



Resumen del trabajo fin de grado:

El trabajo consiste en la realización de la simulación de una célula robotizada, que servirá para el tratamiento de una pieza de aluminio mediante tres máquinas que estarán abastecidas por un robot central; este último será el encargado de gestionar la entrada y salida de piezas en las máquinas, y de realizar el intercambio de piezas entre máquinas, así como el gestionar la entrada y salida de piezas de la célula robotizada.

La simulación se llevará a cabo con el programa RobotStudio, propiedad de ABB, que dispone de una amplia biblioteca de robots con la que poder trabajar.

Palabras clave:

- Simulación
- Célula robotizada
- Proceso industrial
- RobotStudio



Trabajo Fin de Grado

Autor: Gonzalo Muínelo Garrido

Grado en Ingeniería en
Electrónica Industrial y
Automática

Fecha: 2015





INDICE

1. Entorno del proyecto, objetivos y breve descripción.....	5
1.1 Entorno.....	5
1.2 Objetivo	5
1.3 Breve descripción y explicación de la célula robotizada.....	6
2. Robótica y limitaciones, RobotStudio y Robot ABB	9
2.1 Introducción a los paquetes de software de simulación robótica.....	9
Sistemas propietarios:.....	9
Sistemas no propietarios:	12
2.2 Robot utilizado en la célula robotiza. Características técnicas.....	14
2.3 Limitaciones a tener en cuenta en aplicaciones con robots.....	16
Volumen de trabajo	16
Grados de libertad de un robot	17
Capacidad de carga	18
Puntos singulares.....	18
2.4 Introducción a RobotStudio y manual de utilización	18
3. Elementos y máquinas de nuestra célula robotizada	21
3.1 Elementos que no afectan directamente al tratamiento de la pieza	21
Herramienta de doble pinza.....	21
Transportadores	21
Dispensador y recogedor de máscaras	23
Salida de control	24
Controlador	24
Barreras protectoras.....	25
Mesa del robot.....	26
3.2 Elementos que afectan directamente al tratamiento de la pieza	27
Maquina 1: Mecanizado	27
Maquina 2: Horno	28
Maquina 3: Tratamiento especial.....	29
4. Diferentes hipótesis en el diseño de nuestra célula robotizada	31
4.1 Hipótesis referentes a la herramienta	31
1ª hipótesis - Pinza única:	31
2ª hipótesis - Mesa de carga-descarga, y volteador en la herramienta.	32
3ª hipótesis: Herramienta de doble pinza con volteador.....	32
4.2 Hipótesis referentes a la distribución de máquinas y manutención.....	33
1ª hipótesis: Transportadores a cada lado del robot.	33
2ª hipótesis: Ambos transportadores al mismo lado.....	34
3ª hipótesis: transportadores al lado de la máquina 1 y de la máquina 2, e introducción de la salida para el control de piezas.....	35



4ª hipótesis: Robot elevado con transportadores por debajo y nueva situación de manutención de máscaras.....	36
5ª hipótesis: distribución definitiva.	37
5. Desarrollo y programación de la célula robotizada en RobotStudio	39
Configuración del transportador de entrada	40
Configuración del transportador de salida.....	43
Configuración inteligente del dispensador y recogedor.....	45
Configuración de la herramienta con doble pinza	50
Configuración de la maquina 1:.....	55
Configuración de la maquina 2: Horno	60
Configuración de la maquina 3: Tratamiento especial.....	74
Configuración de la salida de control:.....	78
6. Conexión y comunicación Robot-Elementos.....	83
7. Estudio del tiempo de ciclo	87
7.1 Consideraciones temporales para el estudio	87
7.2 Diagrama de Gantt.....	88
Primera iteración:	89
Segunda iteración:.....	90
Tercera iteración	90
Cuarta iteración:.....	91
Quinta iteración.....	91
8. Evaluación y análisis de riesgos reales de nuestra célula robotizada.	93
8.1 Aspectos a tener en cuenta en la seguridad	94
8.2 Análisis de riesgos.....	95
8.3 Identificación y estimación de los riesgos.	97
8.4 Acciones correctoras y reevaluación del riesgo	98
9. Conclusiones.....	101
Bibliografía	103
Anexos	105
ANEXO I - Programa Rapid	105
ANEXO II – Manual de RobotStudio elaborado por el autor del proyecto.....	105



1. Entorno del proyecto, objetivos y breve descripción

1.1 Entorno

El proyecto ha sido desarrollado en un entorno industrial, trabajando en el diseño de una célula robotizada y su simulación en la herramienta RobotStudio, software oficial de ABB que permite trabajar a través de simulaciones con los productos de ABB, así como luego exportar esas simulaciones a la realidad mediante la programación propia de los robots. Este trabajo a través de simulación permite, de una forma segura y fiable, diseñar, configurar y probar un sinfín de posibilidades y alternativas hasta encontrar la solución que se adapte de manera idónea a nuestras necesidades.

1.2 Objetivo

El objetivo del proyecto ha sido la aplicación de los conocimientos adquiridos durante la formación en el Grado de Ingeniería Industrial en Electrónica y Automática impartido en la Universidad de Valladolid, y la ampliación de estos conocimientos, concretamente en el mundo de la robótica de una manera más práctica y aplicada a la realidad industrial de hoy en día en el que la robótica está adquiriendo un protagonismo muy importante tanto en el mundo del automóvil como en todas las empresas de carácter industrial que lleven asociada un proceso productivo.

Otro de los objetivos de este proyecto es la utilización y el dominio de la herramienta RobotStudio, software de simulación propio de ABB, utilizándolo para desarrollar una simulación completa de una célula robotizada, para estudiar la viabilidad técnica a la hora de implantarla en un proceso productivo real.

En la simulación se deberán de situar todos los elementos de manera que estén perfectamente coordinados y ninguno limite o influya en el funcionamiento del resto de elementos y consecuentemente en el funcionamiento de la célula robotizada.

También como objetivo está marcado es el estudio, mediante la simulación, del tiempo de ciclo e intentar cumplir las exigencias, que se imponen desde un nivel superior, referente a cumplir el tiempo de ciclo



poniendo las medidas y recursos necesarios para cumplir este objetivo que es de vital importancia en todos los procesos productivos.

1.3 Breve descripción y explicación de la célula robotizada

El trabajo consiste en una simulación mediante RobotStudio de una célula robotizada cuya finalidad de la célula robotizada es que una pieza sea tratada pasando por tres procesos diferentes. El recorrido que seguirá la pieza será el siguiente:

- Pieza en bruto entra a la célula robotizada gracias a los transportadores.
- Pieza en bruto es cogida del transportador por la herramienta del extremo del robot e introducida en la primera máquina (reflejada en la figura 1) para realizar una tarea de mecanizado sobre la pieza.
- La pieza tratada por la primera máquina es recogida por el robot y transportada e introducida en la segunda máquina (reflejada en la figura 1) que es un horno que calentará la pieza para un posterior tratamiento en la tercera máquina.
- La pieza tratada por la segunda maquina es recogida por el robot y transportada hasta la tercera maquina (reflejada en la figura 1) ; por el camino, la herramienta del extremo del robot recogerá una máscara protectora (un elemento necesario que hay que introducir en la tercera maquina junto con la pieza que lleva el robot). El robot introducirá la pieza y la máscara protectora en la tercera maquina para realizar un tratamiento especial sobre la pieza.
- Una vez finalizado el tratamiento especial en la tercera máquina, el robot recogerá la máscara y la pieza ya trabajada; la pieza trabajada se dejará en el transportador de salida, mientras que la máscara la retendrá hasta que en el siguiente ciclo tenga que coger una nueva.
- Eventualmente se puede solicitar de una manera externa a la célula la salida de una de las piezas entre la tercera máquina y la salida final para realizar un control de las características de la pieza tratada

El resultado final de la simulación será el reflejado en la figura 1:

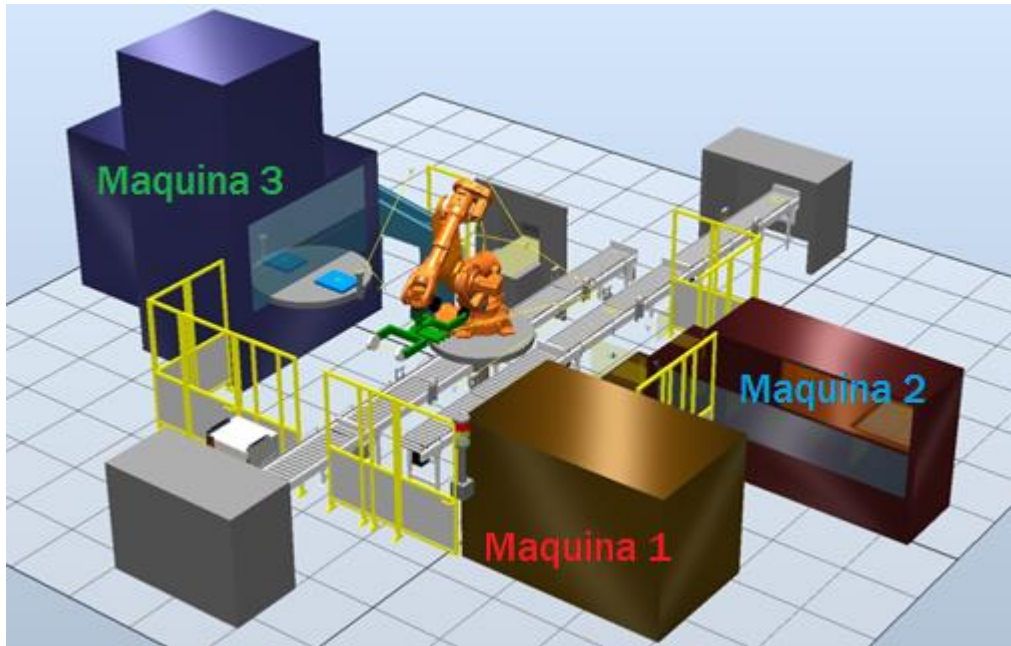


Figura 1



Trabajo Fin de Grado

Autor: Gonzalo Muínelo Garrido

Grado en Ingeniería en
Electrónica Industrial y
Automática

Fecha: 2015





2. Robótica y limitaciones, RobotStudio y Robot ABB

2.1 Introducción a los paquetes de software de simulación robótica

En la actualidad existen una gran cantidad de empresas dedicadas a la robótica industrial, lo cual indica el nivel de automatización en todo proceso productivo. Es por ello que la simulación cobra gran importancia a la hora de aportar soluciones previas a su implementación como detectar errores para una implementación más eficiente, o realizar pruebas o cambios sin parar la producción.

Es por ello que tanto las propias empresas de robots industriales, como empresas especialidad han desarrollado paquetes de software de simulación para sus propios sistemas, o de una manera más general en el caso de aquellos que periten implementar robots de distintas empresas. Es por ello que podríamos hacer una clasificación general atendiendo a los criterios anteriormente citados:

- Sistemas propietarios.
- Sistemas no propietarios.

Sistemas propietarios:

Son aquellos propios de las empresas de producción de robots industriales y que solo permiten trabajar con los robots de la empresa y marca del fabricante. Algunos ejemplos de sistemas propietarios son los siguientes:

- **INSER ROBÓTICA**

Empresa nacida en 1986 y la primera en cuanto a instalación de sistemas robotizados en España. Especializada en los sistemas robotizados encargados de soldadura, paletizado y carga y descarga. Es la distribuidora oficial en la península ibérica de Kawasaki y Panasonic.

Se puede ver el sistema representado como en la realidad (CAD), además de programa para observar el funcionamiento (CAM), y obtener la información necesaria y hacer valoraciones de planificación (CAE).

INSER ROBOTICA entrega el programa personalizado al cliente. Tienen dos herramientas dependiendo del fabricante de robot.



PC ROSET → Kawasaki. Reflejados en las figuras 2 y 3.

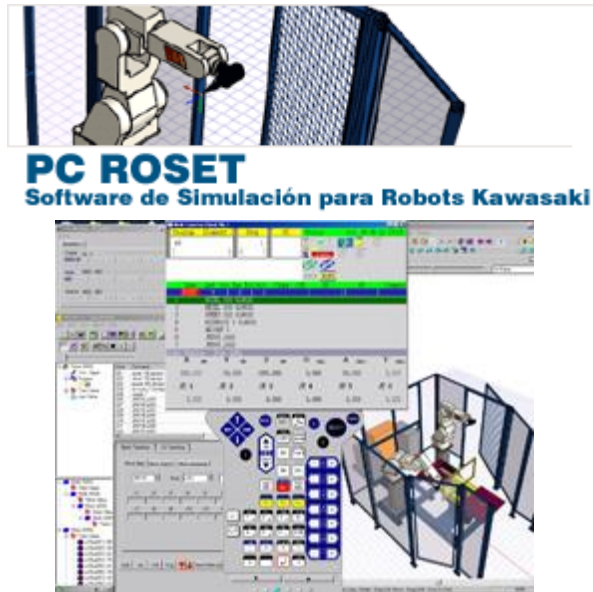


Figura 2

Figura 3

DTPS II → Panasonic. Reflejados en las figuras 4 y 5.



Figura 4

Figura 5

El inconveniente de estas dos herramientas es la poca flexibilidad ya que todo viene perfectamente preparado para el uso.

- KUKA



Empresa puntera y pionera en cuanto al mercado de robots industriales, siendo la primera empresa que introdujo en el mercado el control de un robot industrial basado en PC.

La plataforma actual de control basado en PC es KR C con ventajas como contar con interfaz en Windows, bus de campo, etc.

Su software de simulación es KUKA.Sim y está compuesto por 4 aplicaciones diferentes, dependiendo de las necesidades del usuario.

KUKA.Sim Pro

Es la aplicación de programación offline. Reflejado en la figura 6.



Figura 6

KUKA.Sim Layout

Es la aplicación de modelado de diseños 3D para las instalaciones en las que irá integrado el robot KUKA. Reflejado en la figura 7.

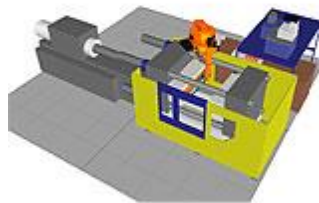


Figura 7

KUKA.Sim Viewer

Es la aplicación que permite la visualización de simulaciones desarrolladas con las dos aplicaciones anteriores. Reflejado en la figura 8.



Figura 8

KUKA.OfficeLite

Es la aplicación que permite crear programas para robots Kuka en cualquier PC. Gracias al uso de esta aplicación el manejo y la programación fuera de línea coincide exactamente con los del robot. Reflejado en la figura 9.

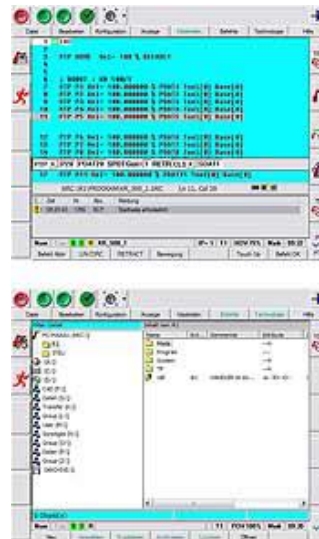


Figura 9

- **RobotStudio:**
Es el software que utilizaremos para el desarrollo de nuestra célula robotizada, y del que hablaremos más adelante.

Sistemas no propietarios:

- **Workspace 5**
Ofrece las ventajas de la simulación offline pero pudiendo usar cualquier robot de cualquiera fabricante. Dos de sus prioridades a la hora de prestaciones son facilidad de uso y facilidad de



aprendizaje. Este programa ofrece la posibilidad de crear trayectorias a partir de una superficie. La flexibilidad es el principal valor de este programa ya que es capaz de usar robots de diferentes marcas como ABB, Fanuc, Panasonic, Comau, etc, y además ofrece distintos lenguajes de programación en función del robot utilizado. Reflejado en la figura 10.

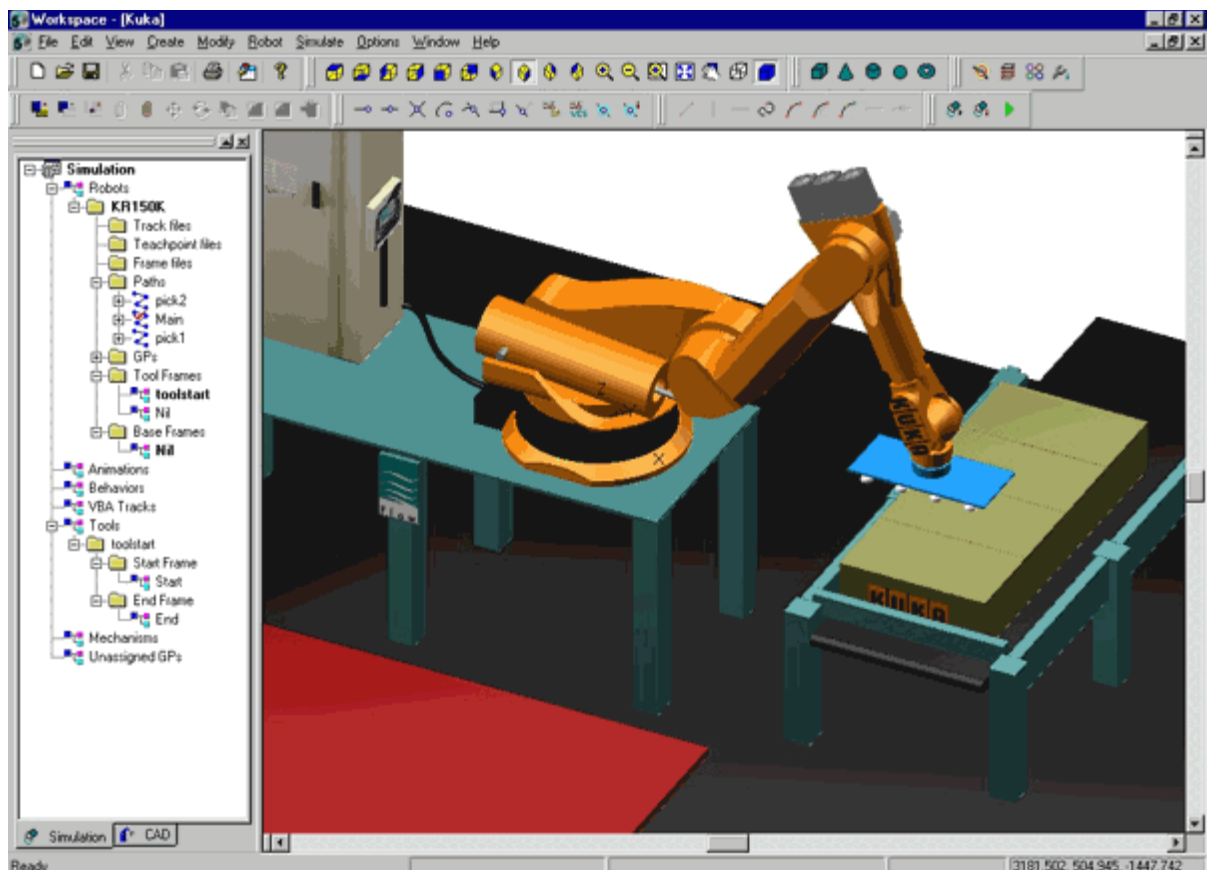


Figura 10

- **Roboworks:**
Desarrollado por Newtonium, es adecuado para el modelado 3D, gran colección de modelos de robots. Es un software muy intuitivo, con posibilidad de animación, capaz de añadir estos modelos a MatLab, MathCad y LabView. Su principal característica es que está dotado de un carácter muy generalista. Reflejado en la figura 11.

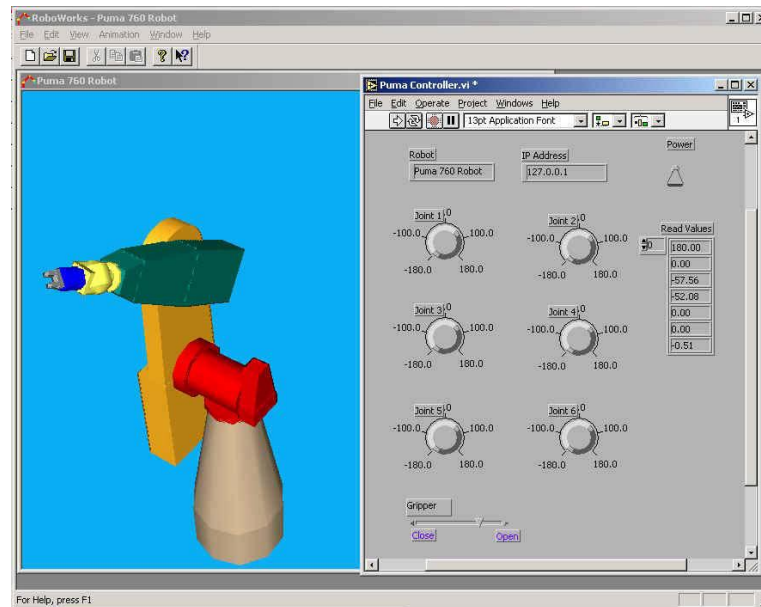


Figura 11

2.2 Robot utilizado en la célula robotiza. Características técnicas

El robot utilizado para nuestra célula robotizada será el robot IRB6640 con un alcance de 2.8 metros y una carga máxima de 185 kilos. La elección de este robot viene impuesta por la empresa, ya bien sea por la reutilización de un robot de este tipo existente en otra célula de la línea que se va a quedar obsoleta o ya bien sea por la disponibilidad del fabricante. Si se quisiese hacer la elección del robot habría que seguir las siguientes pautas reflejadas en la figura 12:

	Implantación	Pinza y/o pieza	Precisión del proceso	Tiempo de ciclo
Área de trabajo	X			
Carga		X		
Precisión			X	
Velocidad				X
Precio	?	?	?	?

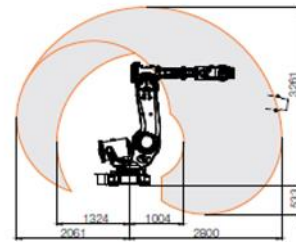
Figura 12

Las características que debemos conocer del robot que vamos a utilizar son las que podemos encontrar en la siguiente tabla.



IRB 6640 - 1.85/2.8

IRB 6640-185/2.8



Especificaciones		Conexiones eléctricas	
Centro de gravedad	300 mm	Voltaje de suministro	200-600 V 50/60 Hz
Momento de rotación de muñeca	1206 Nm		
Número de ejes	6	Consumo de electricidad	ISO-Cube 2.7 Kw
Protección	Complete IP 67		
Montaje	En suelo o plataforma		
Funcionamiento		Dimensiones	
Funcionamiento	Repetitividad de posiciones: 0.07mm	Dimensiones de la base	1107 x 720 mm
	Repetitividad de trayectorias: 0.7mm	Peso del robot	1310 - 1405 kg
Ejes	Rango de Ejes	Entorno (Medio ambiente)	
Eje 1 (Rotación)	+170° / -170°	Temperatura ambiente para la unidad mecánica	
Eje 2 (Brazo)	+85° / -65°	Durante operación	+5°C a +50°C (*)
Eje 3 (Brazo)	+70° / -180°	Durante transporte y almacenaje durante períodos cortos	-25°C a 55°C
Eje 4 (Muñeca)	300° / -300°	Humedad relativa	Máx 95%
Eje 5 (Curva)	120° / -120°	Nivel ruido	Máx 71 dB
Eje 6 (Vuelta)	360° / -360°	Seguridad	Doble circuito con supervisión, paradas de la emergencia y funciones de



			seguridad, 3 posiciones permite el dispositivo
--	--	--	--

(*)En una aplicación de prensa a alta velocidad, la temperatura ambiente de máximo es 40°C

2.3 Limitaciones a tener en cuenta en aplicaciones con robots.

Volumen de trabajo

Es importante conocer las limitaciones que tendrá nuestro robot, que vienen dadas por el rango de giro de cada uno de los ejes, ya que son un aspecto importante a la hora del posicionamiento de nuestro robot en cualquier tipo de aplicación, que aunque son mecanismos que ofrecen muchas posibilidades para llegar a un punto del espacio, no es posible llegar a todos los que deseáramos en algunos casos.

Para esto deberemos de tener en cuenta la volumen de trabajo del robot y el punto muerto. Llamamos volumen de trabajo del robot al espacio físico a los que nuestro robot puede acceder; en el caso de nuestro robot IRB 6640 -1.85 / 2.8 el volumen de trabajo sin herramienta (ya que una herramienta nos ampliaría esa envolvente) sería la reflejada en la figura 13:

IRB 6640-185/2.8

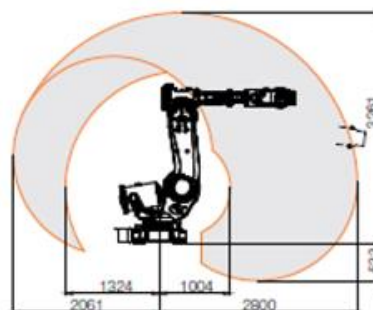


Figura 13

Ese volumen de trabajo es posible visualizarla en el RobotStudio lo cual nos facilita en gran medida conocer hasta donde es capaz de llegar nuestro robot de una manera muy visual, tal y como esta reflejado en la figura 14.

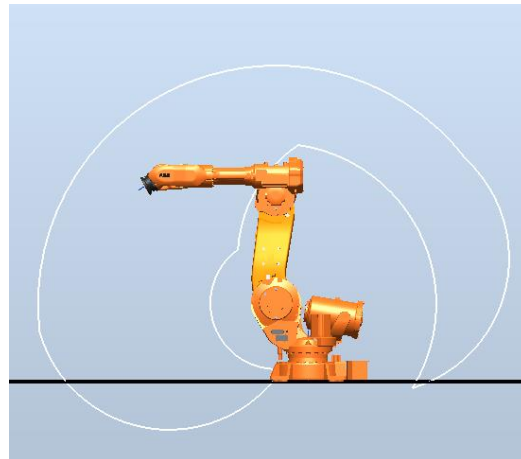


Figura 14

También debemos de tener en cuenta la limitación que nos supone el Eje 1 al no disponer de un giro completo de 360° sino que va desde +170° hasta -170°. Tenemos 20° a los que no podremos acceder con la mismo volumen de trabajo que tenemos en la parte delantera de nuestro robot. Todo esto reflejado en las figuras 15 y 16.

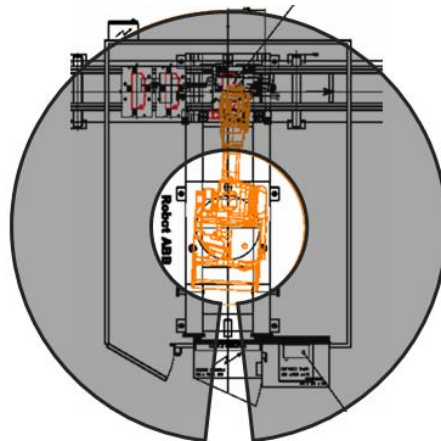


Figura 15

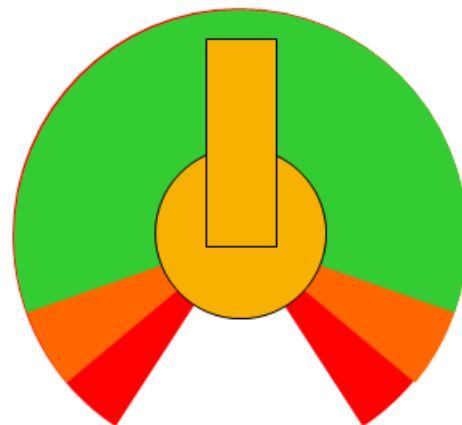
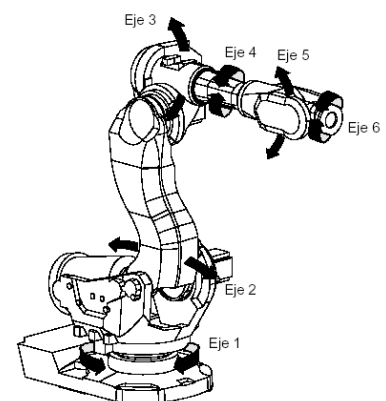


Figura 16

Grados de libertad de un robot

Se denomina grado de libertad a cada uno de los movimientos independientes que puede realizar una articulación de un robot respecto a la anterior.





Esto nos determinará la accesibilidad de un robot y su capacidad para orientar su herramienta. Esto suele coincidir con el número de articulaciones del robot. Se puede observar en la figura 17.

Figura 17

Capacidad de carga

Es la carga que es capaz de manipular el robot y depende del tamaño, la configuración y el sistema de accionamiento del robot.

Además hay que tener en cuenta el peso de la herramienta y el de la pieza a la vez. E incluso en determinados casos hay que tener cuenta los momentos de inercia. Se puede observar en la figura 18.

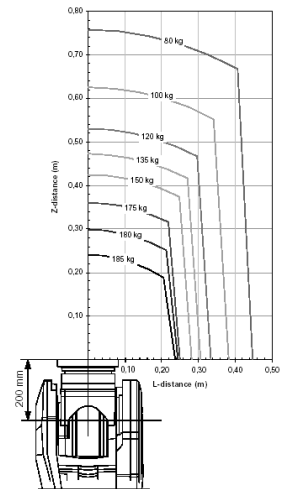


Figura 18

Puntos singulares

Por ultimo deberemos de tener en cuenta los llamados “puntos singulares”, que son puntos a los que, pese a estar dentro del volumen de trabajo, no es posible acceder ya que el algoritmo de trayectorias del robot no tiene solución (la explicación teórica sería que el determinante de la matriz jacobiana es 0, y por lo tanto no se puede determinar una configuración correcta de los ejes del robot). Estos puntos singulares normalmente se dan cuando se intenta realizar un movimiento lineal en una dirección en la que dos o más ejes están alineados.

2.4 Introducción a RobotStudio y manual de utilización

El software RobotStudio es propio de ABB y su utilidad es realizar simulaciones con productos ABB, en concreto con los robots propios de la marca. Las posibilidades que nos ofrece este software van desde lo más sencillo, como podría ser situar un robot y programar una serie de movimientos, hasta lo más complejo como podría ser simular un proceso productivo continuo dependiente de las entradas y salidas que le



llegasen a los elementos de la estación, pasando por el diseño y el modelado de componentes tanto para el robot como para la célula robotizada. Este software nos ofrece una interfaz bastante intuitiva para el desarrollo de aplicaciones sencillas como lo anteriormente hablado de situar un robot y programar unas trayectorias que debe de seguir, pero si se quisiese realizar una simulación más completa y visual requeriría unos conocimientos mayores del software. Para una correcta utilización del software se ha redactado un manual situado en el apartado anexos (Anexo II), que trata los siguientes puntos:

Introducir elementos en la estación (robot, equipamiento, geometrías, etc)

Mover y tipos de movimiento del robot

Introducir una herramienta y situarla en el robot

Crear un objeto de trabajo

Programar objetivo y crear trayectorias

Programación en RAPID

Comunicaciones I/O

Diseño de Hardware



Trabajo Fin de Grado

Autor: Gonzalo Muínelo Garrido

Grado en Ingeniería en
Electrónica Industrial y
Automática

Fecha: 2015





3. Elementos y máquinas de nuestra célula robotizada

Nuestra célula robotizada al constar de varios procesos que trataran una misma pieza tendrá 3 máquinas diferentes que deberán de ser abastecidas por el mismo robot, que además se abastecerá de la manutención entrante y abastecerá a la manutención de salida.

3.1 Elementos que no afectan directamente al tratamiento de la pieza

Herramienta de doble pinza

El extremo de nuestro robot cuenta con una pinza doble para coger y dejar las piezas y las máscaras en las diferentes posiciones para un correcto tratamiento dentro de las máquinas. Esta herramienta tiene en cada pinza un girador para voltear la pieza, ya que la posición de entrada en el horno la pieza debe de estar al revés de como se recoge en la maquina 1; también se dispone de otra doble pinza más pequeña para coger y dejar las máscaras necesarias para el tratamiento de la pieza en la máquina 3.

Esta herramienta ha sido diseñada también como un componente inteligente para dotarla de movimiento y que pueda coger las piezas. La geometría es la reflejada en la figura 19.

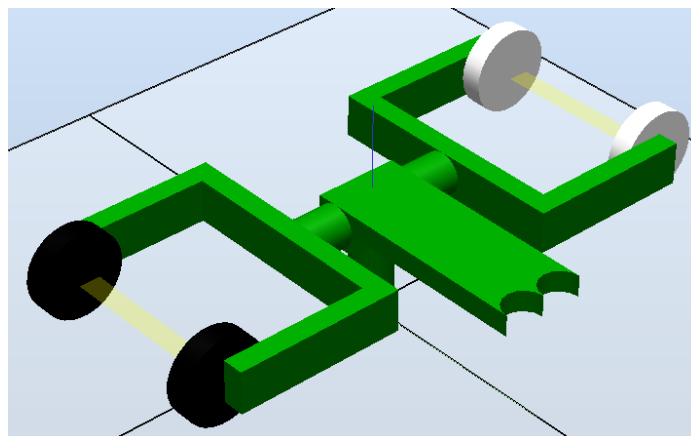


Figura 19

Transportadores

Estos serán los encargados de llevar las piezas desde el exterior de la célula robotizada hasta dentro de la misma, mediante unos rodillos



monitorizados que moverán los soportes en los que van apoyados los piezas hasta las posiciones estipuladas para la cogida y dejada de piezas a través del robot. Las piezas irán situadas de una manera concreta sobre los soportes que viene impuesta por el trabajo que realizarán el robot y las máquinas. Los transportadores en la simulación son los que están disponibles en la biblioteca equipamiento de RobotStudio.

La geometría es la reflejada en la figura 20.

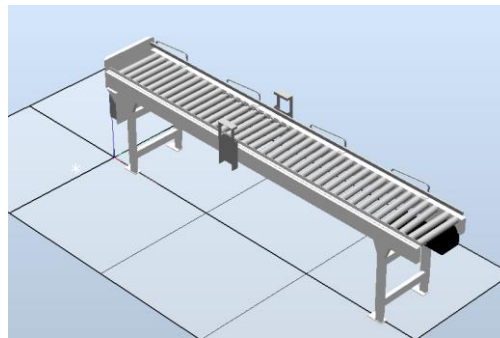


Figura 20

Estos transportadores, disponibles en la biblioteca de ABB, no disponen de ese movimiento de rodillos que hacen que se mueva el soporte donde van nuestras piezas en la realidad, por lo que se han creado dos “componentes inteligentes” con los que conseguimos crear una flujo de piezas.

El primero de ellos será la cola de entrada con un flujo que nos proporcionará una pieza cada cuatro segundos, siempre y cuando el sensor de posición 2 que indica que hay una pieza para recoger por el robot no esté activo, y sensor de posición 1 de la entrada esté activado para darnos la seguridad de que no entrarán dos piezas a la vez provocando una colisión entre ambas. Podemos observar los planos sensores en la figura 21.

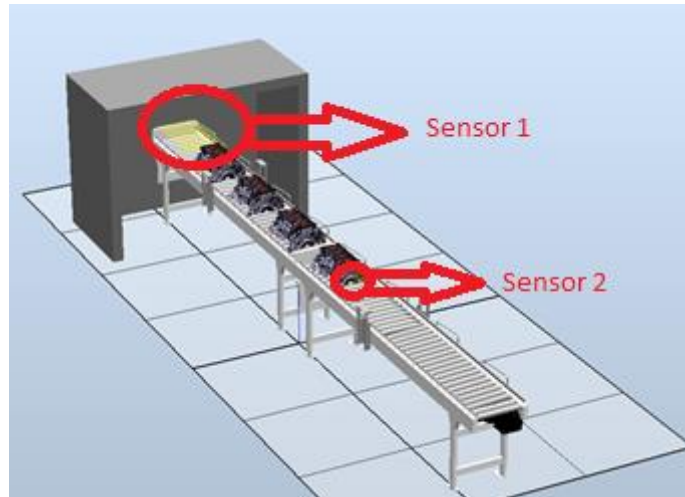


Figura 21

El segundo de ellos nos desplazará las piezas ya tratada por el transportador para salir de la célula robotizada. La geometría es la reflejada en la figura 22.

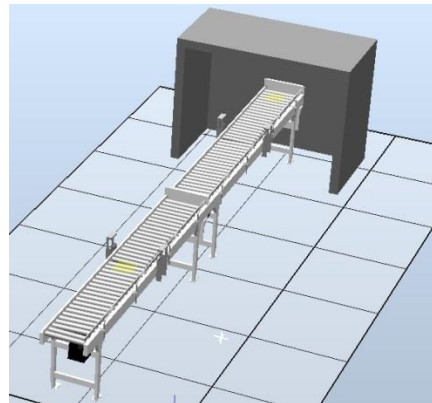


Figura 22

Dispensador y recogedor de máscaras

Este elemento consiste en cuatro carriles de rampas y un carro; dos de los carriles sirven para la entrada de máscaras limpias, y los otros dos son para la salida de las máscaras sucias ya usadas, que irán a parar hasta un carro de máscaras sucias. Las máscaras al ser cilíndricas, entrarán y saldrán rodando por las rampas. La geometría es la reflejada en la figura 23.

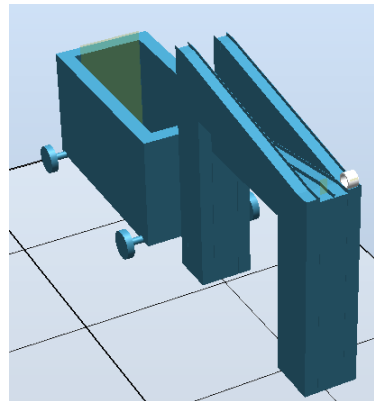


Figura 23

Salida de control

Utilizada para sacar piezas de la célula de las que deseamos hacer comprobaciones o medidas, y para ello tendrá una puerta por la que saldrán las piezas al exterior para su posible control. La geometría es la reflejada en la figura 24.

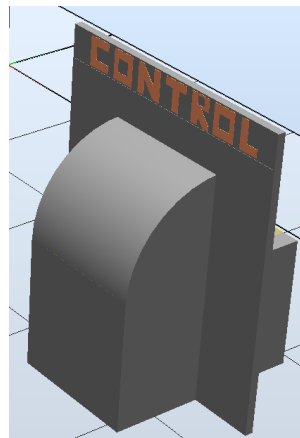


Figura 24

Controlador

Es un elemento que se sitúa fuera de los límites de la célula robotizada. La geometría es la reflejada en la figura 25.

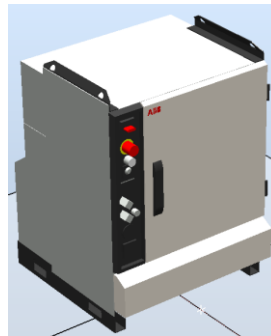


Figura 25

Barreras protectoras

El espacio en la que se encontrarán todas las entradas de las maquinas, el robot y los puestos de cogida y dejada, estará delimitado por unas barreras protectoras con dos puntos de acceso a la célula robotizada. Las barreras protectoras en la simulación son las que están disponibles en la biblioteca equipamiento de RobotStudio. La geometría es la reflejada en la figura 26.

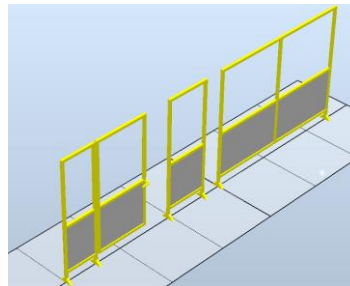


Figura 26

Las barreras protectoras delimitarán el espacio de la célula robotizada con las dimensiones reflejadas en la figura 27.

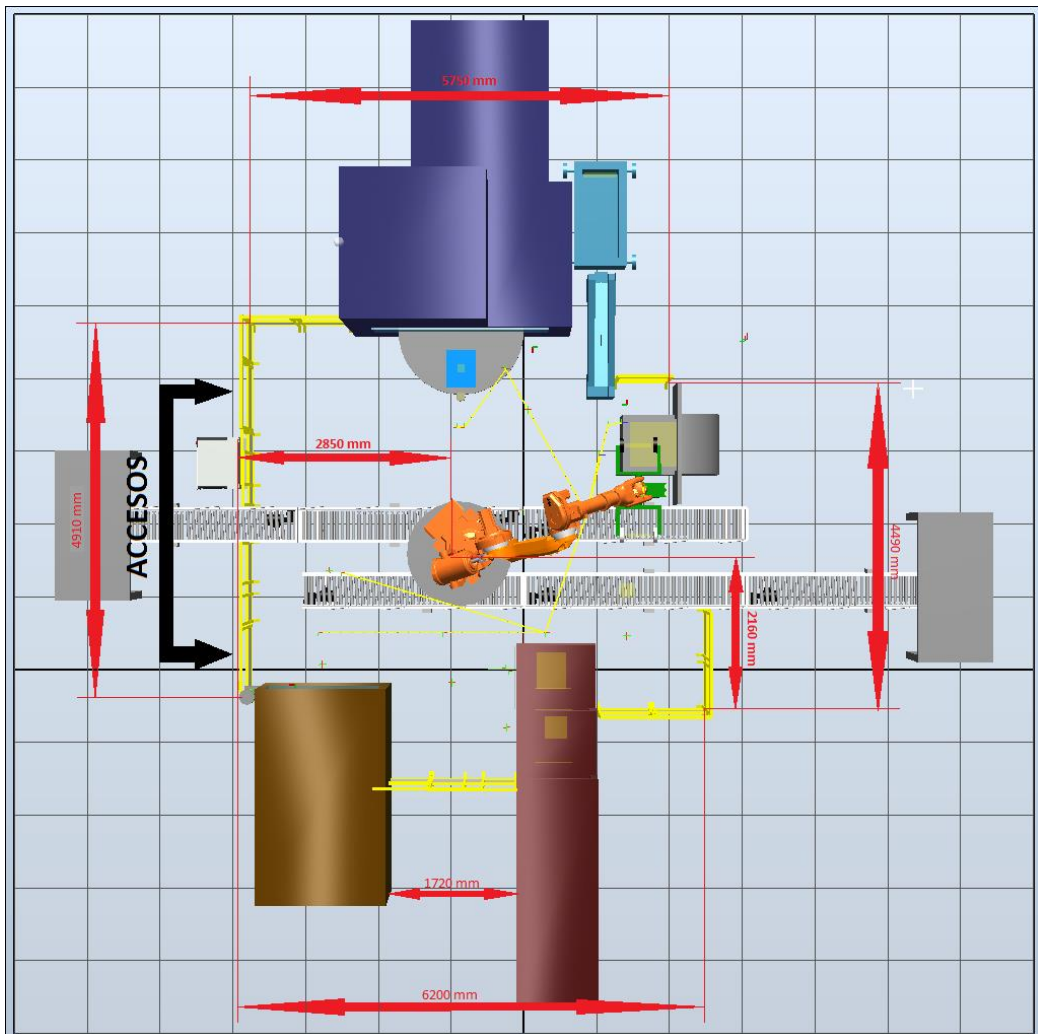


Figura 27

Mesa del robot

Es un elemento que aumenta la altura a la que está la base del robot hasta 1,3 metros para facilitar la tarea del robot con el fin de no colisionar con ningún otro elemento. La geometría es la reflejada en la figura 28.

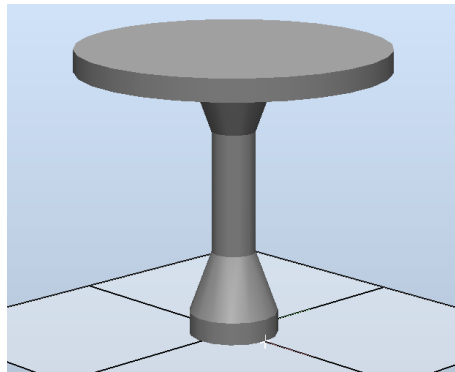


Figura 28

3.2 Elementos que afectan directamente al tratamiento de la pieza

Con elementos que afectan directamente al tratamiento de las máquinas nos referimos a todos aquellos que modificarán las características y propiedades de nuestra pieza, es decir nuestras tres máquinas que serán abastecidas por el robot.

Maquina 1: Mecanizado

Esta máquina se encarga de la primera operación que deben de pasar las piezas que entran en la célula robotizada. Consta de un único puesto de trabajo en el que se deberán de coger y dejar las piezas. La máquina tiene un tiempo de operación real de 130.2 segundos por pieza. Esta máquina tiene una puerta que se abre y se cierra en función de las señales digitales que recibe, y también tiene un aviso luminoso que se pondrá en verde si la puerta está abierta, en amarillo si la puerta se está cerrando o abriendo, y se pondrá en rojo si la puerta está cerrada. La geometría es la reflejada en la figura 29.

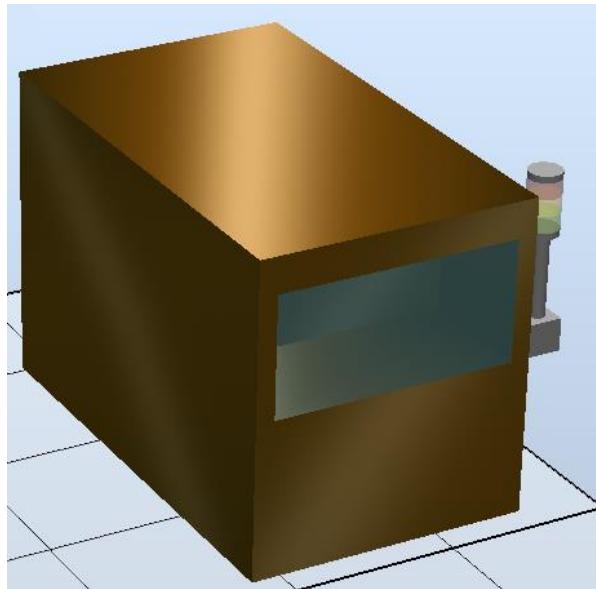


Figura 29

Maquina 2: Horno

La segunda máquina consiste en un horno que calentará la pieza hasta una temperatura determinada para un posterior tratamiento en la maquina 3. Esta segunda maquina tendrá una entrada por la parte inferior y tendrá una salida por la parte superior. Esta máquina nos permitirá que dentro de ella no solamente pueda haber una sola pieza calentándose, sino que puede tener varias piezas a la vez formando una cola de salida que el robot irá liberando. El horno, en la realidad, tardaría en calentar la pieza 138 segundos. La geometría es la reflejada en la figura 30.

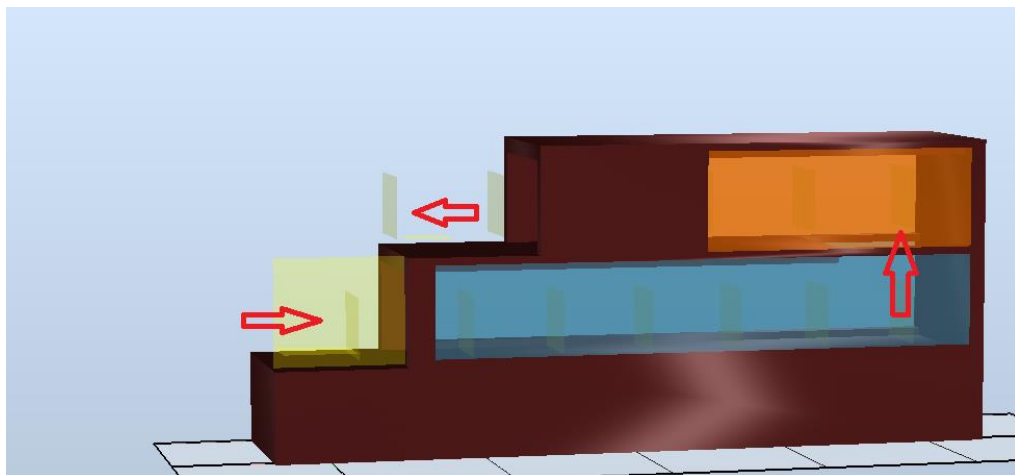


Figura 30



Maquina 3: Tratamiento especial

Esta máquina realizará una operación especial sobre la pieza para mejorar sus futuras prestaciones. En esta máquina se utilizarán las máscaras que se habrán cogido anteriormente, utilizándolas para proteger los agujeros de nuestra pieza en el tratamiento. La máquina dispone de una mesa giratoria en la que se depositará la pieza procedente del horno, y un soporte en la parte delantera en la que se dejará la máscara para que, una vez hecho esto, la puerta de la máquina se abra y gire la mesa, situando la pieza dentro, lista para ser tratada y dejando fuera o un hueco disponible para una nueva pieza, o bien una pieza ya tratada con su correspondiente máscara sucia. El tiempo de operación de esta máquina en la realidad es de 144 segundos. La geometría es la reflejada en la figura 31.

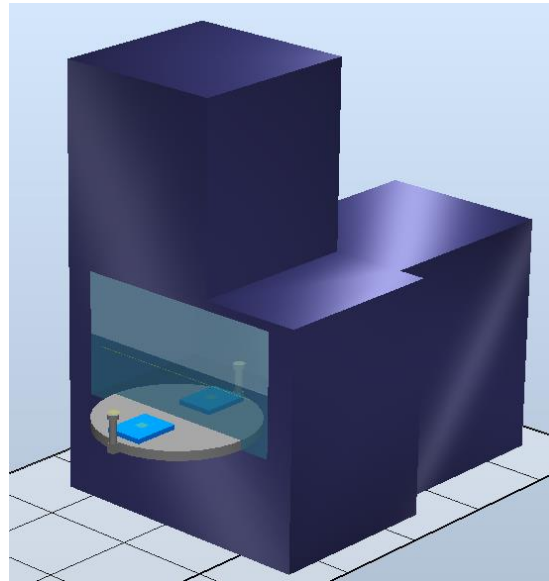


Figura 31



La distribución final de máquinas será la mostrada en la figura 31.1:

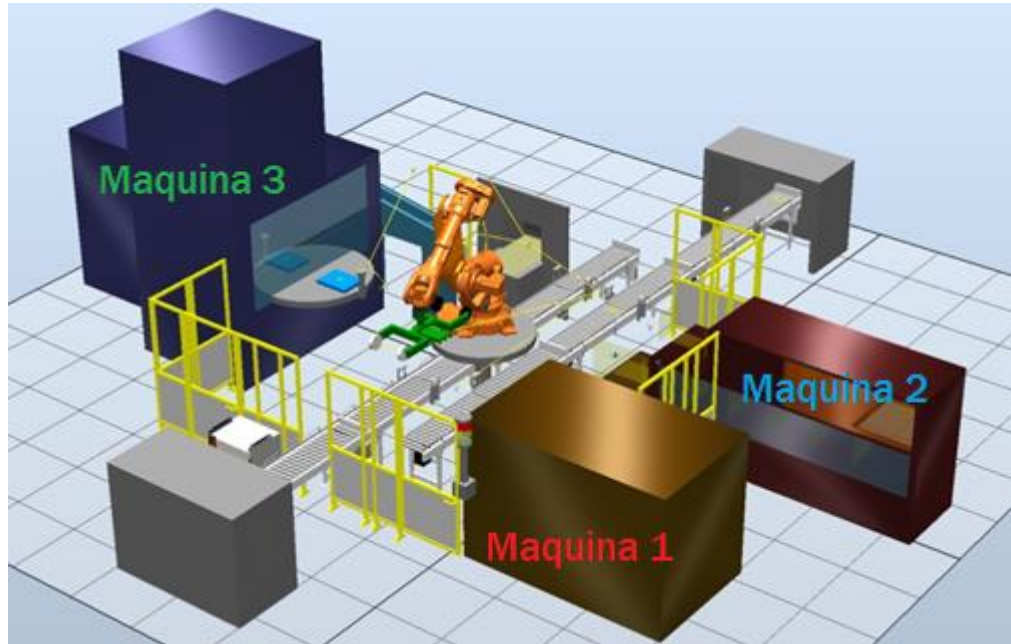


Figura 31.1



4. Diferentes hipótesis en el diseño de nuestra célula robotizada.

Una vez tenemos la idea inicial de la función que queremos que tenga nuestra célula robotizada y sabemos que habrá 3 máquinas que tratarán la pieza, debemos de plantearnos diferentes hipótesis de diseño hasta intentar llegar a la solución que creamos que es más óptima para nuestra célula robotizada.

4.1 Hipótesis referentes a la herramienta

1ª hipótesis - Pinza única:

Trabajar con una herramienta que disponga de una pinza en el extremo con la única función de coger y dejar la pieza en las máquinas o mantenimientos pertinentes.

Ventajas:

Puede ser una herramienta estandarizada y ya conocida por proveedores de herramientas ya que la única función que debe de cumplir es la de coger y dejar las piezas.

Al ser ya conocida y que exista la posibilidad de que esté estandarizada, su coste puede ser el menos elevado.

Fácil acceso de la herramienta a las máquinas ya que será la opción más pequeña en cuanto a herramienta se refiere.

Desventajas:

Al tener una única pinza el intercambio entre pieza trabajada y pieza en bruto o trabajada por la máquina anterior, no es inmediato. Habría que esperar siempre a que la última máquina terminase, liberarla, recoger la pieza de la máquina anterior, y cargar la máquina. En esta operación de carga y descarga se pierde demasiado tiempo.

Es posible que algunas máquinas exijan entrar volteadas en algunas máquinas, y con esta pinza no nos es posible sin que el robot tenga que tener configuraciones de ejes muy complejas y forzadas.



2ª hipótesis - Mesa de carga-descarga, y volteador en la herramienta.

Respondiendo e intentando dar soluciones a las desventajas de la hipótesis anterior, se plantea la posibilidad de situar una mesa delante de cada máquina para realizar el intercambio de piezas y un volteador en la propia herramienta que gire la pieza.

Ventajas:

Se reduce el tiempo de carga y descarga, ya que el tener una mesa nos permite situar la pieza que queremos meter en la máquina cerca de la entrada de esta y nos acortamos tiempos de carga y descarga.

La posibilidad de voltear las piezas en la propia pinza nos facilita la tarea de introducir las piezas con una orientación definida en cada máquina.

Desventaja:

El tiempo de carga y descarga seguiría siendo elevado ya que el intercambio sigue sin ser "inmediato" y el robot tendrá que hacer demasiados movimientos para desempeñar la tarea que queremos.

3ª hipótesis: Herramienta de doble pinza con volteador.

Respondiendo e intentando dar soluciones a las desventajas de la hipótesis anterior, se plantea la posibilidad de tener una herramienta de doble pinza que sea capaz de cargar y descargar la máquina, y que además cuente con un volteador de pieza en cada pinza.

Ventaja:

La principal ventaja es que se gana mucho tiempo en la carga y descarga, lo que nos facilitará alcanzar el objetivo del tiempo de ciclo.

La posibilidad de voltear las piezas en la propia pinza nos facilita la tarea de introducir las piezas con una orientación definida en cada máquina.

Desventajas:



El tamaño de la herramienta sería considerablemente más grande y podríamos colisionar más fácilmente con máquinas, transportadores e incluso el propio robot a la hora de hacer giros.

Entre las 3 hipótesis, vemos que la más interesante y la que nos permite cumplir con un objetivo tan importante como es el tiempo de ciclo es la hipótesis de la herramienta de doble pinza con volteador, con lo cual, nuestra célula robotizada contará con este tipo de herramienta en el extremo del robot. La geometría es la reflejada en la figura 32.

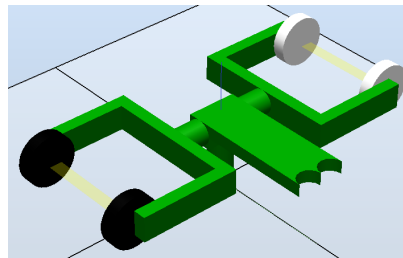


Figura 32

4.2 Hipótesis referentes a la distribución de máquinas y manutención.

1ª hipótesis: Transportadores a cada lado del robot.

Con unas dimensiones aproximadas de las maquina dos y la máquina tres dimensionadas ya, podemos ir haciendo pruebas de la distribución, a falta de saber la dimensión del horno. Decidimos considerar la primera hipótesis poniendo el robot en medio de los dos transportadores, la maquina tres a un lado y las otras dos máquinas al otro. La geometría es la reflejada en la figura 33.

Ventajas:

Conseguimos acceder a todas las maquinas sin colisionar

Desventajas:

Los puestos de dejada de piezas y cogida de piezas están muy alejados y del fin de un ciclo al principio de otro tardamos demasiado.

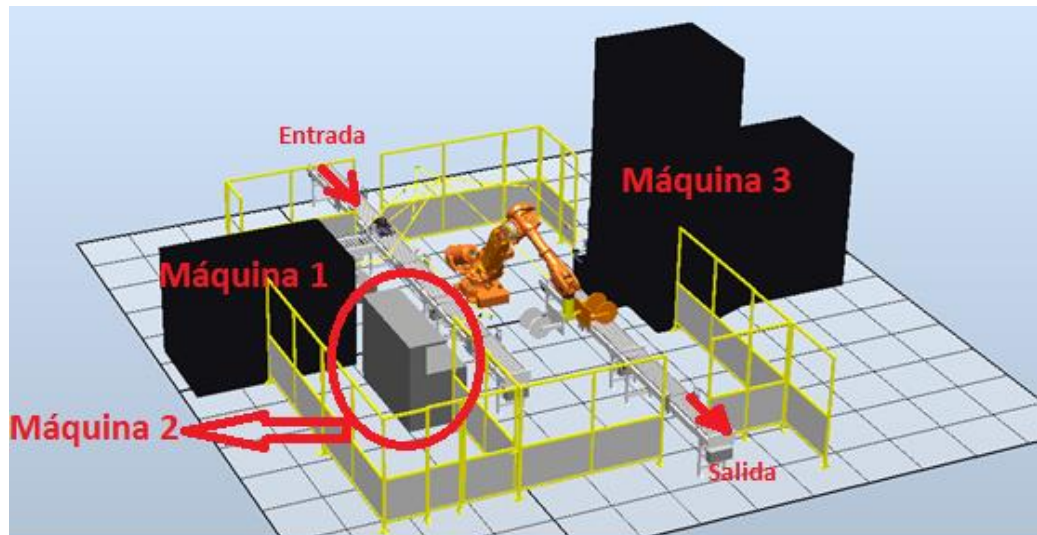


Figura 33

2ª hipótesis: Ambos transportadores al mismo lado

Los transportadores están en el mismo lado que la máquina 1 y la máquina 2, modificamos los puntos de dejada de piezas, y disminuimos el tamaño de la máquina 2 en anchura para hacerlo más real. La geometría es la reflejada en la figura 34.

Ventajas:

- Podemos llegar a todos los sitios con el robot y la herramienta.
- Puestos de cogida y dejada más cercanos.
- Podríamos acerca aún más el LDS y reducimos el espacio de la célula, lo cual optimizaría el espacio dentro de la fábrica

Desventajas:

- Si tenemos una herramienta de doble pinza, cuando dejamos una pieza en uno de los transportadores, podríamos encontrar otra pieza en el otro transportador y colisionar con ella.

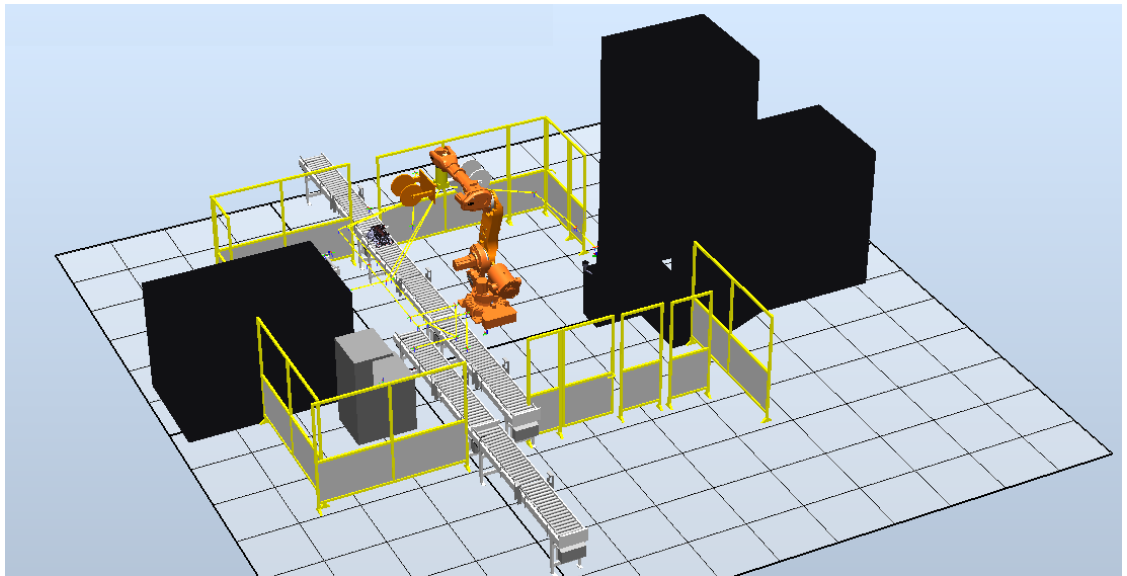


Figura 34

3ª hipótesis: transportadores al lado de la máquina 1 y de la máquina 2, e introducción de la salida para el control de piezas.

Es necesario situar una salida de control para examinar a petición del operario las piezas salientes de la máquina 3 (una nueva exigencia de un nivel superior)

Se han desplazado las máquina 1 y 2. También desplazamos la máquina 3 para dar espacio a la salida de control, y dejamos espacio para la manutención de máscaras que tenemos que colocar encima de la pieza. La geometría es la reflejada en la figura 35.

Ventajas:

Los puestos de cogida y dejada están situados relativamente cerca.

El puesto de cogida de piezas entrantes está cerca de la primera operación (máquina 1).

Desventajas:

No optimizamos las trayectorias ya que de esta disposición hacemos las operaciones 1 y 2 y tenemos que cambiar el sentido de la trayectoria lo que no nos permite optimizar del todo las trayectorias.

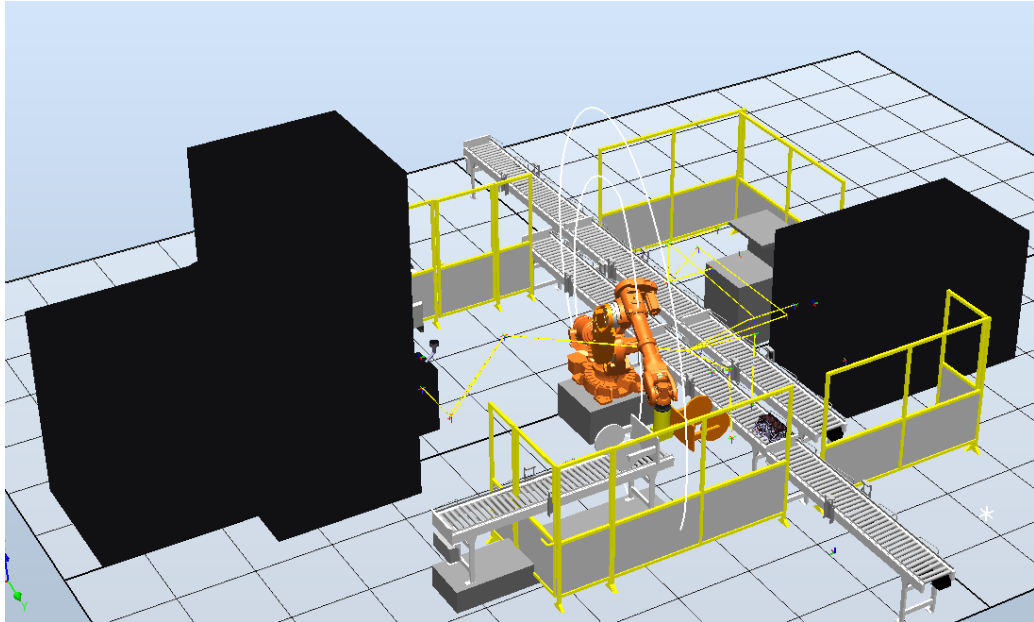


Figura 35

4ª hipótesis: Robot elevado con transportadores por debajo y nueva situación de manutención de máscaras.

Se ha decidido elevar el robot de altura y situar en uno de los laterales de la maquina 3 la manutención de máscaras. La geometría es la reflejada en la figura 36.

Ventajas:

- Es posible llegar con el robot a todos los sitios.
- Las máscaras están cerca de la maquina 3.
- El ciclo está en un solo sentido: Maquina 1 → Maquina 2 → Máscaras → Maquina 3

Desventajas:

- El puesto de control está alejado y se podría encontrar una mejor ubicación.

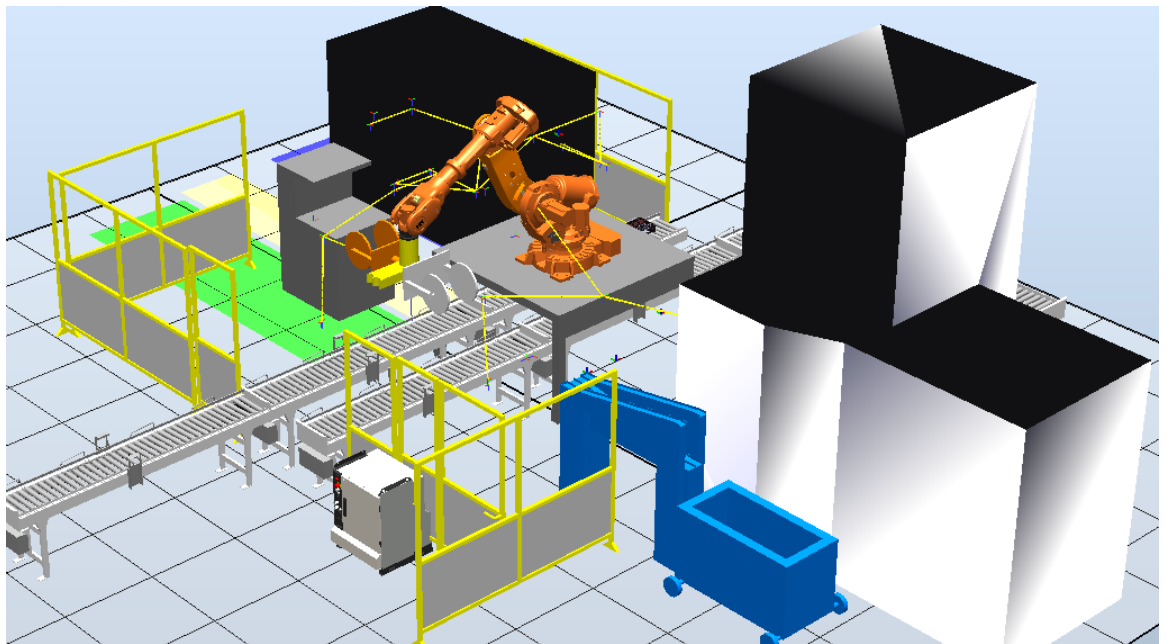


Figura 36

5ª hipótesis: distribución definitiva.

Se sitúa el control en el mismo lado que la manutención de máquinas. La geometría es la reflejada en la figura 37.

Ventajas:

Se consigue optimizar las trayectorias del robot.

El puesto de cogida está cerca de la máquina 1.

El puesto de dejada, expresado en un término coloquial, “nos pilla de paso” a la vuelta de haber realizado todas las operaciones.

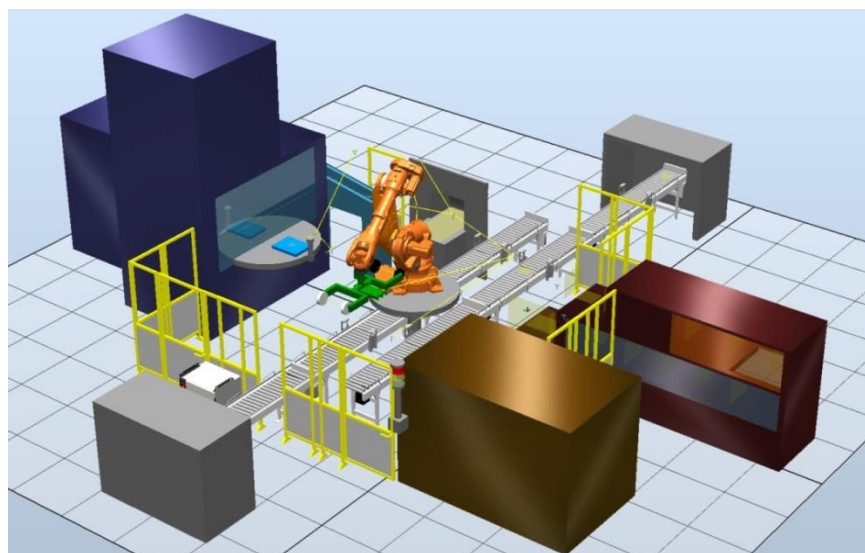


Figura 37



Trabajo Fin de Grado

Autor: Gonzalo Muínelo Garrido

Grado en Ingeniería en
Electrónica Industrial y
Automática

Fecha: 2015





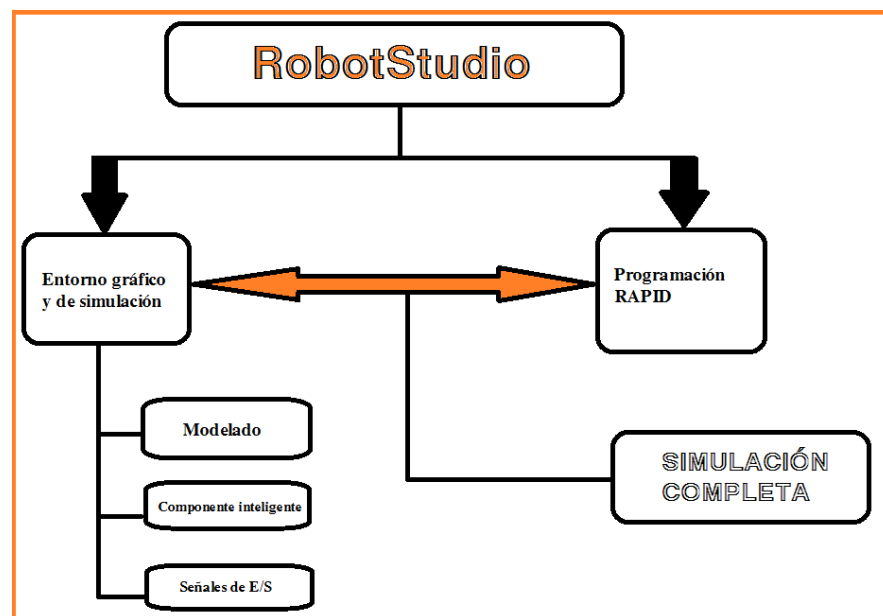
5. Desarrollo y programación de la célula robotizada en RobotStudio

A la hora de desarrollar y programar una célula robotizada en RobotStudio es necesario distinguir dos entornos que van ligados:

Entorno visual y de modelado de RobotStudio.

Entorno de programación RAPID.

En el siguiente diagrama queda reflejado el entorno de RobotStudio.



La interrelación de estos dos entornos nos proporciona como resultado una simulación completa y real.

La parte correspondiente a la programación RAPID se encuentra en el Anexo II.

Los componentes inteligente consisten en elementos que son capaces de activar y desactivar señales digitales, mover elementos, cambiarlos de color, eliminarlos, etc. Todo esto se hace mediante distintos bloques (de sensor, manipuladores, de acciones, paramétricos...). Dado la infinidad de posibilidades que ofrece RobotStudio con este tipo de componentes, existe la posibilidad de configurar cada uno de una forma totalmente distinta, con lo cual la manera de que realicen cierta función puede ser diferente si lo configura una persona u otra. En este apartado se va a explicar cómo están



creados los de nuestra estación, aunque existe una explicación más detallada de los componentes inteligentes en el manual anexo de RobotStudio que se ha elaborado para una mejor comprensión de la herramienta.

Cabe destacar que a la hora de explicar cada componente se va a denominar bloque a un conjunto de bloques que estén relacionados entre sí para desarrollar una función final.

Configuración del transportador de entrada

La geometría es la reflejada en la figura 38.

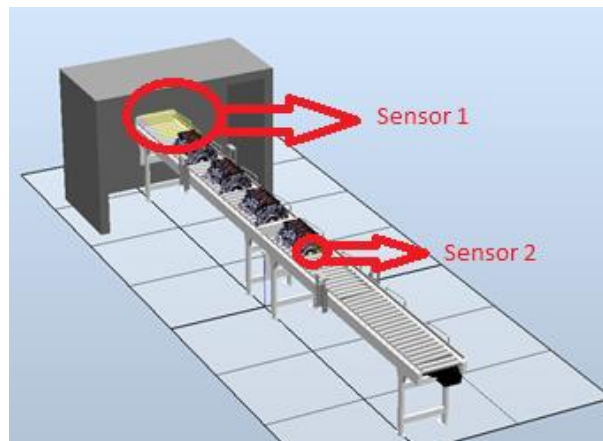


Figura 38

Nuestro “Componente Inteligente” de entrada de piezas está constituido por varios bloques que están conectados mediante entradas y salidas digitales y relacionados en algunos casos en propiedades. El esquema de funcionamiento será el reflejado en la figura 39:

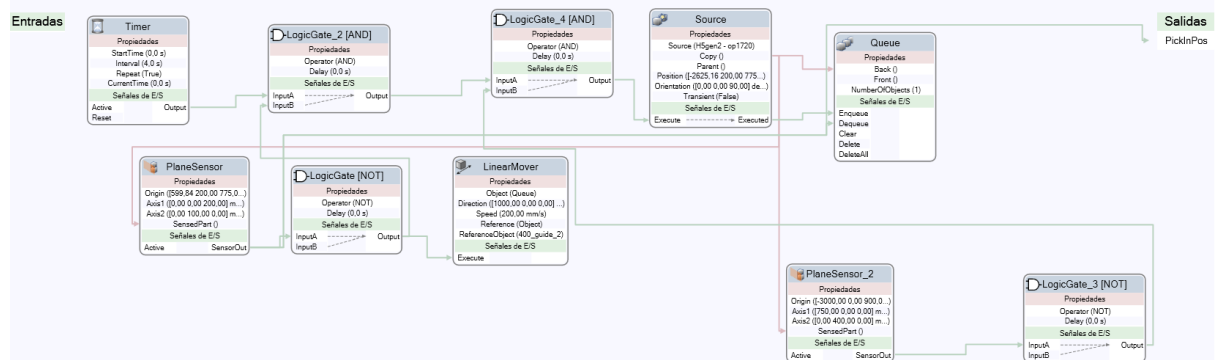


Figura 39



TEMPORIZADOR: que emitirá cada 4 segundos una señal (Output = 1) que irá conectado a una puerta lógica AND (LogicGate_2). Este elemento nos servirá para que cada 4 segundos, si se cumple otra serie de condiciones, se introducirá una nueva pieza a la manutención. El bloque es el reflejado en la figura 40.

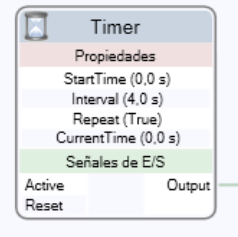


Figura 40

BLOQUE SENSOR-NOT (1): está compuesto de un sensor de plano (sensor situado en la posición de cogida de la pieza) y una puerta lógica NOT. Los bloques que lo componen son los reflejados en la figura 41. La finalidad de este elemento es que mientras el plano de sensor no sea cortado por ningún elemento estará emitiendo un 0 en su Output, que será recibido por la puerta lógica NOT de tal manera que el funcionamiento será el siguiente:

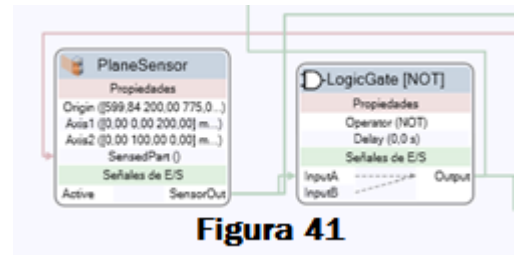


Figura 41

Sensor de plano			Puerta lógica NOT	
Plano no cortado	Output = 0	→	Input A = 0	Output = 1
Plano cortado	Output = 1	→	Input A = 1	Output = 0

PUERTA LÓGICA AND: los dos elementos anteriores van conectados a las dos entradas de una puerta lógica AND (LogicGate_2) que siempre y cuando esté recibiendo en las dos entradas un 1, emitirá una señal Output = 1; si alguna de las entradas fuese distinta de 1, entonces emitirá una señal Output = 0. Los bloques que lo componen son los reflejados en la figura 42.

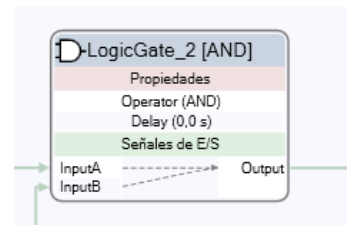


Figura 42



BLOQUE SENSOR-NOT (2): está compuesto de un sensor de plano (sensor situado en entrada para asegurar que no hay colisión entre una pieza parada en el carro y otra nueva entrante en el flujo) y una puerta lógica NOT. La finalidad de este elemento es que mientras el plano de sensor esté cortado, no permita introducir una nueva pieza. El funcionamiento de este elemento será igual que el explicado anteriormente de un bloque idéntico. Los bloques que los componen son los reflejados en la figura 43.

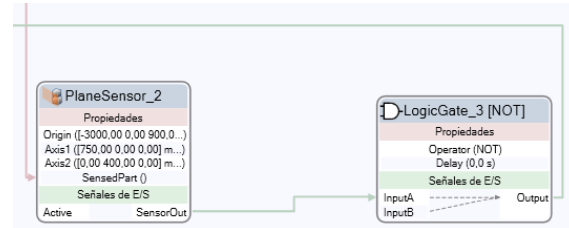


Figura 43

PUERTA LÓGICA AND: los dos elementos anteriores van conectados a las dos entradas de una puerta lógica AND (LogicGate_4) que siempre y cuando esté recibiendo en las dos entradas un 1, emitirá una señal Output = 1; si alguna de las entradas fuese distinta de 1, entonces emitirá una señal Output = 0. La finalidad de este es cortar la entrada de piezas que se sitúan en la cola. Los bloques que los compone son los reflejados en la figura 44.

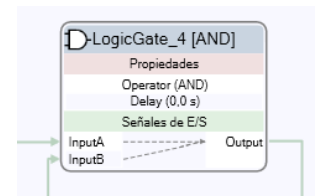


Figura 44

BLOQUE SOURCE-QUEUE (2): está compuesto de source (crea un elemento gráfico idéntico a uno de referencia) y una cola. La finalidad de este elemento es crear ese flujo de piezas entrantes. Siempre que se cree un nuevo elemento se añadirá a la cola y siempre que se llegue hasta la posición de cogida por el robot ese elemento de la posición de cogida dejará de pertenecer a la cola. Los bloques que lo componen son los reflejados en la figura 45.

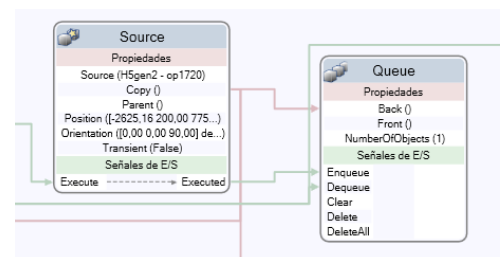


Figura 45

LINEAR MOVER: está compuesto de un bloque LinearMover que moverá en una determinada dirección el elemento que se le diga, en este caso la cola de piezas. La finalidad de este elemento es crear ese movimiento en el flujo de pieza. Siempre que el plano de sensor de la pieza en posición de cogida por el robot no esté cortado por ningún elemento, la cola se moverá. Los bloques que los compone son los reflejados en la figura 46.

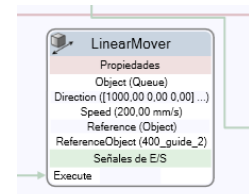


Figura 46



Para entender mejor la relación de entradas y salidas entre bloques (flechas verdes), es decir, que salidas corresponden con que entradas, las conexiones están reflejadas en la siguiente tabla:

Conexiones de EIS

Objeto de origen	Señal de origen	Objeto de destino	Señal de destino
Source	Executed	Queue	Enqueue
PlaneSensor	SensorOut	LogicGate [NOT]	InputA
LogicGate [NOT]	Output	LinearMover	Execute
PlaneSensor	SensorOut	Componente inteligente_1	PickInPos
PlaneSensor	SensorOut	Queue	Dequeue
Timer	Output	LogicGate_2 [AND]	InputA
LogicGate [NOT]	Output	LogicGate_2 [AND]	InputB
PlaneSensor_2	SensorOut	LogicGate_3 [NOT]	InputA
LogicGate_2 [AND]	Output	LogicGate_4 [AND]	InputA
LogicGate_3 [NOT]	Output	LogicGate_4 [AND]	InputB
LogicGate_4 [AND]	Output	Source	Execute

(*)Existe una señal digital de salida llamada PickInPos que será una salida hacia el exterior del componente inteligente indicando que hay pieza disponible para que el robot la pueda coger.

Para entender mejor la relación de propiedades entre bloques (flechas granates), es decir, que propiedades van de un bloque a otro, las conexiones están reflejadas en la siguiente tabla:

Enlazamientos de propiedad

Objeto de origen	Propiedad de origen	Objeto de destino	Propiedad de destino
Source	Copy	Queue	Back
Source	Copy	PlaneSensor	SensedPart
Source	Copy	PlaneSensor_2	SensedPart

Configuración del transportador de salida

La geometría es la reflejada en la figura 48.

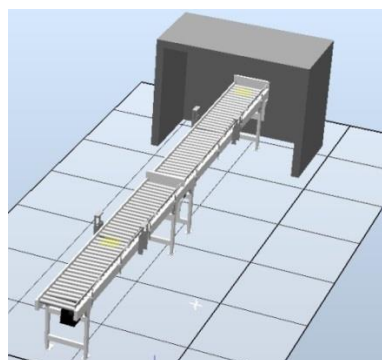


Figura 48



La función de este será la de que sacar de la célula robotizada las piezas que ya hayan pasado por todas las máquinas, con sus respectivos tiempos de operación en cada máquina.

Nuestro “Componente Inteligente” de salida de piezas está constituido por varios bloques que están conectados mediante entradas y salidas digitales y relacionados en algunos casos en propiedades. El esquema de funcionamiento está reflejado en la figura 49.:

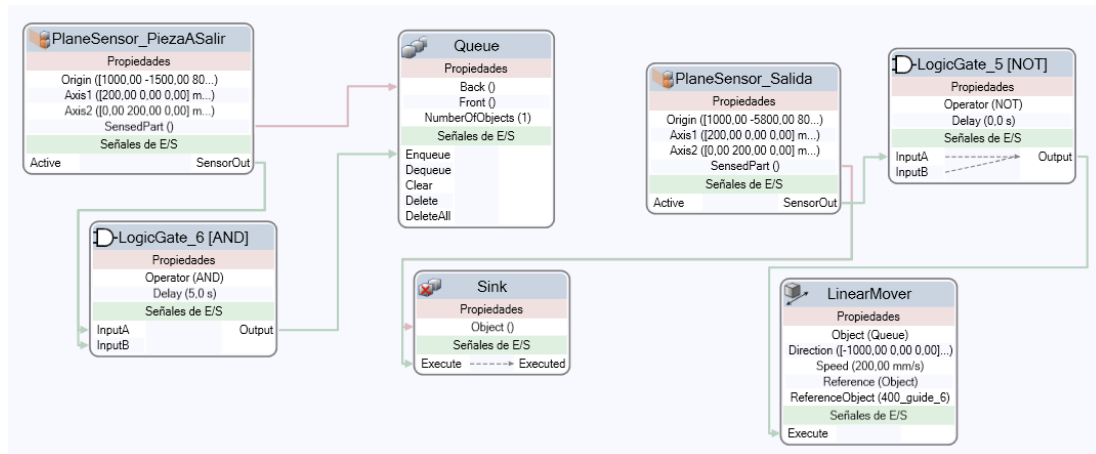


Figura 49

BLOQUE SENSOR-COLA: está compuesto de un sensor de plano (sensor situado en la posición de dejada de la pieza por el robot), una puerta lógica AND cuya única finalidad es que cuando la pieza a salir corte al plano, nuestra pieza tarde 5 segundos en ser colocada en la cola (delay = 5.0 s), es decir, se tarda 5 segundos desde que la pieza corta al sensor de plano, hasta que se activa la señal “Enqueue” que es la que coloca la pieza en la cola. Para saber que elemento debe de situar en la cola, se relaciona la propiedad de SensedPart (elemento detectado, es decir, nuestra pieza) con Back (elemento a situar en la cola). Los bloques que lo componen son los reflejados en la figura 50.

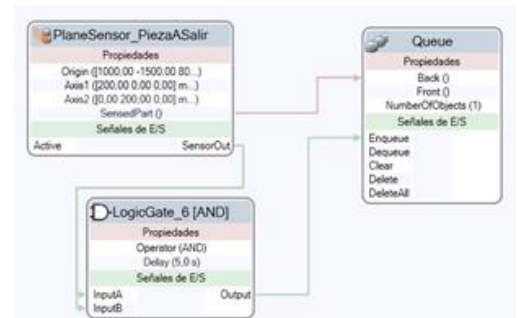


Figura 50



BLOQUE SENSOR-MOVER-ELIMINAR: está compuesto de un sensor de plano (sensor situado al final del transportador), una puerta lógica NOT, un elemento de movimiento y otro que elimina (sink) el componente gráfico, en nuestro caso, la pieza ya tratada. La finalidad de este bloque es que mientras el sensor de plano no sea cortado por nada desplazará la cola creada en el bloque anterior; una vez que el plano sensor sea cortado por algún elemento, este será eliminado y nuestro plano sensor ya no será cortado por nada, lo que permitirá que la cola se siga desplazando hacia el final del transportador. Los bloques que lo componen son los reflejados en la figura 51.

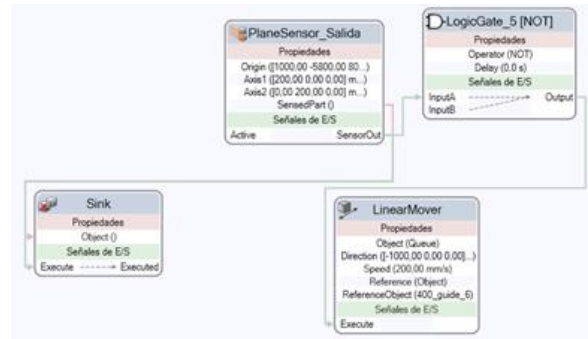


Figura 51

Para entender mejor la relación de entradas y salidas entre bloques (flechas verdes), es decir, que salidas corresponden con que entradas, las conexiones están reflejadas en la siguiente tabla:

Conexiones de E/S

Objeto de origen	Señal de origen	Objeto de destino	Señal de destino
PlaneSensor_Salida	SensorOut	LogicGate_5 [NOT]	InputA
LogicGate_5 [NOT]	Output	LinearMover	Execute
PlaneSensor_Salida	SensorOut	Sink	Execute
PlaneSensor_PiezaASalir	SensorOut	LogicGate_6 [AND]	InputA
PlaneSensor_PiezaASalir	SensorOut	LogicGate_6 [AND]	InputB
LogicGate_6 [AND]	Output	Queue	Enqueue

Para entender mejor la relación de propiedades entre bloques (flechas granates), es decir, que propiedades van de un bloque a otro, las conexiones están reflejadas en la siguiente tabla:

Enlazamientos de propiedad

Objeto de origen	Propiedad de origen	Objeto de destino	Propiedad de destino
PlaneSensor_PiezaASalir	SensedPart	Queue	Back
PlaneSensor_Salida	SensedPart	Sink	Object

Configuración inteligente del dispensador y recogedor

La geometría es la reflejada en la figura 52.

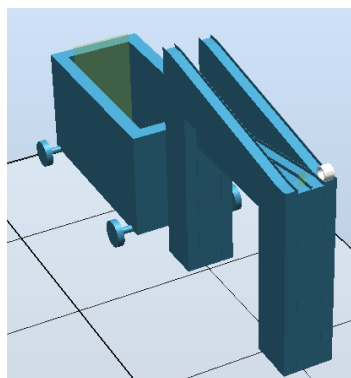


Figura 52

Este elemento se ha creado con las herramientas de modelado de RobotStudio, y se ha hecho como anteriormente con la cola de entrada un componente inteligente. Los bloques que lo componen son los reflejados en la figura 53

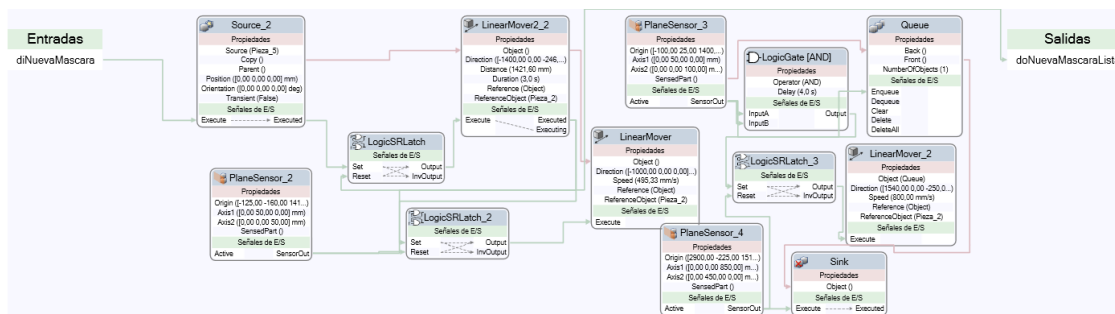


Figura 53

En este elemento tendríamos dos líneas de funcionamiento independiente: una para la entrada de máscaras limpias, y otra para la salida de máscaras sucias.

1. LINEA MÁSCARAS LIMPIAS:

Los bloques que lo componen son los reflejados en la figura 54.

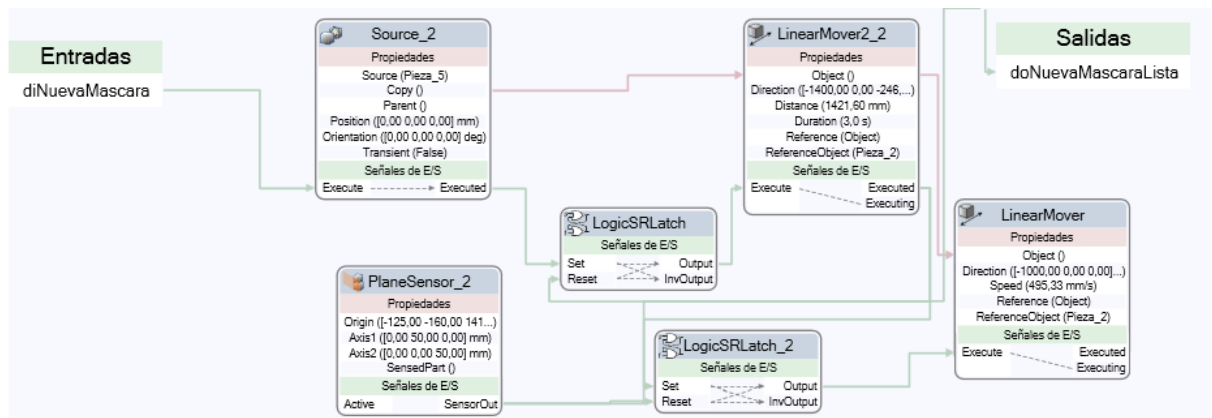


Figura 54

BLOQUE SOURCE: está compuesto de “Source_2” (crea un elemento gráfico idéntico a uno de referencia). La finalidad de este elemento es crear una máscara cuando le sea solicitado a través de la entrada “diNuevaMascara”. La máscara creada empezará desde el punto más alto del dispensador y se deslizará hasta la posición de cogida. Los bloques que lo componen son los reflejados en la figura 55.

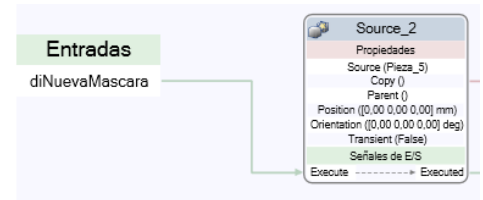


Figura 55

BLOQUE DESLIZAMIENTO: Está compuesto un sensor de plano, dos set-reset, y dos acciones de movimiento. La idea es que cuando el “bloque source” anterior ha creado la nueva máscara, se le hace descender con la misma inclinación y longitud que la rampa del dispensador. Una vez realizado esto, realiza un movimiento lineal paralelo al suelo, cortando al sensor de plano, que frena el movimiento y nuestra máscara queda perfectamente situada en la posición de cogida activando la salida “doNuevaMascaraLista” al exterior del componente inteligente que nos servirá para comunicar que ya hay una nueva máscara limpia. Los bloques que lo componen son los reflejados en la figura 56.

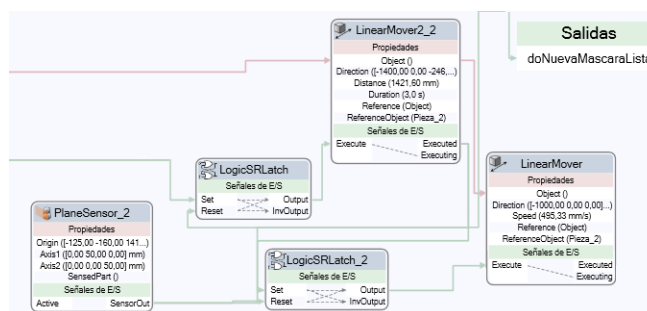


Figura 56



(*)La relación de entradas y salidas entre bloques (flechas verdes), se encuentra al final del apartado del dispensador de máscaras.

(**)La relación de propiedades entre bloques (flechas granates), se encuentra al final del apartado del dispensador de máscaras.

2. LINEA MÁSCARAS SUCIAS:

Los bloques que lo componen son los reflejados en la figura 57.

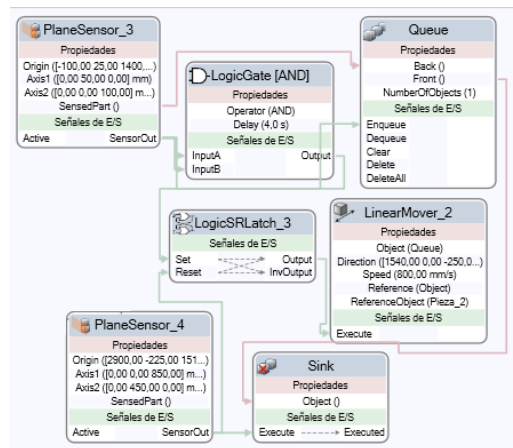


Figura 57

BLOQUE DESLIZAMIENTO: Está compuesto un sensor de plano, un set-reset, una puerta lógica AND (con el único fin de introducir un retraso de 4 segundos desde la detección del sensor) y una acción de movimiento. La idea es que cuando se corta el sensor de plano situado donde se dejan las máscaras se espera 4 segundos y se empieza a deslizar la máscara hacia el carro de recogida. Los bloques que lo componen son los reflejados en la figura 58.

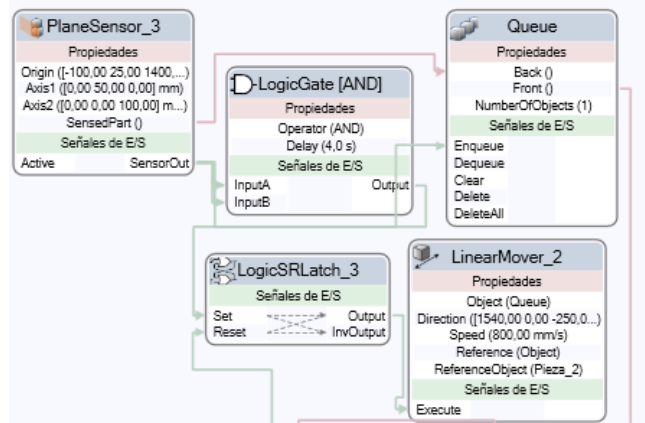


Figura 58

BLOQUE CARRO: Está compuesto un sensor de plano y un sink (elimina un elemento gráfico). La idea es que cuando se corta el sensor de plano situado en una de las caras del carro, la máscara sucia desaparece simulando que esa máscara ya no es utilizable. Los bloques que lo componen son los reflejados en la figura 59.

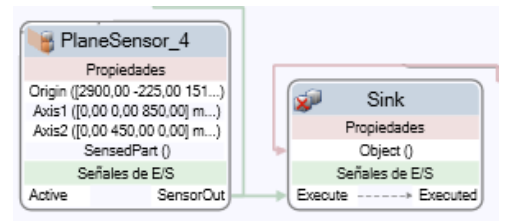


Figura 59

Para entender mejor la relación de entradas y salidas entre bloques (flechas verdes), es decir, que salidas corresponden con que entradas, las conexiones están reflejadas en la siguiente tabla:

Conexiones de E/S

Objeto de origen	Señal de origen	Objeto de destino	Señal de destino
Mascaras	diNuevaMascara	Source_2	Execute
Source_2	Executed	LogicSRLatch	Set
LogicSRLatch	Output	LinearMover2_2	Execute
LinearMover2_2	Executed	LogicSRLatch	Reset
LinearMover2_2	Executed	LogicSRLatch_2	Set
PlaneSensor_2	SensorOut	LogicSRLatch_2	Reset
LogicSRLatch_2	Output	LinearMover	Execute
PlaneSensor_4	SensorOut	Sink	Execute
PlaneSensor_4	SensorOut	LogicSRLatch_3	Reset
LogicSRLatch_3	Output	LinearMover_2	Execute
PlaneSensor_3	SensorOut	Queue	Enqueue
PlaneSensor_2	SensorOut	Mascaras	doNuevaMascaraLista
PlaneSensor_3	SensorOut	LogicGate [AND]	InputA
PlaneSensor_3	SensorOut	LogicGate [AND]	InputB
LogicGate [AND]	Output	LogicSRLatch_3	Set



(*)Existe una señal digital de salida llamada doNuevaMascaraLista que será una salida hacia el exterior del componente inteligente indicando que hay máscara disponible para que el robot la pueda coger.

Para entender mejor la relación de propiedades entre bloques (flechas granates), es decir, que propiedades van de un bloque a otro, las conexiones están reflejadas en la siguiente tabla:

Enlazamientos de propiedad

Objeto de origen	Propiedad de origen	Objeto de destino	Propiedad de destino
Source_2	Copy	LinearMover2_2	Object
LinearMover2_2	Object	LinearMover	Object
PlaneSensor_3	SensedPart	Queue	Back
Queue	Front	Sink	Object

Configuración de la herramienta con doble pinza

La geometría es la reflejada en la figura 60.

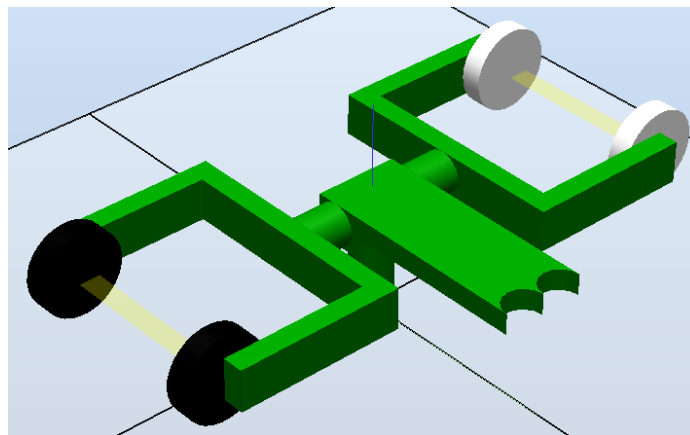


Figura 60

Este elemento se ha creado con las herramientas de modelado de RobotStudio, y se ha hecho como anteriormente un componente inteligente, cuyo esquema de bloques para su funcionamiento es reflejado en la figura 61:

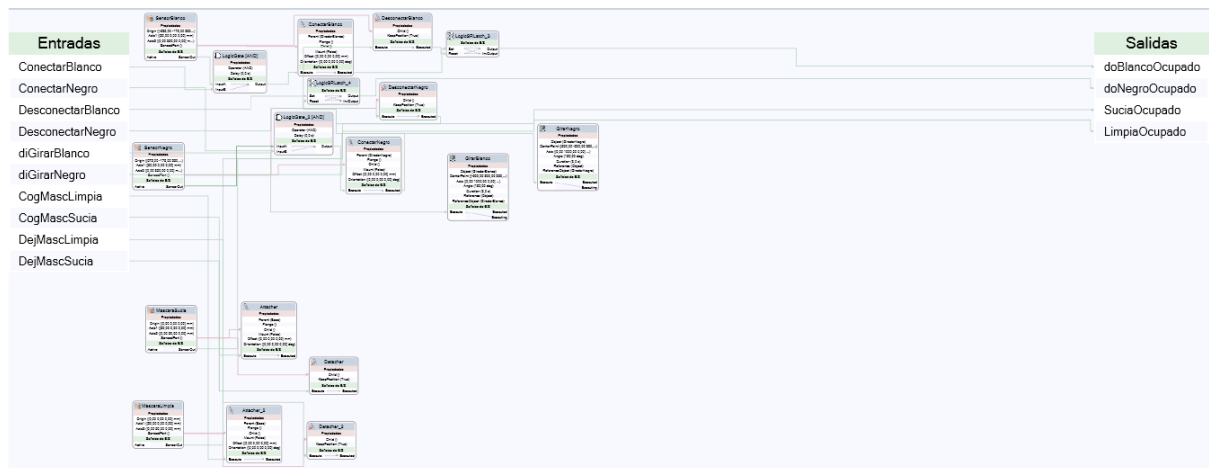


Figura 61

Este componente inteligente tendría dos líneas de funcionamiento independientes: una para la cogida y dejada de las piezas a mecanizar (con posibilidad de voltearlas), y la otra para la cogida y dejada de las máscaras necesarias en la máquina 3.

1. LINEA COGIDA Y DEJADA DE PIEZAS:

Los bloques que lo componen son los reflejados en la figura 62.

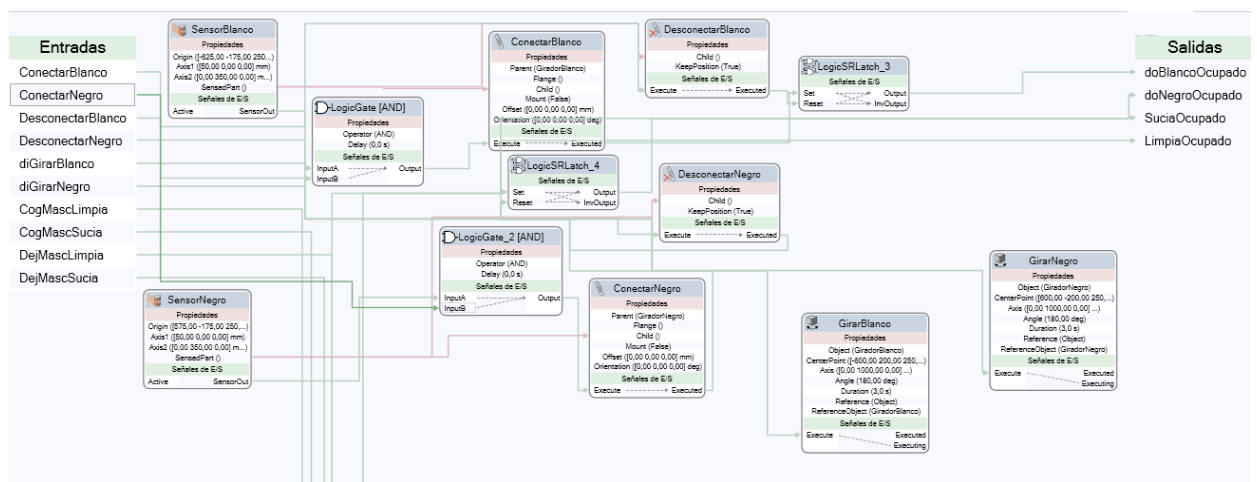


Figura 62



Para entender el funcionamiento debemos de tener en cuenta que es como si tuviésemos un mismo elemento por duplicado, que sería una única pinza. Esta pinza lo que tiene es un sensor que detecta que hay piza, un conector que une la pieza a la pinza, un girador para voltear la pieza y un desconector para desunir la piza de la pinza. Esto viene regido por unas entradas que nos dicen cuando hay que conectar y desconectar, y cuando debemos de girar.

BLOQUE SENSOR PIEZA EN PINZA: Está compuesto un sensor de plano que va de extremo a extremo de la pinza y que detecta si hay alguna pieza que corte ese plano. Los bloques que lo componen son los reflejados en la figura 63.

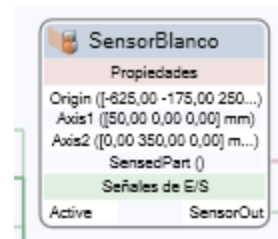


Figura 63

BLOQUE CONECTOR PIEZA EN PINZA: Está compuesto una puerta lógica AND (LogicGate) que reciben en el Input A la señal del bloque sensor indicando que hay una pieza y en el Input B la señal entrante del exterior del componente inteligente “ConectarBlanco” o “ConectarNegro” (en función de la pinza que deseemos conectar. Los bloques que lo componen son los reflejados en la figura 64.

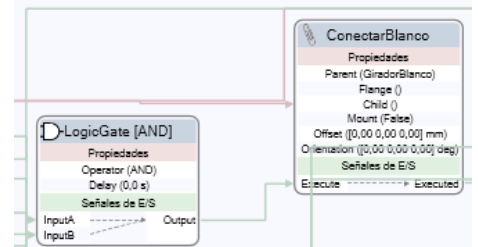


Figura 64

BLOQUE GIRADOR DE PIEZA EN PINZA: Está compuesto simplemente por un componente de movimiento de rotación, que lo que hará será girar los dos extremos de la pinza para colocar de una manera u otra la pieza. Este bloque se ejecutará cuando la señal de entrada diGirarBlanco o diGirarNegro se active a 1. Los bloques que lo componen son los reflejados en la figura 65.

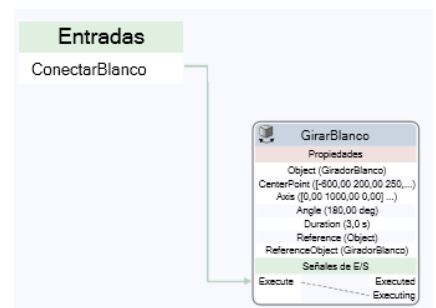


Figura 65



BLOQUE DESCONECTOR DE PIEZA EN PINZA: Está compuesto por un bloque que desconecta dos elementos, en este caso se desconectará la pinza de nuestra pieza, y esto ocurrirá cuando se active una señal de entrada en el componente inteligente llamada “DesconectarBlanco” o “DesconectarNegro”. El bloque es reflejado en la figura 66.



Figura 66

2. LINEA COGIDA Y DEJADA DE MASCARAS

Los bloques que lo componen son los reflejados en la figura 67.

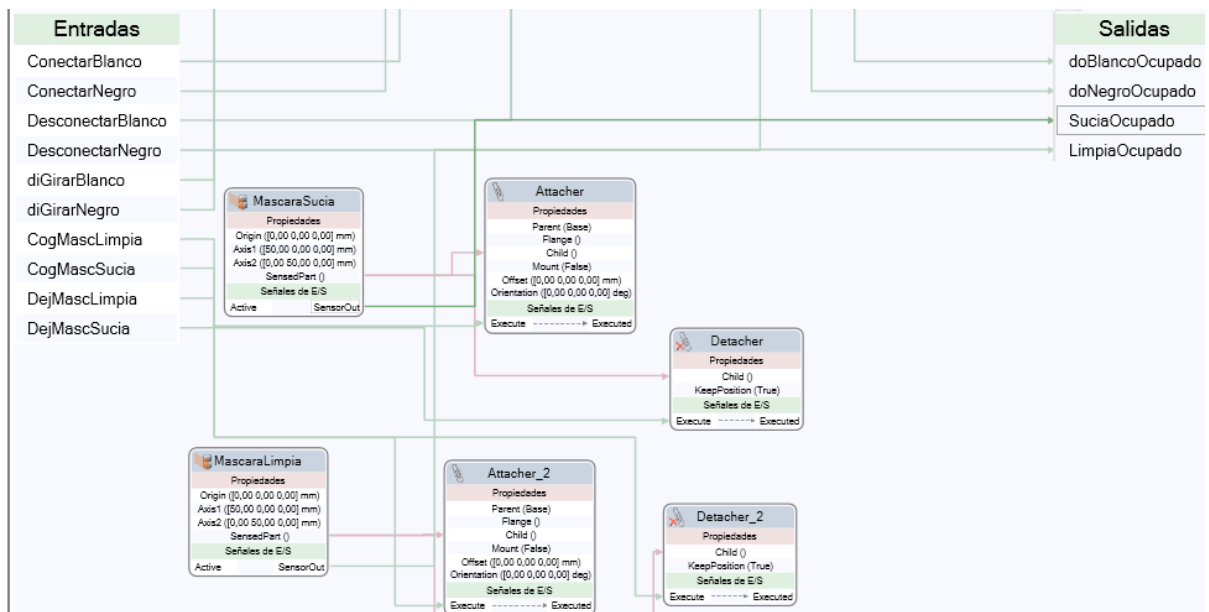


Figura 67

El funcionamiento de como coger y dejar las máscaras es muy sencillo; hay una pinza para máscaras limpias y otra para sucias, una señal digital activa cuando hay que conectar o desconectar la máscara de la pinza, y mediante un sensor se sabe si está ocupada o no la pinza.



BLOQUE CONECTOR Y DESCONECTOR DE PIEZA EN PINZA: el sensor

marca lo que se debe conectar y lo que se debe desconectar mediante la relación de propiedades. La conexión y desconexión va gestionada por señales digitales de entrada. Los bloques que lo componen son los reflejados en la figura 68.

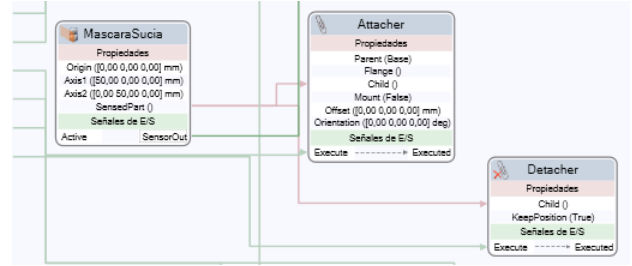


Figura 68

La pinza giratoria tiene una serie de entradas y salidas digitales que gestionan las ordenes que llegan del exterior para realizar acciones o para dar información sobre cuál es el estado actual de la pinza (ocupada, libre, etc). Estas señales son las siguientes:

Señales de E/S

Nombre	Tipo de señal	Valor
ConectarBlanco	DigitalInput	0
ConectarNegro	DigitalInput	0
DesconectarBlanco	DigitalInput	0
DesconectarNegro	DigitalInput	0
doBlancoOcupado	DigitalOutput	0
doNegroOcupado	DigitalOutput	0
diGirarBlanco	DigitalInput	0
diGirarNegro	DigitalInput	0
CogMascLimpia	DigitalInput	0
CogMascSucia	DigitalInput	0
DejMascLimpia	DigitalInput	0
DejMascSucia	DigitalInput	0
SuciaOcupado	DigitalOutput	0
LimpiaOcupado	DigitalOutput	0

Para entender mejor la relación de entradas y salidas entre bloques (flechas verdes), es decir, que salidas corresponden con que entradas, las conexiones están reflejadas en la siguiente tabla:



Conexiones de E/S

Objeto de origen	Señal de origen	Objeto de destino	Señal de destino
SensorBlanco	SensorOut	LogicGate [AND]	InputA
Pinza_Giratoria	ConectarBlanco	LogicGate [AND]	InputB
LogicGate [AND]	Output	ConectarBlanco	Execute
SensorNegro	SensorOut	LogicGate_2 [AND]	InputA
Pinza_Giratoria	ConectarNegro	LogicGate_2 [AND]	InputB
Pinza_Giratoria	DesconectarBlanco	DesconectarBlanco	Execute
Pinza_Giratoria	DesconectarNegro	DesconectarNegro	Execute
ConectarBlanco	Executed	LogicSRLatch_3	Set
DesconectarBlanco	Executed	LogicSRLatch_3	Reset
LogicSRLatch_3	Output	Pinza_Giratoria	doBlancoOcupado
DesconectarNegro	Executed	LogicSRLatch_4	Reset
LogicSRLatch_4	Output	Pinza_Giratoria	doNegroOcupado
Pinza_Giratoria	diGirarNegro	GirarNegro	Execute
Pinza_Giratoria	diGirarBlanco	GirarBlanco	Execute
LogicGate_2 [AND]	Output	ConectarNegro	Execute
ConectarNegro	Executed	LogicSRLatch_4	Set
Pinza_Giratoria	CogMascSucia	Attacher	Execute
Pinza_Giratoria	CogMascLimpia	Attacher_2	Execute
Pinza_Giratoria	DejMascLimpia	Detacher_2	Execute
Pinza_Giratoria	DejMascSucia	Detacher	Execute
MascaraSucia	SensorOut	Pinza_Giratoria	SuciaOcupado
MascaraLimpia	SensorOut	Pinza_Giratoria	LimpiaOcupado

Para entender mejor la relación de propiedades entre bloques (flechas granates), es decir, que propiedades van de un bloque a otro, las conexiones están reflejadas en la siguiente tabla:

Enlazamientos de propiedad

Objeto de origen	Propiedad de origen	Objeto de destino	Propiedad de destino
SensorBlanco	SensedPart	ConectarBlanco	Child
SensorBlanco	SensedPart	DesconectarBlanco	Child
SensorNegro	SensedPart	DesconectarNegro	Child
SensorNegro	SensedPart	ConectarNegro	Child
MascaraSucia	SensedPart	Attacher	Child
MascaraLimpia	SensedPart	Attacher_2	Child
MascaraSucia	SensedPart	Detacher	Child
MascaraLimpia	SensedPart	Detacher_2	Child

Configuración de la maquina 1:

Ya hemos hablado de que esta máquina se ocupara de una operación de mecanizado de la pieza. La geometría es la reflejada en la figura 69.

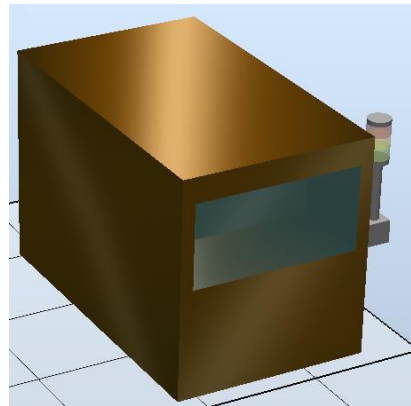


Figura 69

Este elemento se ha creado con las herramientas de modelado de RobotStudio, y se ha hecho como los anteriores con un componente inteligente, que llamaremos componente inteligente de nivel superior (ya que este cuenta con otro componente inteligente de nivel inferior), cuyo esquema de bloques para su funcionamiento es reflejado en la figura 70.

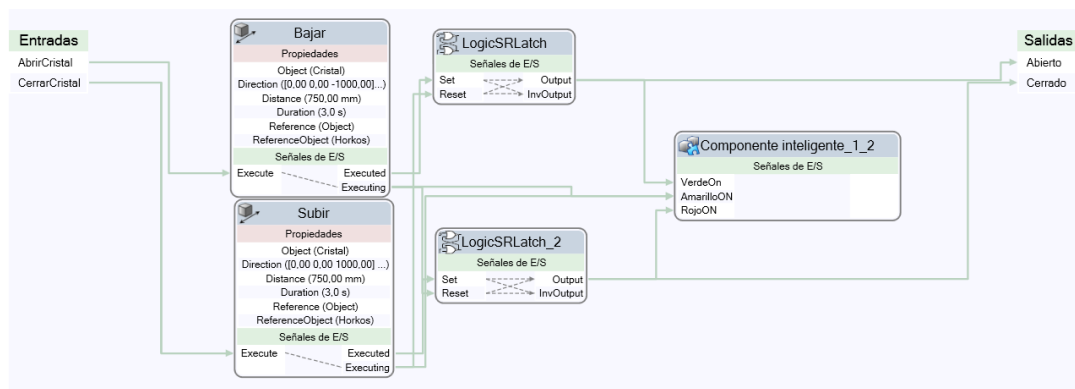


Figura 70

Podríamos diferenciar tres líneas de funcionamiento. La primera de ellas se encargará de abrir la puerta de la máquina, la segunda se encargará de cerrar la puerta, y la tercera será un componente inteligente de nivel inferior, que se encargará de encender y apagar luces de puerta abierta, puerta cerrada, abriendo o cerrando puerta.

1. LINEA ABERTURA DE LA PUERTA:

Los bloques que lo componen son los reflejados en la figura 71.

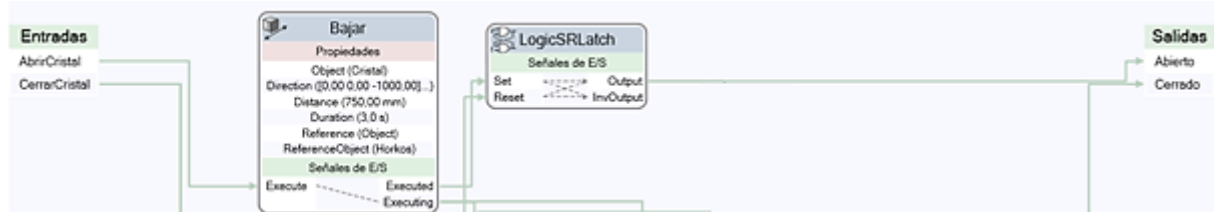


Figura 71

El funcionamiento será muy sencillo, la entrada del componente inteligente “AbrirCristal” será la encargada de hacer ejecutar la acción de bajar el cristal, y una vez hecho mandará a través del LogicSRLatch la señal de que la puerta está abierta.

2. LINEA CIERRE DE LA PUERTA:

Los bloques que lo componen son los reflejados en la figura 72.

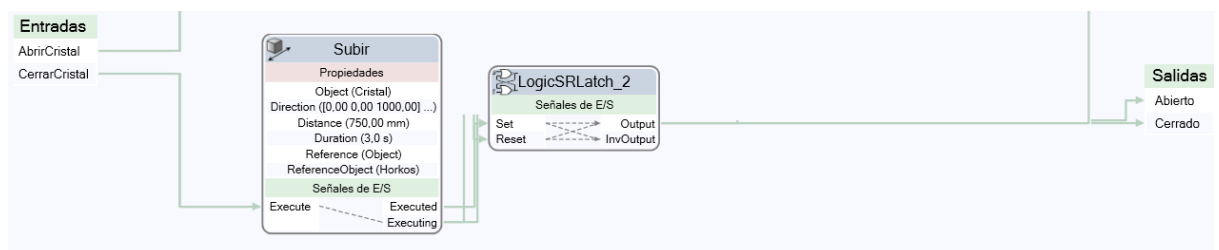


Figura 72

El funcionamiento será también muy sencillo, la entrada del componente inteligente “CerrarCristal” será la encargada de hacer ejecutar la acción de subir el cristal, y una vez hecho mandará a través del LogicSRLatch la señal de que la puerta está Cerrada.

3. LINEA LUCES DE ESTADO:

Este componente es algo especial, ya que interactúa con las dos líneas de las que hemos hablado anteriormente, siendo a su vez un componente inteligente de nivel inferior; una explicación sencilla sería decir que es un componente inteligente dentro de otro componente inteligente de nivel superior. La relación que tiene con las dos líneas anteriores queda reflejada en la figura 73:

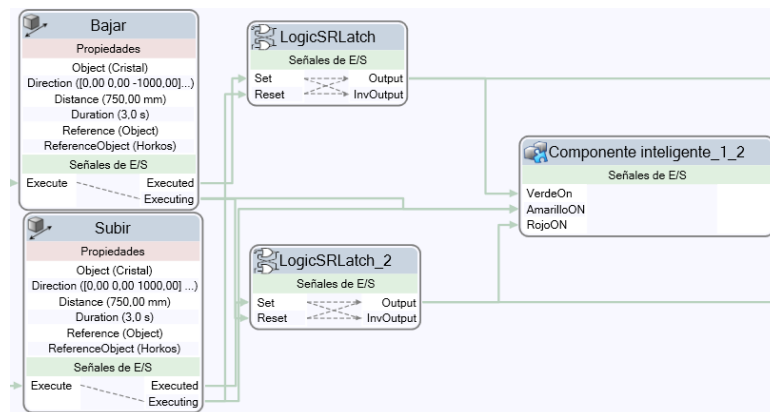


Figura 73

Tiene tres entradas que recibirán la señal de que la puerta se ha bajado o se ha subido (VerdeOn y RojoOn respectivamente), y la señal que la puerta está bajándose o subiéndose (AmarilloON).

Entrando en la propia configuración de este componente inteligente de luces de estado nos encontramos el siguiente esquema de funcionamiento reflejado en la figura 74:

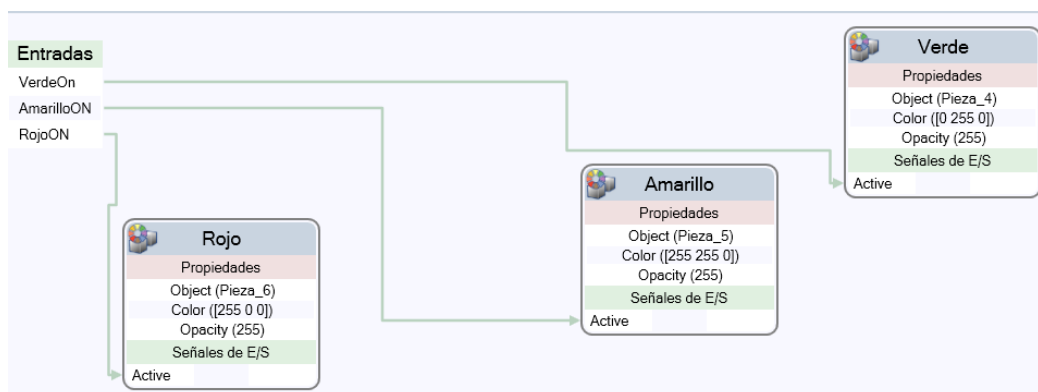


Figura 74

Estos bloques activarán el cambio de color a un color opaco en función de las entradas recibidas, que están reflejadas en la siguiente tabla:

Conexiones de E/S

Objeto de origen	Señal de origen	Objeto de destino	Señal de destino
Componente inteligente_1_2	RojoON	Rojo	Active
Componente inteligente_1_2	AmarilloON	Amarillo	Active
Componente inteligente_1_2	VerdeOn	Verde	Active

Con la finalidad de que se entienda mejor, vamos a ver un esquema del componente inteligente de la maquina 1 de mecanizado, indicando



donde se sitúa el esquema de funcionamiento de las luces de estado.
Reflejado en la figura 75.

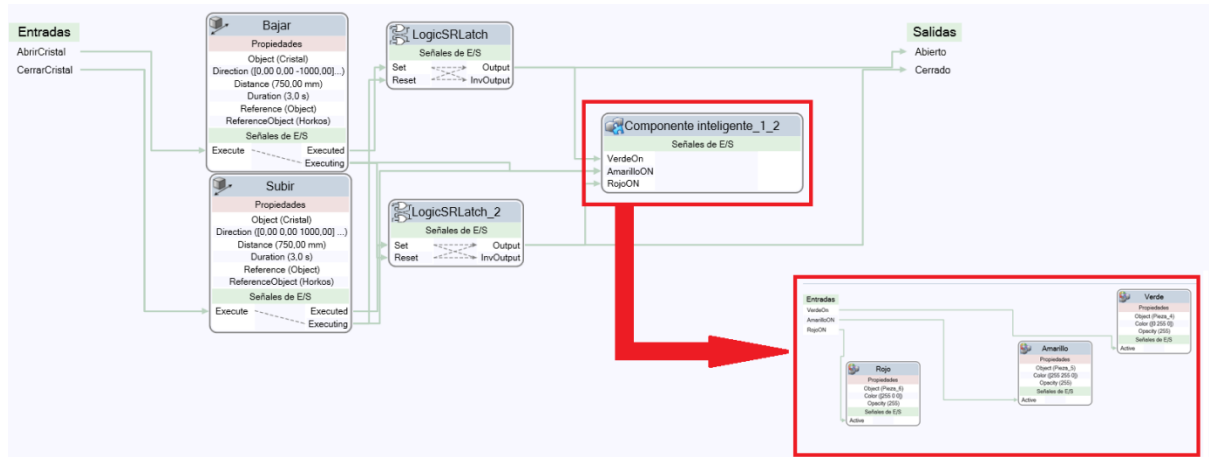


Figura 75

Para entender mejor la relación de entradas y salidas entre bloques (flechas verdes), es decir, que salidas corresponden con que entradas, las conexiones están reflejadas en la siguiente tabla:

Conexiones de E/S

Objeto de origen	Señal de origen	Objeto de destino	Señal de destino
Bajar	Executing	Componente inteligente_1_2	AmarilloON
Subir	Executing	Componente inteligente_1_2	AmarilloON
Horkos	AbrirCristal	Bajar	Execute
Horkos	CerrarCristal	Subir	Execute
Subir	Executed	LogicSRLatch_2	Set
LogicSRLatch_2	Output	Componente inteligente_1_2	RojoON
Bajar	Executed	LogicSRLatch	Set
LogicSRLatch	Output	Componente inteligente_1_2	VerdeON
LogicSRLatch	Output	Horkos	Abierto
LogicSRLatch_2	Output	Horkos	Cerrado
Bajar	Executing	LogicSRLatch_2	Reset
Subir	Executing	LogicSRLatch	Reset

(*)Existen dos señales digitales de salida llamadas **Abierto** y **Cerrado** que serán señales de salida hacia el exterior del componente inteligente indicando que la maquina está abierta o la maquina está cerrada.



Configuración de la maquina 2: Horno

La geometría es la reflejada en la figura 76.

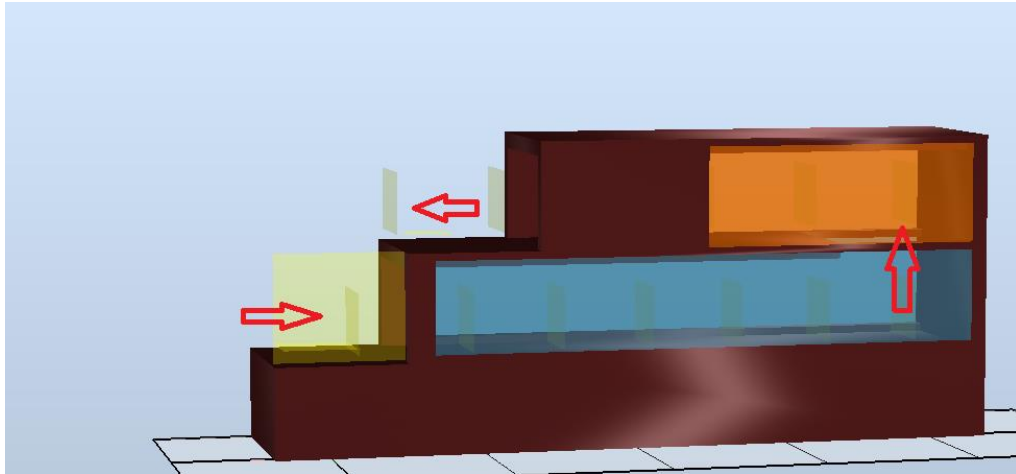


Figura 76

Este es el componente inteligente más complejo ya que está diseñado de tal manera que se puedan introducir más de una pieza de tal forma que se puedan calentar más de una pieza a la vez, y una vez la posición de salida esté libre, todas las piezas avancen una posición. El ejemplo visual de funcionamiento sería el reflejado en la figura 77:

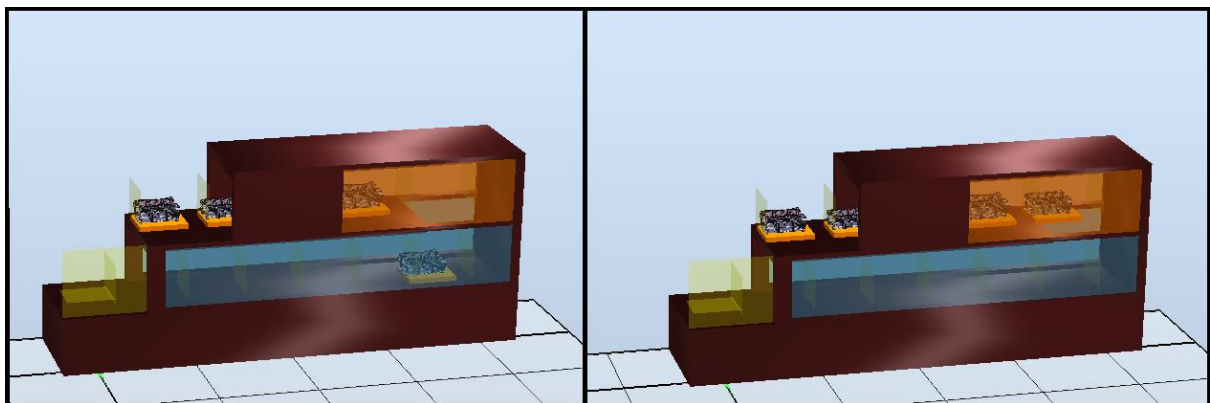


Figura 77

En la primera imagen de la figura 77 se ve como hay una serie de piezas esperando en la parte superior del horno, mientras una va avanzando por la parte inferior hasta subir y situarse en la última posición libre como se ve en la segunda imagen. Una vez el robot se lleve la pieza del puesto de cogida, todas las piezas avanzarán hasta que la posición de cogida vuelva a estar ocupado.

El esquema de funcionamiento del horno es el reflejado en la figura 78:

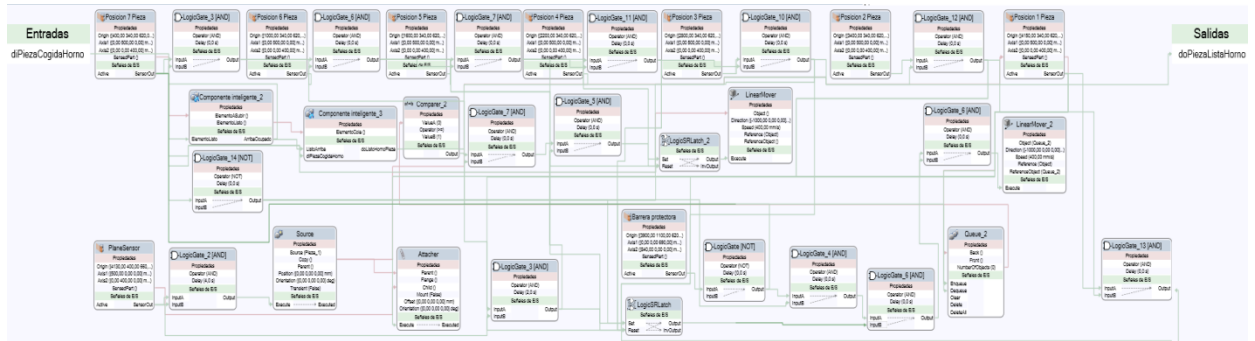


Figura 78

Al haber una cantidad bastante numerosa de sensores en el horno inferior, vamos a señalar los sensores de plano que indican las distintas posiciones en las figuras 79 y 80:

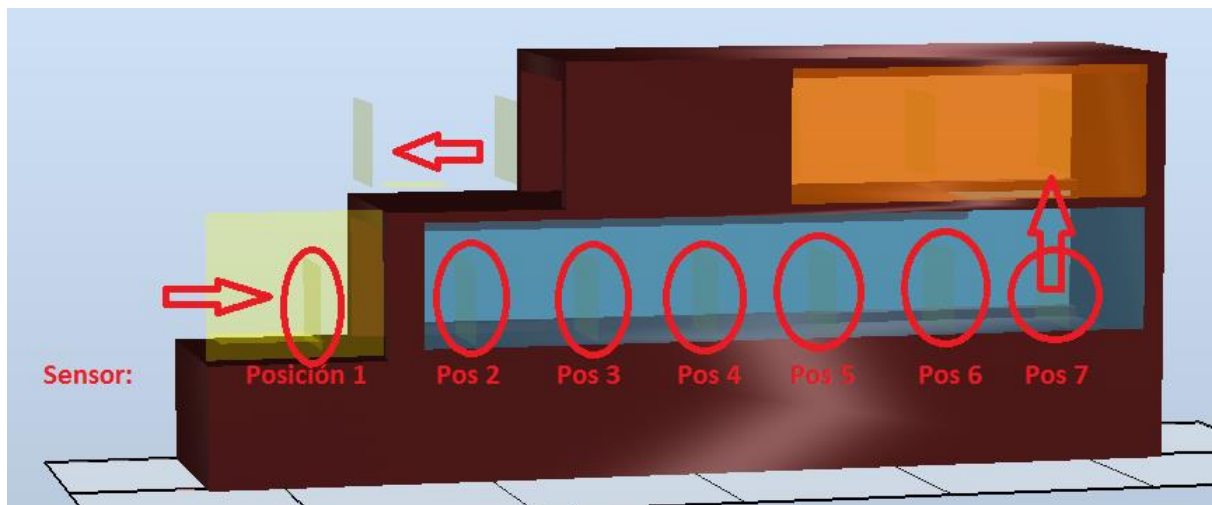


Figura 79

Además de estos en el nivel inferior, habrá otros dos planos sensores:

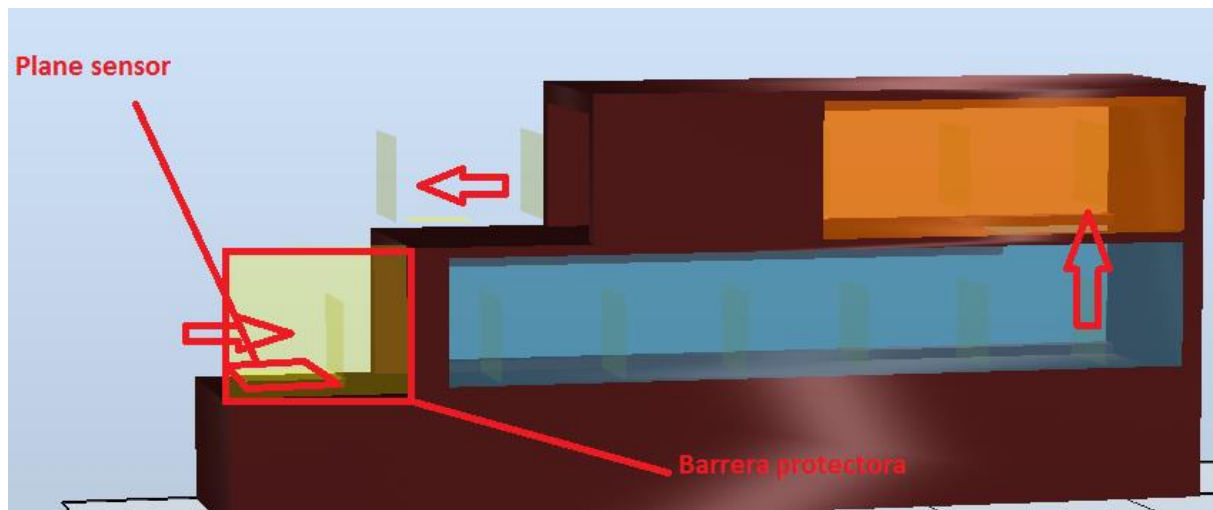


Figura 80

BLOQUE SOPORTE PARA PIEZAS: Este bloque creará un soporte naranja al detectar que hemos puesto una pieza a la entrada del horno. Detectará que hay una pieza a la entrada del horno ya que la pieza cortará un sensor de plano horno (plane sensor); creará el soporte (con un retraso de 4 segundos desde que el sensor detecta que hay pieza a entrar en el horno), conectará la pieza con el soporte, detectará que ya está el soporte, que lo situará en la cola (la cola será movida por otro bloque) y cuando el sensor 7 esté activo quitará el primer elemento de la cola. Los bloques que lo componen son los reflejados en la figura 81.

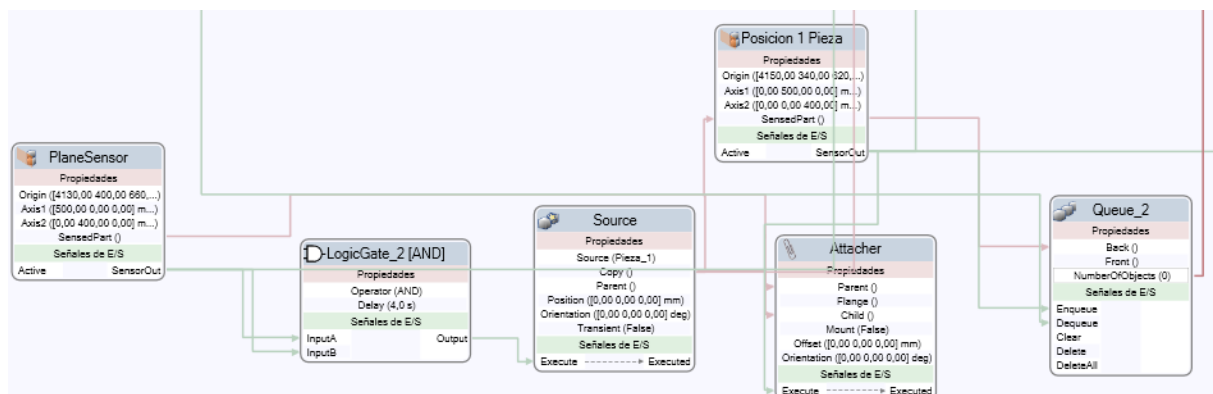


Figura 81

BLOQUE PIEZAS EN HORNO INFERIOR: Este bloque de lo que se encarga es de comprobar que la posición siguiente a la que podría avanzar la pieza junto con su soporte está libre. Esto lo haremos mediante una serie de planos sensores y puertas AND debidamente



conectados entre sí mediante las señales digitales. Los bloques que lo componen son los reflejados en la figura 82.

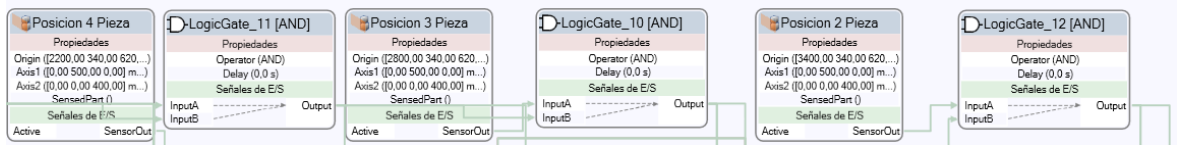
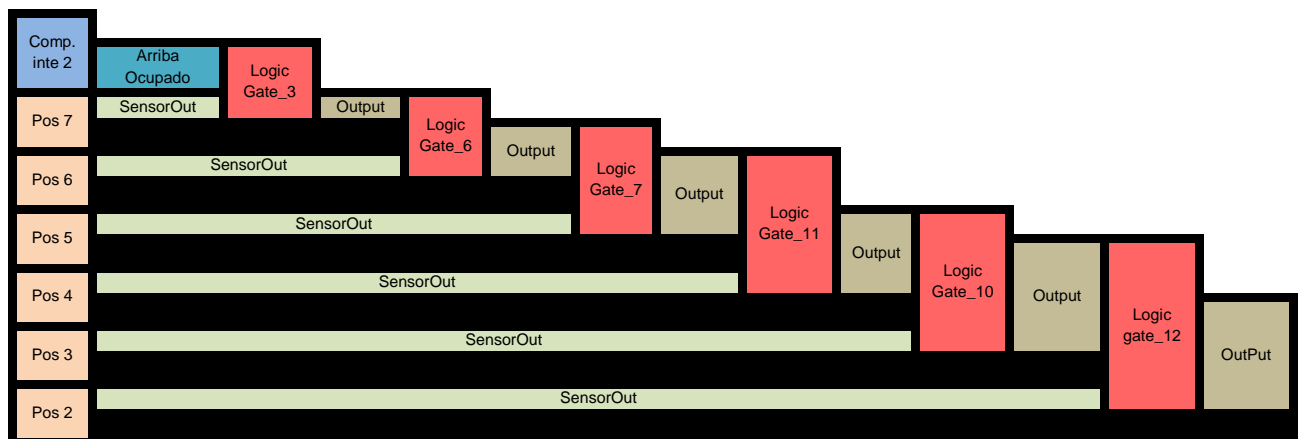


Figura 82

Las puertas AND nos servirán para indicar si las posiciones siguientes están ocupadas. El esquema solo va desde la posición 4 hasta la posición 2, pero la conexión y el funcionamiento serían igual para todos.

Para entender mejor el funcionamiento de este bloque vamos a ver en una tabla las salidas y entras de nuestros sensores y de nuestras puertas lógicas (todas ellas AND).



Las salidas (Output) de las puertas lógicas AND (LogicGate_X) serán las que frenarán el movimiento de las piezas (obviamente debidamente conectadas) por el horno inferior si las posiciones siguientes están ocupadas.

BLOQUE MOVIMIENTO PIEZAS EN HORNO INFERIOR: Este bloque de lo que se encarga es de mover las piezas por el horno inferior siempre y cuando se cumplan una serie de condiciones; el movimiento se hará de



todas las piezas que estén en el horno inferior. Los bloques que lo componen son los reflejados en la figura 83.

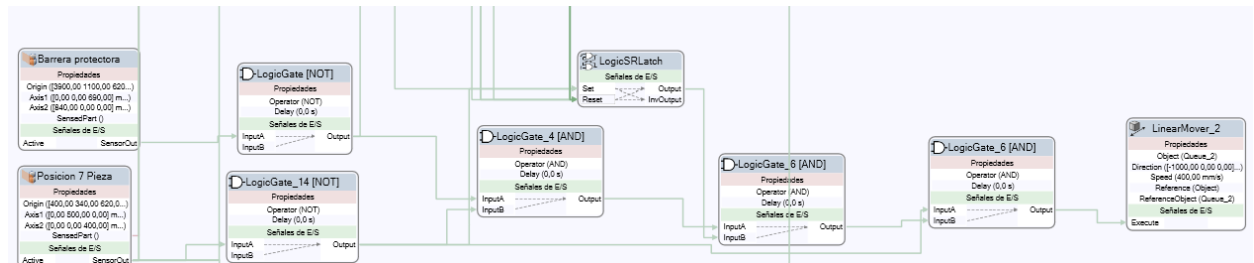


Figura 83

En este movimiento influyen dos sensores de plano, el primero es la berrara protectora que hemos podido ver anteriormente en el esquema de los sensores, que lo que hará será evitar activar el movimiento de las piezas en el horno mientras que el robot este dentro del volumen de trabajo del horno; el segundo plano es el de posición 7, que mientras no sea cortado por nada permitirá el movimiento.

Estas dos condiciones irán unidas a una puerta AND que irá a su vez unida a otra AND que tendrá una nueva condición impuesta por la LogicSRLatch que tiene una condición de set, y es que generará un Output = 1 cuando se produzca un cambio de 0 a 1, y generará un Output = 0 cuando cualquiera de las puertas AND explicadas en el bloque anterior cambie de 0 a 1, lo que significará que hay las posiciones siguientes están ocupadas por piezas y no avanzará más la cola.

BLOQUE MOVIMIENTO PIEZAS ENTRANTES CON POSICION 7 OCUPADA: Este bloque de lo que se encarga es de mover la pieza entrante a través del horno inferior cuando la última posición de pieza del horno inferior esté ocupada y con lo cual no moverá la cola; para ello lo que se hace es mover el elemento hasta que se encuentre un plano sensor libre y el siguiente a ese ocupado. Los bloques que lo componen son los reflejados en la figura 84.

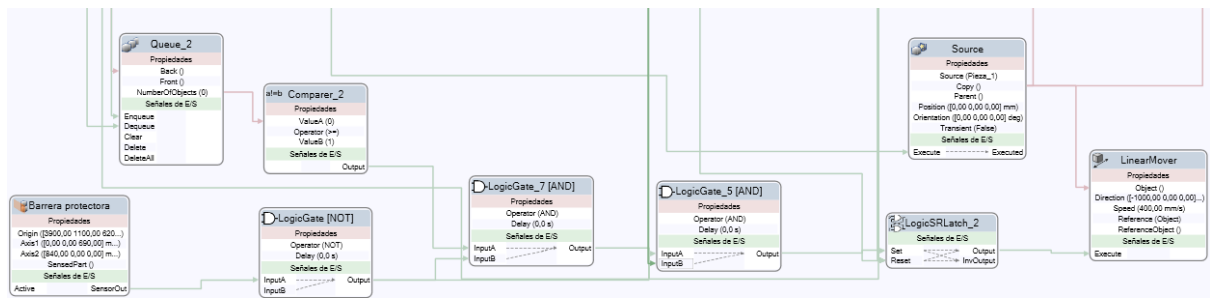


Figura 84

Una de las condiciones que tendrá que cumplir es que en la cola haya **1** o más elementos para que se ejecute esta acción, y lo que moveremos será el soporte creado conectado a la pieza a tratar. Aquí también tenemos una condición impuesta por la **LogicSRLatch_2** que tiene una condición de set, y es que generará un **Output = 1** cuando se produzca un cambio de 0 a 1 en la puerta lógica **AND 3**, y generará un **Output = 0** cuando cualquiera de las puertas **AND** explicadas en el bloque anterior cambie de 0 a 1, lo que significará que hay las posiciones siguientes están ocupadas por piezas y no avanzará más la cola.

Para entender mejor el funcionamiento, las conexiones de entradas y salidas digitales dentro del propio horno y la que tenemos de entrada (**diPiezaCogidaHorno**) y la de salida (**doPiezaListaHorno**) están reflejadas en la siguiente tabla junto con sus relaciones y conexiones entre ellas.



Conexiones de E/S

Objeto de origen	Señal de origen	Objeto de destino	Señal de destino
Barrera protectora	SensorOut	LogicGate [NOT]	InputA
PlaneSensor	SensorOut	LogicGate_2 [AND]	InputA
PlaneSensor	SensorOut	LogicGate_2 [AND]	InputB
LogicGate_2 [AND]	Output	Source	Execute
Posicion 1 Pieza	SensorOut	Attacher	Execute
PlaneSensor	SensorOut	LogicGate_3 [AND]	InputB
Posicion 1 Pieza	SensorOut	LogicGate_3 [AND]	InputA
LogicGate [NOT]	Output	LogicGate_4 [AND]	InputA
Posicion 6 Pieza	SensorOut	LogicGate_6 [AND]	InputB
LogicGate_6 [AND]	Output	LogicSRLatch	Reset
LogicGate_6 [AND]	Output	LogicGate_7 [AND]	InputB
Posicion 5 Pieza	SensorOut	LogicGate_7 [AND]	InputA
LogicGate_7 [AND]	Output	LogicSRLatch	Reset
LogicGate_7 [AND]	Output	LogicGate_11 [AND]	InputB
Posicion 4 Pieza	SensorOut	LogicGate_11 [AND]	InputA
LogicGate_11 [AND]	Output	LogicGate_10 [AND]	InputB
Posicion 3 Pieza	SensorOut	LogicGate_10 [AND]	InputA
LogicGate_10 [AND]	Output	LogicGate_12 [AND]	InputB
Posicion 2 Pieza	SensorOut	LogicGate_12 [AND]	InputA
LogicGate_12 [AND]	Output	LogicGate_13 [AND]	InputB
Posicion 1 Pieza	SensorOut	LogicGate_13 [AND]	InputA
LogicGate_11 [AND]	Output	LogicSRLatch	Reset
LogicGate_10 [AND]	Output	LogicSRLatch	Reset
LogicGate_12 [AND]	Output	LogicSRLatch	Reset
LogicGate_13 [AND]	Output	LogicSRLatch	Reset
LogicGate_3 [AND]	Output	LogicSRLatch	Set
LogicGate_14 [NOT]	Output	LogicSRLatch	Set
Posicion 1 Pieza	SensorOut	Queue_2	Enqueue
Posicion 7 Pieza	SensorOut	Queue_2	Dequeue
Posicion 7 Pieza	SensorOut	LogicGate_14 [NOT]	InputA
Posicion 7 Pieza	SensorOut	Componente inteligente_2	ElementoListo
Componente inteligente_2	ArribaOcupado	LogicGate_3 [AND]	InputB
Posicion 7 Pieza	SensorOut	LogicGate_3 [AND]	InputA
LogicGate_3 [AND]	Output	LogicGate_6 [AND]	InputA
LogicGate_3 [AND]	Output	LogicSRLatch	Reset
LogicGate_6 [AND]	Output	LinearMover_2	Execute
Comparer_2	Output	LogicGate_7 [AND]	InputA
LogicGate_14 [NOT]	Output	LogicGate_6 [AND]	InputA
LogicGate [NOT]	Output	LogicGate_7 [AND]	InputB
LogicGate_7 [AND]	Output	LogicSRLatch_2	Reset
LogicGate_11 [AND]	Output	LogicSRLatch_2	Reset
LogicGate_10 [AND]	Output	LogicSRLatch_2	Reset
LogicGate_12 [AND]	Output	LogicSRLatch_2	Reset
LogicGate_6 [AND]	Output	LogicSRLatch_2	Reset
LogicGate_14 [NOT]	Output	LogicGate_4 [AND]	InputB
LogicGate_4 [AND]	Output	LogicGate_6 [AND]	InputA
LogicSRLatch	Output	LogicGate_6 [AND]	InputB
LogicGate_6 [AND]	Output	LogicGate_6 [AND]	InputB
Posicion 7 Pieza	SensorOut	LogicSRLatch_2	Reset
LogicSRLatch_2	Output	LinearMover	Execute
LogicGate_3 [AND]	Output	LogicGate_5 [AND]	InputB
LogicGate_7 [AND]	Output	LogicGate_5 [AND]	InputA
LogicGate_5 [AND]	Output	LogicSRLatch_2	Set
Componente inteligente_2	ArribaOcupado	Componente inteligente_3	ListoArriba
Componente inteligente_3	doListoHornoPieza	Horno	doPiezaListaHorno
Horno	diPiezaCogidaHorno	Componente inteligente_3	diPiezaCogidaHorno

Para entender mejor la relación de propiedades entre bloques (flechas granates), es decir, que propiedades van de un bloque a otro, las conexiones están reflejadas en la siguiente tabla:

Enlazamientos de propiedad

Objeto de origen	Propiedad de origen	Objeto de destino	Propiedad de destino
Source	Copy	Posicion 1 Pieza	SensedPart
Source	Copy	Attacher	Parent
PlaneSensor	SensedPart	Attacher	Child
Posicion 1 Pieza	SensedPart	Queue_2	Back
Posicion 7 Pieza	SensedPart	Componente inteligente_2	ElementoASubir
Source	Copy	LinearMover	Object
Queue_2	NumberOfObjects	Comparer_2	ValueA
Componente inteligente_2	ElementoListo	Componente inteligente_3	ElementoCola



Como en el caso de la maquina 1, también tenemos componentes inteligentes de nivel inferior que en este caso, uno se encargará de desarrollar la función “ascensor”, es decir, subir las piezas a la parte superior del horno, y el otro componente inteligente se encargará de mover las piezas por el horno superior.

1. Componente inteligente ascensor

Los bloques que lo componen son los reflejados en la figura 86.

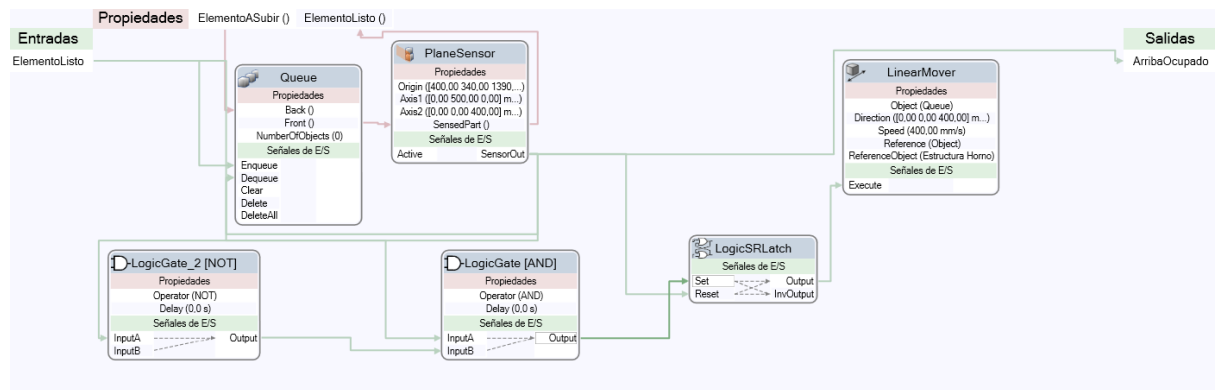


Figura 86

El funcionamiento general de este elemento será el de que cuando la entrada digital “ElementoListo” tenga el valor 1, colocará en la cola el elemento a subir, y esa cola será la que mueva hasta el nivel superior; cuando el elemento llegue al final del recorrido de subida, se desvinculará ese elemento de la cola.

BLOQUE CREADOR DE COLA: Este bloque creará la cola cuando la entrada digital “ElementoListo” tenga el valor 1, y el elemento a añadir en la cola será “ElementoASubir”, y eliminará el elemento de la cola cuando el elemento haya llegado hasta el plano sensor de la parte superior del horno. Los bloques que lo componen son los reflejados en la figura 87.

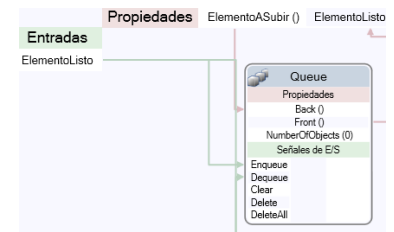


Figura 87

En la siguiente imagen del horno podemos observar la situación del sensor que influirá en la tarea ascensor. La geometría es la reflejada en la figura 88.

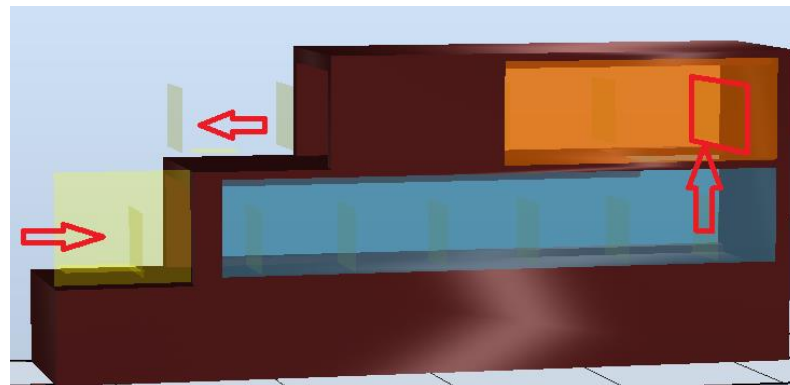


Figura 88

BLOQUE MOVIMIENTO DE COLA: Este bloque moverá la cola desde el horno inferior hasta el horno superior cuando el Output del LogicSRLatch tenga el valor 1 y esto se cumplirá siempre y cuando la salida de la puerta AND cambie de 0 a 1 porque se han cumplido las condiciones de que el sensor de plano indicado anteriormente no está siendo cortado por nada y además la entrada digital “ElementoListo” tiene el valor 1. Dejará de mover la cola cuando el sensor de plano sea cortado. Los bloques que lo componen son los reflejados en la figura 89.

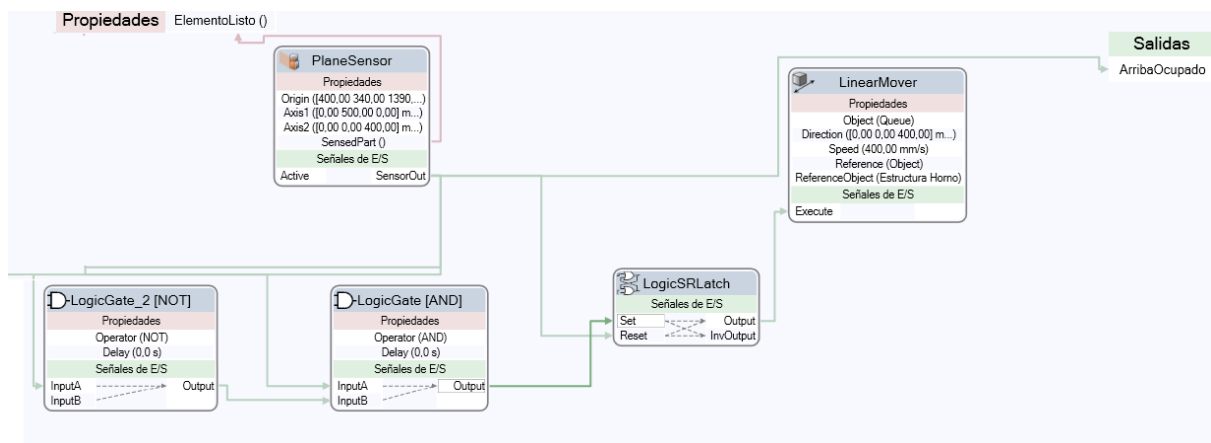


Figura 89

Para entender mejor la relación de entradas y salidas entre bloques (flechas verdes), es decir, que salidas corresponden con que entradas, las conexiones están reflejadas en la siguiente tabla:



Conexiones de E/S

Objeto de origen	Señal de origen	Objeto de destino	Señal de destino
PlaneSensor	SensorOut	Queue	Dequeue
Componente inteligente_2	ElementoListo	Queue	Enqueue
LogicSRLatch	Output	LinearMover	Execute
PlaneSensor	SensorOut	LogicSRLatch	Reset
PlaneSensor	SensorOut	LogicGate_2 [NOT]	InputA
LogicGate_2 [NOT]	Output	LogicGate [AND]	InputB
Componente inteligente_2	ElementoListo	LogicGate [AND]	InputA
LogicGate [AND]	Output	LogicSRLatch	Set
PlaneSensor	SensorOut	Componente inteligente_2	ArribaOcupado

Para entender mejor la relación de propiedades entre bloques (flechas granates), es decir, que propiedades van de un bloque a otro, las conexiones están reflejadas en la siguiente tabla:

Enlazamientos de propiedad

Objeto de origen	Propiedad de origen	Objeto de destino	Propiedad de destino
Queue	Front	PlaneSensor	SensedPart
PlaneSensor	SensedPart	Componente inteligente_2	ElementoListo
Componente inteligente_2	ElementoASubir	Queue	Back

2. Componente inteligente horno superior

Este se encargará de desplazar las piezas desde el fondo del horno hasta la salida de piezas en el horno superior. Para ello tenemos un esquema de funcionamiento reflejado en la figura 90.

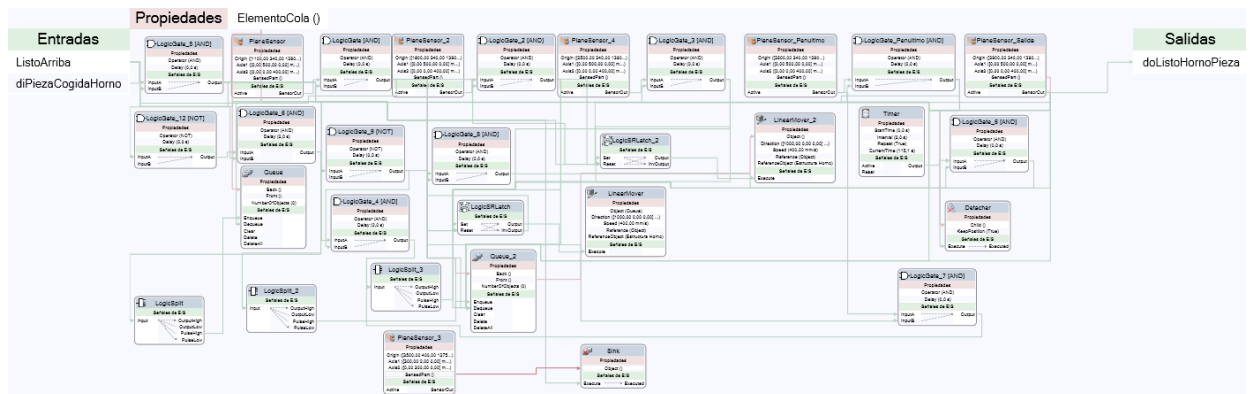


Figura 90

La idea de funcionamiento del nivel superior es la misma que la que se tiene en el nivel inferior del horno; una serie de sensores enlazados a través de puertas AND para frenar el movimiento si las posiciones siguientes están ocupadas.

Los planos que tendremos en este componente inteligente están situados tal y como se especifica en la figura 91.

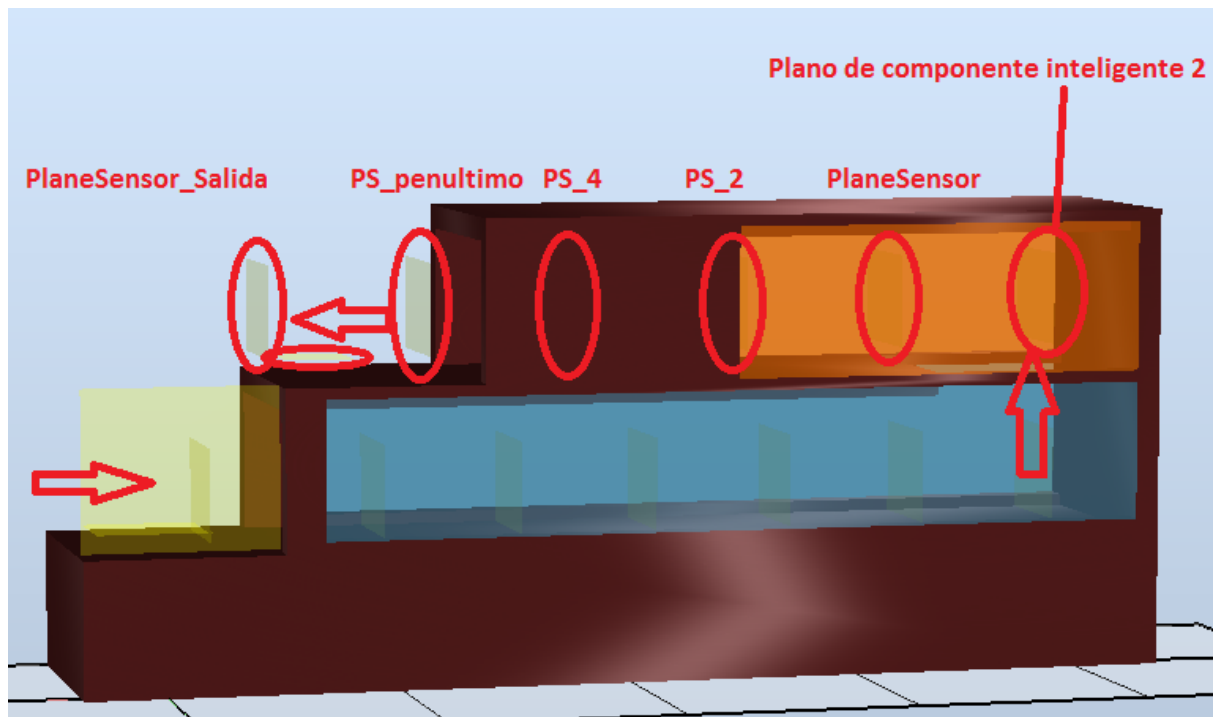


Figura 91

BLOQUE CREADOR DE COLA: Este bloque añadirá un elemento a la cola cuando la entrada digital “ListoArriba” tenga el valor 1, y que no estén ocupadas las primeras posiciones del fondo del horno. Los bloques que lo componen son los reflejados en la figura 92.

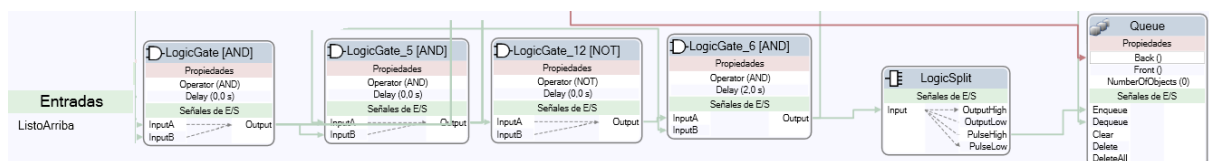


Figura 92

BLOQUE PIEZAS EN HORNO SUPERIOR: Este bloque de lo que se encarga es de comprobar que la posición siguiente a la que podría avanzar la pieza junto con su soporte está libre. Esto lo haremos mediante una serie de planos sensores y puertas AND debidamente conectados entre sí mediante las señales digitales. Los bloques que lo componen son los reflejados en la figura 93.

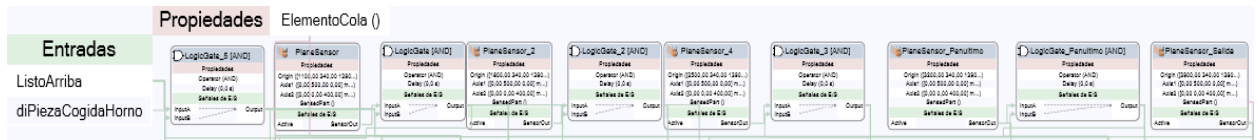
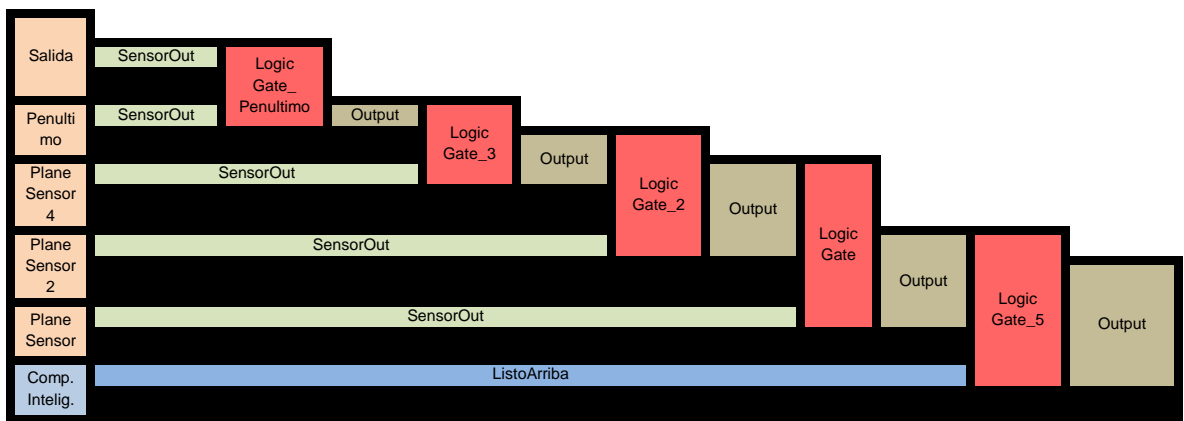


Figura 93

Las puertas AND nos servirán para indicar si las posiciones siguientes están ocupadas.

Para entender mejor el funcionamiento de este bloque vamos a ver en una tabla las salidas y entras de nuestros sensores y de nuestras puertas lógicas (todas ellas AND).



Las salidas (Output) de las puertas lógicas AND (LogicGate_X) serán las que frenarán el movimiento de las piezas (obviamente debidamente conectadas) por el horno inferior si las posiciones siguientes están ocupadas.

BLOQUE MOVIMIENTO PIEZAS EN HORNO SUPERIOR: Este bloque de lo que se encarga es de mover las piezas por el horno superior siempre y cuando se cumplan una serie de condiciones. Los bloques que lo componen son los reflejados en la figura 94.

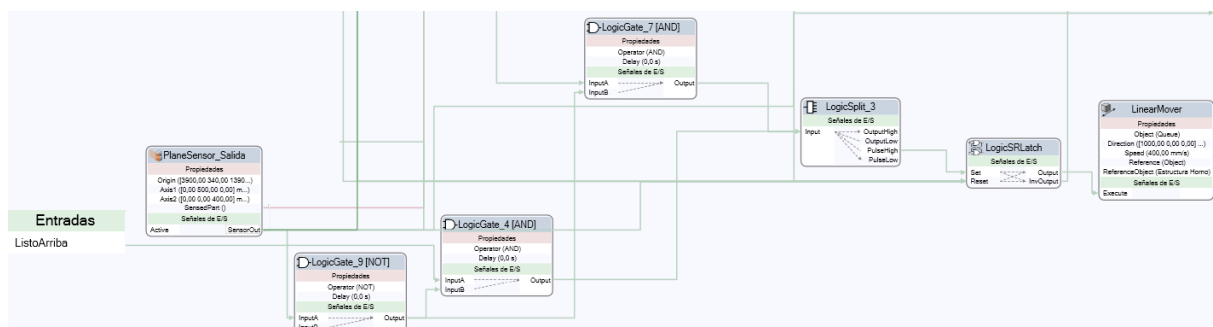


Figura 94



Hay dos condiciones que pueden activar el movimiento de la cola de piezas del horno superior. Una de ellas es la entrada digital “ListoArriba” tenga el valor 1 y que no haya ninguna pieza en la posición del sensor de plano de salida, y la otra condición es que la penúltima posición donde pueden estar las piezas este ocupada y que no haya ninguna pieza en la posición del sensor de plano de salida.

La forma de frenar el movimiento de las piezas es de la misma forma que se hacía en el horno inferior, un cambio en las salidas de las puertas lógicas AND de 0 a 1, reseteará el LogicSRLatch y frenará el movimiento.

BLOQUE MOVIMIENTO PIEZAS CON ULTIMAS POSICIONES OCUPADAS:

Este bloque de lo que se encarga es de mover piezas de manera individual por el horno superior si la posición de salida y la penúltima posición están ocupadas, ya que si moviese la cola entera desplazaría piezas ya debidamente posicionadas en el interior del horno. Los bloques que lo componen son los reflejados en la figura 95.

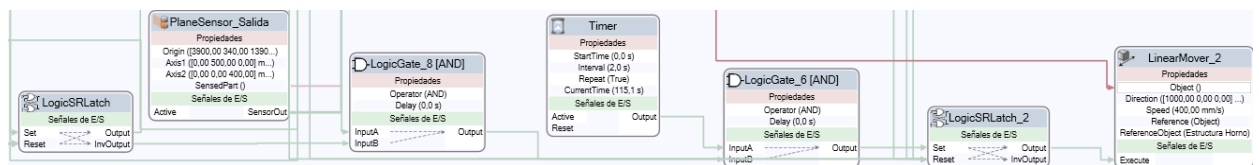


Figura 95

Para que el movimiento se produzca, cada 2 segundos un temporizador envía una señal de 1 para la puerta lógica AND, cuya otra condición que se debe de cumplir es que la salida esté ocupada, y no esté produciéndose un movimiento de la cola.

BLOQUE DESCONECTOR Y ELIMINACIÓN DEL SOPORTE:

Este bloque desconectará el soporte de la pieza, y cuando el robot se haya llevado la pieza, eliminará gráficamente el soporte para dejar la posición de salida libre.

Desconectará la pieza del soporte cuando el plano sensor salida se active, y eliminará el soporte cuando se active la entrada diPiezaCogidaHorno. Los bloques que lo componen son los reflejados en la figura 96.

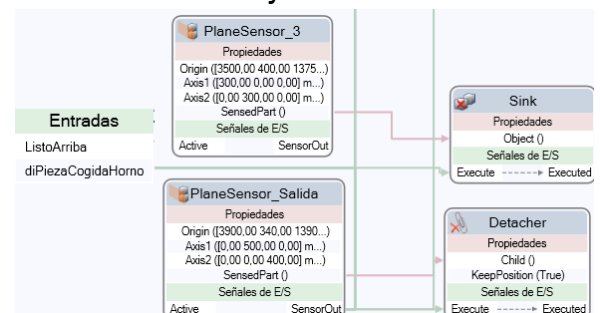


Figura 96



Para entender mejor la relación de entradas y salidas entre bloques (flechas verdes), es decir, que salidas corresponden con que entradas, las conexiones están reflejadas en la siguiente tabla:

Conexiones de E/S

Objeto de origen	Señal de origen	Objeto de destino	Señal de destino
LogicSRLatch	Output	LinearMover	Execute
LogicGate_6 [AND]	Output	LogicSplit	Input
LogicSplit	PulseHigh	Queue	Enqueue
PlaneSensor_Salida	SensorOut	LogicGate_Penultimo [AND]	InputB
PlaneSensor_2	SensorOut	LogicGate_2 [AND]	InputA
LogicGate_2 [AND]	Output	LogicGate [AND]	InputB
PlaneSensor	SensorOut	LogicGate [AND]	InputA
PlaneSensor_Penultimo	SensorOut	LogicGate_Penultimo [AND]	InputA
Componente inteligente_3	ListoArriba	LogicGate_5 [AND]	InputA
LogicGate [AND]	Output	LogicGate_5 [AND]	InputB
LogicSRLatch_2	Output	LinearMover_2	Execute
PlaneSensor_Salida	SensorOut	LogicGate_8 [AND]	InputA
LogicSRLatch	InvOutput	LogicGate_8 [AND]	InputB
LogicGate_2 [AND]	Output	LogicSRLatch_2	Reset
LogicGate [AND]	Output	LogicSRLatch_2	Reset
LogicGate_5 [AND]	Output	LogicSRLatch_2	Reset
Componente inteligente_3	ListoArriba	LogicGate_6 [AND]	InputB
PlaneSensor_Salida	SensorOut	Queue	Dequeue
LogicGate_6 [AND]	Output	LogicSplit_2	Input
LogicSplit_2	PulseHigh	Queue_2	Enqueue
LogicGate_Penultimo [AND]	Output	Queue_2	Dequeue
LogicGate_2 [AND]	Output	Queue_2	Dequeue
LogicGate [AND]	Output	Queue_2	Dequeue
LogicGate_5 [AND]	Output	Queue_2	Dequeue
PlaneSensor_Salida	SensorOut	Queue_2	Dequeue
Timer	Output	LogicGate_6 [AND]	InputA
LogicGate_8 [AND]	Output	LogicGate_6 [AND]	InputB
LogicGate_6 [AND]	Output	LogicSRLatch_2	Set
LogicGate_5 [AND]	Output	LogicGate_12 [NOT]	InputA
LogicGate_12 [NOT]	Output	LogicGate_6 [AND]	InputA
PlaneSensor_4	SensorOut	LogicGate_3 [AND]	InputA
LogicGate_3 [AND]	Output	LogicGate_2 [AND]	InputB
LogicGate_Penultimo [AND]	Output	LogicGate_3 [AND]	InputB
LogicGate_3 [AND]	Output	LogicSRLatch_2	Reset
LogicGate_3 [AND]	Output	Queue_2	Dequeue
LogicGate_Penultimo [AND]	Output	LogicSRLatch_2	Reset
LogicSplit_3	PulseHigh	LogicSRLatch	Set
PlaneSensor_Salida	SensorOut	LogicSRLatch	Reset
LogicGate_Penultimo [AND]	Output	LogicSRLatch	Reset
LogicGate_3 [AND]	Output	LogicSRLatch	Reset
LogicGate_2 [AND]	Output	LogicSRLatch	Reset
LogicGate [AND]	Output	LogicSRLatch	Reset
LogicGate_5 [AND]	Output	LogicSRLatch	Reset
PlaneSensor_Salida	SensorOut	Detach	Execute
PlaneSensor_Salida	SensorOut	LogicGate_9 [NOT]	InputA
LogicGate_9 [NOT]	Output	LogicGate_4 [AND]	InputB
Componente inteligente_3	ListoArriba	LogicGate_4 [AND]	InputA
LogicGate_9 [NOT]	Output	LogicGate_7 [AND]	InputB
PlaneSensor_Penultimo	SensorOut	LogicGate_7 [AND]	InputA
LogicGate_7 [AND]	Output	LogicSplit_3	Input
LogicGate_4 [AND]	Output	LogicSplit_3	Input
PlaneSensor_Salida	SensorOut	Componente inteligente_3	doListoHornoPieza
Componente inteligente_3	diPiezaCogidaHorno	Sink	Execute

Para entender mejor la relación de propiedades entre bloques (flechas granates), es decir, que propiedades van de un bloque a otro, las conexiones están reflejadas en la siguiente tabla:



Enlazamientos de propiedad

Objeto de origen	Propiedad de origen	Objeto de destino	Propiedad de destino
PlaneSensor_Salida	SensedPart	Detacher	Child
Componente inteligente_3	ElementoCola	Queue	Back
Componente inteligente_3	ElementoCola	Queue_2	Back
Queue_2	Front	LinearMover_2	Object
PlaneSensor_3	SensedPart	Sink	Object

Configuración de la maquina 3: Tratamiento especial

La geometría es la reflejada en la figura 97.

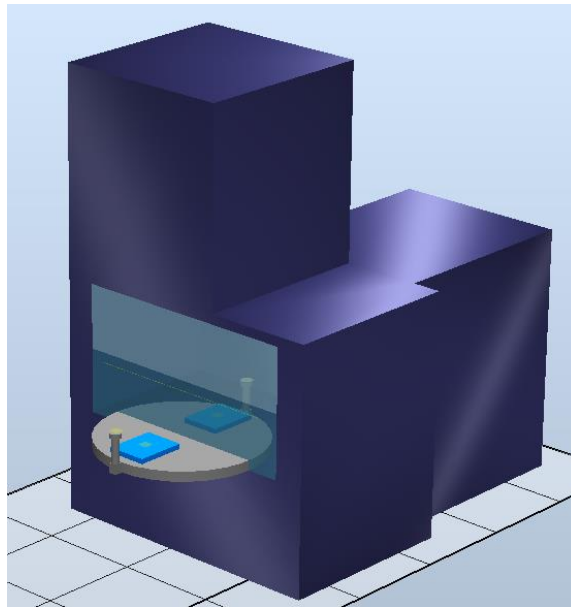


Figura 97

Este es un componente inteligente también complejo ya que está diseñado para que podamos situar una pieza en la entrada, la máscara de entrada en el soporte de máscaras y que cuando todo esto está realizado la mesa gire para dejar la pieza que anteriormente ha trabajado junto con la máscara sucia disponible para que el robot pueda recogerlo y realizar la siguiente operación (de control o salida). El ejemplo visual de funcionamiento sería el reflejado en la figura 98:

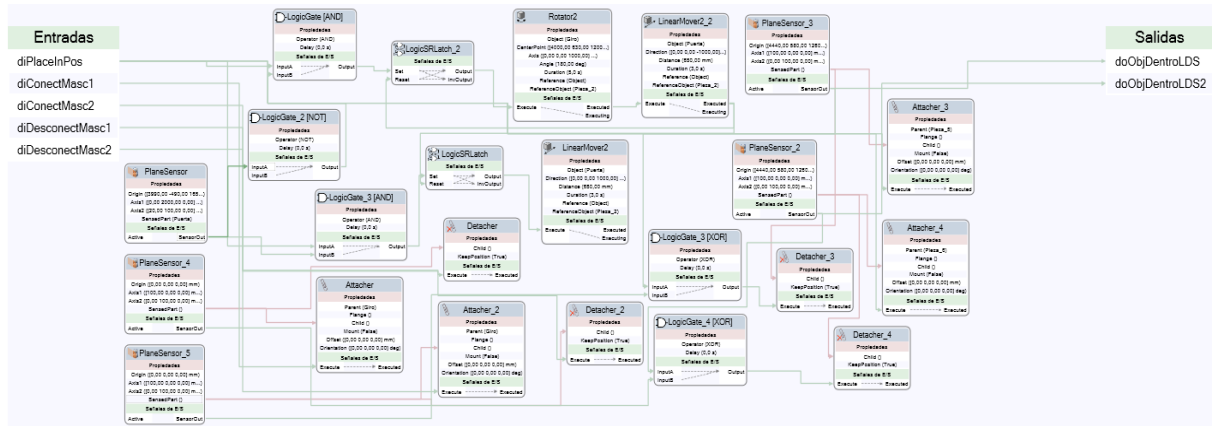


Figura 98

BLOQUE CONECTOR DE PIEZA CON SOPORTE: Este bloque conectará la pieza dejada en el soporte al soporte, para que cuando la mesa gire, la pieza gire también. La conexión se hará gracias a unos sensores de plano que detectarán la pieza que hay que conectar, y a través de señales digitales de entrada se activará la función de conectar. Los bloques que lo componen son los reflejados en la figura 99.

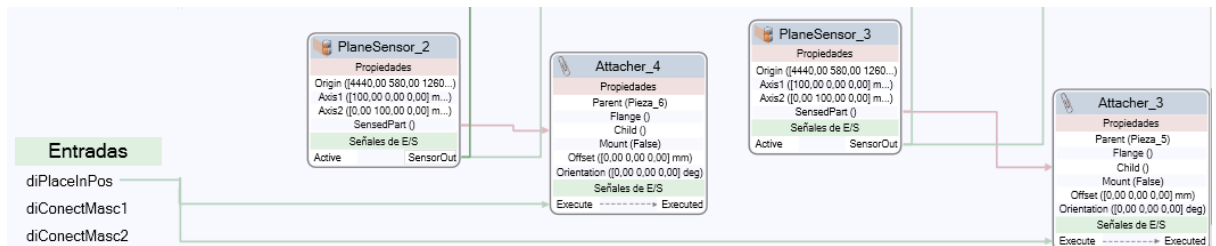


Figura 99

BLOQUE CONECTOR DE MASCARAS CON SOPORTE: Este bloque conectará las máscaras dejadas en el soporte al soporte, para que cuando la mesa gire, las máscaras giren también. La conexión se hará gracias a unos sensores de plano que detectarán la máscara que hay que conectar, y a través de señales digitales de entrada se activará la función de conectar. Los bloques que lo componen son los reflejados en la figura 100.

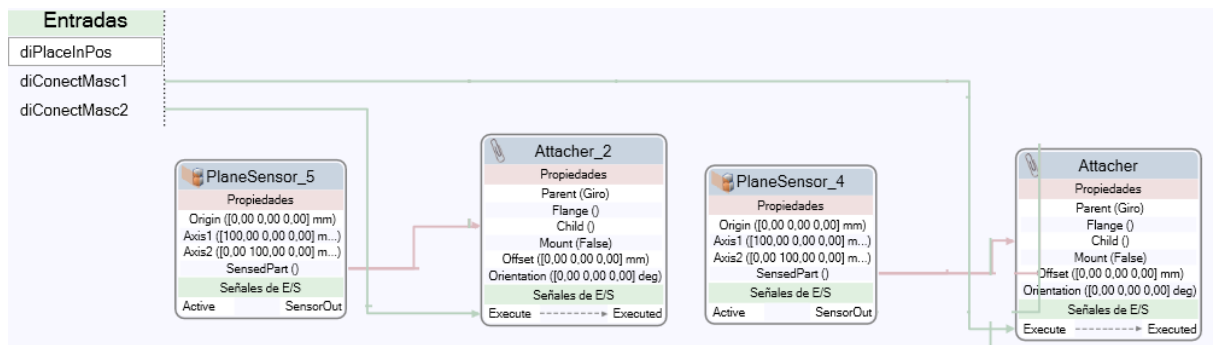


Figura 100

BLOQUE APERTURA PUERTA: Este bloque abre la puerta de la maquina 3 cuando recibe la señal de que el robot ha situado la pieza en el soporte de entrada y que además comprueba si la puerta está cerrada; cumpliendo estas dos condiciones, abrirá la puerta para posteriormente permitir el giro de la mesa. Los bloques que lo componen son los reflejados en la figura 101.

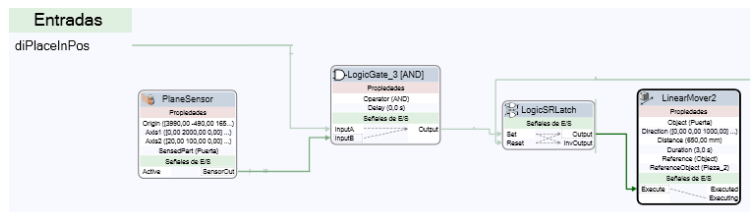


Figura 101

BLOQUE GIRADOR DE MESA Y CIERRE DE PUERTA: Este bloque se encarga de realizar el giro de la mesa siempre y cuando la señal “diPlacelnPos” tenga el valor 1, y la puerta esté abierta. Una vez ejecutado el giro de la mesa, la puerta se cerrará. Los bloques que lo componen son los reflejados en la figura 102.

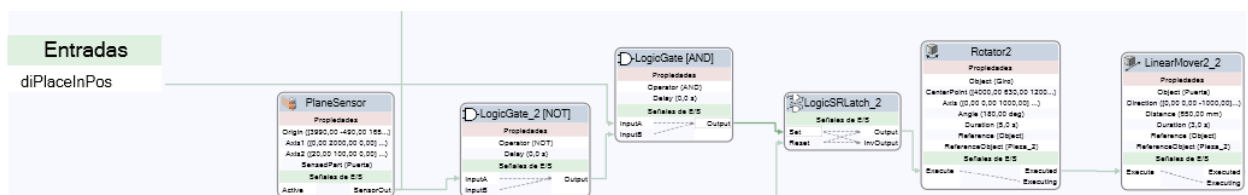


Figura 102



BLOQUE DESCONECTOR DE MASCARA CON SOPORTE: Este bloque desconectará la máscara ya usada del soporte, para que el robot pueda coger la sucia y llevársela a la salida de máscaras. Esto se producirá cuando se active la entrada digital “diDesconnectMasc(1 o 2)”. Los bloques que lo componen son los reflejados en la figura 103.

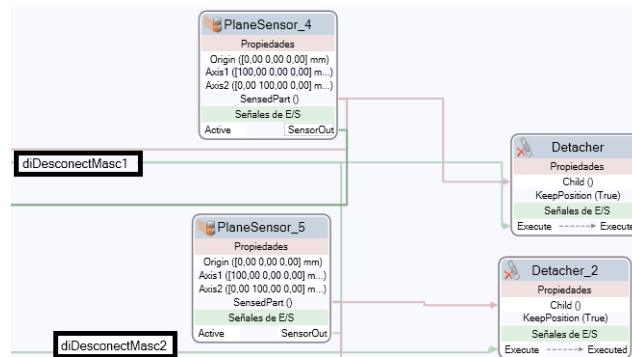


Figura 103

BLOQUE DESCONECTOR DE PIEZA CON SOPORTE: Este bloque desconectará la pieza dejada en el soporte del soporte, para que el robot pueda llevarse la pieza ya tratada sin que siga vinculada al soporte. Esta desconexión se producirá gracias a los planos sensores de máscaras y de posición de pieza relacionados mediante una puerta XOR. Los bloques que lo componen son los reflejados en la figura 104.

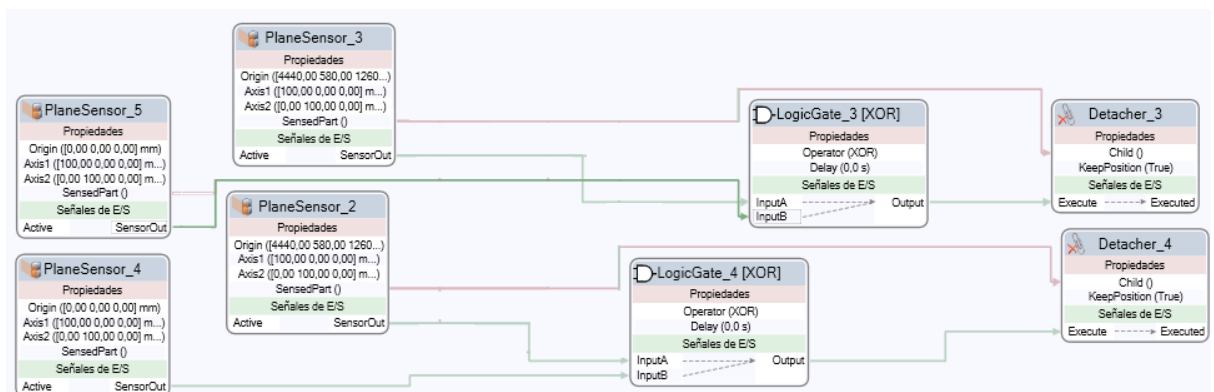


Figura 104

Para entender mejor la relación de entradas y salidas entre bloques (flechas verdes), es decir, que salidas corresponden con que entradas, las conexiones están reflejadas en la siguiente tabla:



Conexiones de E/S

Objeto de origen	Señal de origen	Objeto de destino	Señal de destino
LDS	diPlacelnPos	LogicGate [AND]	InputA
PlaneSensor	SensorOut	LogicGate_2 [NOT]	InputA
LogicGate_2 [NOT]	Output	LogicGate [AND]	InputB
Rotator2	Executed	LinearMover2_2	Execute
PlaneSensor	SensorOut	LogicGate_3 [AND]	InputB
LDS	diPlacelnPos	LogicGate_3 [AND]	InputA
LogicGate_3 [AND]	Output	LogicSRLatch	Set
LogicSRLatch	Output	LinearMover2	Execute
LogicGate [AND]	Output	LogicSRLatch_2	Set
LogicSRLatch_2	Output	Rotator2	Execute
LinearMover2_2	Executed	LogicSRLatch	Reset
LinearMover2_2	Executed	LogicSRLatch_2	Reset
PlaneSensor_3	SensorOut	LDS	doObjDentroLDS
PlaneSensor_2	SensorOut	LDS	doObjDentroLDS2
LDS	diConectMasc1	Attacher	Execute
LDS	diConectMasc2	Attacher_2	Execute
LDS	diDesconnectMasc2	Detacher	Execute
LDS	diDesconnectMasc1	Detacher_2	Execute
LDS	diPlacelnPos	Attacher_3	Execute
LDS	diPlacelnPos	Attacher_4	Execute
PlaneSensor_3	SensorOut	LogicGate_3 [XOR]	InputA
PlaneSensor_2	SensorOut	LogicGate_4 [XOR]	InputA
PlaneSensor_4	SensorOut	LogicGate_4 [XOR]	InputB
PlaneSensor_5	SensorOut	LogicGate_3 [XOR]	InputB
LogicGate_3 [XOR]	Output	Detacher_3	Execute
LogicGate_4 [XOR]	Output	Detacher_4	Execute

Para entender mejor la relación de propiedades entre bloques (flechas granas), es decir, que propiedades van de un bloque a otro, las conexiones están reflejadas en la siguiente tabla:

Enlazamientos de propiedad

Objeto de origen	Propiedad de origen	Objeto de destino	Propiedad de destino
PlaneSensor_4	SensedPart	Attacher	Child
PlaneSensor_5	SensedPart	Attacher_2	Child
PlaneSensor_4	SensedPart	Detacher	Child
PlaneSensor_5	SensedPart	Detacher_2	Child
PlaneSensor_3	SensedPart	Attacher_3	Child
PlaneSensor_2	SensedPart	Attacher_4	Child
PlaneSensor_2	SensedPart	Detacher_4	Child
PlaneSensor_3	SensedPart	Detacher_3	Child

Configuración de la salida de control:

La geometría es la reflejada en la figura 105.

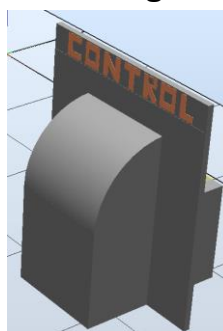


Figura 105

También configurada como un componente inteligente debido a que debemos de ser capaces de variar el número de veces y el momento en el que se produce la salida a control (actualmente configurada cada 300



segundos), y que tenga una compuerta que nos permita tener cierta seguridad para la salida de piezas. El esquema de bloques para su funcionamiento es el reflejado en la figura 106:

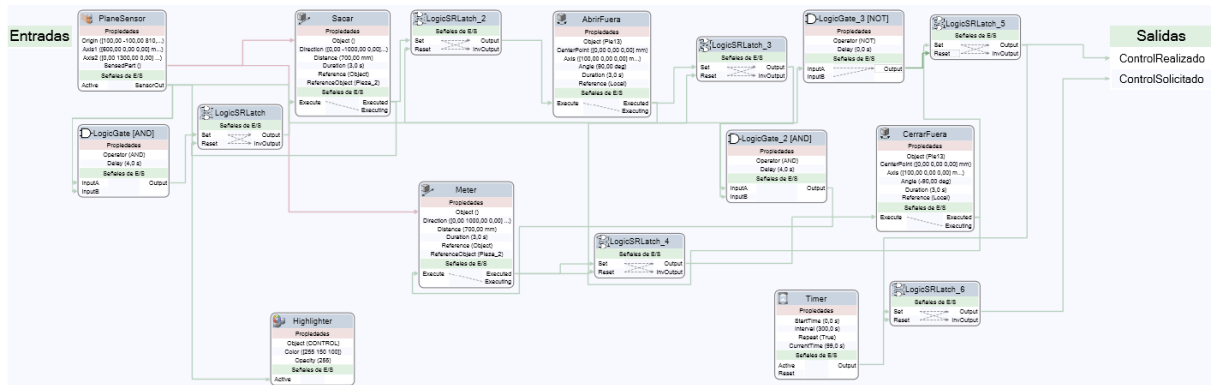


Figura 106

El propio componente inteligente de control genera cada 300 segundos una señal llamada “ControlSolicitado”, y una vez realizado el control se generará otra señal de salida llamada “ControlRealizado”.

BLOQUE LUZ LUMINOSA DE CONTROL: Este bloque sirve para que en cuanto el sensor de plano detecta que se ha dejado una pieza en control, se nos ilumine una señal luminosa de CONTROL. Los bloques que lo componen son los reflejados en la figura 107.

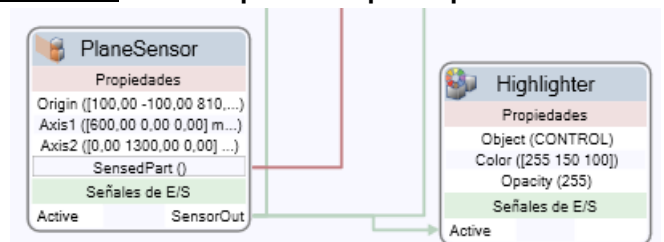


Figura 107

BLOQUE SALIDA DE PIEZA Y APERTURA DE PUERTA: Este bloque nos moverá la pieza hasta una posición adecuada para la salida de la misma de la célula, y posteriormente nos abrirá la puerta para que el operario pueda realizar con ella las operaciones las realiza de manera secuencial. Los bloques que lo componen son los reflejados en la figura 108.

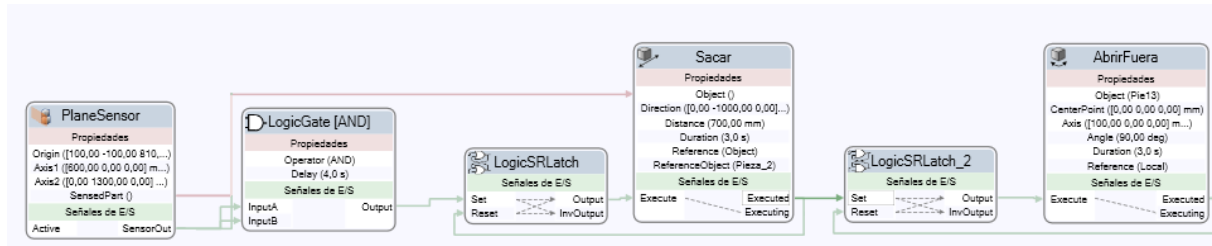


Figura 108

BLOQUE RETORNO DE PIEZA Y CIERRE DE PUERTA: Este bloque nos moverá la pieza hasta llegar otra vez dentro de la célula y simultáneamente nos cerrará la puerta. Estas dos operaciones las realiza de manera simultánea. Los bloques que lo componen son los reflejados en la figura 109.

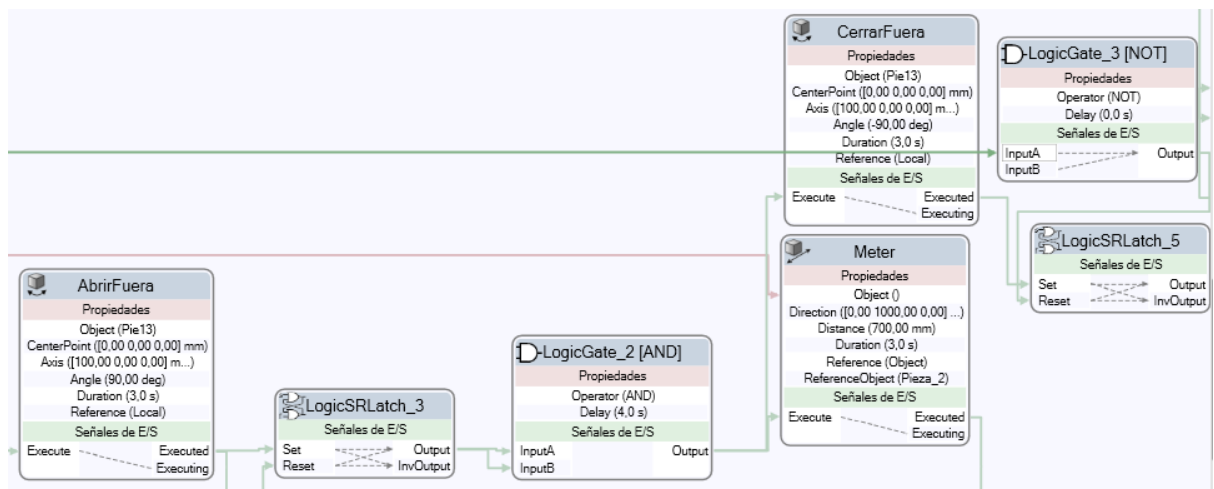


Figura 109

BLOQUE DE SOLICITUD DE CONTROL: Este bloque cada 30 segundos nos generará una salida “ControlSolicitado” que, gracias a la programación en RAPID, hará que la próxima pieza saliente de la maquina 3, pase por control. Los bloques que lo componen son los reflejados en la figura 110.

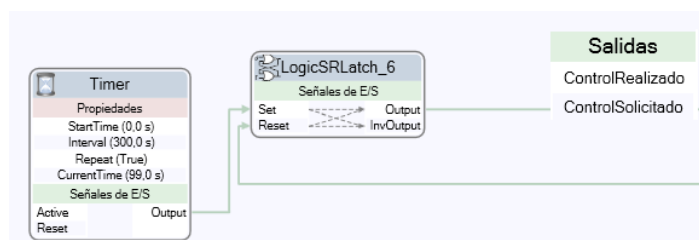


Figura 110



Para entender mejor la relación de entradas y salidas entre bloques (flechas verdes), es decir, que salidas corresponden con que entradas, las conexiones están reflejadas en la siguiente tabla:

Conexiones de E/S

Objeto de origen	Señal de origen	Objeto de destino	Señal de destino
LogicSRLatch	Output	Sacar	Execute
Sacar	Executed	LogicSRLatch	Reset
Sacar	Executed	LogicSRLatch_2	Set
LogicSRLatch_2	Output	AbrirFuera	Execute
AbrirFuera	Executed	LogicSRLatch_2	Reset
PlaneSensor	SensorOut	LogicGate [AND]	InputA
PlaneSensor	SensorOut	LogicGate [AND]	InputB
LogicGate [AND]	Output	LogicSRLatch	Set
AbrirFuera	Executed	LogicSRLatch_3	Set
LogicSRLatch_3	Output	LogicGate_2 [AND]	InputA
LogicSRLatch_3	Output	LogicGate_2 [AND]	InputB
LogicGate_2 [AND]	Output	Meter	Execute
Meter	Executed	LogicSRLatch_4	Set
Meter	Executed	LogicSRLatch_3	Reset
LogicSRLatch_4	Output	CerrarFuera	Execute
CerrarFuera	Executed	LogicSRLatch_4	Reset
PlaneSensor	SensorOut	Highlighter	Active
CerrarFuera	Executed	LogicSRLatch_5	Set
PlaneSensor	SensorOut	LogicGate_3 [NOT]	InputA
LogicGate_3 [NOT]	Output	LogicSRLatch_5	Reset
LogicSRLatch_5	Output	Control	ControlRealizado
Timer	Output	LogicSRLatch_6	Set
LogicSRLatch_6	Output	Control	ControlSolicitado
LogicSRLatch_5	Output	LogicSRLatch_6	Reset

Para entender mejor la relación de propiedades entre bloques (flechas granates), es decir, que propiedades van de un bloque a otro, las conexiones están reflejadas en la siguiente tabla:

Enlazamientos de propiedad

Objeto de origen	Propiedad de origen	Objeto de destino	Propiedad de destino
PlaneSensor	SensedPart	Sacar	Object
PlaneSensor	SensedPart	Meter	Object



Trabajo Fin de Grado

Autor: Gonzalo Muínelo Garrido

Grado en Ingeniería en
Electrónica Industrial y
Automática

Fecha: 2015





6. Conexión y comunicación Robot-Elementos.

Una vez explicados todos los elementos de la célula robotizada en cuanto a su función, funcionamiento y señales digitales, es importante saber que todos los elementos deben de estar perfectamente sincronizados entre sí para un correcto funcionamiento de la célula.

La figura 111 nos da una idea global de cómo se relacionan los elementos de la célula con el robot mediante los inputs y outputs de cada elemento.

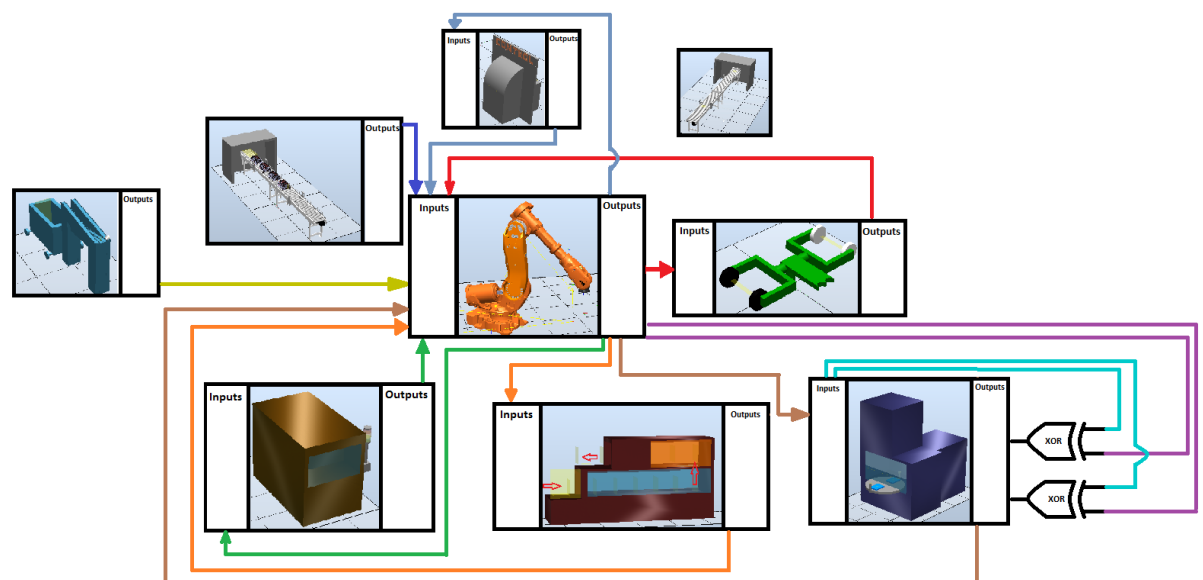


Figura 111

Cada flecha corresponde a un conjunto de señales digitales de entrada y salida de cada elemento; en la mayoría de los casos interactúan enviando y recibiendo información con el robot, aunque algunos (como la cola de entrada) solo envían información al robot y no reciben, o (como la cola de salida) que no interactúa en cuanto al envío de información con el robot.

Para ser más concretos, en el siguiente esquema de la figura 112 se pueden observar todos los elementos interactuando entre sí, a través de diferentes entradas y salidas.

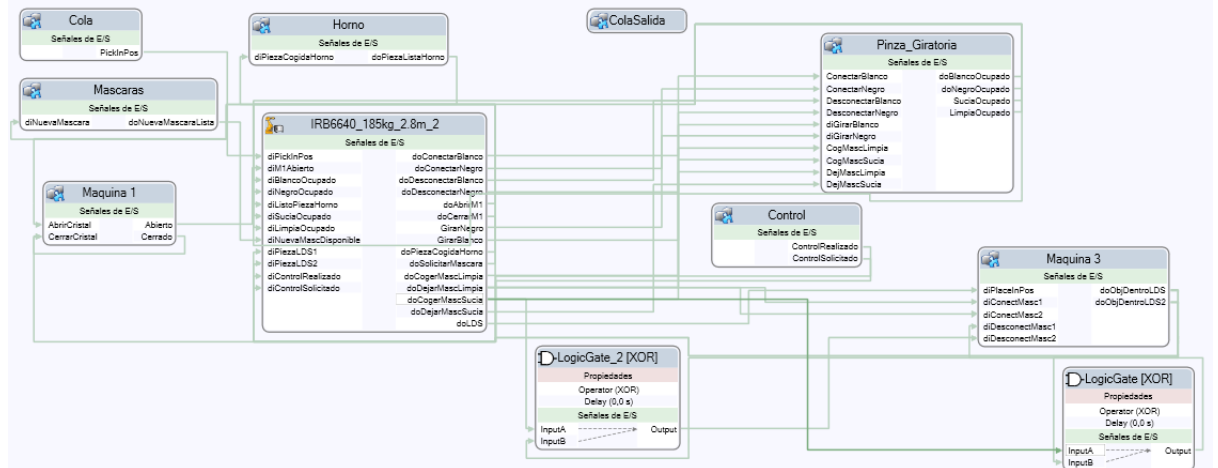


Figura 112

En la siguiente tabla se recogen que entrada corresponde con que salida de una manera mucho más clara que en el esquema anterior.

Conexiones de E/S

Objeto de origen	Señal de origen	Objeto de destino	Señal de destino
Cola	PickInPos	IRB6640_185kg_2.8m_2	diPickInPos
IRB6640_185kg_2.8m_2	doConectarBlanco	Pinza_Giratoria	ConectarBlanco
IRB6640_185kg_2.8m_2	doConectarNegro	Pinza_Giratoria	ConectarNegro
IRB6640_185kg_2.8m_2	doDesconectarBlanco	Pinza_Giratoria	DesconectarBlanco
IRB6640_185kg_2.8m_2	doDesconectarNegro	Pinza_Giratoria	DesconectarNegro
IRB6640_185kg_2.8m_2	doAbrirM1	Maquina 1	AbrirCristal
IRB6640_185kg_2.8m_2	doCerrarM1	Maquina 1	CerrarCristal
Maquina 1	Abierto	IRB6640_185kg_2.8m_2	diM1Abierto
Maquina 1	Cerrado	Maquina 1	CerrarCristal
Pinza_Giratoria	doBlancoOcupado	IRB6640_185kg_2.8m_2	diBlancoOcupado
Pinza_Giratoria	doNegroOcupado	IRB6640_185kg_2.8m_2	diNegroOcupado
IRB6640_185kg_2.8m_2	GirarNegro	Pinza_Giratoria	diGirarNegro
IRB6640_185kg_2.8m_2	GirarBlanco	Pinza_Giratoria	diGirarBlanco
Horno	doPiezaListaHorno	IRB6640_185kg_2.8m_2	diListaPiezaHorno
IRB6640_185kg_2.8m_2	doPiezaCogidaHorno	Horno	diPiezaCogidaHorno
IRB6640_185kg_2.8m_2	doSolicitarMascara	Mascaras	diNuevaMascara
IRB6640_185kg_2.8m_2	doCogerMascLimpia	Pinza_Giratoria	CogMascLimpia
IRB6640_185kg_2.8m_2	doDejarMascLimpia	Pinza_Giratoria	DejMascLimpia
IRB6640_185kg_2.8m_2	doCogerMascSucia	Pinza_Giratoria	CogMascSucia
IRB6640_185kg_2.8m_2	doDejarMascSucia	Pinza_Giratoria	DejMascSucia
Pinza_Giratoria	SuciaOcupado	IRB6640_185kg_2.8m_2	diSuciaOcupado
Pinza_Giratoria	LimpiaOcupado	IRB6640_185kg_2.8m_2	diLimpiaOcupado
Mascaras	doNuevaMascaraLista	IRB6640_185kg_2.8m_2	diNuevaMascDisponible
IRB6640_185kg_2.8m_2	doDejarMascLimpia	Maquina 3	diConectMasc1
IRB6640_185kg_2.8m_2	doDejarMascLimpia	Maquina 3	diConectMasc2
IRB6640_185kg_2.8m_2	doLDS	Maquina 3	diPlacelnPos
Maquina 3	doObjDentroLDS	IRB6640_185kg_2.8m_2	diPiezaLDS1
Maquina 3	doObjDentroLDS2	IRB6640_185kg_2.8m_2	diPiezaLDS2
IRB6640_185kg_2.8m_2	doCogerMascSucia	LogicGate [XOR]	InputA
IRB6640_185kg_2.8m_2	doCogerMascSucia	LogicGate_2 [XOR]	InputA
Maquina 3	doObjDentroLDS	LogicGate [XOR]	InputB
Maquina 3	doObjDentroLDS2	LogicGate_2 [XOR]	InputB
LogicGate [XOR]	Output	Maquina 3	diDesconnectMasc1
LogicGate_2 [XOR]	Output	Maquina 3	diDesconnectMasc2
Control	ControlRealizado	IRB6640_185kg_2.8m_2	diControlRealizado
Control	ControlSolicitado	IRB6640_185kg_2.8m_2	diControlSolicitado

En cuanto al Robot de nuestra célula, se ha definido una tarjeta virtual de entradas y de salidas llamada ABB.



Trabajo Fin de Grado

Autor: Gonzalo Muinelo Garrido

Grado en Ingeniería en
Electrónica Industrial y
Automática

Fecha: 2015



Name	Type of Signal	Assigned to Unit	Unit Mapping	Access Level	Signal Value at System Failure and Power Fail	Default Value
doCogerMascLimpia	Digital Output	ABB	14	All	Keep Current Value (no change)	0
doCerrarM1	Digital Output	ABB	8	All	Keep Current Value (no change)	0
doConectarBlanco	Digital Output	ABB	1	All	Keep Current Value (no change)	0
doCogerMascSucia	Digital Output	ABB	16	All	Keep Current Value (no change)	0
diPiezaLDS2	Digital Input	ABB	25	All	N/D	0
diPiezaLDS1	Digital Input	ABB	24	All	N/D	0
doAbrirM1	Digital Output	ABB	7	All	Keep Current Value (no change)	0
diSuciaOcupado	Digital Input	ABB	18	All	N/D	0
doLDS	Digital Output	ABB	23	All	Keep Current Value (no change)	0
doDesconectarNegro	Digital Output	ABB	4	All	Keep Current Value (no change)	0
doSolicitarMascara	Digital Output	ABB	22	All	Keep Current Value (no change)	0
doPiezaCogidaHorno	Digital Output	ABB	11	All	Keep Current Value (no change)	0
doDejarMascLimpia	Digital Output	ABB	15	All	Keep Current Value (no change)	0
doConectarNegro	Digital Output	ABB	2	All	Keep Current Value (no change)	0
doDesconectarBlanco	Digital Output	ABB	3	All	Keep Current Value (no change)	0
doDejarMascSucia	Digital Output	ABB	17	All	Keep Current Value (no change)	0
diBlancoOcupado	Digital Input	ABB	5	All	N/D	0
diControlRealizado	Digital Input	ABB	27	All	N/D	0
diControlSolicitado	Digital Input	ABB	26	All	N/D	0
GirarNegro	Digital Output	ABB	13	All	Keep Current Value (no change)	0
GirarBlanco	Digital Output	ABB	12	All	Keep Current Value (no change)	0
FMaq2Arriba	Digital Output	ABB	9	All	Keep Current Value (no change)	0
diLimpiaOcupado	Digital Input	ABB	19	All	N/D	0
diNuevaMascDisponible	Digital Input	ABB	20	All	N/D	0
diPickInPos	Digital Input	ABB	0	All	N/D	0
diNegroOcupado	Digital Input	ABB	6	All	N/D	0
diM1Abierto	Digital Input	ABB	9	All	N/D	0
diListoPiezaHorno	Digital Input	ABB	21	All	N/D	0
diM1Cerrado	Digital Input	ABB	10	All	N/D	0



Trabajo Fin de Grado

Autor: Gonzalo Muínelo Garrido

Grado en Ingeniería en
Electrónica Industrial y
Automática

Fecha: 2015





7. Estudio del tiempo de ciclo

Uno de los principales objetivos a cumplir en el proyecto es conseguir un tiempo de ciclo aceptable que nos permita que el resto del proceso productivo se desarrolle correctamente. Para que esto suceda debemos de conseguir que cada 2.6 minutos (156 segundos) salga una pieza de la célula robotizada en un funcionamiento normal, en el que no ha habido ninguna avería de ninguna de las maquinas o de la manutención y que no se ha solicitado realizar un control, ya que el control está considerado como un evento esporádico y que no está incluido en el funcionamiento habitual del proceso de nuestra célula robotizada. Este estudio lo realizaremos mediante un diagrama de Gantt con los tiempos obtenidos de la simulación y los tiempos de trabajo de la máquina.

7.1 Consideraciones temporales para el estudio

Para este estudio del tiempo de ciclo se ha tenido en cuenta los siguientes tiempos de trabajo de la máquina:

MAQUINA	TIEMPO EN MINUTOS	TIEMPO EN SEGUNDOS
Máquina 1 (mecanizado)	2.17	130.2
Máquina 2 (horno)	2.3	138
Máquina 3 (tratamiento especial)	2.4	144

También hemos considerado que siempre vamos a tener disponible una máscara limpia y siempre vamos a poder dejar la máscara sucia. Se ha considerado también que el tiempo que tarda la pinza en coger o dejar la pieza o la máscara son 3 segundos, siendo este un tiempo algo elevado respecto de la realidad (ponernos el caso más desfavorable).

De la simulación hemos obtenido los siguientes tiempos en los movimientos y acciones del robot:



ACCIÓN	TIEMPO
Coger de pieza entrante	6.5
Descarga y carga M1	17.1
Transición M1 a M2	2.3
Dejar pieza M2	11.3
Coger pieza M2	7.9
Transición M2 a mascarar	3.1
Intercambio mascarar	15.8
Descarga y carga M3 (piezas y mascarar)	35.4
Transición M3 hasta dejar pieza	9.7
Salida hasta Inicio	5.9

En un primer estudio del tiempo de ciclo se obtuvo un tiempo de ciclo de 158 segundos, muy cerca del objetivo pero por encima del objetivo. Para reducir este primer tiempo de ciclo se realizaron las siguientes acciones:

Optimización de trayectorias, eliminando o modificando puntos intermedios en los movimientos de carga y descarga de las máquinas.

Optimización de trayectorias, eliminando o modificando puntos intermedios en los movimientos de transición de una máquina a otra.

Eliminación de puntos de espera previos a los movimientos de carga y descarga.

7.2 Diagrama de Gantt

Para estudiar el tiempo de ciclo utilizaremos la herramienta Excel en la que se ha programado una hoja en la que introduciendo la duración del trabajo de las máquinas y los tiempos que emplea el robot en movimientos, obtendremos un diagrama de Gantt para estudiar el tiempo de ciclo.

En la figura 113 se refleja el diagrama de Gantt realizado para estudiar el tiempo de ciclo, empezando por un ciclo "ideal" en el que el robot va a llegar a las máquinas y se va a encontrar que tiene una pieza lista para ser descargada de la máquina y poder cargar la máquina con una nueva



pieza. Se han hecho tantas iteraciones de un ciclo hasta observar que el tiempo de ciclo se ha estabilizado. Para una mejor visualización del diagrama, se puede acceder a el a través del formato digital.

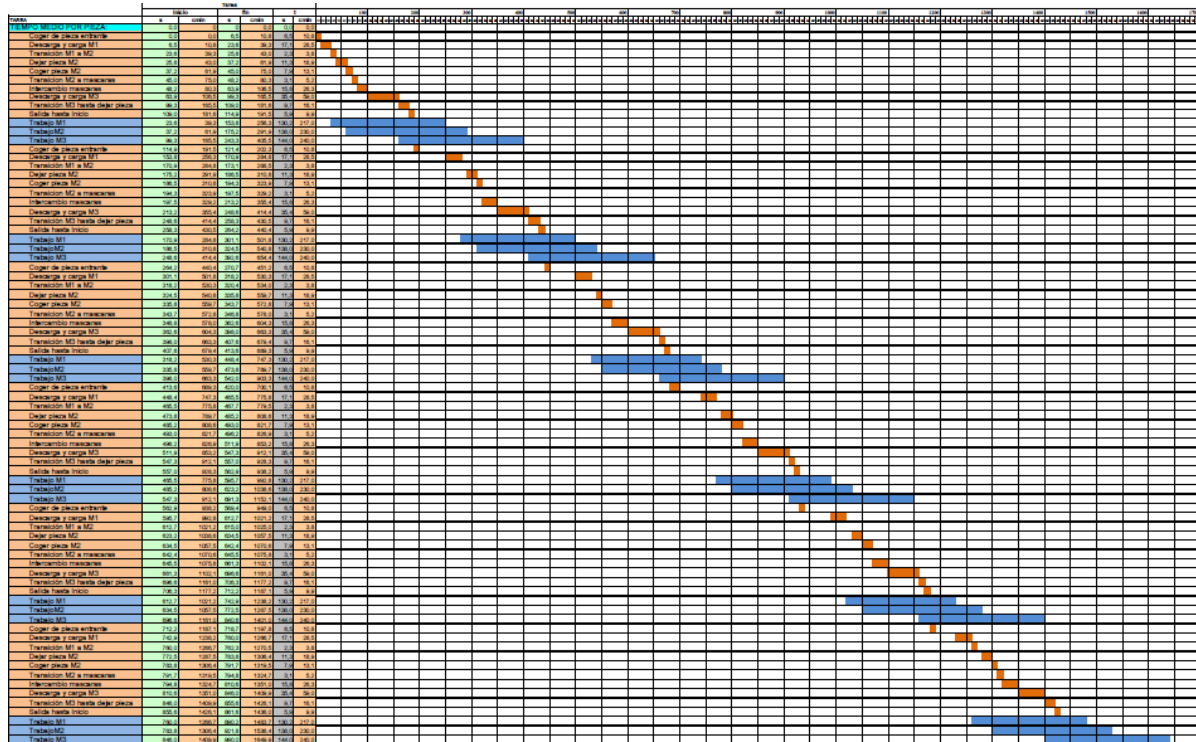


Figura 113

Primera iteración:

Se ha considerado que en la primera iteración todas las maquinas tienen una pieza disponible para realizar la carga y descarga y que el pinza de máscaras tiene una máscara sucia que deberá de dar salida. La primera iteración queda reflejada en la figura 114:

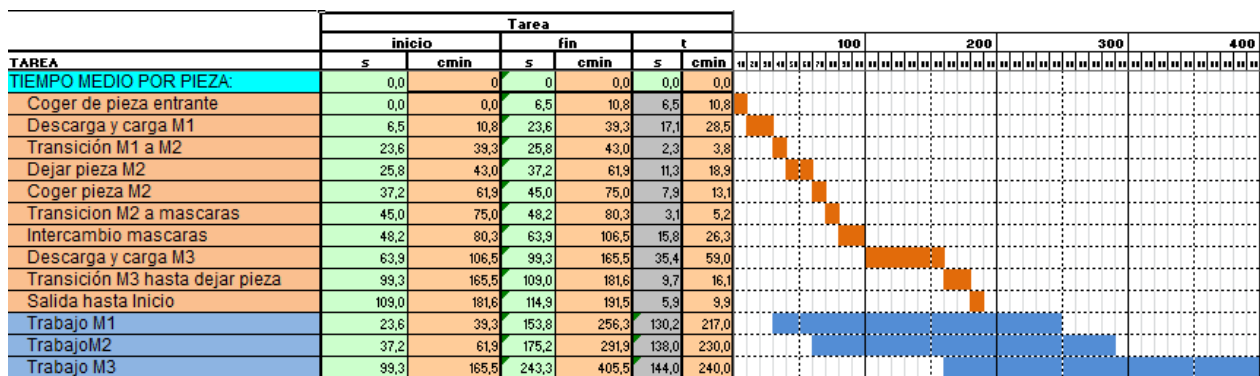


Figura 114



En esta primera iteración se obtiene un tiempo de ciclo de 114.9 segundos, pero obviamente no es una situación que podamos tener en cuenta porque ningún ciclo real va a tener las condiciones tan óptimas.

Segunda iteración:

En esta segunda iteración, reflejada en la figura 115, el robot tendrá que esperar a que la maquina 1 termine de trabajar de cuando ha empezado en la iteración anterior y también esperará a la maquina 2 (aunque un tiempo mucho menor), ya que el robot al no haber tenido que esperar en ningún otro sitio ha llegado a la posición del principio antes de que la maquina 1 terminase de trabajar.

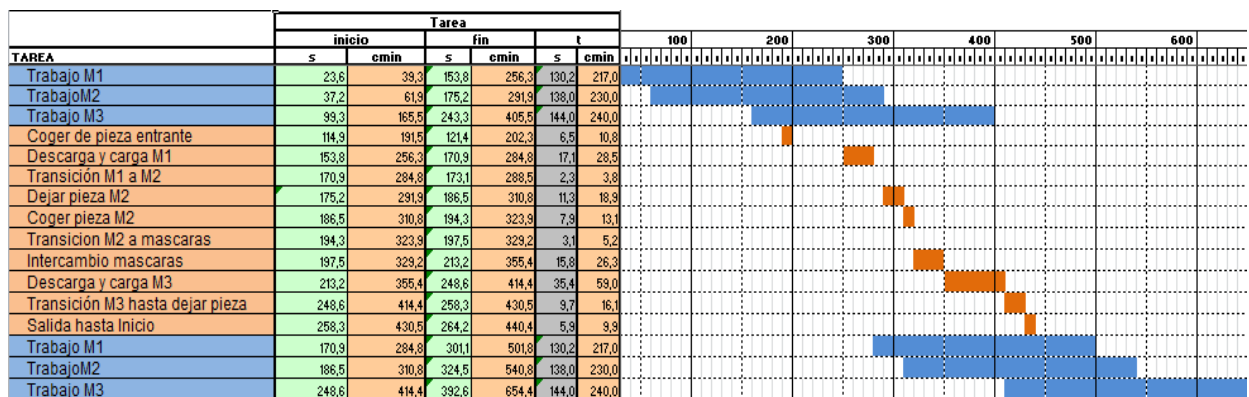


Figura 115

En esta segunda iteración se obtiene un tiempo de ciclo de 149.3 segundos, un tiempo de ciclo en el que cumplimos el objetivo, pero debemos de hacer mínimo 3 iteraciones más para verificar que este tiempo se repite y es nuestro tiempo de ciclo real.

Tercera iteración

Realizamos la tercera iteración, reflejada en la figura 116, observando que el robot volverá a esperar a la maquina 1 y a la maquina 2.



8. Evaluación y análisis de riesgos reales de nuestra célula robotizada.

Cabe destacar como aspecto crucial e imprescindible en las células robotizadas el poner todas las medidas necesarias para garantizar la seguridad de las personas que puedan interactuar con la célula y el entorno de una u otra manera.

En cuanto a leyes y normas de seguridad debemos de tener muy en cuenta las siguientes:

Real Decreto 1644 / 2008: de 10 de octubre:

Por el que se establecen las normas para la comercialización y puesta en servicio de las máquinas

Norma UNE-EN ISO 10218 – 1:

Robots para entornos industriales - Requisitos de seguridad - Parte 1: Robot

Norma UNE-EN ISO 10218 – 2:

Robots para entornos industriales - Requisitos de seguridad - Parte 2.

Norma UNE-EN ISO 12100 – 1:

Seguridad de las máquinas- conceptos básicos, principios generales para el diseño. Parte 1: Terminología básica.

Norma UNE-EN ISO 12100 – 2:

Seguridad de las máquinas- conceptos básicos, principios generales para el diseño. Parte 2: principios técnicos.

Norma UNE –EN ISO 13849 – 1:

Seguridad de las maquinas- partes de los sistemas de mando relativas a la seguridad. Parte 1: principios generales para el diseño.

Norma UNE –EN ISO 13849 – 2:



Seguridad de las maquinas- partes de los sistemas de mando relativas a la seguridad. Parte 2: Validación.

El cumplimiento de las normas nos supone una presunción de conformidad con el cumplimiento de la ley.

8.1 Aspectos a tener en cuenta en la seguridad

Debemos de conocer ciertas definiciones para poder identificar los riesgos y garantizar la seguridad:

Peligro: Fuente de posible lesión o daño a la salud.

Zona Peligrosa: Cualquier zona dentro y/o alrededor de una maquina en la cual la presencia de una persona suponga un riesgo para su seguridad o salud.

Persona expuesta: Cualquier persona que se encuentre, enteramente o en parte, en una zona peligrosa.

Operador: Persona o personas encargadas de instalar, manejar, regular, mantener, limpiar, reparar o desplazar una máquina.

Riesgo: Combinación de la probabilidad y la gravedad de una lesión o de un daño a la salud que pueda producirse en una situación peligrosa.

Resguardo: Elemento de la maquina utilizado específicamente para proporcionar protección por medio de una barrera física.

Dispositivo de protección: Dispositivo (distinto de un resguardo) que reduce el riesgo, por si solo o asociado con un resguardo.

Uso previsto: Uso de la máquina de acuerdo con la información proporcionada en las instrucciones para la utilización.

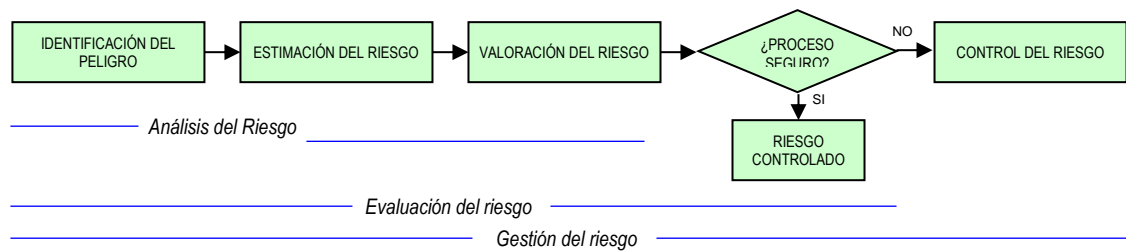
Mal uso razonablemente previsible: Uso de la máquina de una forma no propuesta en las instrucciones para la utilización, pero



que puede resultar de un comportamiento humano fácilmente previsible.

8.2 Análisis de riesgos

Para realizar el análisis, evaluación y gestión de riesgos se ha utilizado el método para la Evaluación de Riesgos establecido por el Instituto Nacional de Seguridad e Higiene en el Trabajo (INSHT) en el documento "Guías para la acción preventiva - Trabajo en Oficinas-".



El riesgo asociado a una situación peligrosa concreta es función de la probabilidad de que se produzca dicho daño (NP) y de la gravedad del mismo (NG):

$$ER = NP \cdot NG$$

La probabilidad (NP) viene dada por la combinación de los siguientes factores:

$$NP = NE \cdot ND \cdot F$$

NG - GRAVEDAD		
Mortal	Muerte	100
Muy Grave	Lesiones graves que pueden ser irreparables.	60
Grave	Lesiones con incapacidad laboral transitoria.	25
Leve	Pequeñas lesiones que no requieren tratamiento urgente.	10



NE - NIVEL DE EXPOSICIÓN		
Continuada	Continuamente. Varias veces en la jornada laboral con tiempo prolongado.	4
Frecuente	Varias veces en la jornada laboral, aunque sea con tiempos cortos.	3
Ocasional	Alguna vez durante la jornada laboral y con periodo corto de tiempo.	2
Esporádica	Irregularmente.	1

F - POSIBILIDAD DE EVITAR EL DAÑO	
Factor menor o igual a 1 que recoge las circunstancias que influyen en la disminución de la probabilidad de que se produzca un daño, como puede ser una alta cualificación, presencia de señales de advertencia, experiencia y conocimientos prácticos de la maquinaria o la rapidez con la que en una situación peligrosa se puede producir un daño	(0,5-1)

ND - NIVEL DE DEFICIENCIA		
Muy deficiente	Se han detectado factores que determinan como muy posible la materialización del peligro. El conjunto de medidas preventivas existentes respecto al peligro resulta ineficaz	10
Deficiente	Se ha detectado algún factor significativo en el peligro que precisa ser corregido. La eficacia del conjunto de medidas preventivas existentes se ve reducida de forma apreciable.	6
Sin deficiencia	La eficacia del conjunto de medidas preventivas existentes respecto al peligro no se ve reducida	2

RIESGO	4000 - 600	Riesgo muy alto, no tolerable
	< 600 - 150	Riesgo alto, no tolerable
	< 150 - 60	Riesgo medio residual, se deben establecer medidas preventivas
	< 60	Riesgo bajo residual, se deben establecer medidas preventivas



8.3 Identificación y estimación de los riesgos.

El análisis de riesgos se hará no solo durante el funcionamiento de la máquina, si no que se hará de todo el ciclo de vida de la célula robotizada, incluyendo el transporte de los elementos hasta el lugar de la implantación y una futura retirada de la célula.

En la siguiente tabla se reflejará los riesgos identificados de todo el ciclo de vida de la célula robotizada, y la evaluación de los riesgos en base a los parámetros anteriormente fijados.

Identificación del riesgo					Estimación del riesgo					
Nº	Ciclo de vida	Tarea	Mal uso razonablemente previsible - Comportamiento involuntario	Origen, consecuencia y descripción del riesgo.	Gravedad	NE	ND	F	Valor	Valoración
1	Transporte	Elevación y carga de instalación	Comportamiento que resulta de una falta de concentración o de atención	Caída de objetos. Realización de los trabajos mecánicos sin las herramientas o medios adecuados	60	1	2	0.7	84	Medio
2	Montaje, instalación y puesta en servicio de la célula.	Montaje electro-neumático de la célula.	Comportamiento que resulta de una falta de concentración o de atención	Contactos directos o indirectos con partes activas. Realización de los trabajos eléctricos sin las herramientas o medios adecuados	60	1	2	0.7	84	Medio
3	Montaje, instalación y puesta en servicio de la célula.	Montaje mecánico de la célula.	Comportamiento que resulta de una falta de concentración o de atención	Caída de objetos. Golpes, atrapamientos por realización de trabajos sin herramientas ni medios adecuados	60	1	2	0.7	84	Medio
4	Reglaje. Programa. Cambio de proceso	Fijar trayectoria del robot. Fijar parámetro (Velocidad, límites, etc)	Comportamiento que resulta de una falta de concentración o de atención	Golpe por robot	60	2	6	0.6	432	Alto
5	Reglaje de herramienta. Programa	Fijar los parámetros de apertura de pinza	Comportamiento que resulta de una falta de concentración o de atención	Atrapamiento	60	2	6	0.6	432	Alto
6	Funcionamiento	Verificación del producto final.	Comportamiento que resulta de presiones a dejar mantener la máquina funcionando en cualquier circunstancia.	Golpe o rozadura de la parte móvil contra el cerramiento de seguridad mientras sale la pieza terminada	25	2	6	0.7	210	Alto



7	Funcionamiento	Intervenciones menores Apertura de puerta de cerramiento sin autorización	Comportamiento que resulta de presiones a dejar mantener la máquina funcionando en cualquier circunstancia	Golpe o atrapamiento con elementos móviles y/o robots en el interior del cerramiento	60	2	6	0.8	576	Alto
8	Mantenimiento	Reglaje o ajuste de cilindros neumáticos o accionamientos eléctricos	Comportamiento que resulta de presiones a dejar mantener la máquina funcionando en cualquier circunstancia	Golpe o atrapamiento al manipular los accionamientos neumáticos o eléctricos, en caso de fallo sistema de seguridad previsto	60	2	6	0.8	576	Alto
9	Búsqueda y detección de averías.	Reglaje o ajuste de cilindros neumáticos o accionamientos eléctricos	Comportamiento que resulta de presiones a dejar mantener la máquina funcionando en cualquier circunstancia	Golpe o atrapamiento al manipular los accionamientos neumáticos o eléctricos, en caso de fallo sistema de seguridad previsto	60	2	6	0.8	576	Alto
10	Desinstalación de la célula.	Desmontaje mecánico de la célula.	Comportamiento que resulta de una falta de concentración o de atención	Caída de objetos. Golpes, atrapamientos por realización de trabajos sin herramientas ni medios adecuados	60	1	2	0.7	84	Medio
11	Desinstalación de la célula.	Desmontaje electro-neumático de la célula.	Comportamiento que resulta de una falta de concentración o de atención	Contactos directos o indirectos con partes activas. Realización de los trabajos eléctricos sin las herramientas o medios adecuados	60	1	2	0.7	84	Medio
12	Transporte	Elevación y carga de instalación	Comportamiento que resulta de una falta de concentración o de atención	Caída de objetos. Realización de los trabajos mecánicos sin las herramientas o medios adecuados	60	1	2	0.7	84	Medio

8.4 Acciones correctoras y reevaluación del riesgo

Ante los riesgos identificados y la evaluación de los mismos, se deben de interponer unas acciones correctoras que disminuyan el valor del riesgo y nos permita tener una instalación mucho más segura.



Estimación del riesgo			Reevaluación del riesgo						Riesgo Residual
Nº	Acción correctora	Prioridad	Gravedad	NE	ND	F	Valor	Valoración	
1	Utilización Equipos de Protección Individuales de acuerdo con puesto de trabajo: guantes contra riesgo mecánico/eléctrico/corte chapa/químico; gafas de seguridad; casqueta de protección; calzado de seguridad; cascos de protección auditiva; etc.	Alta	25	1	2	0.5	25	Bajo	Definir los Riesgos Residuales de la máquina NO eliminados e informar acerca de los mismos. Realización trabajos sólo por personal formado técnicamente y autorizado.
2	Utilización Equipos de Protección Individuales de acuerdo con puesto de trabajo: guantes contra riesgo mecánico/eléctrico/corte chapa/químico; gafas de seguridad; casqueta de protección; calzado de seguridad; cascos de protección auditiva; etc.	Alta	25	1	2	0.5	25	Bajo	Definir los Riesgos Residuales de la máquina NO eliminados e informar acerca de los mismos. Realización trabajos sólo por personal formado técnicamente y autorizado.
3	Utilización Equipos de Protección Individuales de acuerdo con puesto de trabajo: guantes contra riesgo mecánico/eléctrico/corte chapa/químico; gafas de seguridad; casqueta de protección; calzado de seguridad; cascos de protección auditiva; etc.	Alta	25	1	2	0.5	25	Bajo	Definir los Riesgos Residuales de la máquina NO eliminados e informar acerca de los mismos. Realización trabajos sólo por personal formado técnicamente y autorizado.
4	Medida técnica específica (por ej. colocación de paneles y bimanual de accionamiento para activación de movimientos; corte de energías parada de la máquina, medios para evacuar cargas electrostáticas, etc.).	Alta	10	2	2	0.6	24	Bajo	Definir los Riesgos Residuales de la máquina NO eliminados e informar acerca de los mismos. Realización trabajos sólo por personal formado técnicamente y autorizado
5	Medida técnica específica (por ej. colocación de paneles y bimanual de accionamiento para activación de movimientos; corte de energías parada de la máquina, medios para evacuar cargas electrostáticas, etc.).	Alta	10	2	2	0.6	24	Bajo	Definir los Riesgos Residuales de la máquina NO eliminados e informar acerca de los mismos. Realización trabajos sólo por personal formado técnicamente y autorizado
6	Elección de dispositivos de protección adecuados al riesgo: fijos o móviles con enclavamiento.	Alta	25	2	2	0.7	70	Medio	Definir los Riesgos Residuales de la máquina NO eliminados e informar acerca de los mismos. Realización trabajos sólo por personal formado técnicamente y autorizado
7	Medida técnica específica (por ej. colocación de paneles y bimanual de accionamiento para activación de movimientos; corte de energías parada de la máquina, medios para evacuar cargas electrostáticas, etc.).	Alta	25	2	2	0.5	50	Bajo	Definir los Riesgos Residuales de la máquina NO eliminados e informar acerca de los mismos. Realización trabajos sólo por personal formado técnicamente y autorizado



8	Medida técnica específica (por ej. colocación de paneles y bimanual de accionamiento para activación de movimientos; corte de energías parada de la máquina, medios para evacuar cargas electrostáticas, etc.).	Alta	10	2	2	0.5	20	Bajo	Definir los Riesgos Residuales de la máquina NO eliminados e informar acerca de los mismos. Realización trabajos sólo por personal formado técnicamente y autorizado
9	Medida técnica específica (por ej. colocación de paneles y bimanual de accionamiento para activación de movimientos; corte de energías parada de la máquina, medios para evacuar cargas electrostáticas, etc.).	Alta	10	2	2	0.5	20	Bajo	Definir los Riesgos Residuales de la máquina NO eliminados e informar acerca de los mismos. Realización trabajos sólo por personal formado técnicamente y autorizado
10	Utilización Equipos de Protección Individuales de acuerdo con puesto de trabajo: guantes contra riesgo mecánico/eléctrico/corte chapa/químico; gafas de seguridad; casqueta de protección; calzado de seguridad; cascos de protección auditiva; etc.	Alta	25	1	2	0.5	25	Bajo	Definir los Riesgos Residuales de la máquina NO eliminados e informar acerca de los mismos. Realización trabajos sólo por personal formado técnicamente y autorizado.
11	Utilización Equipos de Protección Individuales de acuerdo con puesto de trabajo: guantes contra riesgo mecánico/eléctrico/corte chapa/químico; gafas de seguridad; casqueta de protección; calzado de seguridad; cascos de protección auditiva; etc.	Alta	25	1	2	0.5	25	Bajo	Definir los Riesgos Residuales de la máquina NO eliminados e informar acerca de los mismos. Realización trabajos sólo por personal formado técnicamente y autorizado.
12	Utilización Equipos de Protección Individuales de acuerdo con puesto de trabajo: guantes contra riesgo mecánico/eléctrico/corte chapa/químico; gafas de seguridad; casqueta de protección; calzado de seguridad; cascos de protección auditiva; etc.	Alta	25	1	2	0.5	25	Bajo	Definir los Riesgos Residuales de la máquina NO eliminados e informar acerca de los mismos. Realización trabajos sólo por personal formado técnicamente y autorizado.



9. Conclusiones

Una vez finalizado el trabajo fin de grado, se pueden extraer una serie de conclusiones:

- El software RobotStudio nos permite realizar simulaciones muy cercanas a la realidad, y que pueden ser de gran utilidad a la hora de optimizar los trabajos de implementar células robotizadas nuevas.
- La simulación robótica nos permite reflejar de manera muy visual el resultado que queremos obtener, pudiendo plantear distintas hipótesis de distribución de elementos dentro de nuestro entorno robotizado.
- Mediante la simulación robótica podemos hacer un estudio exhaustivo de los movimientos del robot, intentando optimizarlos sin tener que estar físicamente trabajando con el robot.
- Supone una gran ventaja el poder trabajar previamente con la programación Rapid del robot desde un ordenador, sin necesidad de estar trabajando físicamente con el robot.
- Es necesario para la utilización de este tipo de software unos conocimientos previos sobre robótica, y además un trabajo amplio para poder entender el funcionamiento completo de dicho programa.
- Para la función de programación en Rapid a través de dicho programa, sería necesario una gran biblioteca de elementos y modelos gráficos con los que vaya a interactuar el robot.

Como opinión personal, creo que este programa será de gran utilidad en el futuro, y una herramienta básica en los trabajos que se realicen con robots de la marca ABB.



Trabajo Fin de Grado

Autor: Gonzalo Muínelo Garrido

Grado en Ingeniería en
Electrónica Industrial y
Automática

Fecha: 2015





Trabajo Fin de Grado

Autor: Gonzalo Muinelo Garrido

Grado en Ingeniería en
Electrónica Industrial y
Automática

Fecha: 2015



Bibliografía

ABB Robotics Products - IRB 6640 -
<http://new.abb.com/products/robotics/es/robots-industriales/irb-6640>

ABB Robotics Products - Manual de programación en Rapid -
<http://personal.biada.org/~jhorriilo/RAPID%20Manual%20operador.pdf>

ABB Robotics Products - Manual del Operador -
[http://www09.abb.com/global/scot/scot352.nsf/veritydisplay/0f50aa77049a50d9c1257d9e0046cd84/\\$file/3HAC032104-es.pdf](http://www09.abb.com/global/scot/scot352.nsf/veritydisplay/0f50aa77049a50d9c1257d9e0046cd84/$file/3HAC032104-es.pdf)

Instituto Nacional de Salud e Higiene en el Trabajo - Ficha técnica sobre metodología del análisis de riesgos -
http://www.insht.es/InshtWeb/Contenidos/Documentacion/FichasTecnicas/NTP/Ficheros/301a400/ntp_330.pdf

Victor Martínez León - Simulación de procesos de producción robotizados mediante el programa RobotStudio. Proyecto fin de carrera. Universidad Carlos III de Madrid. Julio 2008.



Trabajo Fin de Grado

Autor: Gonzalo Muínelo Garrido

Grado en Ingeniería en
Electrónica Industrial y
Automática

Fecha: 2015





Anexos

ANEXO I - Programa Rapid

Todo el programa RAPID desarrollado para realizar la simulación del sistema robotizado, se encuentra en el CD que acompaña a esta memoria.

ANEXO II – Manual de RobotStudio elaborado por el autor del proyecto.

Tabla de contenido

1. Introducir elementos en la estación	106
Introducir un robot en la estación	106
Introducir robot desde la biblioteca ABB	106
Introducir una estación con controlador de robot.	107
Fijar la posición de un robot.....	109
Introducir un elemento de equipamiento o una geometría	110
Fijar la posición de un elemento no robot.	111
2. Mover y tipos de movimiento del robot.....	112
Movimiento eje a eje del robot.....	112
Movimiento lineal del robot.	113
Movimiento de reorientación	114
3. Introducir una herramienta y situarla en el robot.....	116
Crear dato de herramienta.....	116
Crear herramienta a partir de geometría existente.....	117
Crear una herramienta con mecanismo.....	120
4. Crear un objeto de trabajo.....	124
5. Programar objetivo y crear trayectorias.	126
Mostrar volumen de trabajo	130
6. Programación en RAPID.....	131
7. Comunicaciones I/O.....	133
8. Diseño de Hardware.....	140
9. Componente inteligente	143



1. Introducir elementos en la estación

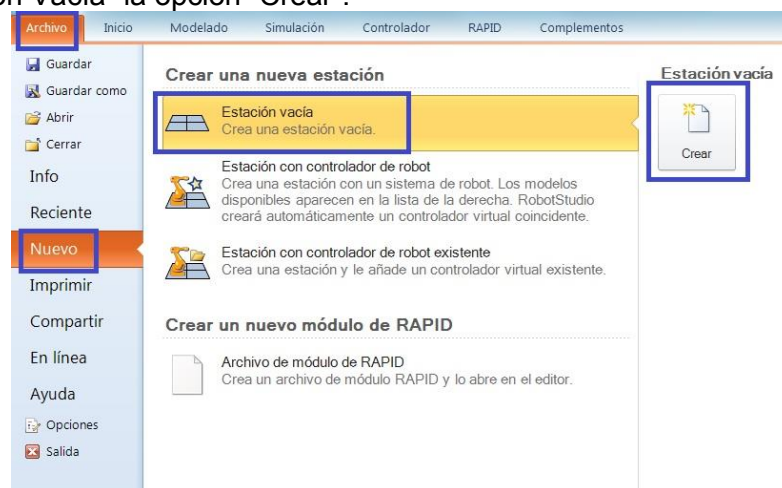
En nuestra estación podremos introducir elementos tales como robots, equipamiento o distintas geometrías.

Introducir un robot en la estación

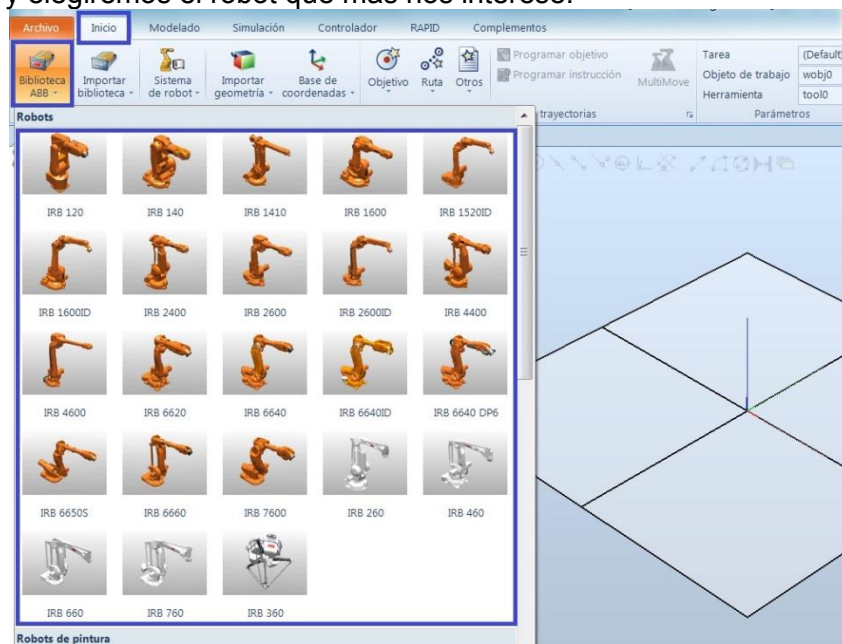
Hay varias opciones para introducir un robot en la estación, la diferencia entre ambas son si el controlador está asignado o hay que asignarle un controlador existente o crear un sistema robot.

Introducir robot desde la biblioteca ABB

Cuando abrimos RobotStudio nos aparecerá en la ventana principal, en la pestaña “Archivo”, la opción de “Nuevo”, de la cual seleccionaremos “Estación Vacía” la opción “Crear”.



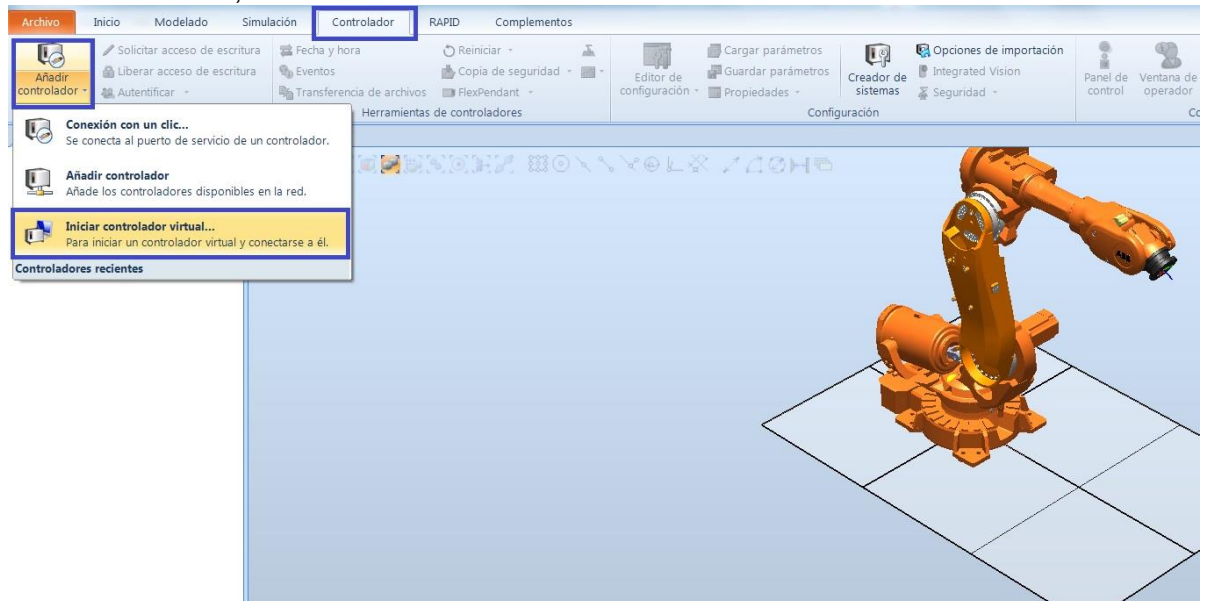
Se nos creará una Estación Vacía, en la que tendremos distintas pestañas. Situándonos en la pestaña “Inicio”, seleccionaremos la opción “Biblioteca ABB” y elegiremos el robot que más nos interese.



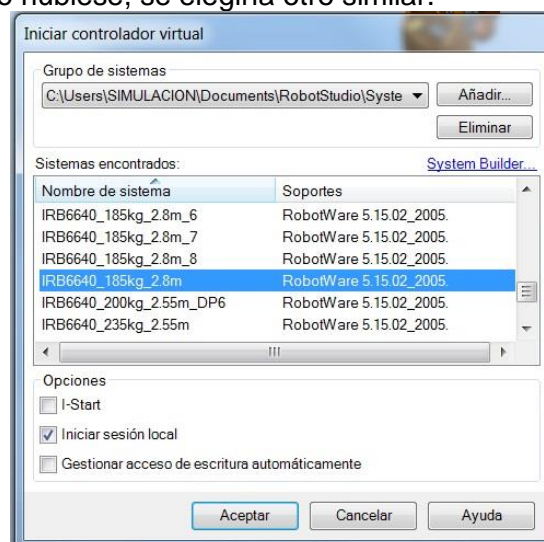


Lo siguiente que deberíamos sería añadirle un controlador ya que en el paso anterior lo que hemos hecho simplemente ha sido introducir un robot sin controlador.

En el mismo nivel que la pestaña “Inicio” tendremos una pestaña llamada “Controlador”, en la cual seleccionaremos “Iniciar controlador virtual...”



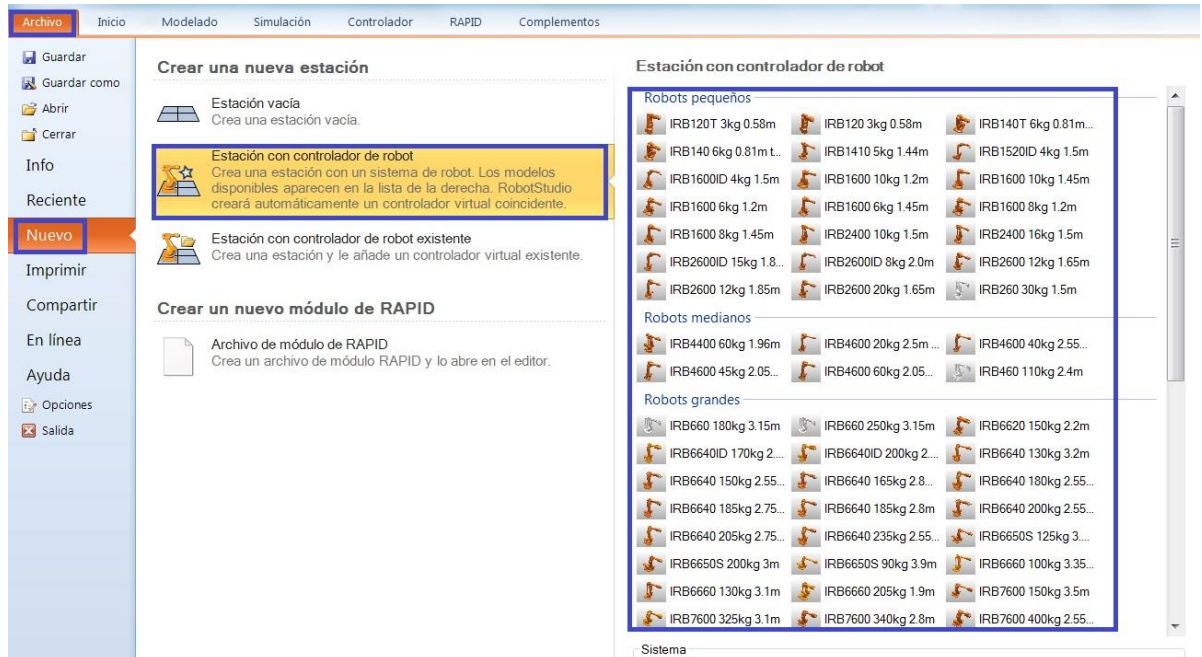
Se nos abrirá una ventana emergente en la que elegiremos el controlador que coincida con nuestro robot, es decir, si hemos seleccionado un robot IRB664_185kg_2.8m, buscaremos el controlador que se corresponda, en el caso de que no lo hubiese, se elegiría otro similar.



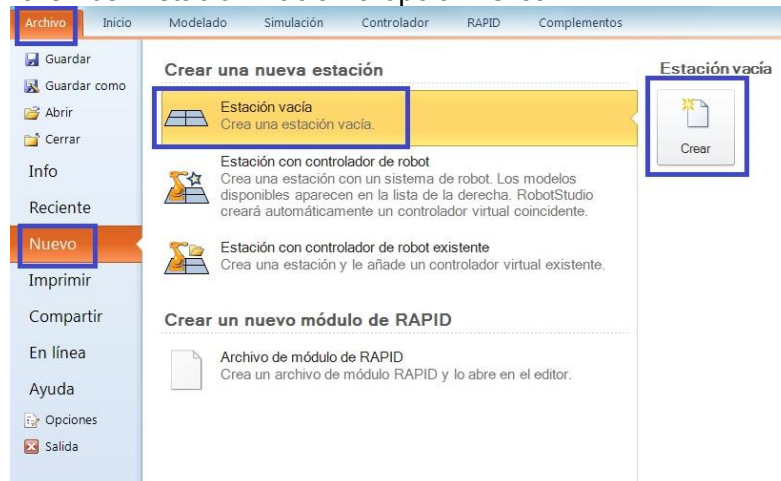
Ya tendríamos un robot con su respectivo controlador virtual asociado.

Introducir una estación con controlador de robot.

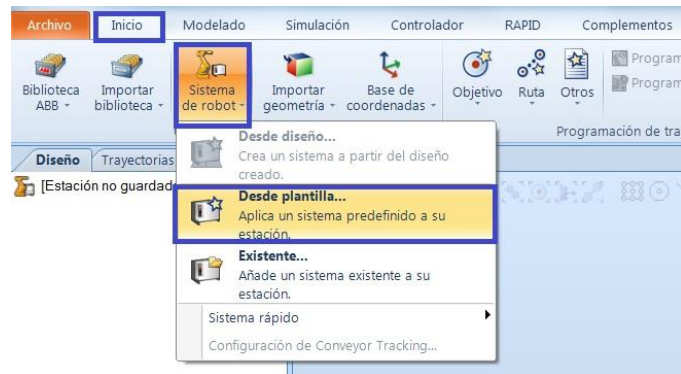
Cuando abrimos RobotStudio nos aparecerá en la ventana principal, en la pestaña “Archivo”, la opción de “Nuevo”, de la cual seleccionaremos “Estación con controlador de robot” y de entre la variedad de robots introduciremos el que más nos convenga para nuestra aplicación.



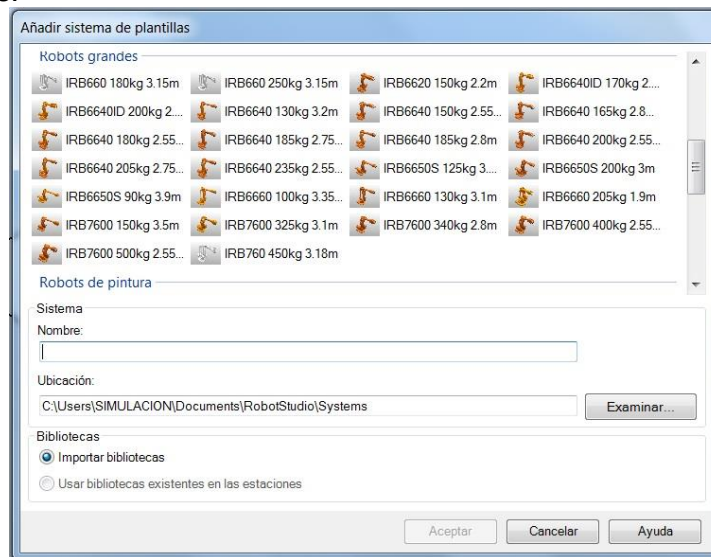
Una vez hecho esto ya tendríamos un robot con su respectivo controlador. Esta sería la forma más rápida de empezar a trabajar. Otra forma de hacer lo mismo podría ser en la pestaña "Archivo", la opción de "Nuevo", de la cual seleccionaremos "Estación Vacía" la opción "Crear".



Ya en la estación vacía, desde la pestaña inicio seleccionaremos "Sistema de Robot" → "Desde plantilla", y podremos elegir el robot que más nos convenga de la ventana emergente que se nos abrirá, que vendrá ya con un controlador asociado.



Se nos abrirá una ventana emergente con todos los sistemas robots disponibles.



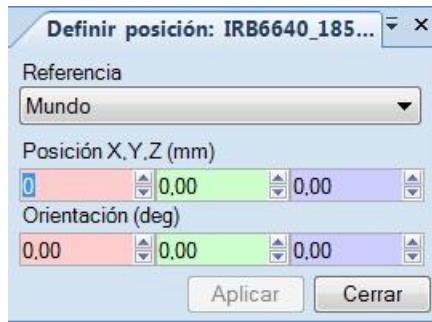
Fijar la posición de un robot

El robot inicialmente se situará en el origen de coordenadas y nosotros podemos mover el robot hasta la posición que más nos convenga, así como elegir la orientación de dicho robot.

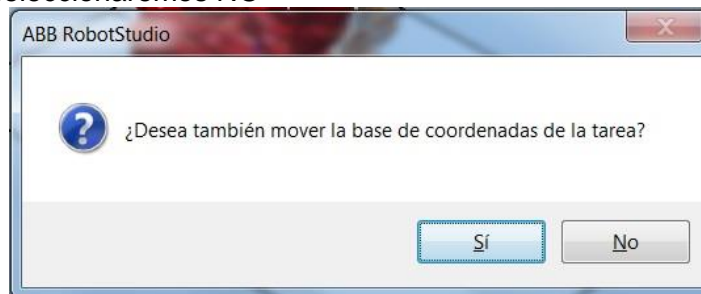
Si queremos fijar la posición a través de coordenadas en el panel izquierdo nos aparecerá en la parte superior una pestaña de "diseño" con todos los elementos que tenga nuestra estación. Situándonos con el cursor encima de nuestro robot y haciendo clic con el botón derecho se nos abrirá un desplegable con varias opciones, seleccionaremos la opción "fijar posición".

Se nos abrirá un panel en el que podremos introducir coordenadas X, Y, Z, así como introducir orientación respecto a cada uno de los ejes.

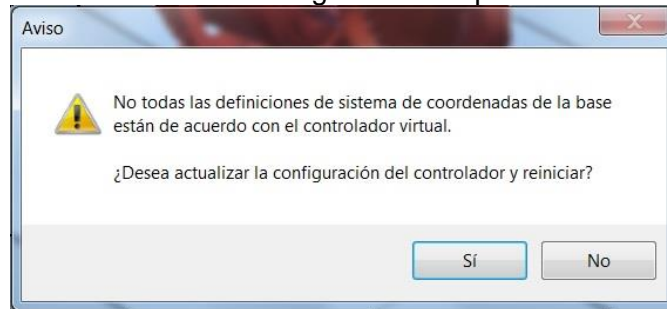




Una vez introducidas las coordenadas se nos abrirá una ventana emergente, de la que seleccionaremos NO



Aparecerá una nueva ventana emergente de la que seleccionaremos SI



La otra opción que tendríamos a la hora de mover un robot sería hacer clic sobre el elemento que queremos mover, y hacerlo a mano alzada seleccionando en la pestaña "Inicio", dentro del apartado "Mano alzada" una de estos dos iconos:



El primero que encontramos serviría para desplazarnos en un movimiento lineal, mediante unos ejes que nos aparecerán en el origen de coordenadas de nuestro elemento, y el otro sería para girar.

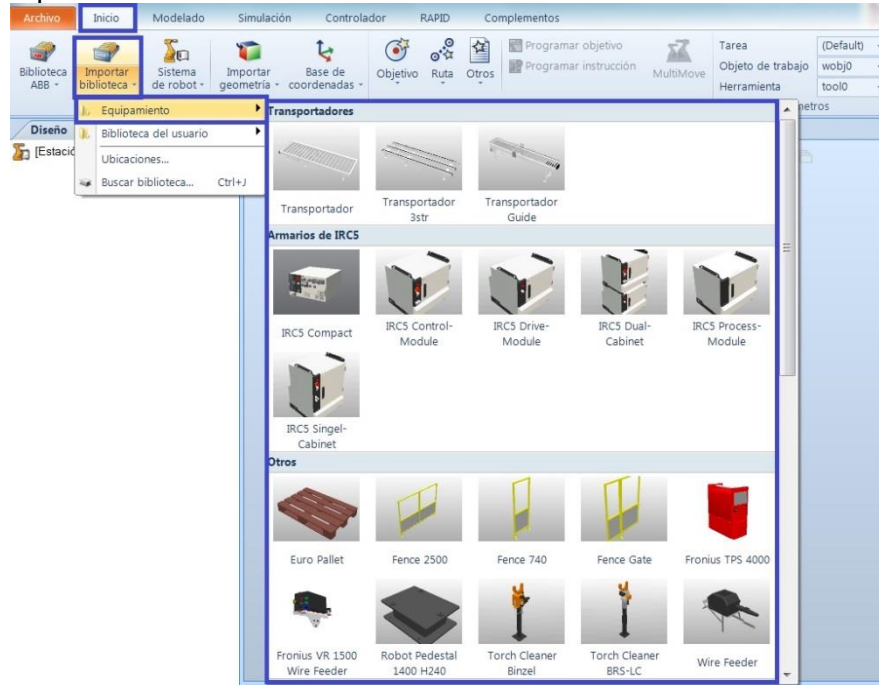
Introducir un elemento de equipamiento o una geometría

A parte de introducir robots, también deberemos de poder introducir ciertos elementos que permitan una adecuada simulación, estos elementos puede ser elementos predefinidos por RobotStudio como los elementos de



equipamiento, o podemos buscar una biblioteca desde una memoria externa; así como importar geometrías en formatos vmrl o algún otro formato con RobotStudio.

Para introducir un elemento de equipamiento lo que deberemos hacer será situarnos en la pestaña “Inicio”, y seleccionar “Importar biblioteca”; ahí tendríamos la opción de seleccionar “Equipamiento” y elegimos el que más se adapte a nuestras necesidades.



El mismo procedimiento haríamos si quisiésemos introducir un elemento de “Biblioteca del usuario”.

El procedimiento para “Buscar en biblioteca” sería seleccionando un elemento que tenemos en alguna memoria externa o en alguna carpeta diferente, esto se haría seleccionando la opción “Buscar en biblioteca”, y sería como en cualquier programa cuando queremos buscar y abrir un archivo.

Fijar la posición de un elemento no robot.

Los procedimientos serían los mismos que a la hora de mover un robot, tanto por coordenadas como a mano alzada.



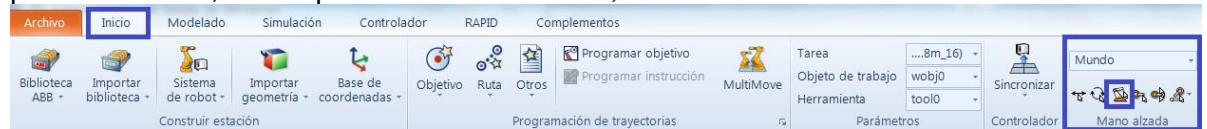
2. Mover y tipos de movimiento del robot.

Podemos mover el robot de distintas formas (eje a eje, movimiento lineal o movimiento de reorientación).

Movimiento eje a eje del robot.

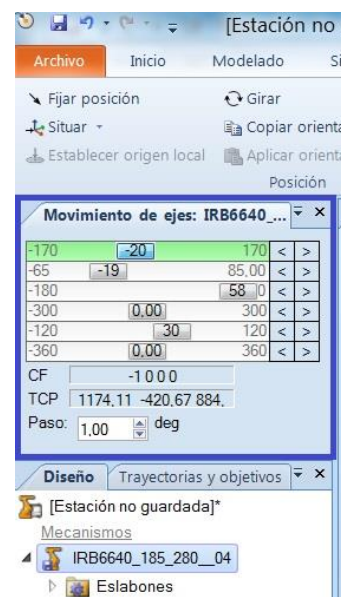
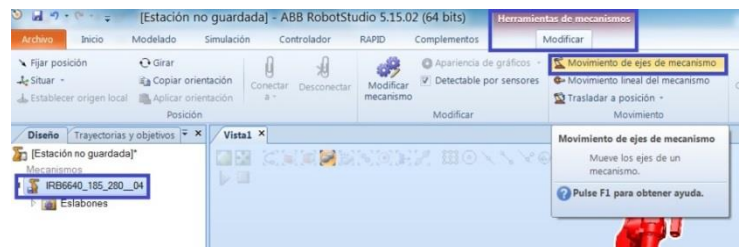
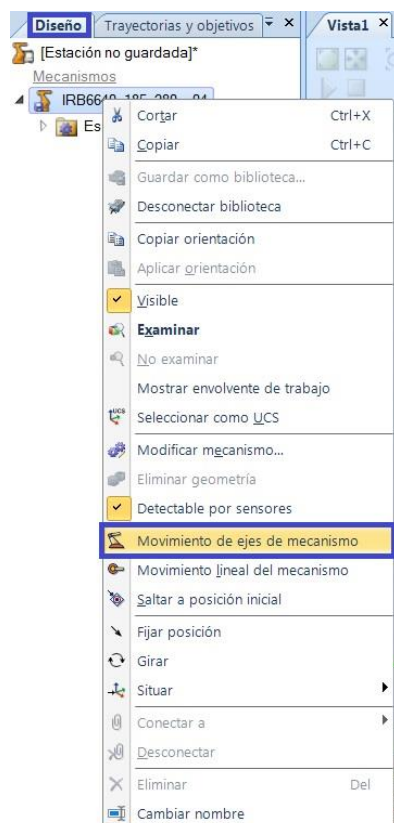
En el movimiento eje a eje del robot podremos hacerlo o a mano alzada o introduciendo datos a cada eje del robot dentro de sus límites.

Si decidimos hacerlo a mano alzada lo que haremos será seleccionar en la pestaña “Inicio”, en el apartado “Mano alzada”, haremos clic en el icono



A partir de ahí lo que haremos será hacer clic en el eslabón del robot que queramos mover, y haciendo uso del ratón lo moveremos.


Si decidimos movernos introduciendo datos a cada uno de los ejes del robot dentro de sus límites podremos hacerlo, seleccionando con el botón derecho del ratón el robot en la pestaña diseño y ahí seleccionando “Movimiento de ejes de mecanismo”. O bien seleccionando el robot en la pestaña diseño, lo cual hará aparecer al nivel de la pestaña “Inicio” otra pestaña de herramientas de mecanismos denominada “Modificar”, dentro de la que encontraremos la misma opción que antes “Movimiento de ejes de mecanismo”. Ambas opciones nos abrirán una ventana en la que poder introducir los datos.

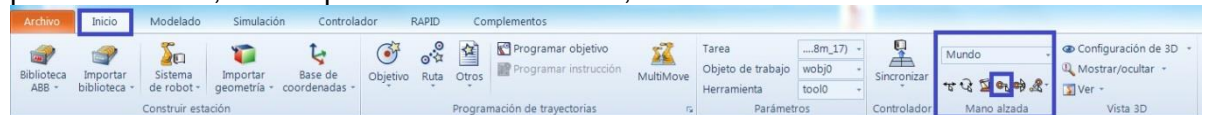




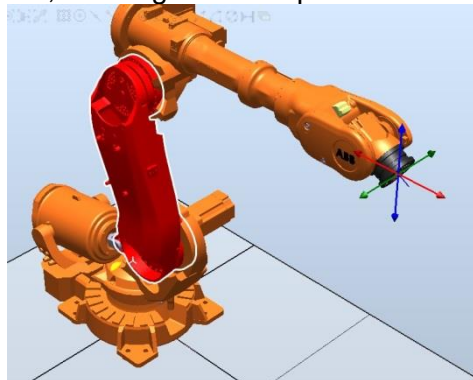
Movimiento lineal del robot.

El movimiento lineal consiste en que el robot desplaza el extremo del robot, o más habitualmente el TCP de la herramienta de manera lineal por los ejes X, Y, Z. Al igual que anteriormente hay dos maneras distintas de hacerlo, una es a mano alzada y la otra es introduciendo datos al robot.

Si decidimos hacerlo a mano alzada lo que haremos será seleccionar en la pestaña “Inicio”, en el apartado “Mano alzada”, haremos clic en el icono 



A partir de ahí lo que haremos será hacer clic en el robot para que nos aparezcan unos ejes X, Y, Z con flechas para que podamos movernos ,haciendo uso del ratón, a lo largo de cualquiera de los ejes.



Al igual que en el apartado de movimiento eje a eje, si decidimos movernos introduciendo datos dentro de sus límites podremos hacerlo, seleccionando con el botón derecho del ratón el robot en la pestaña diseño y ahí seleccionando “Movimiento lineal del mecanismo”. O bien seleccionando el robot en la pestaña diseño, lo cual hará aparecer al nivel de la pestaña “Inicio” otra pestaña de herramientas de mecanismos denominada “Modificar”, dentro de la que encontraremos la misma opción que antes “Movimiento lineal del mecanismo”. Ambas opciones nos abrirán una ventana en la que poder introducir los datos.

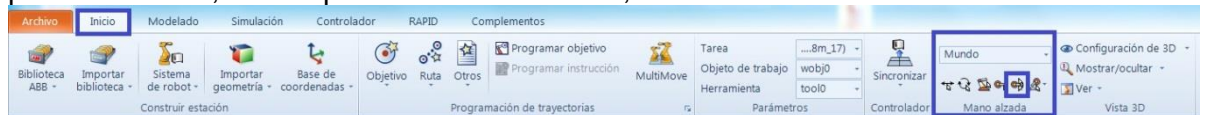




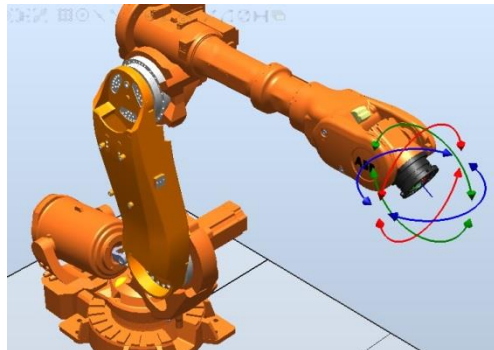
Movimiento de reorientación

Este tipo de movimiento lo que hará será dejar fijo el TCP y mover los ejes del robot de tal manera que se podrá tener la misma posición TCP pero con distinta orientación. Al igual que los otros dos citados anteriormente hay dos maneras distintas de hacerlo, una es a mano alzada y la otra es introduciendo datos al robot.

Si decidimos hacerlo a mano alzada lo que haremos será seleccionar en la pestaña “Inicio”, en el apartado “Mano alzada”, haremos clic en el icono



A partir de ahí lo que haremos será hacer clic en el robot para que nos aparezcan unos flechas para que podamos girar respecto al eje X, Y o Z haciendo uso del ratón.



Si decidimos movernos introduciendo datos a cada uno de los ejes del robot dentro de sus límites podremos hacerlo, seleccionando con el botón derecho del ratón el robot en la pestaña diseño y ahí seleccionando “Movimiento lineal”. O bien seleccionando el robot en la pestaña diseño, lo cual hará aparecer al nivel de la pestaña “Inicio” otra pestaña de herramientas de mecanismos denominada “Modificar”, dentro de la que encontraremos la misma opción que antes “Movimiento lineal”. Ambas opciones nos abrirán una ventana en la que poder introducir los datos. Los datos de reorientación se introducirán en los apartados Rx, Ry o Rz.



Trabajo Fin de Grado

Autor: Gonzalo Muinelo Garrido

Grado en Ingeniería en
Electrónica Industrial y
Automática

Fecha: 2015



Movimiento lineal: IRB6640_1... ⌵ ✕

X	1885,71	<	>
Y	0,00	<	>
Z	1955,00	<	>
RX	-180,00	<	>
RY	60,00	<	>
RZ	-180,00	<	>

Cfg:

⌵

Paso: mm/deg



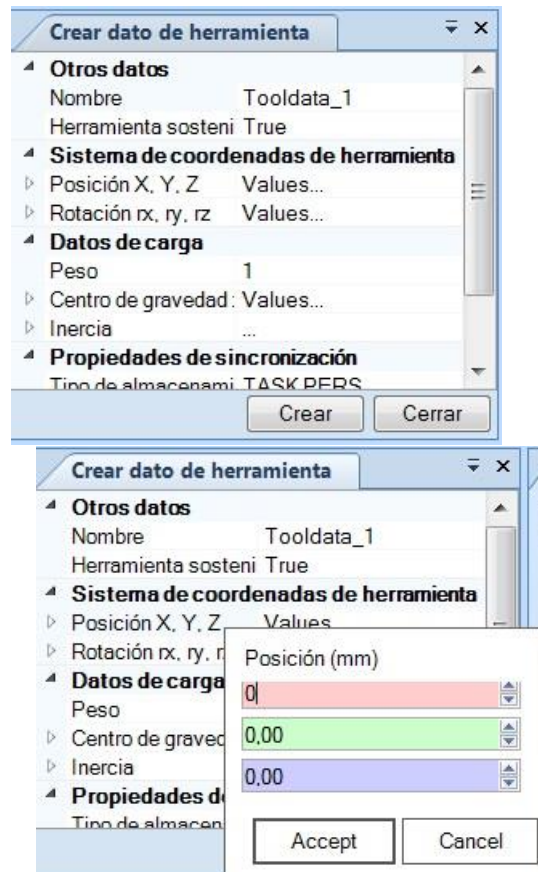
3. Introducir una herramienta y situarla en el robot.

Introducir una herramienta en nuestra estación de trabajo es muy importante ya que casi ningún robot tendría utilidad si no tuviese una herramienta, es por ello, que este apartado ofrece una gran cantidad de posibilidades a la hora de crear, posicionar o definir un TCP, dependiendo de nuestras necesidades.

Crear dato de herramienta

Esta es la primera opción que tenemos a la hora de definir una herramienta y su TCP. La herramienta en este caso será una herramienta genérica en la que simplemente definiremos su TCP. Nos situaremos en la pestaña "Inicio", y en el panel llamado "Programación de trayectorias", en el apartado "Otros" seleccionaremos la opción "Crear dato de herramienta". Una vez seleccionado nos aparecerá una ventana en la que podremos introducir los datos que tendrá nuestra herramienta.

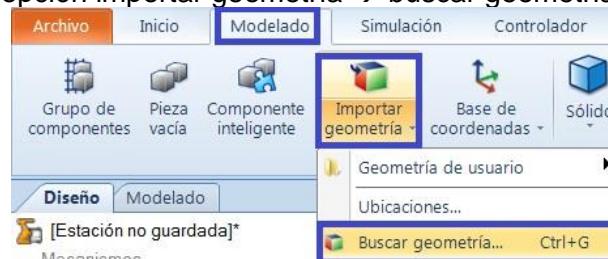




Tendremos que situar las coordenadas de la herramienta, seleccionaremos “Posición X, Y, Z” y ahí introduciremos sus coordenadas manualmente, o haciendo clic en el lugar donde queramos situarlo, también podremos rotar el TCP, introducir peso, centro de gravedad, etc.

Crear herramienta a partir de geometría existente.

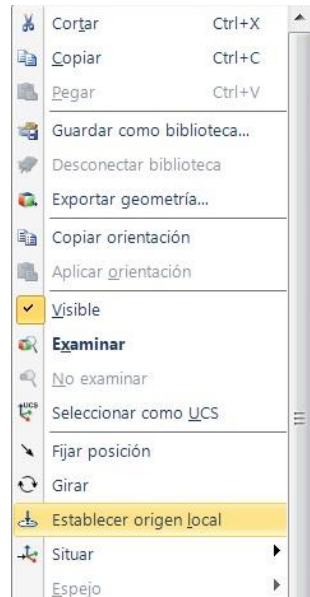
Podremos crear una herramienta a partir de una geometría ya existente. Lo primero que deberemos hacer será importar la geometría desde la pestaña “Modelado”, situada en el mismo nivel que la pestaña “Inicio”, y ahí seleccionar la opción importar geometría → buscar geometría.



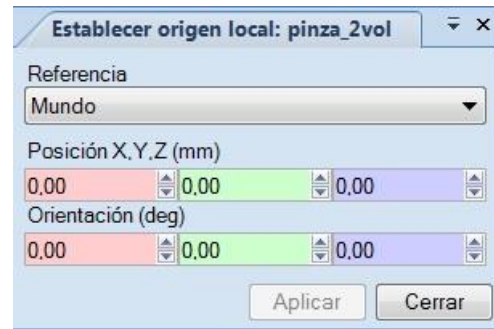
Buscaremos la geometría como buscaríamos cualquiera archivo, y lo cargaremos. Una vez cargado, nos aparecerá en la pantalla situado su origen local en el origen de coordenadas de la estación; hay veces que el origen local puede estar alejado de la pieza en sí, con lo cual a la hora de situar la herramienta en el robot es probable que no nos quede anclada como debería; deberemos con lo cual situar el origen local, para ello o en la propia geometría vista en la pantalla del simulador, o en la pestaña “Diseño”



en el nombre de la geometría, haremos clic con el botón derecho del ratón y se nos desplegará un panel con diferentes opciones y entre todas ellas elegiremos “Establecer origen local”.

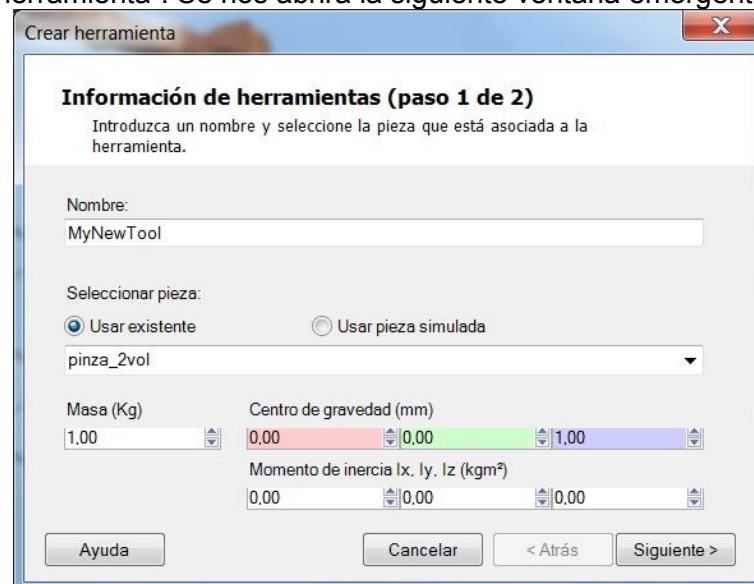


Una vez seleccionado se nos abrirá una ventana para introducir las coordenadas, o bien podremos situar las coordenadas manualmente sobre un punto que queramos en la ventana de simulación.



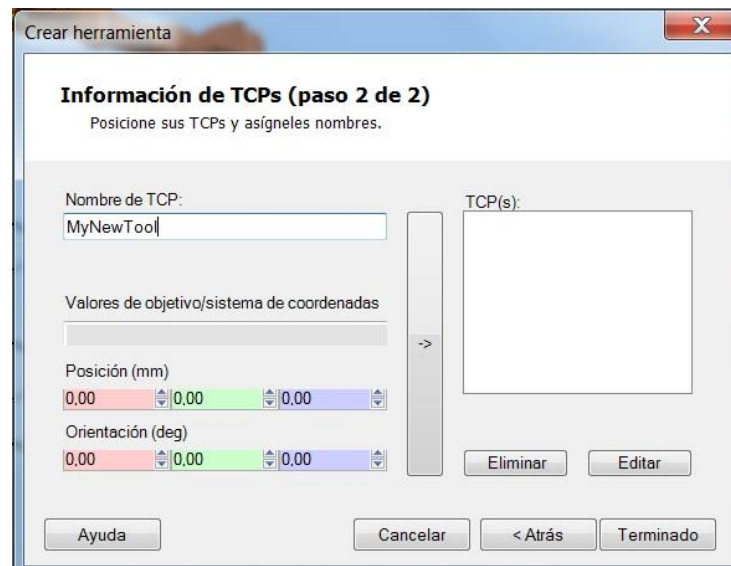
Este paso anterior solo sería necesario si el origen local de nuestra geometría no estuviese correctamente situado.

El siguiente paso a seguir sería seleccionar en la pestaña “Modelado” la opción de “Crear herramienta”. Se nos abrirá la siguiente ventana emergente.



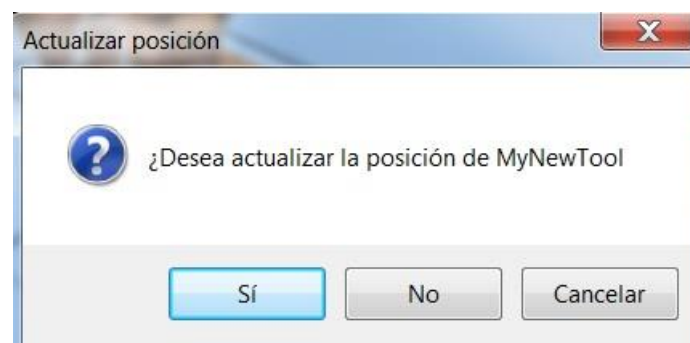
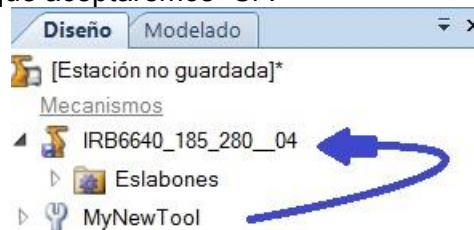
En ella introduciremos el nombre de nuestra herramienta, seleccionaremos la opción de “Usar existente”, seleccionaremos nuestra geometría y pulsaremos Siguiente >.

Esto nos abrirá una nueva ventana emergente que servirá para definir los TCPs de la herramienta:



Introduciremos el nombre del TCP, y podremos introducir las coordenadas del TCP a través de esta ventana indicando los datos, o podremos seleccionar un punto de la ventana de simulación, una vez hecho esto tendremos que añadir este TCP pulsando la flecha “->”, y podremos añadir tantos TCPs como queramos con este procedimiento.

Para terminar, situaremos nuestra herramienta en el extremo del robot, para ello desde la pestaña “Diseño” arrastraremos la herramienta hasta el nuestro Robot. Esto abrirá un aviso que aceptaremos “Sí”.





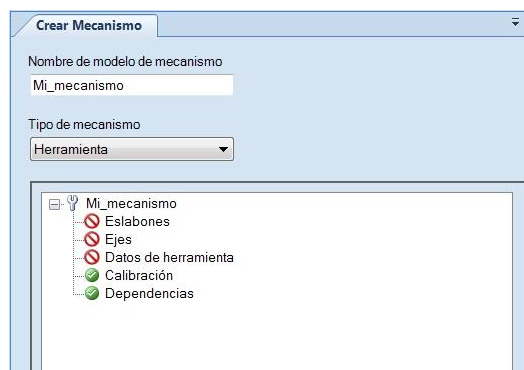
Crear una herramienta con mecanismo.

Existe una última opción si lo que deseamos es crear una herramienta que tenga movimiento propio, es decir una herramienta que tenga uno o varios mecanismos que le permitan tener movimiento. Para esto no podemos tener una sola geometría que sea un sólido rígido, sino que debemos de tener varias geometrías para que unas sean eslabones de otras, y así adquieran movimiento.

Para esto seleccionaremos la opción “Crear mecanismo” situada en la pestaña “Modelado”.

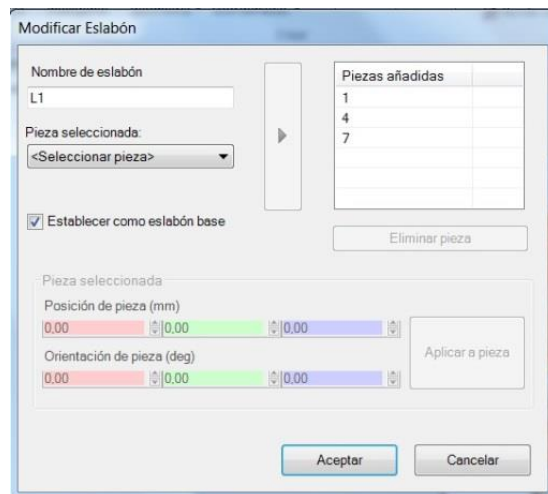


Al seleccionar esta opción se nos abrirá una nueva ventana con la que comenzaremos a modelar el mecanismo.



En esta ventana le daremos un nombre al mecanismo y seleccionaremos entre las opciones de tipo de mecanismo la de “Herramienta”.

A partir de ahí tendremos que definir por este orden: Eslabones, Ejes y Datos de Herramienta.



Empezaremos definiendo los eslabones haciendo doble clic sobre la palabra eslabones lo que hará que se abra una nueva ventana emergente. En esta ventana emergente le daremos nombre a los eslabones y seleccionaremos las piezas que compondrán cada eslabón.

Elegimos el nombre de nuestro eslabón y seleccionamos todas las piezas que queremos que pertenezcan a ese eslabón.

Deberemos de seleccionar un eslabón base, es decir, un eslabón que es fijo y no se mueva, generalmente seleccionaremos los soportes de las herramientas.

Vamos haciendo esto de forma iterativa creando tantos eslabones como sea necesario para nuestra Herramienta con mecanismo. Pulsamos aceptar cuando hayamos creado un eslabón, para poder crear otro.



Ahora procederemos a definir los ejes sobre los que se moverá nuestros eslabones.

Podremos nombre a cada uno de los ejes, y elegiremos si queremos que sea de rotación (giro) o prismático (traslación). Seleccionaremos un eslabón principal y un eslabón secundario que será el que realice el movimiento. La opción activo la seleccionaremos.

Para definir cada eje fijaremos dos puntos y a partir de ellos se generará un eje con esos dos puntos. También podremos darle unos límites de movimiento al mecanismo, uno inferior y otro superior, que puede ser constante o variable, generalmente utilizaremos la opción constante.

Aplicaremos, y crearemos tantos ejes como sea necesario.

Por último definiremos los datos de la herramienta (es decir, su TCP).

Indicando en primer lugar el eslabón al que pertenece, y su posición y orientación al igual que hacíamos cuando definíamos un dato de herramienta.

También podremos definir la masa, su centro de gravedad y su momento de inercia si fuese necesario.

Una vez realizados estos pasos, justo debajo, en la misma ventana tendremos que seleccionar la opción de compilar mecanismo.



Compilado el mecanismo, tendremos que definir las posiciones que va a tener nuestro mecanismo, es decir, memorizar una serie de poses a las que daremos nombres para después acceder fácilmente a ellas, por ejemplo, una pose podría ser si el mecanismo es de rotación, girado completamente en sentido horario hasta su límite. Para ello pulsaremos el botón “Añadir”.

Después de haber definido las poses, definiremos sus tiempos de transición, este tiempo es el tiempo que tarda en pasar de una pose a otra y viceversa, estos datos los introduciremos en una tabla. Para ello deberemos de pulsar el botón “Definir tiempos de transición”, y nos aparecerá en una nueva ventana la tabla.



Definir tiempos de transición

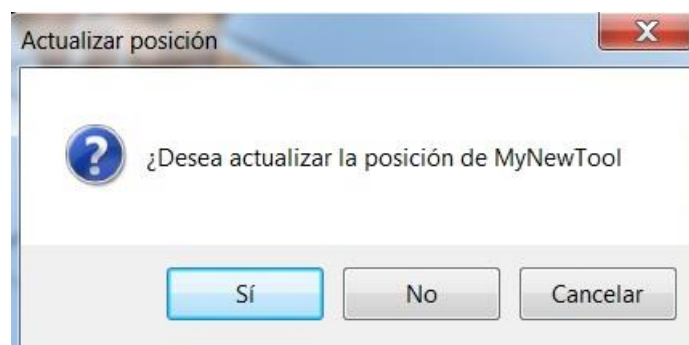
Tiempos de transición (s)

A pose: De pose:

	Pose de	Pose inicial	Pose1	Pose2	Pose3	Pose4	Pose5
► Pose de	0,000	0,000	0,000	0,000	0,000	0,000	0,000
Pose	0,000	0,000	0,000	0,000	0,000	0,000	0,000
Pose1	0,000	0,000	0,000	0,000	0,000	0,000	0,000
Pose2	0,000	0,000	0,000	0,000	0,000	0,000	0,000
Pose3	0,000	0,000	0,000	0,000	0,000	0,000	0,000
Pose4	0,000	0,000	0,000	0,000	0,000	0,000	0,000
Pose5	0,000	0,000	0,000	0,000	0,000	0,000	0,000

Aceptar Cancelar

Cuando hemos hecho todos estos pasos, cerramos el creador de mecanismos y ya tenemos nuestra herramienta con mecanismo. Para terminar, situaremos nuestra herramienta en el extremo del robot, para ello desde la pestaña “Diseño” arrastraremos la herramienta hasta el nuestro Robot. Esto abrirá un aviso que aceptaremos “Sí”.

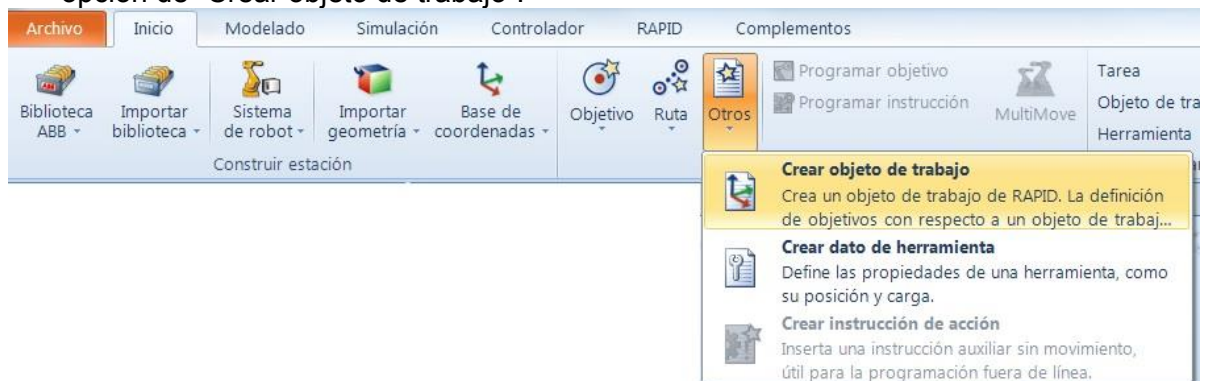




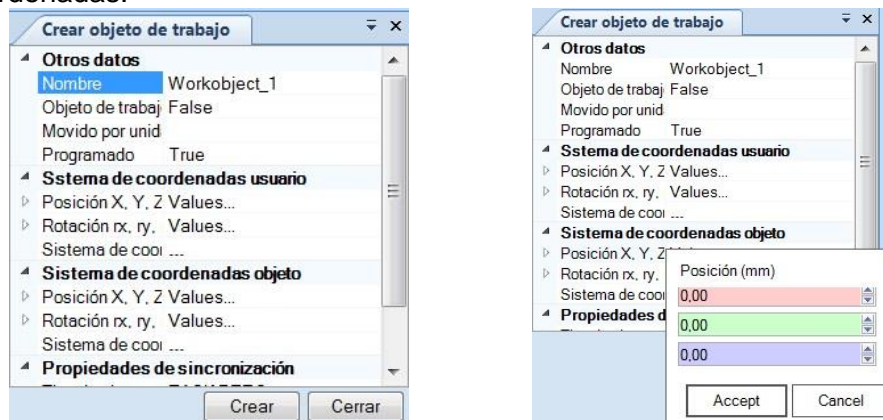
4. Crear un objeto de trabajo

Un objeto de trabajo es un sistema de referencia que definiremos para después programar objetivos y trayectorias. Los objetos de trabajo son interesantes desde el punto de vista de asociarlos o conectarlos a una determinada geometría de nuestra estación, como una mesa, o una máquina, ya que si estas se moviesen, el objeto de trabajo se movería con la geometría asociada, y consecuentemente también lo harían los objetivos programados respecto de ese objeto de trabajo.

En el mismo panel donde teníamos “Crear dato de herramienta”, tenemos la opción de “Crear objeto de trabajo”.



A partir de ahí se nos abrirá una ventana en la que podremos definir el nombre de nuestro objeto de trabajo, así como la posición de sistema de coordenadas.



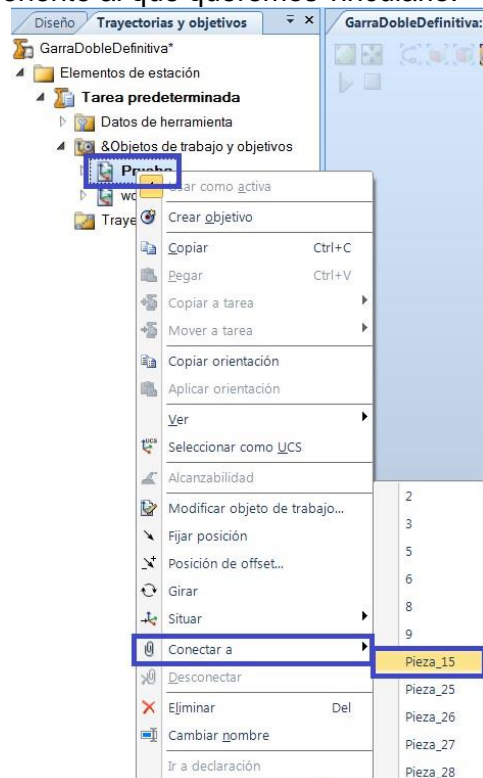
A parte de poder definir la posición de nuestro objeto de trabajo mediante coordenadas, podremos también definir su posición haciendo clic en cualquiera de los 3 apartados para introducir los datos, y posteriormente haciendo clic con el ratón en la ventana de simulación en el lugar en el que queramos situarlo.

Una vez definido nuestro objeto de trabajo, podremos conectarlo a un componente de la estación de trabajo para que nuestro objeto de trabajo se mueva con dicho componente.

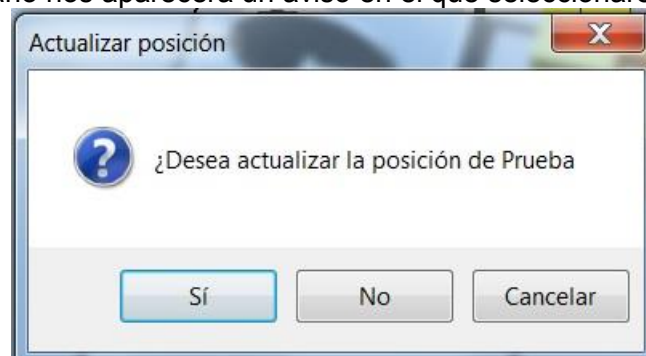
En el panel donde teníamos la pestaña “Diseño” también tenemos el panel “Trayectorias y objetivos”, en ese panel dentro de la carpeta “Objetos de trabajo y objetivos” encontraremos todos los objetos de trabajo que tenga nuestra estación.



Haciendo clic con el botón derecho del ratón en nuestro objeto de trabajo se nos desplegarán una serie de opciones, entre las que está “Conectar a...” y elegiremos el componente al que queremos vincularlo.



Al seleccionarlo nos aparecerá un aviso en el que seleccionaremos “NO”.

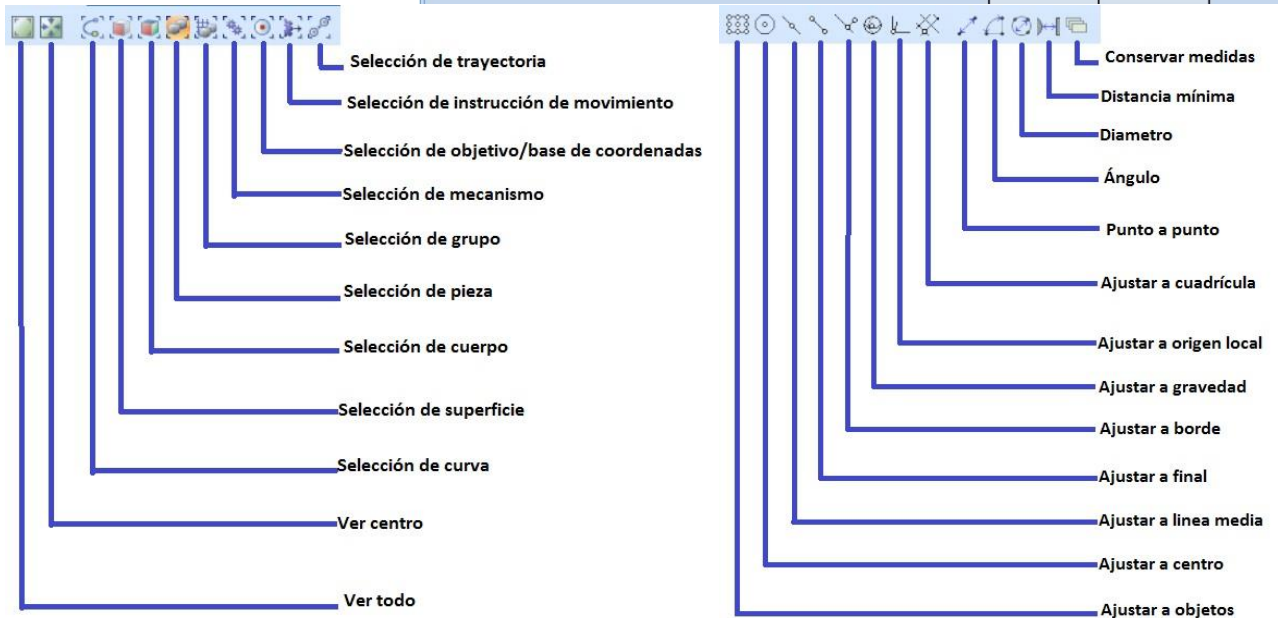
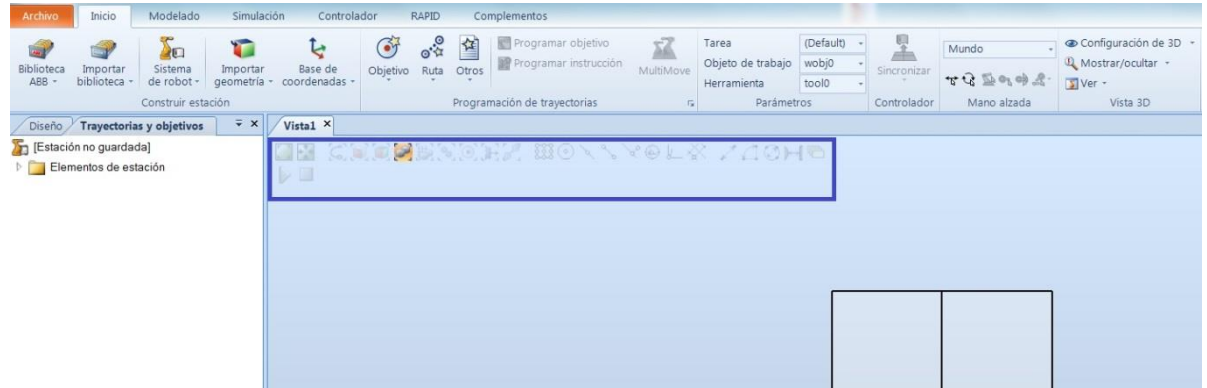




5. Programar objetivo y crear trayectorias.

Después de haber definido herramientas, objetos de trabajo, robot, su controlador y todos los elementos principales, nos interesa saber programar objetivos a los que queremos que llegue el robot y programar sus trayectorias, pero antes de nada se va a explicar un panel que permite configurar la manera de seleccionar puntos, o superficies.

Ese panel se encuentra situado en la ventana de simulación, justo en la parte de arriba.



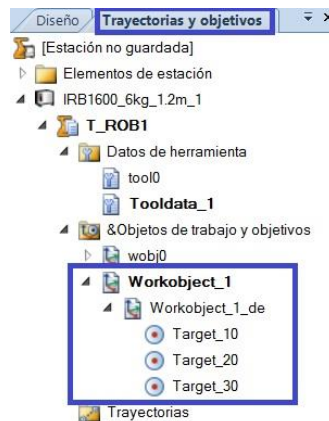
Para empezar a programar objetivos lo que deberemos hacer será situar en el simulador al robot en la posición que nos interese, lo podemos hacer de manera manual (con lo explicado anteriormente) o introduciendo datos de coordenadas. Es importante seleccionar la herramienta y el objeto de trabajo en el que queremos trabajar, podremos seleccionarlo en la pestaña "Inicio" donde encontraremos las dos opciones: "Objeto de trabajo" y "Herramienta".



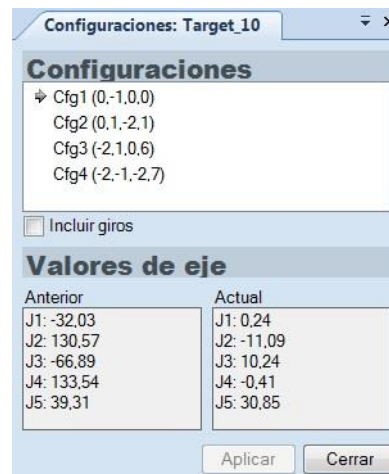
Una vez seleccionado el objeto de trabajo y la herramienta y situado el robot en la posición que queremos, deberemos hacer clic en el botón “Programar objetivo”, esto nos generará un punto con unas coordenadas que estará memorizado dentro de un objeto de trabajo.



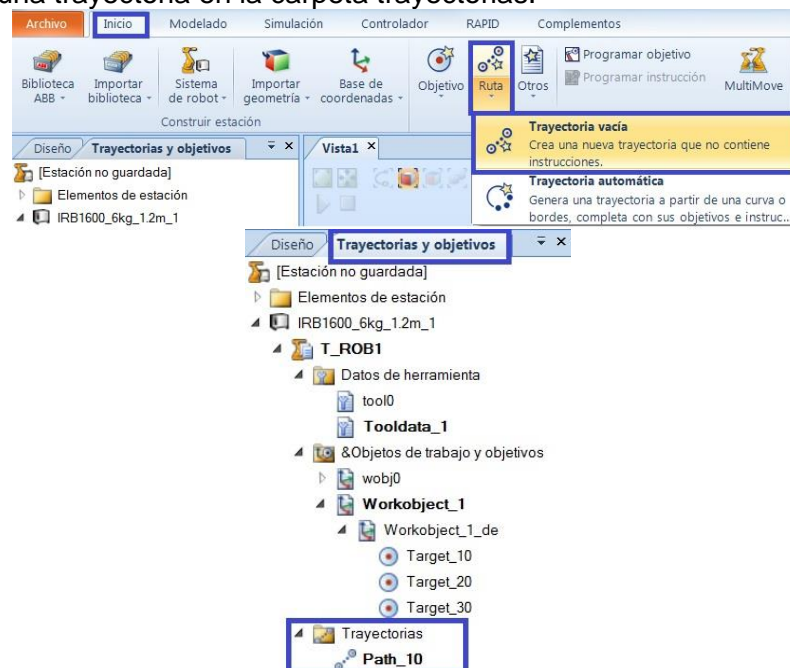
Así podemos programar tantos objetivos como queramos, que se situaran en el objeto de trabajo y con la herramienta que tenemos seleccionada previamente.



Haciendo clic con el botón derecho en cualquiera de los objetivos programados, se nos abrirá un panel con distintas opciones, entre las que está la de “Modificar objetivo” con la que podremos cambiar la posición del objetivo, girarlo o incluso convertir el objetivo en un objeto de trabajo; también encontraremos otras opciones en el panel como la de “Alcanzabilidad” que nos comprobará si el objetivo es alcanzable por el robot y su herramienta, y otra opción muy interesante que es la de “Configuraciones”, esta opción nos ofrecerá la posibilidad de fijar las posiciones de los ejes de una manera u otra para acceder al mismo punto. Eso lo haremos mediante una ventana que se nos abrirá al pulsar “Configuraciones”:

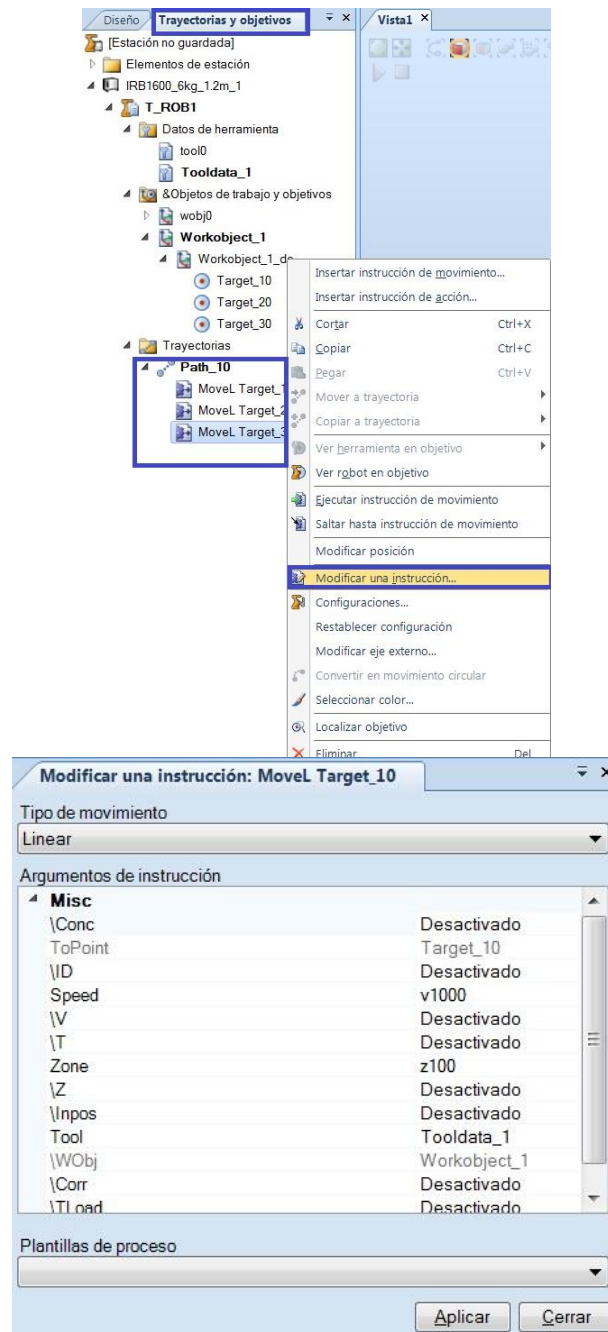


Por último lo que haremos será crear una trayectoria con los objetivos que ya tenemos programados, para ello deberemos de estar en la pestaña “Inicio”, pulsar sobre “Ruta” y seleccionar “Trayectoria vacía”. Esto nos creará una trayectoria en la carpeta trayectorias.

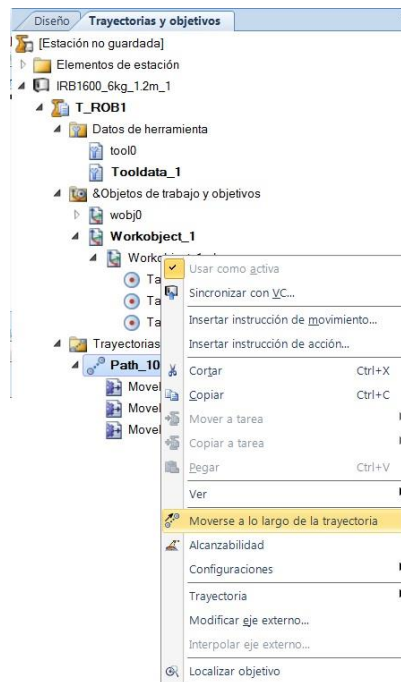


Añadiremos puntos a la trayectoria arrastrando los objetivos hasta la trayectoria de la misma manera que lo hacíamos cuando queríamos conectar la herramienta al robot, esto nos creará una serie de movimientos (MoveL, moveJ, etc) hasta esos puntos.

Si hacemos clic con el botón derecho del ratón sobre estos movimientos de la trayectoria tendremos la opción de modificar la instrucción (cambiar velocidad, cambiar tipo de movimiento, etc).



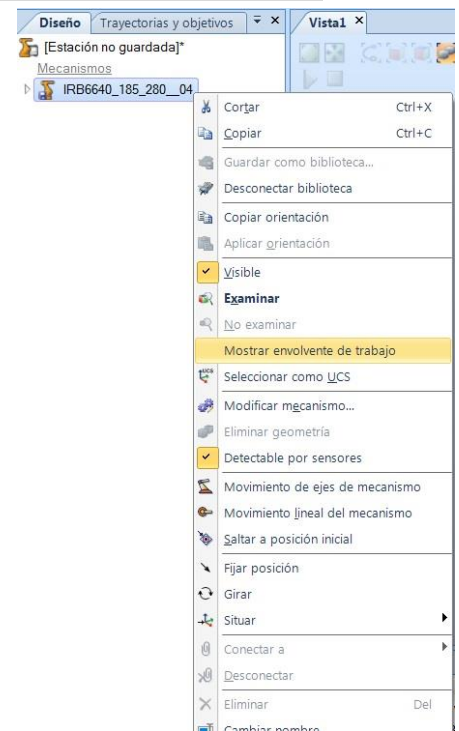
Si deseamos ver el movimiento del robot a lo largo de la trayectoria, lo único que tendríamos que hacer sería pulsar con el botón derecho del ratón sobre el nombre de la trayectoria y seleccionar “Moverse a lo largo de la trayectoria”.



Si modificásemos algún objetivo tendríamos que acceder de nuevo a las configuraciones de dicho objetivo y darle una nueva configuración a los ejes, sino no podrá realizar el movimiento.

Mostrar volumen de trabajo

Puede ser muy útil a la hora de situar el robot o elementos de la estación, dentro del volumen de trabajo del robot, es decir, dentro de un rango en el que se puedan programar objetivos. Para ver este volumen de trabajo haremos clic con el botón derecho del ratón sobre el nombre del robot, situado en la pestaña diseño, y se nos desplegará un panel en el que podremos seleccionar la opción “Mostrar envoltorio de trabajo”.

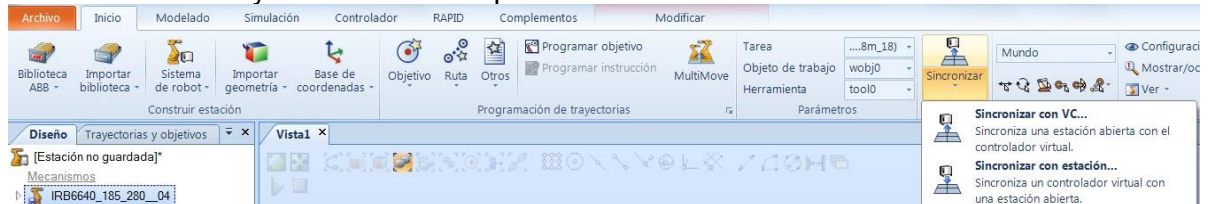




6. Programación en RAPID

Una vez aprendidos todos los objetivos y trayectorias nos interesa desarrollar un programa en RAPID, para ello RobotStudio nos permite que podamos pasar los objetivos y trayectorias desde la ventana de simulación a un programa RAPID.

Para hacer esto, en la pestaña “Inicio” deberemos de pulsar el botón “Sincronizar..” y tendremos dos opciones: “Con VC” o “Con estación”.

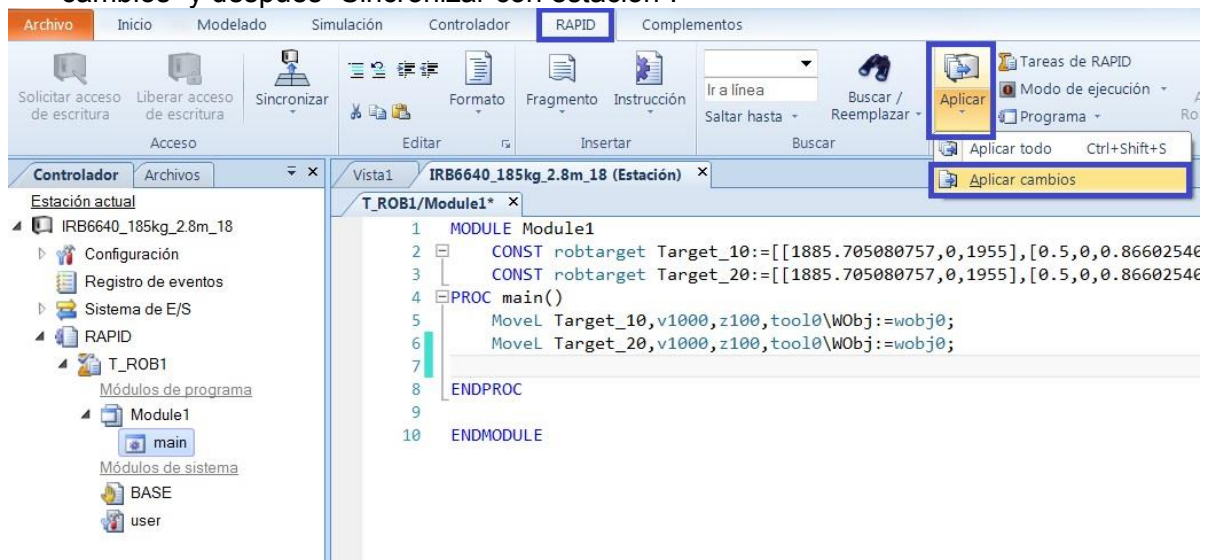


Si estamos en la ventana de simulación y hemos introducido cambios desde la ventana de simulación lo que deberemos hacer será “Sincronizar con VC” para que esos cambios se vean reflejados en el programa RAPID.

Por el contrario si hemos realizado los cambios desde la pestaña “RAPID”, es decir, hemos introducido modificaciones en el programa y queremos que se vea reflejado en la ventana de simulación, deberemos de “Sincronizar con estación”.

En la ventana RAPID podemos hacer modificaciones sobre el programa, para que estas sean aplicadas y se vean en la simulación deberemos de, después de haber hecho los cambios, seleccionar la opción “Aplicar cambios”.

Si deseamos ver nuestra simulación completa, deberemos de crear un main() en el programa RAPID y escribir ahí todo lo que queramos que sea nuestro programa. Una vez hecho esto, deberemos de lo primero “Aplicar cambios” y después “Sincronizar con estación”.



Hecho esto anterior, podemos entrar en la pestaña “Simulación” y darle al PLAY para ver nuestra simulación.



Trabajo Fin de Grado

Autor: Gonzalo Muínelo Garrido

Grado en Ingeniería en
Electrónica Industrial y
Automática

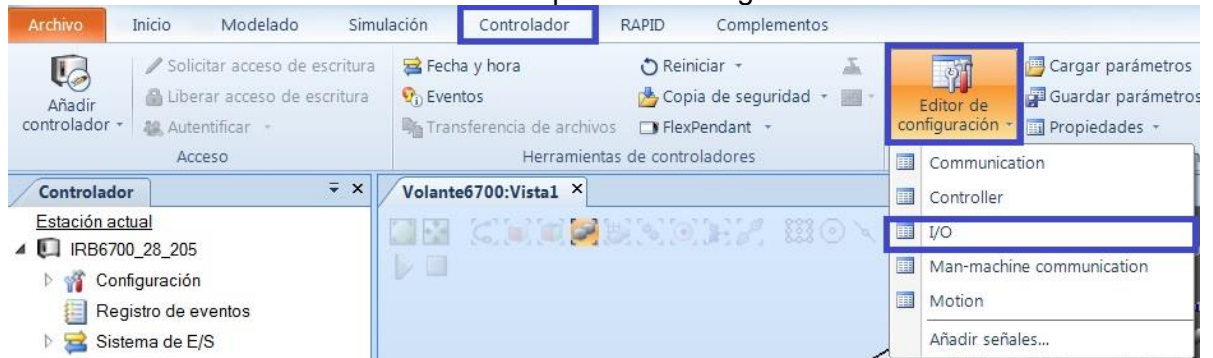
Fecha: 2015



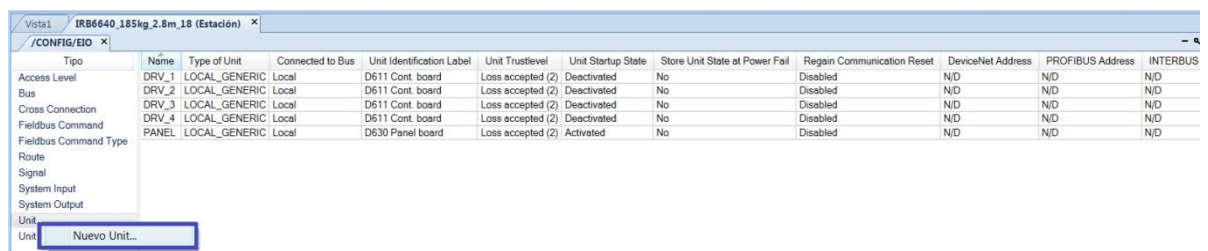


7. Comunicaciones I/O

Podemos configurar entradas y salidas analógicas o digitales, con las que posteriormente secuenciar o controlar parte de la simulación. Lo primero que tenemos que hacer es crear una unidad en la que meteremos nuestras señales. Debemos de entrar en el apartado “Configuración I/O”.



Se nos abrirá una nueva ventana con distintos apartados.



En “Tipo” haremos clic con el botón derecho en “Unit” y seleccionaremos “Nuevo Unit”, se nos abrirá una nueva ventana emergente y la configuraremos de la siguiente manera.



Nombre	Valor	Información
Name	Salidas	Cambiado
Type of Unit	Virtual	Cambiado
Connected to Bus	Virtual1	Cambiado
Unit Identification Label		
Unit Trustlevel	Error when lost (1)	
Unit Startup State	<input type="radio"/> Deactivated <input checked="" type="radio"/> Activated	
Store Unit State at Power Fail	<input type="radio"/> Yes <input checked="" type="radio"/> No	
Regain Communication Reset	<input type="radio"/> Enabled <input checked="" type="radio"/> Disabled	

Value (RAPID)
Los cambios no entrarán en vigor hasta que realice un reinicio del controlador en caliente.
El número máximo de caracteres es 32.

Aceptar Cancelar

Ya tenemos creado nuestra unidad con el nombre que hayamos querido darle. En esta unidad meteremos todas nuestras señales. Ahora deberemos de crear las señales, para ello haremos lo mismo que antes pero haciendo clic con el botón derecho en "Signal", y seleccionaremos "Nuevo Signal"

Tipo	Name	Type of Signal	Assigned to Unit	Signal Identification Label	Unit Mapping	Category	Access Level	Default Value	Signal Value at System Failure an
Access Level	AS1	Digital Input	PANEL	Automatic Stop chain(X5.11 to X5.6) and (X5.9 to X5.1)	13	safety	INTERNAL	0	N/D
Access Level	AS2	Digital Input	PANEL	Automatic Stop chain backup(X5.5 to X5.6) and (X5.3 to X5.1)	14	safety	INTERNAL	0	N/D
Bus	AUTO1	Digital Input	PANEL	Automatic Mode(X9.6)	5	safety	INTERNAL	0	N/D
Cross Connection	AUTO2	Digital Input	PANEL	Automatic Mode backup(X9.2)	6	safety	INTERNAL	0	N/D
Fieldbus Command	CH1	Digital Input	PANEL	Run Chain 1	22	safety	INTERNAL	0	N/D
Fieldbus Command Type	CH2	Digital Input	PANEL	Run Chain 2	23	safety	INTERNAL	0	N/D
Route	DRV1BRAKE	Digital Output	DRV_1	Brake-release coil	2	safety	INTERNAL	0	Keep Current Value (no change)
Signal	Nuevo Signal...		DRV_1	Brake Feedback(X3.6) at Contactor Board	11	safety	INTERNAL	0	N/D
System Input			DRV_1	Brake Voltage OK	15	safety	INTERNAL	0	N/D
System Output	DRV1CHAIN1	Digital Output	DRV_1	Chain 1 Interlocking Circuit	0	safety	INTERNAL	0	Keep Current Value (no change)
Unit	DRV1CHAIN2	Digital Output	DRV_1	Chain 2 Interlocking Circuit	1	safety	INTERNAL	0	Keep Current Value (no change)
Unit Type	DRV1EXTCONT	Digital Input	DRV_1	External customer contactor (X2d) at Contactor Board	4	safety	INTERNAL	0	N/D
	DRV1FAN1	Digital Input	DRV_1	Drive Unit FAN1(X10.3 to X10.4) at Contactor Board	9	safety	INTERNAL	0	N/D
	DRV1FAN2	Digital Input	DRV_1	Drive Unit FAN2(X11.3 to X11.4) at Contactor Board	10	safety	INTERNAL	0	N/D
	DRV1K1	Digital Input	DRV_1	Contactor K1 Read Back chain 1	2	safety	INTERNAL	0	N/D
	DRV1K2	Digital Input	DRV_1	Contactor K2 Read Back chain 2	3	safety	INTERNAL	0	N/D
	DRV1LIM1	Digital Input	DRV_1	Limit Switch 1 (X2a) at Contactor Board	0	safety	INTERNAL	0	N/D
	DRV1LIM2	Digital Input	DRV_1	Limit Switch 2 (X2b) at Contactor Board	1	safety	INTERNAL	0	N/D
	DRV1PANCH1	Digital Input	DRV_1	Drive Voltage contactor coil 1	5	safety	INTERNAL	0	N/D
	DRV1PANCH2	Digital Input	DRV_1	Drive Voltage contactor coil 2	6	safety	INTERNAL	0	N/D
	DRV1PTCEXT	Digital Input	DRV_1	External Motor temperature(X2d.1 to X2d.2)	8	safety	INTERNAL	0	N/D
	DRV1PTCINT	Digital Input	DRV_1	Motor temperature warning(X5.1 to X5.3) at Contactor Board	7	safety	INTERNAL	0	N/D
	DRV1SPEED	Digital Input	DRV_1	Speed Signal(X1.7) at Contactor Board	12	safety	INTERNAL	0	N/D
	DRV1TEST1	Digital Input	DRV_1	Run chain 1 glitch test	13	safety	INTERNAL	0	N/D

Al hacer clic en "Nuevo Signal" se nos abrirá una ventana emergente con la que podremos configurar nuestra señal.



Nombre	Valor	Información
Name	Prueba1	Cambiado
Type of Signal	Digital Output	Cambiado
Assigned to Unit		
Signal Identification Label		
Category		
Access Level		
Default Value	0	
Signal Value at System Failure and Power Fail	<input type="radio"/> Set the Default Value <input checked="" type="radio"/> Keep Current Value (no change)	
Store Signal Value at Power Fail	<input type="radio"/> Yes <input checked="" type="radio"/> No	
Invert Physical Value	<input type="radio"/> Yes <input checked="" type="radio"/> No	

Value (Cadena)
Los cambios no entrarán en vigor hasta que realice un reinicio del controlador en caliente.
El número máximo de caracteres es 5.

Aceptar Cancelar

Seleccionamos el tipo de señal, y seguimos configurando. En este caso hemos configurado una Digital Output.

Nombre	Valor	Información
Name	prueba	Cambiado
Type of Signal	Digital Output	Cambiado
Assigned to Unit	salidas	Cambiado
Signal Identification Label		
Unit Mapping	0	Cambiado
Category		
Access Level	All	Cambiado
Default Value	0	
Signal Value at System Failure and Power Fail	<input type="radio"/> Set the Default Value <input checked="" type="radio"/> Keep Current Value (no change)	
Store Signal Value at Power Fail	<input type="radio"/> Yes <input checked="" type="radio"/> No	
Invert Physical Value	<input type="radio"/> Yes <input checked="" type="radio"/> No	

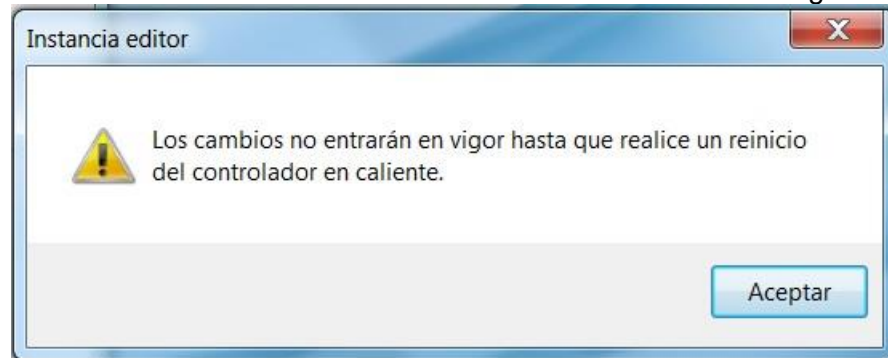
Value (RAPID)
Los cambios no entrarán en vigor hasta que realice un reinicio del controlador en caliente.
El número máximo de caracteres es 32.

Aceptar Cancelar

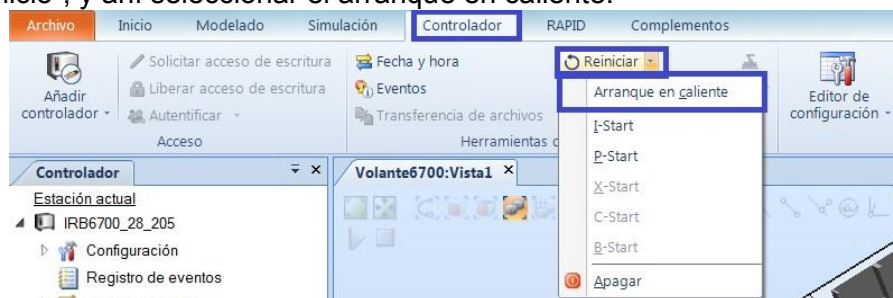
En el apartado Unit Mapping, a cada señal le iremos dando un valor "0, 1, 2,3...,16".



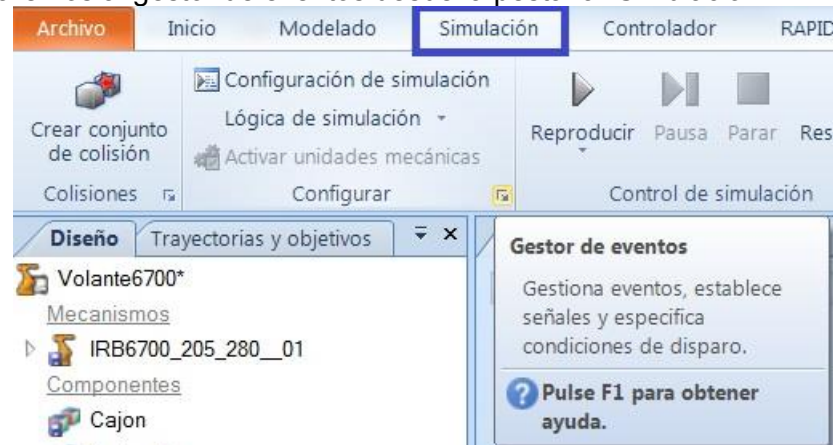
Con cada cambio se nos abrirá una ventana de aviso como la siguiente.



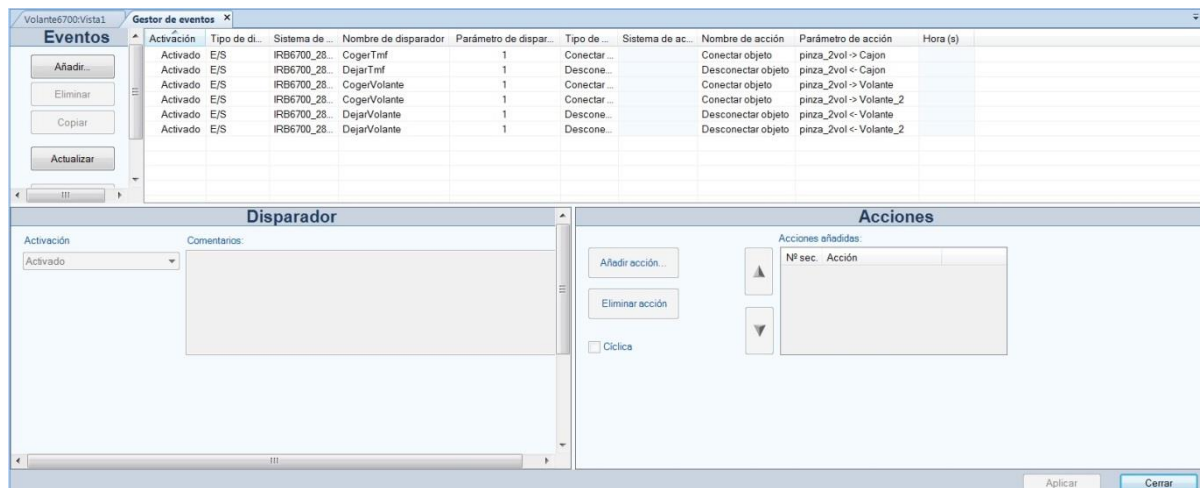
Esto significa que la unidad y las señales que hemos creado no podrán ser utilizadas hasta que no realicemos un “arranque en caliente” del controlador. Para hacer el arranque en caliente, deberemos ir a la pestaña “Controlador”, que se encuentra al mismo nivel que la pestaña “Inicio”, y ahí seleccionar el arranque en caliente.



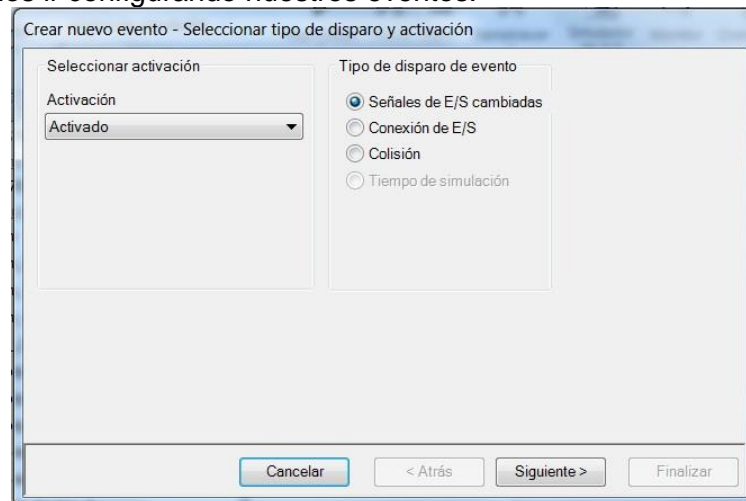
Cuando ya hemos hecho el arranque en caliente, lo que deberemos de hacer será, mediante el gestor de eventos, configurar que acciones queremos que se produzcan con los cambios de entradas y salidas. Entraremos al gestor de eventos desde la pestaña “Simulación”



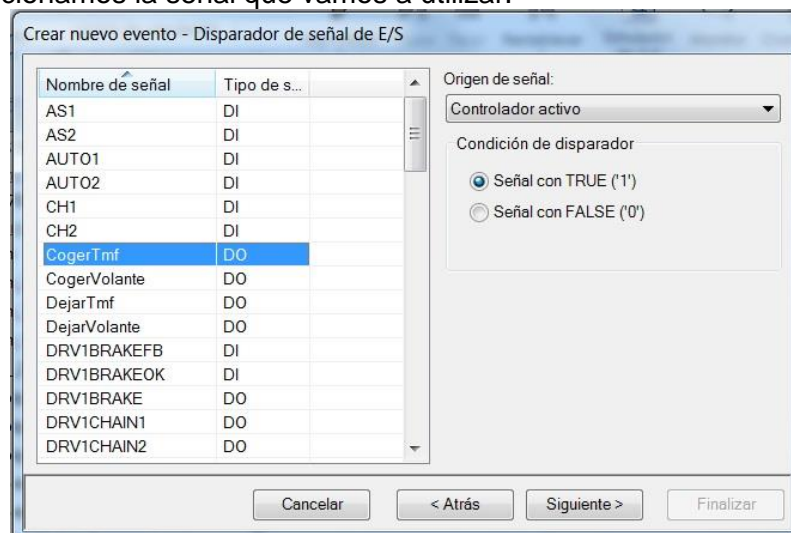
Al abrirse el gestor de eventos, podremos añadir, eliminar, copiar, actualizar, etc eventos.



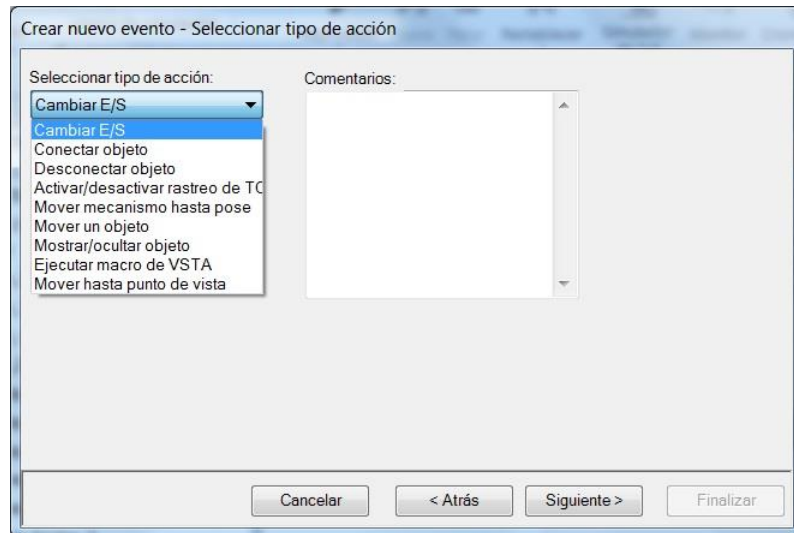
Al pulsar añadir, se nos abrirá una nueva ventana emergente en la que podremos ir configurando nuestros eventos.



Seleccionamos la señal que vamos a utilizar.

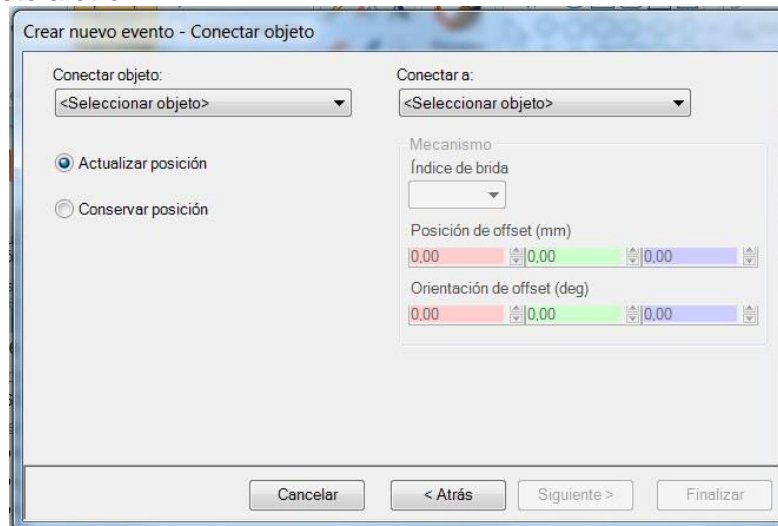


Elegimos el tipo de acción/evento que queremos que se produzca.



Dependiendo del tipo de acción que elijamos, continuaremos a una ventana u otra, sea cual sea, es muy intuitivo ya que será ir seleccionando lo que queremos.

En la siguiente ventana estamos hablando de que la acción es conectar un objeto a otro.



Ya tendríamos creado un evento, y así podríamos crear tantos como queramos.

Ahora para que podamos verlos en la simulación lo que deberemos de hacer será introducir en el programa RAPID instrucciones que hagan que el valor de las entradas o salidas cambien, en este caso particular vemos la utilización del "SetDO", que pone la salida digital a 1, y el RESET que resetea la entrada o salida digital correspondiente.



Trabajo Fin de Grado

Autor: Gonzalo Muinelo Garrido

Grado en Ingeniería en
Electrónica Industrial y
Automática

Fecha: 2015



```
PROC DejarTermoformado1()  
  MoveJ Offs(CogerTermof1,0,0,1000),v1000,z100,Pinza\WObj:=CogerVolantes;  
  MoveL CogerTermof1,v1000,z100,Pinza\WObj:=CogerVolantes;  
  WaitTime 3;  
  SetDO CogerTmf,1;  
  Reset CogerTmf;  
  MoveL Offs(CogerTermof1,0,0,1000),v1000,z100,Pinza\WObj:=CogerVolantes;  
  MoveJ Preparación1_4,v1000,z100,Pinza\WObj:=DejarTermf1;  
  MoveJ Offs( CogerTermof1_3,0,0,1000),v1000,z100,Pinza\WObj:=DejarTermf1;  
  MoveL Offs( CogerTermof1_3,0,0,100),v1000,z100,Pinza\WObj:=DejarTermf1;  
  WaitTime 3;  
  SetDO DejarTmf,1;  
  Reset DejarTmf;  
  MoveL Offs( CogerTermof1_3,0,0,1000),v1000,z100,Pinza\WObj:=DejarTermf1;  
ENDPROC
```



8. Diseño de Hardware

En este apartado detallaremos como crear hardware desde RobotStudio, ya sea una mesa, una pieza, etc.

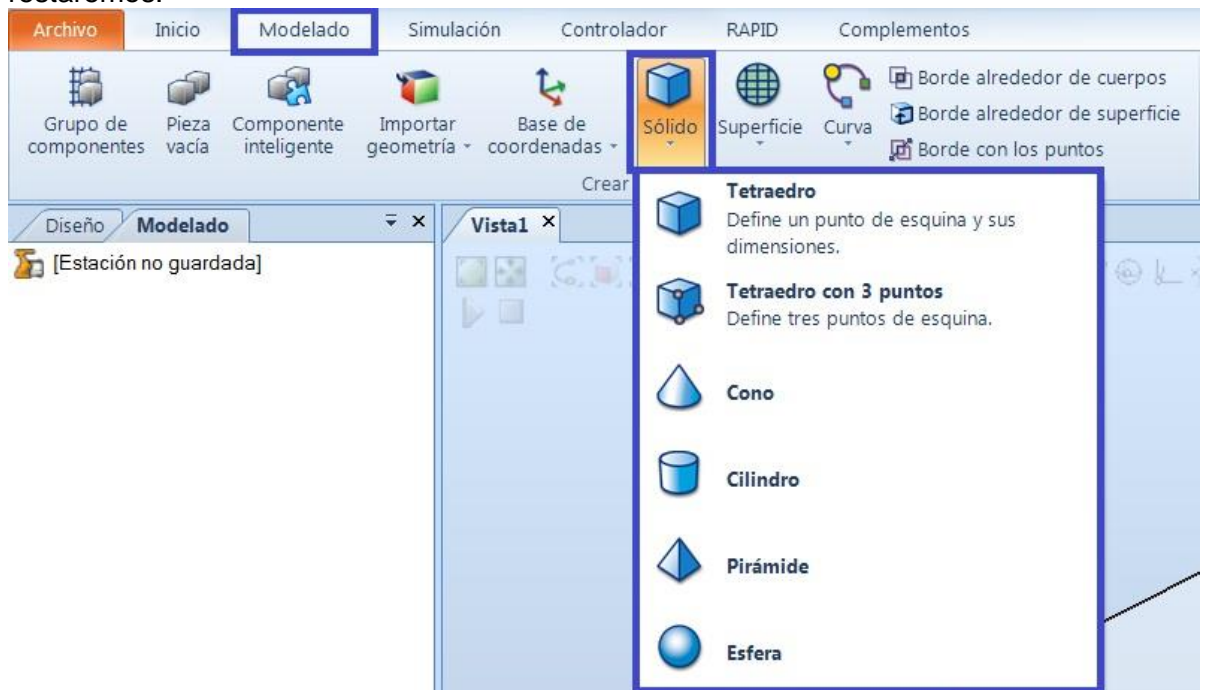
Tendremos que entrar en la pestaña “Modelado” para poder crear nuestros elementos.

Dentro de la pestaña “Modelado” tenemos muchas opciones, pero las más interesantes para nosotros son las marcadas.



A la hora de empezar deberemos introducir un sólido con la geometría base de nuestro elemento. Esto podremos hacerlo desde la opción “Sólido”.


En el ejemplo siguiente introduciremos un tetraedro, un cilindro y los restaremos.



Seleccionamos el sólido (en este caso un tetraedro) y se nos abrirá una ventana en la que podremos elegir las coordenadas donde queremos situar nuestro sólido, y sus dimensiones.



Crear tetraedro



Referencia
Mundo

Punto de esquina (mm)
0,00 0,00 0,00

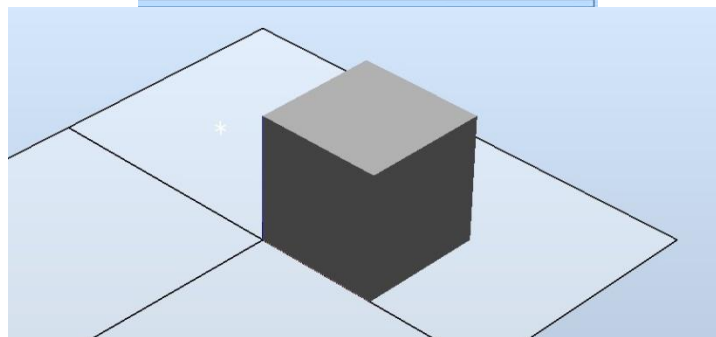
Orientación (deg)
0,00 0,00 0,00

Longitud (mm)
0,00

Anchura (mm)
0,00

Altura (mm)
0,00

Borrar Crear Cerrar



Ahora introducimos un cilindro.

Crear cilindro



Referencia
Mundo

Punto central de la base (mm)
0,00 0,00 0,00

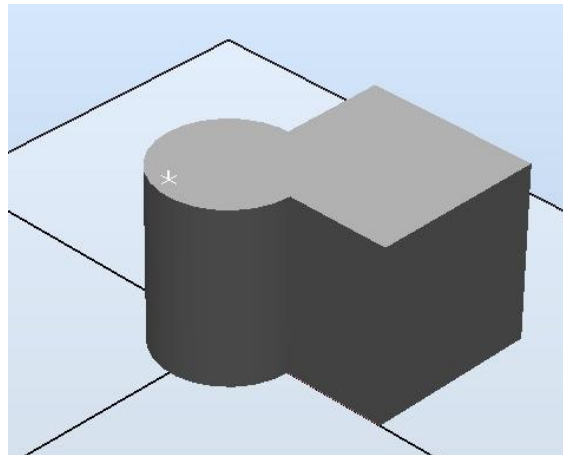
Orientación (deg)
0,00 0,00 0,00

Radio (mm)
200,00

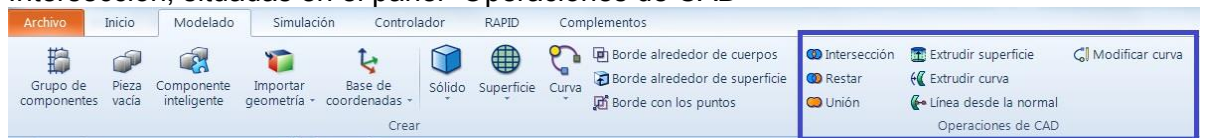
Diámetro (mm)
400,00

Altura (mm)
500

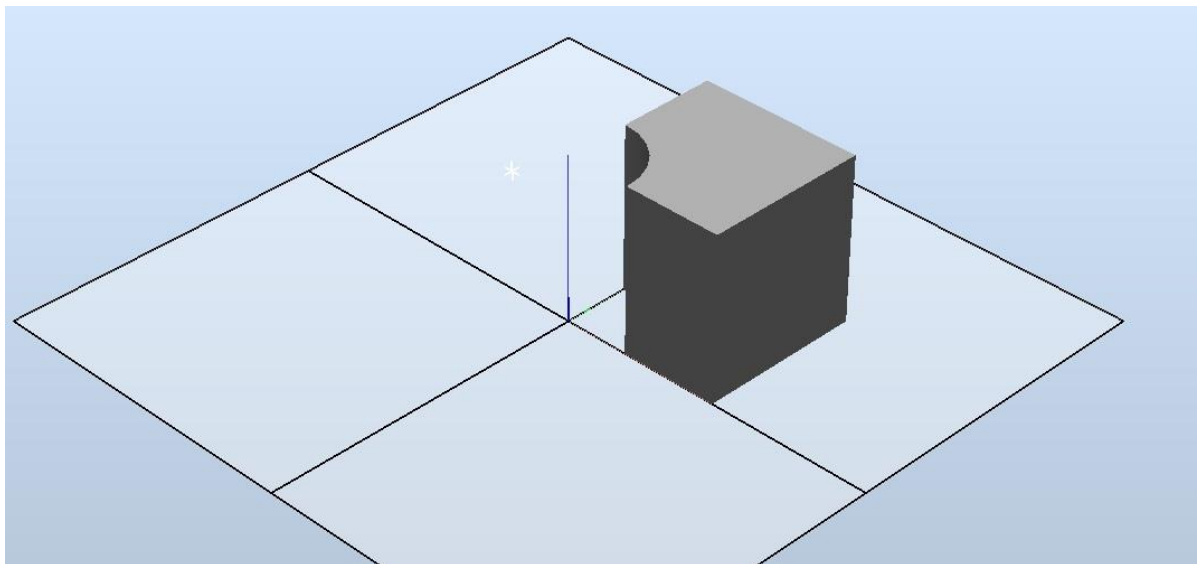
Borrar Crear Cerrar



Ahora utilizaremos una de las funciones de Restar, Unir o crear una Intersección, situadas en el panel “Operaciones de CAD”



En este caso utilizaremos la función restar, seleccionaremos clicando sobre ellos, el primer elemento y después el elemento a restar, hasta que aparezcan en la ventana.



Situando los elementos y utilizando las opciones de intersección, restar o unión, creamos los elementos que necesitemos en nuestro sistema.



9. Componente inteligente

Se define como componente inteligente a un objeto de RobotStudio, con o sin representación gráfica, que presenta el comportamiento que puede o desea implementarse.

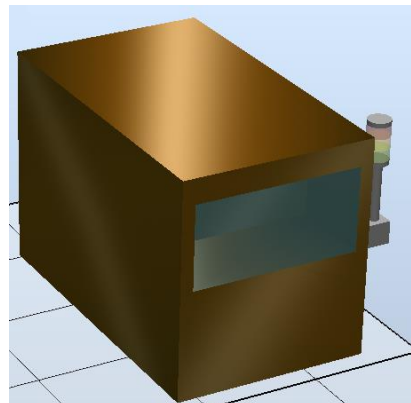
También es importante el término “agregación”, que es el proceso de conectar varios componentes inteligentes mediante enlazamientos y/o conexiones con el fin de implementar un comportamiento más complejo.

En la pestaña modelado seleccionaremos la opción de componente inteligente:

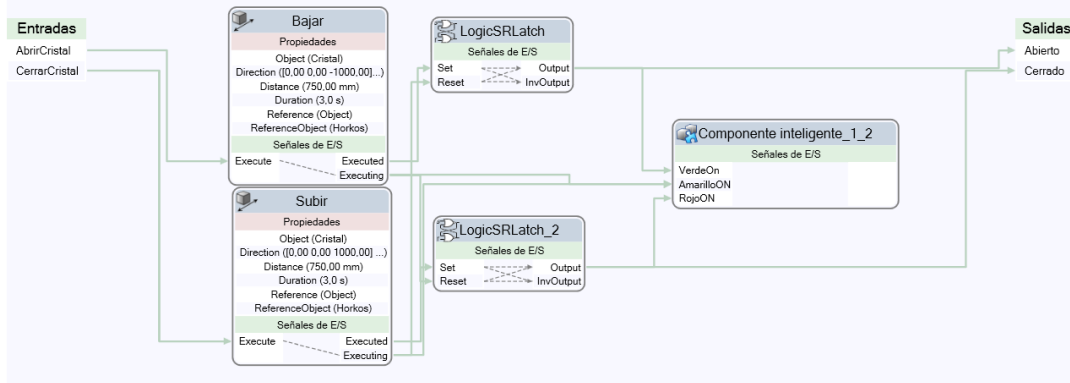


A partir de ahí seleccionaremos la opción de abrir componente e introduciremos todos los elementos necesarios para conseguir el movimiento deseado; conectaremos estos elementos o bloques en la pestaña diseño.

En el siguiente ejemplo tenemos una maquina que abre y cierra la puerta y enciende una luz en función de si la puerta esta abierta, cerrada o en movimiento.



Este elemento se ha creado con las herramientas de modelado de RobotStudio, y se ha hecho como los anteriores con un componente inteligente, que llamaremos componente inteligente de nivel superior (ya que este cuenta con otro componente inteligente de nivel inferior), cuyo esquema de bloques para su funcionamiento es el siguiente:



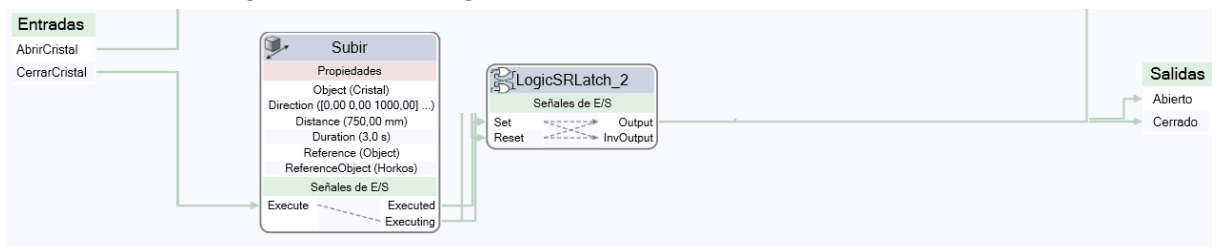
Podríamos diferenciar tres líneas de funcionamiento. La primera de ellas se encargará de abrir la puerta de la máquina, la segunda se encargará de cerrar la puerta, y la tercera será un componente inteligente de nivel inferior, que se encargará de encender y apagar luces de puerta abierta, puerta cerrada, abriendo o cerrando puerta.

4. LINEA ABERTURA DE LA PUERTA:



El funcionamiento será muy sencillo, la entrada del componente inteligente “AbrirCristal” será la encargada de hacer ejecutar la acción de bajar el cristal, y una vez hecho mandará a través del LogicSRLatch la señal de que la puerta está abierta.

5. LINEA CIERRE DE LA PUERTA:

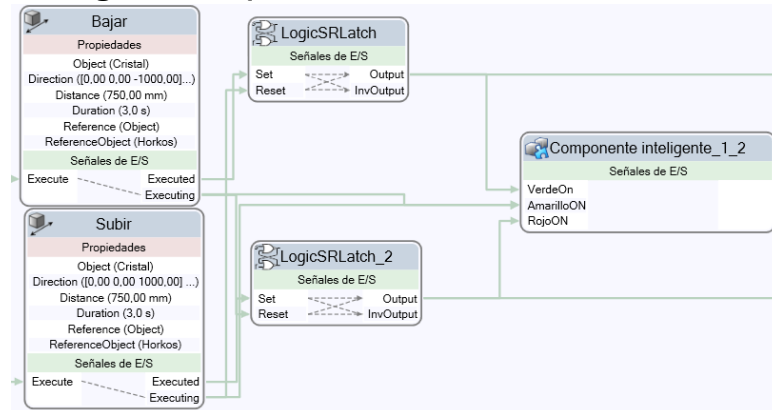


El funcionamiento será también muy sencillo, la entrada del componente inteligente “CerrarCristal” será la encargada de hacer ejecutar la acción de subir el cristal, y una vez hecho mandará a través del LogicSRLatch la señal de que la puerta está Cerrada.



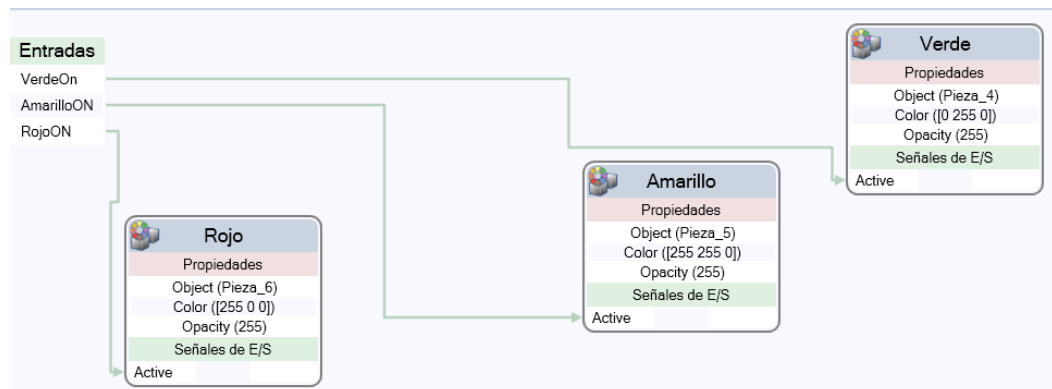
6. LINEA LUCES DE ESTADO:

Este componente es algo especial, ya que interactúa con las dos líneas de las que hemos hablado anteriormente, siendo a su vez un componente inteligente de nivel inferior; una explicación sencilla sería decir que es un componente inteligente dentro de otro componente inteligente de nivel superior. La relación que tiene con las dos líneas anteriores queda reflejada en el siguiente esquema de funcionamiento:



Tiene tres entradas que recibirán la señal de que la puerta se ha bajado o se ha subido (VerdeOn y RojoOn respectivamente), y la señal que la puerta está bajándose o subiéndose (AmarilloON).

Entrando en la propia configuración de este componente inteligente de luces de estado nos encontramos el siguiente esquema:



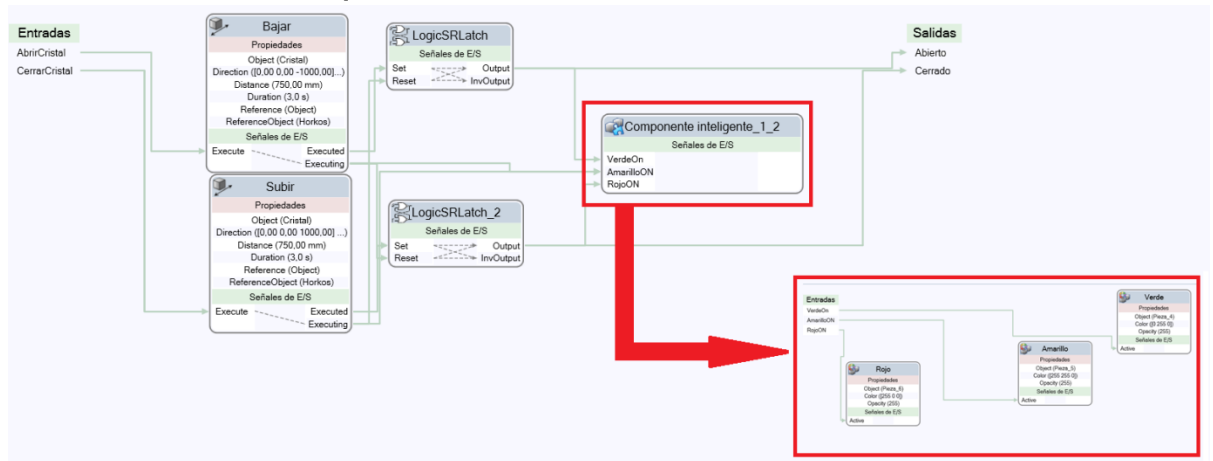
Estos bloques activarán el cambio de color a un color opaco en función de las entradas recibidas, que están reflejadas en la siguiente tabla:

Conexiones de E/S

Objeto de origen	Señal de origen	Objeto de destino	Señal de destino
Componente inteligente_1_2	RojoON	Rojo	Active
Componente inteligente_1_2	AmarilloON	Amarillo	Active
Componente inteligente_1_2	VerdeOn	Verde	Active



Con la finalidad de que se entienda mejor, vamos a ver un esquema del componente inteligente de la maquina 1 de mecanizado, indicando donde se sitúa el esquema de funcionamiento de las luces de estado.



Para entender mejor la relación de entradas y salidas entre bloques (flechas verdes), es decir, que salidas corresponden con que entradas, las conexiones están reflejadas en la siguiente tabla:

Conexiones de E/S

Objeto de origen	Señal de origen	Objeto de destino	Señal de destino
Bajar	Executing	Componente inteligente_1_2	AmarilloON
Subir	Executing	Componente inteligente_1_2	AmarilloON
Horkos	AbriendoCristal	Bajar	Execute
Horkos	CerrandoCristal	Subir	Execute
Subir	Executed	LogicSRLatch_2	RojoON
Bajar	Executed	LogicSRLatch	Set
LogicSRLatch	Output	Componente inteligente_1_2	VerdeON
LogicSRLatch	Output	Horkos	Abierto
LogicSRLatch_2	Output	Horkos	Cerrado
Bajar	Executing	LogicSRLatch_2	Reset
Subir	Executing	LogicSRLatch	Reset

(*)Existen dos señales digitales de salida llamadas Abierto y Cerrado que serán señales de salida hacia el exterior del componente inteligente indicando que la maquina está abierta o la maquina está cerrada.



Trabajo Fin de Grado

Autor: Gonzalo Muínelo Garrido

Grado en Ingeniería en
Electrónica Industrial y
Automática

Fecha: 2015

