puede establecer la comunicación. Para poder probar este tipo de comunicación necesitaríamos tener físicamente tanto la controladora IRC5 como la tarjeta de comunicación de ABB DSQC688.

En la siguiente foto podemos ver como al intentar asignarle el nombre del dispositivo en la red PROFINET no lo encuentra al estar simulado el sistema.

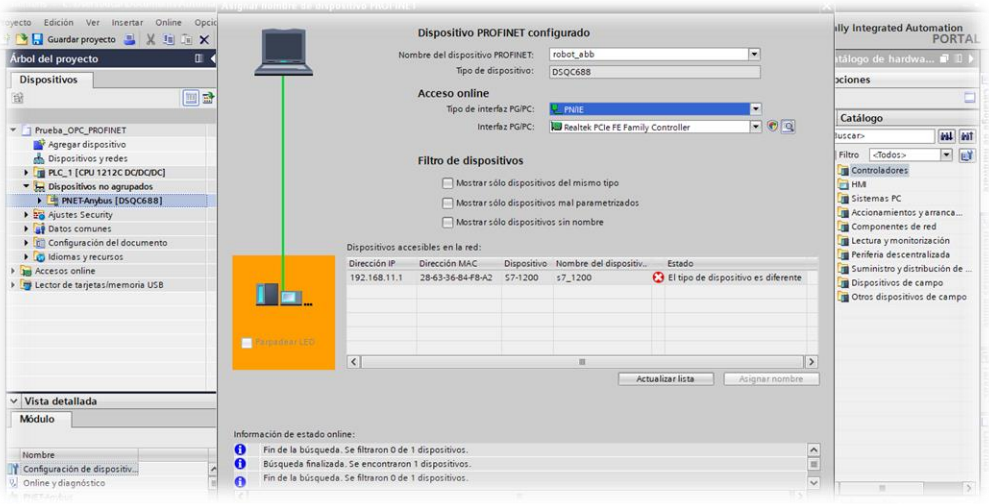

Figura 115 Mensaje error PLC al no encontrar dispositivo IO

4.4.2 Configuración PROFINET en IRC5

Llegados a este punto con la configuración finalizada en TIA PORTAL ya podríamos comenzar la transferencia de los datos mediante el programa residente en el PLC, pero para terminar de completar la configuración de la comunicación faltaría la configuración en la controladora IRC5 de ABB.

El primer paso que tenemos que hacer para configurar la controladora IRC5 es comprobar si en las opciones (Fig. 116) tenemos el módulo de comunicación PROFINET instalado. Si no está instalado tendríamos que añadirlo a la controladora (opción: 840-3 PROFINET Anybus Device) y reiniciarla para poder continuar con la configuración.

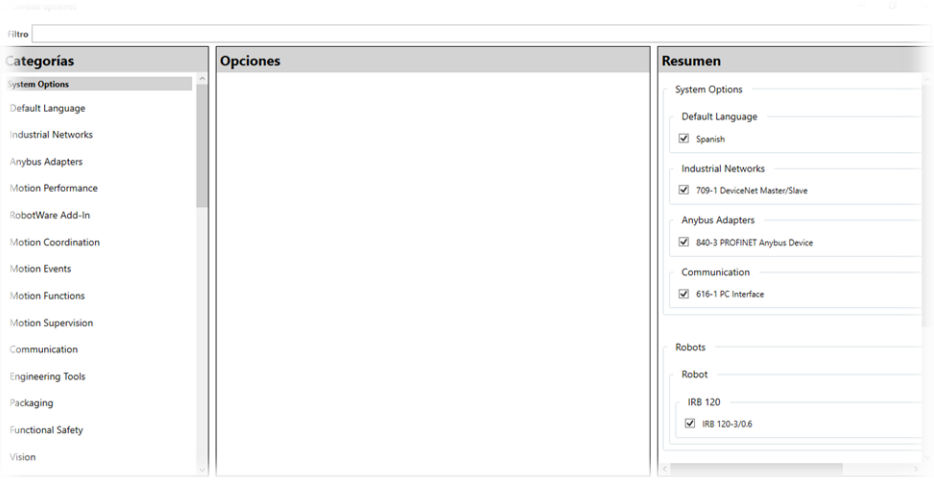

Figura 116 Opciones instaladas en controladora IRC5

Una vez tenemos añadida la opción PROFINET en la controladora, el siguiente paso será el de insertar el módulo de comunicación PROFINET en la misma. Para ello en la pestaña "controladora" en el árbol de la derecha seleccionamos la opción "I/O System" y en la ventana de la derecha (Fig. 117) debería aparecer la posibilidad de comunicación mediante PROFINET.

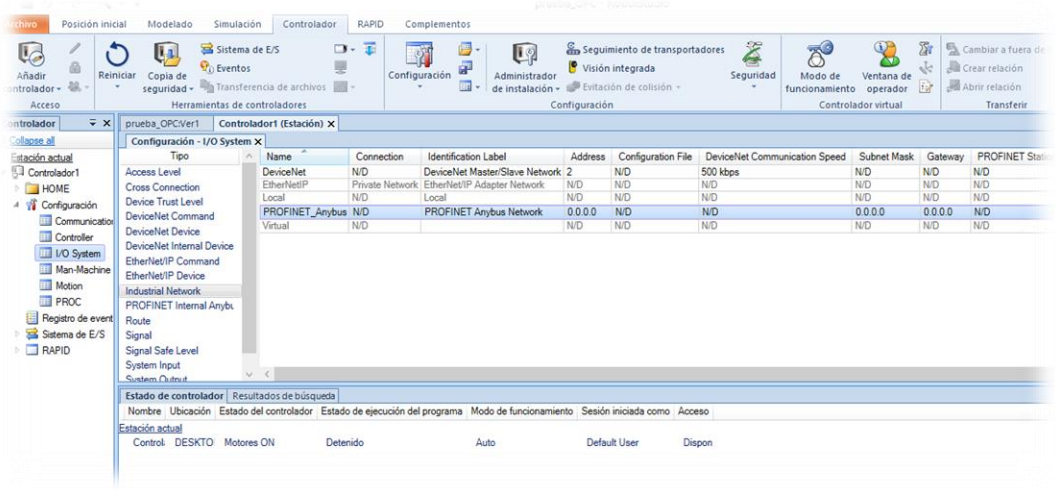

Figura 117 Insertar modulo PROFINET en IRC5

Cuando insertamos el módulo PROFINET el programa nos avisa de que estos cambios requieren el reinicio del controlador (Fig. 118), pero debido a que debemos hacer más modificaciones que requieren dicho reinicio, no lo haremos en este momento y continuaremos con la configuración de la controladora IRC5.

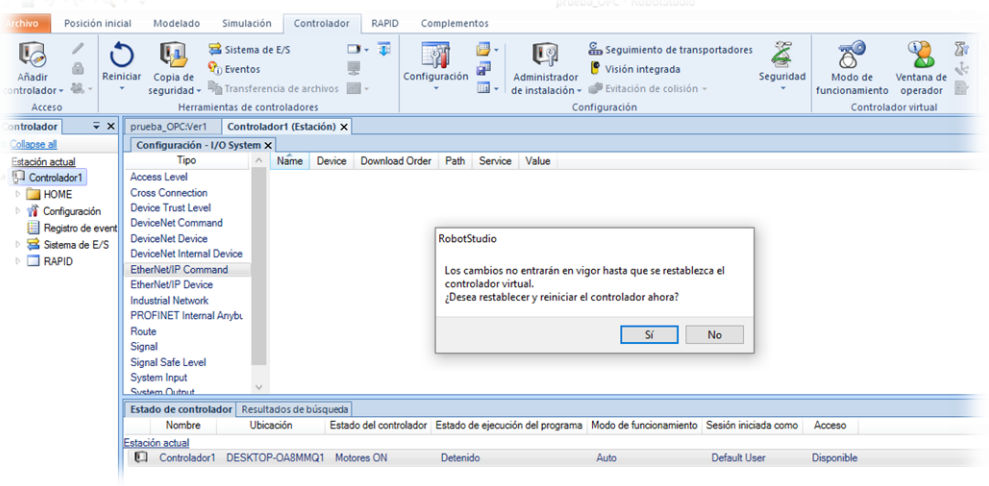

Figura 118 Mensaje controladora una vez añadido Profinet.

Después de crear la comunicación PROFINET debemos configurarla con los mismos datos que la hemos creado en TIA PORTAL para que ambos sistemas sean reconocidos por la red PROFINET creada. Para ello seleccionamos la red PROFINET creada y se nos desplegará un menú (Fig. 119) para su configuración, en el cual introduciremos los mismos datos que hemos puesto en TIA PORTAL.

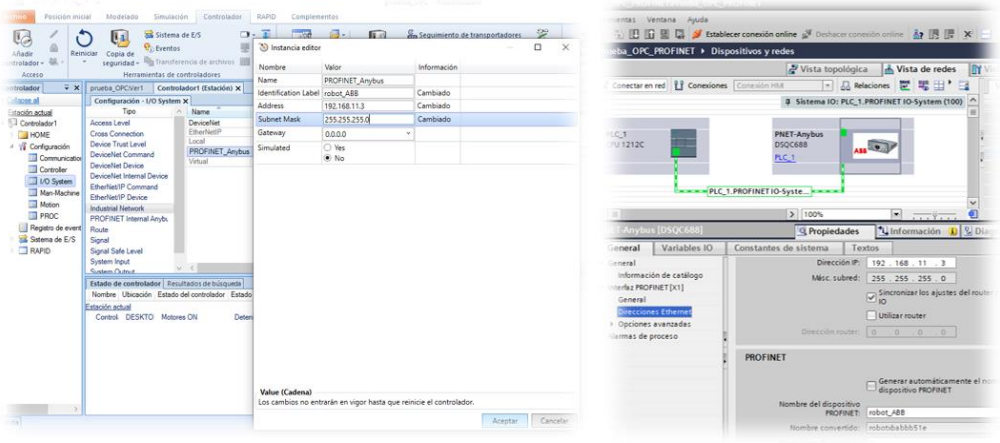

Figura 119 Asignación IP y nombre a tarjeta comunicación ABB

Una vez asignado IP y su nombre en la red, el siguiente paso es asignar los datos a transferir mediante la red PROFINET (Fig. 120). Al igual que en la asignación de IP y nombre de red, tendremos que seleccionar el mismo tamaño de datos que lo indicado en TIA PORTAL.

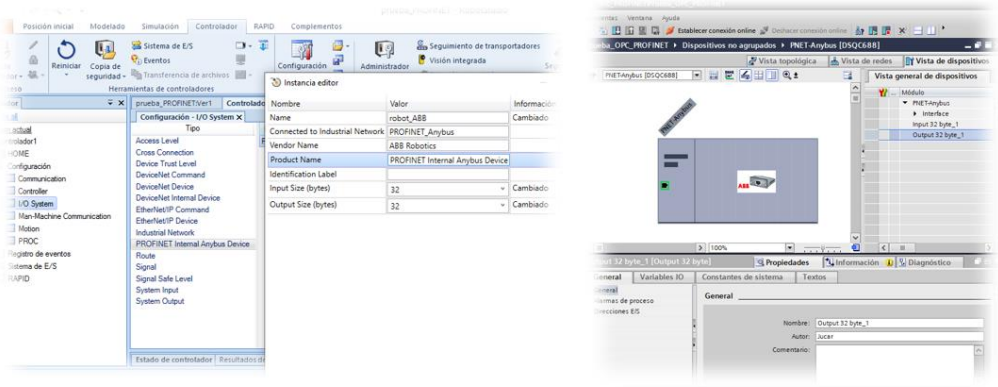

Figura 120 Asignación bytes de comunicación ABB-PLC

4.4.2.1 Creacion de señales entre robot-PLC

Configurada ya la comunicación y dimensionados los bytes de intercambio, el siguiente paso es crear las señales de intercambio entre ROBOT y PLC.

Esto lo podemos hacer da varias maneras y en diferentes menus de opciones de RobotStudio.

4.4.2.1.1 Crear señales desde pestaña “controlador”

Para crear las señales desde la pestaña “controlador” accederemos al desplegable de la izquierda de la pantalla y seleccionaremos la pestaña “I/O System” y porteriormente en el desplegable que aparece a la derecha seleccionamos “signal” (Fig. 121) e iremos añadiendo una a una todas las señales necesarias, configurando individualmente el tipo de señal que es.

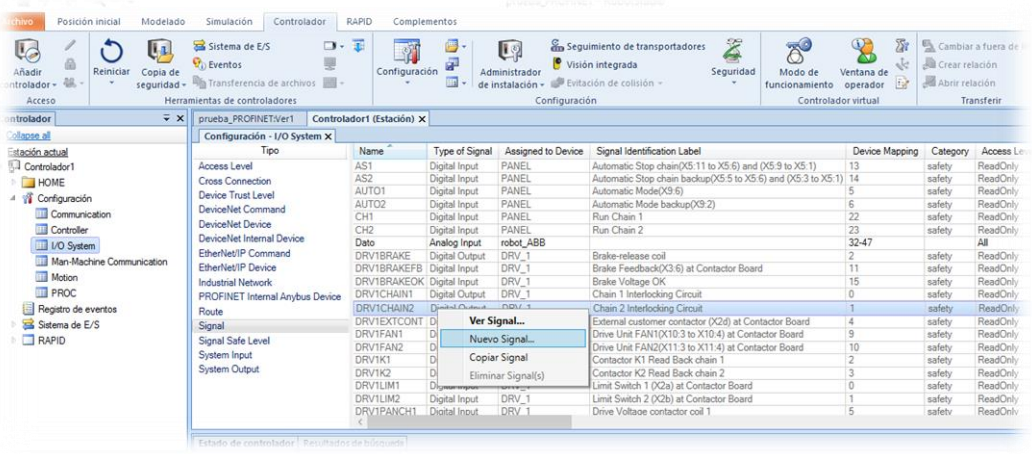

Figura 121 Creación señales necesarias para comunicación

Una cosa que debemos tener en cuenta a la hora de crear señales son los bits necesarios para cada señal y su correcto mapeo a la hora de crearla.

- *Una señal digital necesitará 1 bit.*
- *Una señal tipo Word necesitará 16 bits.*
- *Una DWord 32 bits.*

Estas consideraciones las deberemos tener en cuenta a la hora de establecer el mapeo de las señales nuevas para evitar que entren en conflicto con otras ya creadas.

Para evitar estos conflictos reservaremos los 2 primeros Bytes (0-15) para las entradas digitales (16 entradas) y usaremos los siguientes para las entradas Word y Dword necesarias. La primera que se declare comenzará en el bit de mapeo 16.

En nuestro caso hemos configurado la comunicación con 32 bytes de entrada y salida, por lo que no tendremos problemas a la hora de mapear las señales creadas.

En primer lugar, crearemos una señal digital según las consideraciones antes comentadas (Fig.122).

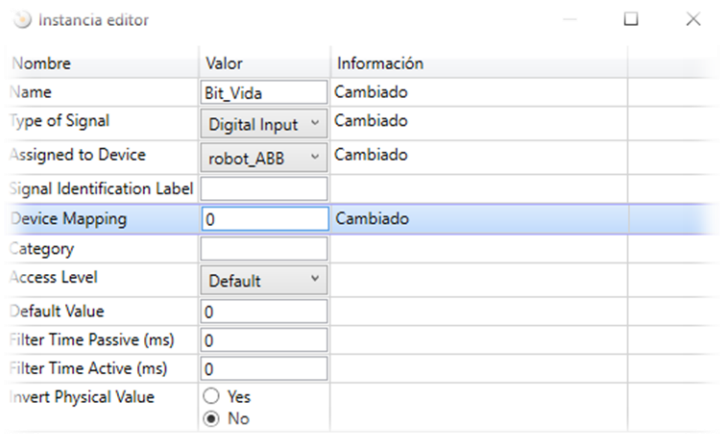

Figura 122 Declaración de señal digital ABB

Ahora creamos una señal analógica. Se hará de la misma manera que para las señales digitales, pero esta vez al ser una señal analógica de tipo Dword (Fig. 123) tendremos que reservar 4 bytes para ella, teniendo en cuenta los bytes reservados para las señales digitales.

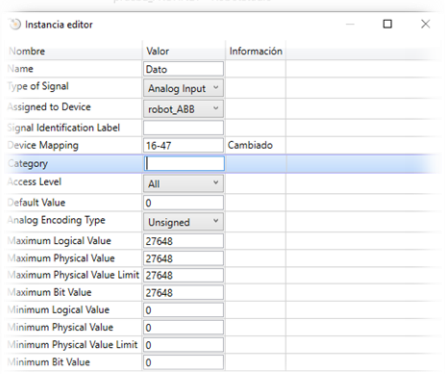

Figura 123 Declaración de señal analógica ABB

De esta manera iremos creando todas las señales necesarias para la comunicación y una vez finalizada la creacion de todas ellas será cuando reiniciemos el controlador (Fig. 124) para que todos estos cambios tengan efecto en el mismo.

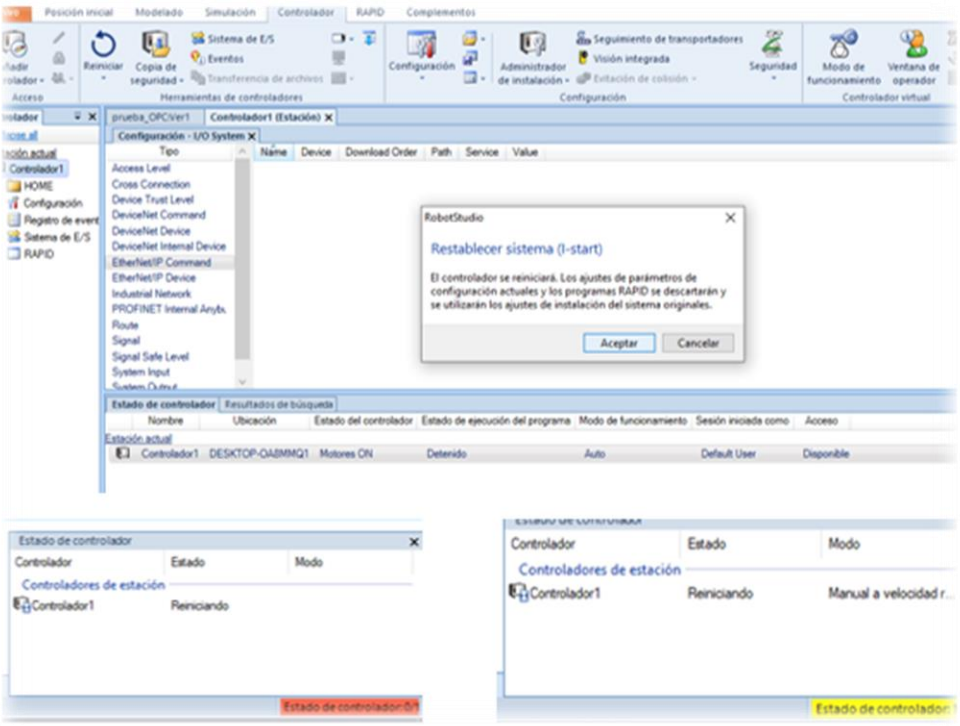

Figura 124 Reinicio controlador después crear señales

4.4.2.1.2 Crear señales desde pestana “Configurador E/S”

Otra forma de declarar las señales de comunicación es desde la pestaña “configuracion E/S” (Fig. 125). En el arbol que sale a la izquierda debemos seleccionar “PROFINET_Anybus”.

De esta forma podemos ver todos los datos correspondientes a la configuracion PROFINET establecida en el controlador y podremos añadir las señales necesarias de la misma manera que lo hemos realizado anteriormente.

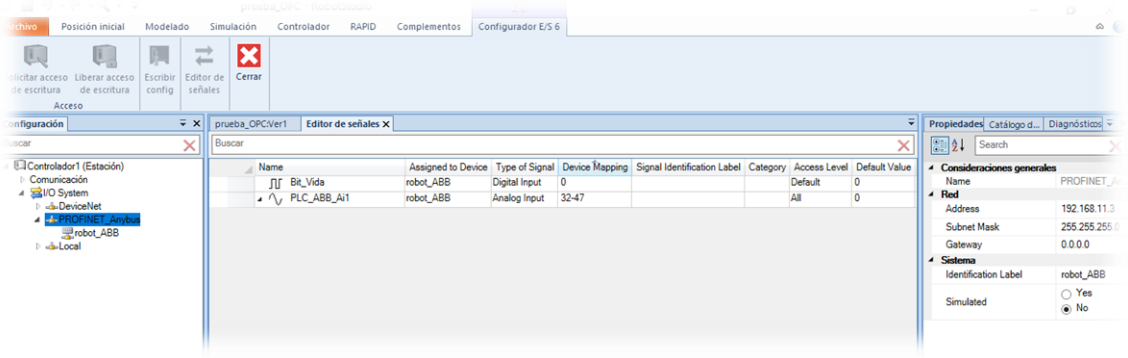

Figura 125 Declaración de señales desde E/S ABB

4.4.3 Transferencia bytes PLC-ABB

Una cosa a tener en cuenta es que al transferir los datos entre PLC y ABB los bytes no están en el mismo orden. Es por este motivo por lo que tendremos que invertir el orden de los bytes en uno de los sistemas. Por comodidad y facilidad se hará en el PLC (Fig. 126) mediante función SWAP. De esta manera ya estarán en el mismo orden en los 2 sistemas.

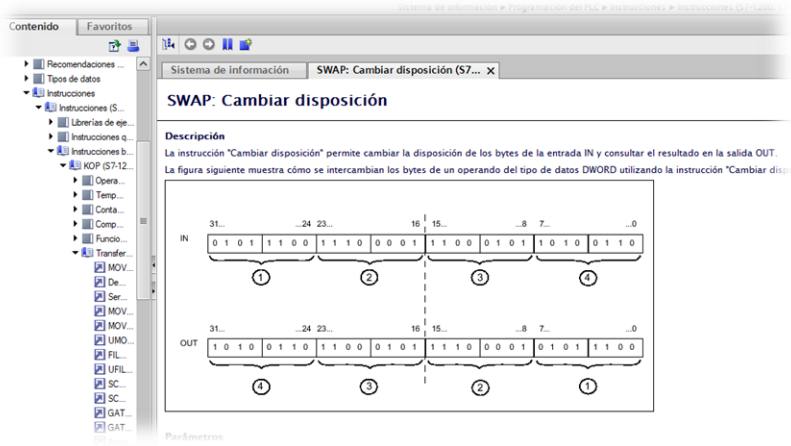

Figura 126 Función SWAP en TIA PORTAL

4.4.4 Señales intercambio en RAPID

Una vez que tenemos finalizada toda la configuración de PROFINET, declaradas las señales necesarias y reiniciado el controlador ya podremos hacer uso de las señales de intercambio a la hora de realizar la programación en RAPID. Las señales se incluirán durante la programación y control del programa RAPID.

Durante la programación RAPID (Fig. 127) podemos ver que es posible acceder a cualquiera de las señales declaradas.

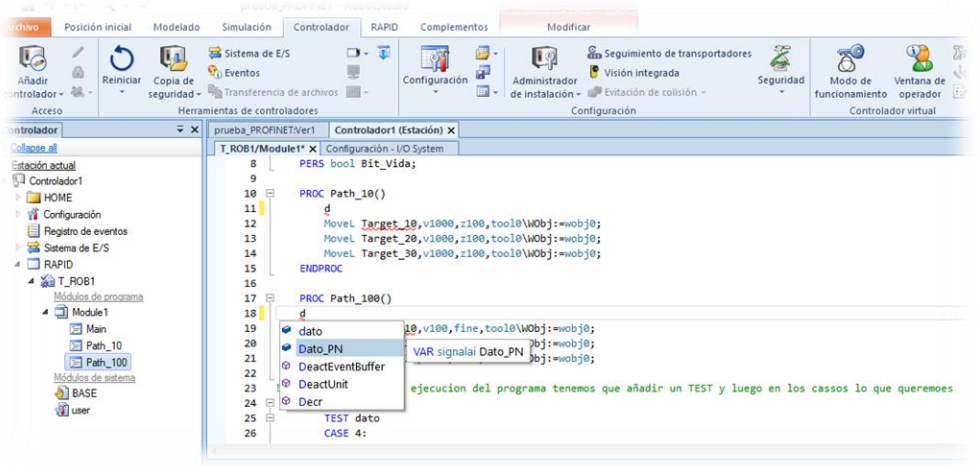

Figura 127 Disponibilidad de señales en RAPID

4.5 Desarrollo Aplicación ABB-PLC

Una vez realizadas las configuraciones necesarias para establecer la comunicación OPC entre el robot y el conjunto PLC-HMI, el siguiente paso realizado es el de la creación de un SCADA para conseguir la interactuación entre robot y operador; para ello usamos una pantalla táctil de 4” (SIEMENS KTP 400).

El objetivo es que el operario de la línea de producción pueda modificar ciertos parámetros de la configuración del robot para poder adaptarlo, si fuese necesario, al proceso productivo.

Para la programación del SCADA utilizamos el software TIA PORTAL. En el proyecto creado para la configuración OPC y PROFINET integramos ahora una pantalla táctil.

Una vez integrada la pantalla, comenzamos a diseñar el entorno gráfico sobre el que actuará el operario.

El diseño de la pantalla constará de un menú principal (Fig. 128) en el cual el operario podrá seleccionar uno de los tres procesos programados. En esta pantalla principal

también tendremos unos campos de texto que mostrarán la configuración actual enviada al robot y tendremos los botones de “Marcha” y “Paro” (Fig. 129) para controlar la tarea a realizar por el robot.

Figura 128 Menú principal SCADA

Figura 129 Menú principal con paro SCADA

Desde el menú principal, al seleccionar el proceso deseado, se abrirá la pantalla de configuración de las variables que se pueden modificar en el robot. Para la prueba se han puesto tanto la velocidad del TCP del robot como la aproximación a los puntos de la trayectoria (fig. 130).

Una vez seleccionados los valores deseados para enviarlos al robot, pulsaremos el botón de transferir. Una vez pulsado, se transfieren al programa RAPID los valores

seleccionados y será este el encargado de condicionar la ejecución de la tarea según los datos transferidos.

Una vez transferido tendremos que pulsar el botón de Marcha para que el robot inicie la tarea.

Figura 130 Variables modificables mediante SCADA

El paso de los datos entre PLC y robot se hace, como ya se comentó, asociando las variables declaradas tanto en PLC como en RAPID mediante el software LinkMaster (Fig. 131).

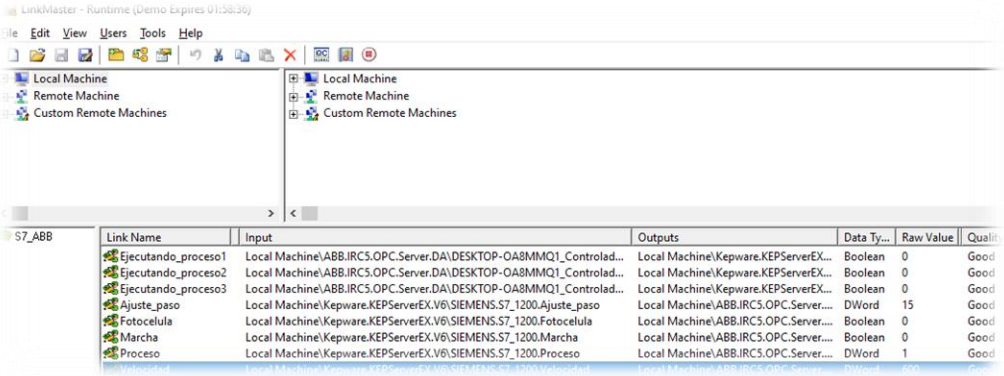

Figura 131 Configuración variables en LinkMaster

Una vez que tenemos configurado LinkMaster y la conexión está correctamente realizada, modificamos el código RAPID para condicionarlo según las señales recibidas del PLC.

Los primero que realizamos en RAPID es la declaración de esas variables (Fig. 132) de intercambio PLC-robot.

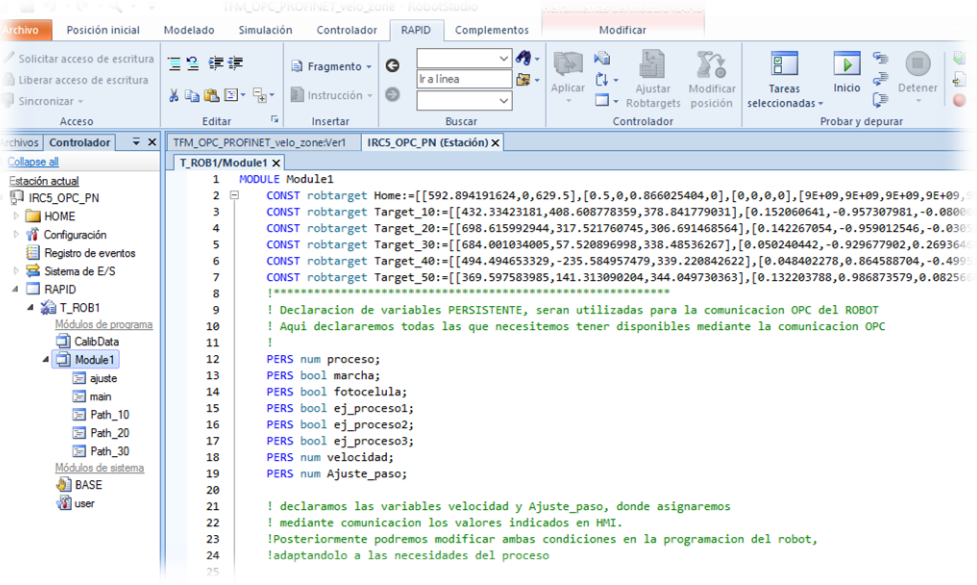

Figura 132 Declaración variables en RAPID

Las modificaciones realizadas en el código RAPID para condicionarlo a las señales recibidas son:

- En primer lugar, declaramos dos variables (Fig. 133) que las asociaremos a las del sistema “speeddata” y “zonedata” para posteriormente modificar los datos necesarios.

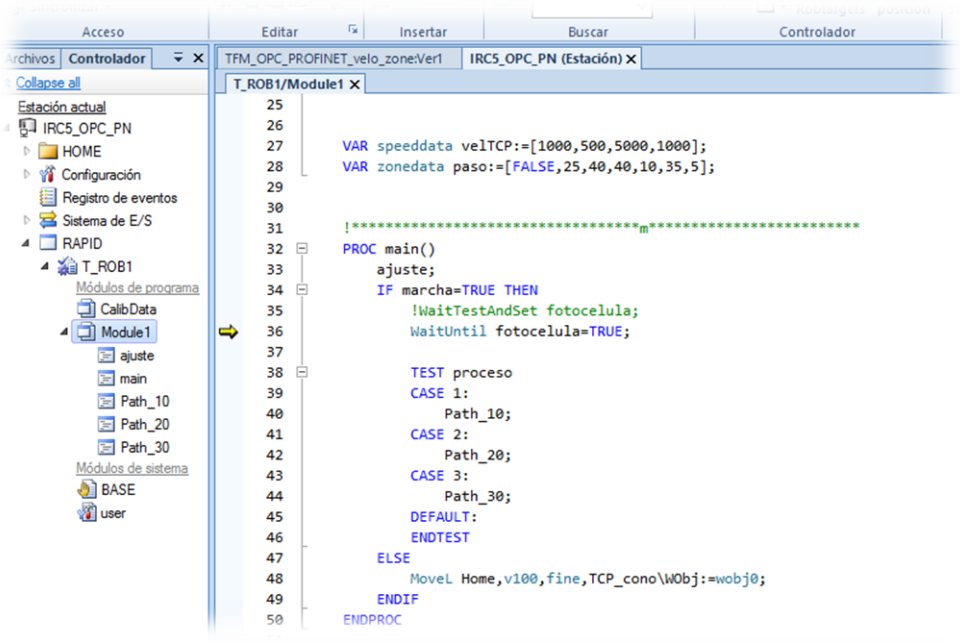

Figura 133 Declaración de variables asociadas a speeddata y zonedata

- En el siguiente paso creamos una rutina (Fig. 134) en la que definiremos la variable asociada a "zonedata" para adaptarla a los datos recibidos del HMI. En esta rutina asociamos el dato recibido del punto de aproximación al valor "pzone_tcp" que se aplicará al movimiento programado.

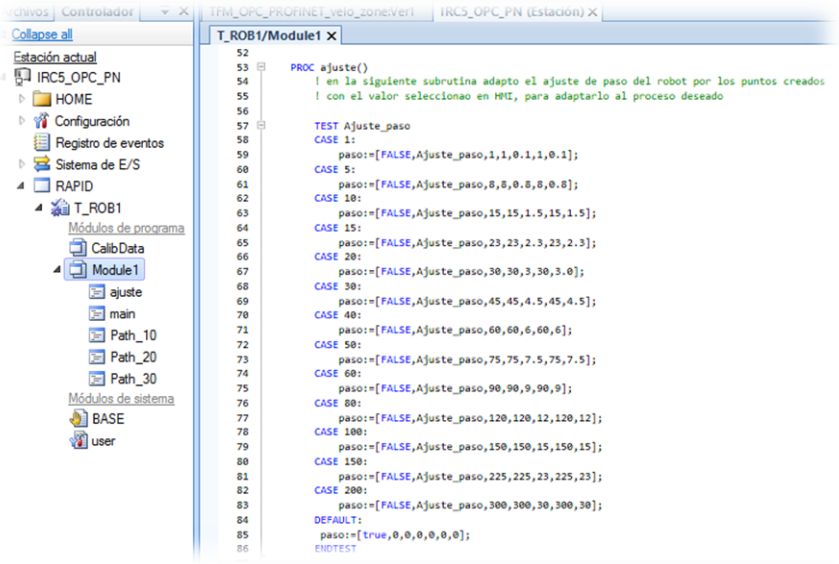

Figura 134 Rutina para asignar valor de HMI a "pzone_tcp"

- A continuación, condicionamos la ejecución de la rutina "Main" (Fig. 135) con las condiciones de las variables recibidas del PLC.
  En concreto condicionaremos la ejecución del código a que la señal de marcha esté activada; posteriormente el programa esperará a que la señal de la fotocélula

se active y por último, dependiendo del valor del proceso a ejecutar, se realizará la tarea programada.

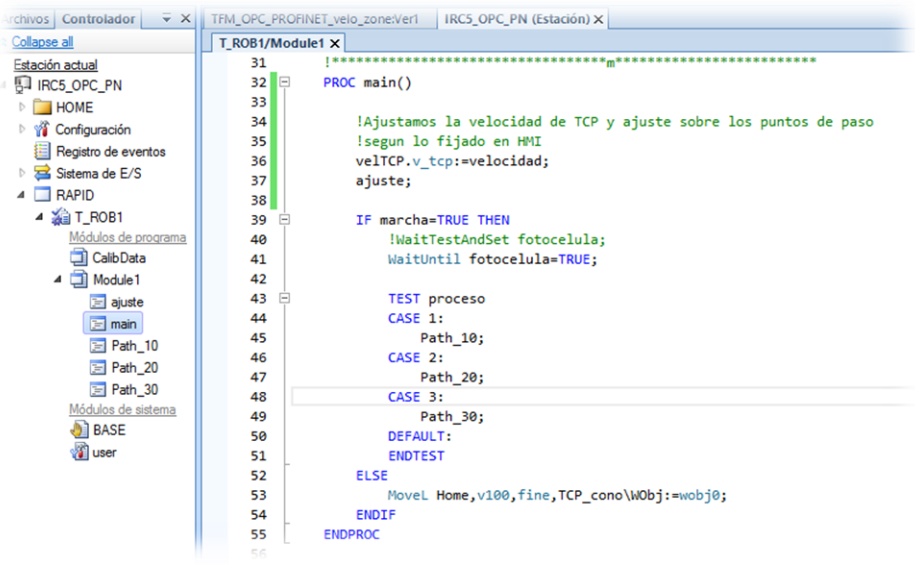

Figura 135 Rutina Main condicionada

- Por último, se programan las rutinas (Fig. 136) de las tareas a ejecutar. Para ello se modifican las líneas de código condicionándolas según los valores recibidos del PLC.

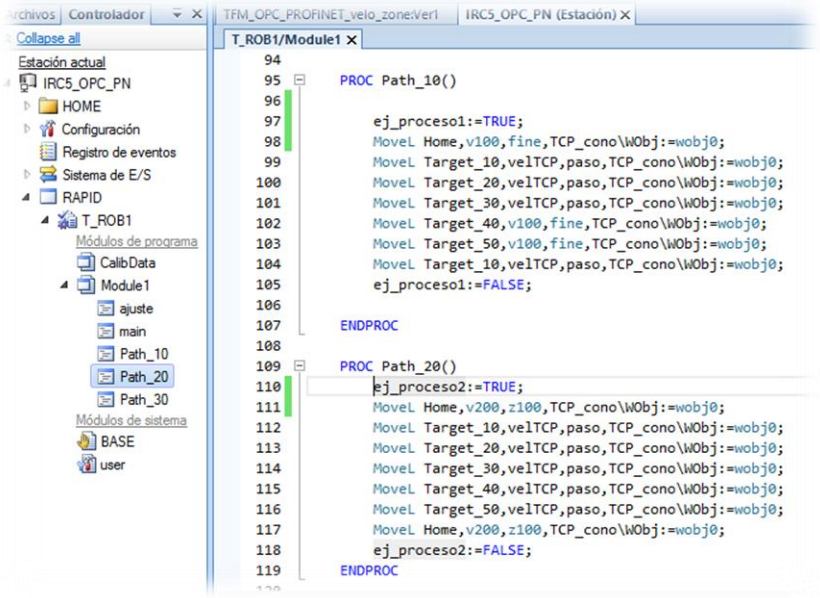

Figura 136 Rutina tareas condicionadas

5 Conclusiones y futuros proyectos

5.1 Conclusiones

La realización de este TFM tenía como objetivos principales:

- Adquirir el conocimiento necesario para crear, programar y simular una estación robotizada mediante el software de RobotStudio de ABB.
- Adquirir conocimientos necesarios para realizar una programación RAPID de la estación de trabajo creada anteriormente.
- Establecer una comunicación directa (OPC o PROFINET) entre la estación robótica y la línea de producción donde tiene que desarrollar su tarea.

Considero que se han cumplido los principales objetivos de este TFM, pero dado que tanto el software RobotStudio como la programación RAPID abarcan mucho campo y ofrecen muchas posibilidades tanto de diseño como de programación, no ha sido posible conocerlos en profundidad durante la realización de este TFM.

También se ha cumplido el objetivo de establecer la comunicación del robot con el entorno, permitiendo al ficticio operador de la línea de producción realizar modificaciones sobre el comportamiento de la tarea a realizar por el robot según las exigencias en las condiciones de la línea de producción. Esto deja una puerta abierta a mejorar los sistemas donde se aplica una estación robótica con este tipo de comunicaciones, pudiéndosele añadir multitud de dispositivos que permitan elevar la productividad y calidad de la línea de producción.

5.2 Líneas futuras de trabajo

Como principales líneas futuras de trabajo se podrían incluir la profundización en el diseño en RobotStuido y programación de lenguaje RAPID. Este sería muy interesante para conseguir, mediante la comunicación con los dispositivos de campo, una mejora y optimización de la tarea realizada por el robot.

Pero donde personalmente veo más posibles líneas de trabajo es en ampliar el abanico de dispositivos que pueden conectarse a la línea de producción. Estos dispositivos (sensores, sondas, Visión Artificial, etc.) podrían conectarse mediante cualquiera de las comunicaciones vistas y poder así mejorar la línea de producción y sobre todo ganar en calidad de producto final.

6 Bibliografía

[1] https://es.wikipedia.org/wiki/Rob%C3%B3tica

[2] http://laroboticaronal.blogspot.com/2008/04/

[4] https://www.youtube.com/watch?v=4Q3vfQKaCQI

[3] https://scielo.isciii.es/pdf/aue/v31n3/v31n3a02.pdf

[5] https://www.edsrobotics.com/blog/evolucion-robotica-industrial/

[6] https://robotsinaction.com/el-primer-robot-industrial-unimate/

[7] https://es.wikipedia.org/wiki/Unimate

[8] https://www.edsrobotics.com/blog/tipos-robots-industriales-usos/

[9] https://new.abb.com/products/robotics/es/robotstudio

[10] Manual del operador RobotStudio 2021.2 ID de documento: HAC032104-005 Rev. AH

[11] http://reea-blog.blogspot.com/p/robotstudio-abb.html

[12] Manual de referencia técnica Instrucciones, funciones y tipos de datos de RAPID

[13] ABB Flexible Automation AB Guía de Referencia RAPID On-line 3.0

[14] https://www.youtube.com/watch?v=_N_iYMBG0fM

[15] http://personal.biada.org/~jhorrillo/INTRODUCCIO%20RAPID.pdf

[16] https://opcfoundation.org/about/what-is-opc/

[17] https://www.kepserverexopc.com/que-es-opc-y-que-es-un-opc-server/

[18]https://es.wikipedia.org/wiki/OPC

Varios OPC

[19] Pérez, A. (2016). Sistema OPC para automatización mediante redes de estado [Tesis de maestría, Universidad de Sevilla]. e-REDING trabajos y proyectos fin de estudios de la E.T.S.I.

[20] https://es.scribd.com/document/76605188/REDES-DE-COMUNICACION-INDUSTRIAL-PROFINET

[21] https://www.conectronica.com/ethernet-industrial/profinet-la-revolucion-industrial-de-ethernet

[22] https://us.profinet.com/guia-de-profinet-para-principiantes/

[23] https://www.youtube.com/watch?v=dvHl6daktGY

[24] https://us.profinet.com/canales-de-comunicacion-profinet/

[25] https://cache.industry.siemens.com/dl/files/593/109741593/att_895707/v1/s71200_system_manual_es-ES_es-ES.pdf

[26] https://www.kepserverexopc.com/wp-content/uploads/2018/05/TNLK034KEP-Gui%CC%81a-de-comunicacio%CC%81n-remota-por-OPC-DA-con-KEPServerEX.pdf

[27] https://www.youtube.com/watch?v=j3cLFLBMW6Q7

[28] https://www.youtube.com/watch?v=pjqKbI3-g-c&list=PLucpG56BySyT1ePytDpPH7Sp91OTZxeGl&index=2

[29] https://www.kepware.com/getattachment/ee5f9114-2b1f-40b7-953e-d08832c3497d/linkmaster-manual.pdf

[30] https://abb.sluzba.cz/Pages/Public/IRC5RoboticsDocumentationRW6/Controllers/IRC5/FlexPendant/es/3HAC050941-005.pdf

7 Anexo

7.1 Código RAPID

```
MODULE Module1
    CONST robtarget
Home:=[[592.894191624,0,629.5],[0.5,0,0.866025404,0],[0,0,0,0],[9E+09,9E+09,9E+0
9,9E+09,9E+09,9E+09]];
    CONST robtarget
Target_10:=[[432.33423181,408.608778359,378.841779031],[0.152060641,-
0.957307981,-0.08006604,-0.232440145],[0,-1,-
2,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget
Target_20:=[[698.615992944,317.521760745,306.691468564],[0.142267054,-
0.959012546,-0.030522634,-0.243153018],[0,-1,-
2,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget
Target_30:=[[684.001034005,57.520896998,338.48536267],[0.050240442,-
0.929677902,0.269364667,-0.24620636],[0,-1,-
2,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget Target_40:=[[494.494653329,-
235.584957479,339.220842622],[0.048402278,0.864588704,-
0.499516275,0.025041616],[-1,-1,-2,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarget
Target_50:=[[369.597583985,141.313090204,344.049730363],[0.132203788,0.986873
579,0.08256608,0.042255656],[0,-1,-
2,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    !*****************************************************
    ! Declaración de variables PERSISTENTE, serán utilizadas para la comunicación
OPC del ROBOT
    ! Aquí declararemos todas las que necesitemos tener disponibles mediante la
comunicación OPC
    PERS num proceso;
    PERS bool marcha;
    PERS bool fotocelula;
    PERS bool ej_proceso1;
    PERS bool ej_proceso2;
    PERS bool ej_proceso3;
    PERS num velocidad;
    PERS num Ajuste_paso;

    ! declaramos las variables velocidad y Ajuste paso, donde asignaremos
    ! mediante comunicación los valores indicados en HMI.
    ! Posteriormente podremos modificar ambas condiciones en la programación del
robot, adaptándolo a las necesidades del proceso


    VAR speeddata velTCP:=[1000,500,5000,1000];
```

```
VAR zonedata paso:=[FALSE,25,40,40,10,35,5];

!***********************************************************
PROC main()
   ajuste;
   IF marcha=TRUE THEN

      WaitUntil fotocelula=TRUE;

      TEST proceso
      CASE 1:
         Path_10;
      CASE 2:
         Path_20;
      CASE 3:
         Path_30;
      DEFAULT:
      ENDTEST
   ELSE
      MoveL Home,v100,fine,TCP_cono\WObj:=wobj0;
   ENDIF
ENDPROC


PROC ajuste()
   ! en la siguiente subrutina adapto el ajuste de paso del robot por los puntos creados
   ! con el valor seleccionao en HMI, para adaptarlo al proceso deseado

   TEST Ajuste_paso
   CASE 1:
      paso:=[FALSE,Ajuste_paso,1,1,0.1,1,0.1];
   CASE 5:
      paso:=[FALSE,Ajuste_paso,8,8,0.8,8,0.8];
   CASE 10:
      paso:=[FALSE,Ajuste_paso,15,15,1.5,15,1.5];
   CASE 15:
      paso:=[FALSE,Ajuste_paso,23,23,2.3,23,2.3];
   CASE 20:
      paso:=[FALSE,Ajuste_paso,30,30,3,30,3.0];
   CASE 30:
      paso:=[FALSE,Ajuste_paso,45,45,4.5,45,4.5];
   CASE 40:
      paso:=[FALSE,Ajuste_paso,60,60,6,60,6];
   CASE 50:
      paso:=[FALSE,Ajuste_paso,75,75,7.5,75,7.5];
   CASE 60:
      paso:=[FALSE,Ajuste_paso,90,90,9,90,9];
   CASE 80:
      paso:=[FALSE,Ajuste_paso,120,120,12,120,12];
   CASE 100:
```

```
        paso:=[FALSE,Ajuste_paso,150,150,15,150,15];
    CASE 150:
        paso:=[FALSE,Ajuste_paso,225,225,23,225,23];
    CASE 200:
        paso:=[FALSE,Ajuste_paso,300,300,30,300,30];
    DEFAULT:
     paso:=[true,0,0,0,0,0,0];
    ENDTEST

ENDPROC

PROC Path_10()
    ej_proceso1:=TRUE;
    velTCP.v_tcp:=velocidad;

    MoveL Home,v100,fine,TCP_cono\WObj:=wobj0;
    MoveL Target_10,velTCP,paso,TCP_cono\WObj:=wobj0;
    MoveL Target_20,velTCP,paso,TCP_cono\WObj:=wobj0;
    MoveL Target_30,velTCP,paso,TCP_cono\WObj:=wobj0;
    MoveL Target_40,v100,fine,TCP_cono\WObj:=wobj0;
    MoveL Target_50,v100,fine,TCP_cono\WObj:=wobj0;
    MoveL Target_10,velTCP,paso,TCP_cono\WObj:=wobj0;
    ej_proceso1:=FALSE;

ENDPROC

PROC Path_20()
    ej_proceso2:=TRUE;
    MoveL Home,v200,z100,TCP_cono\WObj:=wobj0;
    MoveL Target_10,velTCP,paso,TCP_cono\WObj:=wobj0;
    MoveL Target_20,velTCP,paso,TCP_cono\WObj:=wobj0;
    MoveL Target_30,velTCP,paso,TCP_cono\WObj:=wobj0;
    MoveL Target_40,velTCP,paso,TCP_cono\WObj:=wobj0;
    MoveL Target_50,velTCP,paso,TCP_cono\WObj:=wobj0;
    MoveL Home,v200,z100,TCP_cono\WObj:=wobj0;
    ej_proceso2:=FALSE;
ENDPROC

PROC Path_30()
    ej_proceso3:=TRUE;
    MoveL Home,v100,fine,TCP_cono\WObj:=wobj0;
    MoveL Target_10,v100,fine,TCP_cono\WObj:=wobj0;
    MoveC Target_20,Target_30,velTCP,paso,TCP_cono\WObj:=wobj0;
    MoveL Target_40,v100,fine,TCP_cono\WObj:=wobj0;
    MoveL Target_50,v100,fine,TCP_cono\WObj:=wobj0;
    MoveL Target_10,v100,fine,TCP_cono\WObj:=wobj0;
    ej_proceso3:=FALSE;
ENDPROC

ENDMODULE
```

7.2 Hojas características

A continuación, se hace referencia a los DataSheet de los principales dispositivos del robot ABB, así como del PLC S7 1200 y la pantalla HMI.

7.2.1 IRC5

https://search.abb.com/library/Download.aspx?DocumentID=ROB0295EN&LanguageCode=en&DocumentPartId=&Action=Launch

7.2.2 IRB140

https://new.abb.com/products/robotics/es/robots-industriales/irb-140/datos

7.2.3 CPU S7-1200

https://media.automation24.com/datasheet/nl/6ES72121BE400XB0_en.pdf

7.2.4 KPT 900

https://media.automation24.com/datasheet/es/6AV21232JB030AX0.pdf

7.2.5 DSQC688

https://new.abb.com/products/es/3HAC031670-001/3hac031670-001