TRABAJO 2ª Ev. ROBÓTICA INDUSTRIAL

**I.E.S Santiago Apóstol**

**2º A.R.I.**

**Promoción 2013-2015**

**18/03/2015**

Fº Javier Pérez González

En el presente documento se describen las actividades realizadas en la asignatura de Robótica Industrial a lo largo de todo el año. Vienen descritos tanto programas en código Rapid, programa en célula entrenadora en SFC y conexionado de E/S de Robot

ÍNDICE

[1.- Programa maquina en SFC asociada a robot 2](#_Toc414995137)

[2.- Explicación de ejercicio 3](#_Toc414995138)

[3.- Programa en código Rapid 4](#_Toc414995139)

[3.1.- Explicación 4](#_Toc414995140)

[3.2.- Programa principal ( main() ) 7](#_Toc414995141)

[3.3.- Programa completo 8](#_Toc414995142)

[3.4.- Diagrama de Flujo del Programa 18](#_Toc414995143)

[18](#_Toc414995144)

[4.- Como guardar un punto con FlexPendant 19](#_Toc414995145)

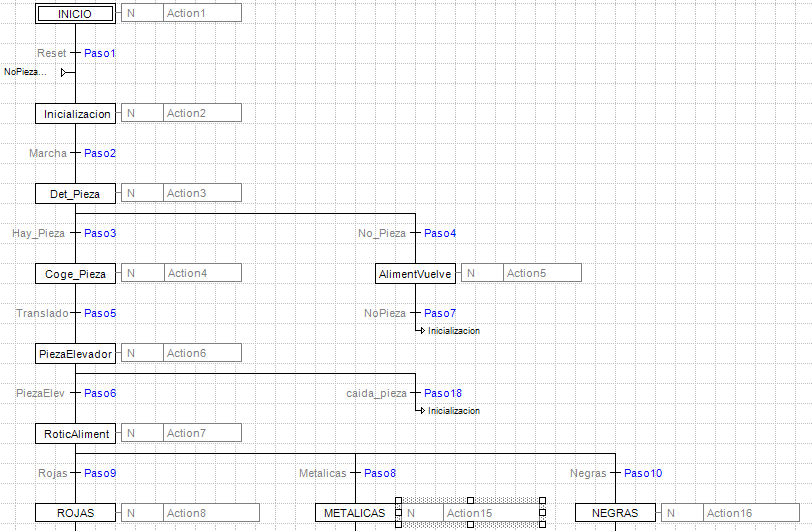
[5.- Simulación de Entradas y Salidas en RobotStudio 23](#_Toc414995146)

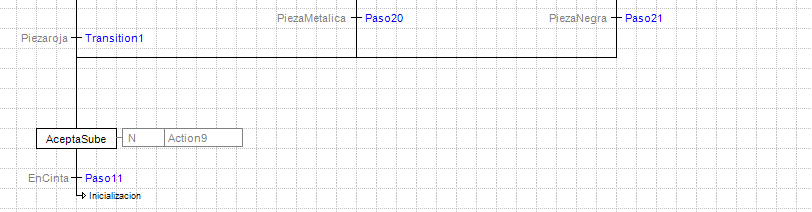
[6.- Como cargar un programa desde RobotStudio 27](#_Toc414995147)

[7.- Variar las coordenadas de un Punto en Ejecución 28](#_Toc414995148)

[7.- Plano de Conexiones del Robot 29](#_Toc414995149)

# 1.- Programa maquina en SFC asociada a robot



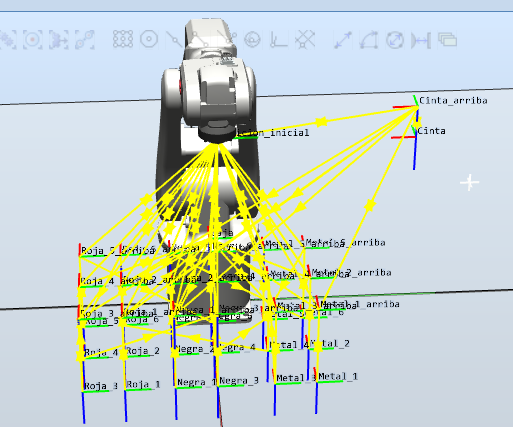


Hemos elegido las siguientes salidas del autómata de la máquina para la activación de las entradas del robot:

* Salida 2.00 corresponde a la entrada DI10\_1 del Robot. Se activará al detectarse **pieza roja.**
* Salida 2.01 corresponde a la entrada DI10\_3 del Robot. Se activará al detectarse **pieza negra.**
* Salida 2.02 corresponde a la entrada DI10\_2 del Robot. Se activará al detectarse **pieza metal.**
* Salida 3.00 corresponde a la entrada DI10\_4 del Robot. Se activará al detectarse **llegada de pieza** a su posición de recogida.

# 2.- Explicación de ejercicio

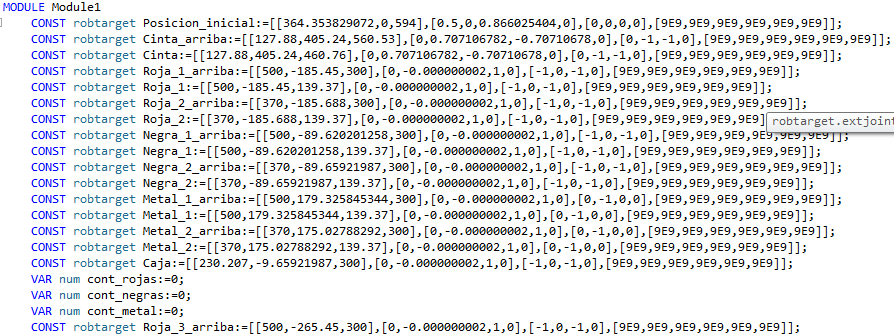
Nuestro debe recoger las piezas que le van llegando en la cinta y completar 3 pallets de 6 piezas cada uno todas. Cada pallet tendrá piezas de un determinado color: Roja, Negras o Metálicas. En el momento en el que nos llegué una pieza cuyo pallet esté completo con 6 unidades, debemos entrar en un bucle de espera para poder retirarlas.

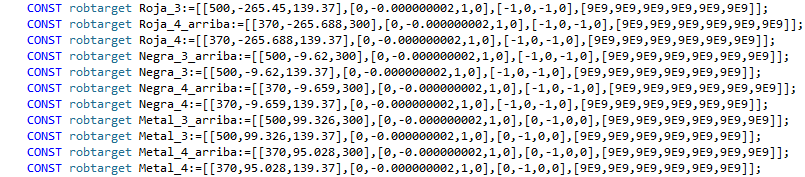


# 3.- Programa en código Rapid

## 3.1.- Explicación

En la primera parte del programa tenemos la cabecera del mismo. En ella tenemos los puntos que hemos elegido y las variables que usamos.





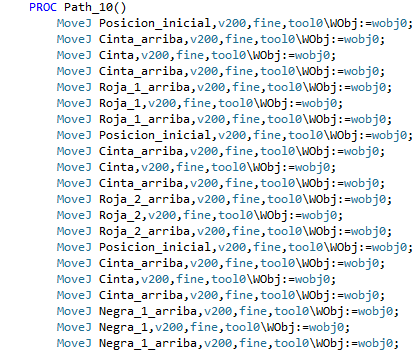
En cada punto, la primera terna de valores que tenemos corresponden con las coordenadas XYZ del punto que corresponda.

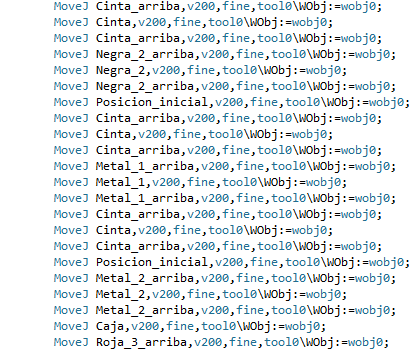


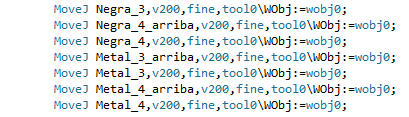
Los restantes datos son valores que corresponden a la configuración de ejes del robot en dicho punto. En nuestro caso son los que vienen indicados al crear un punto en RobotStudio.



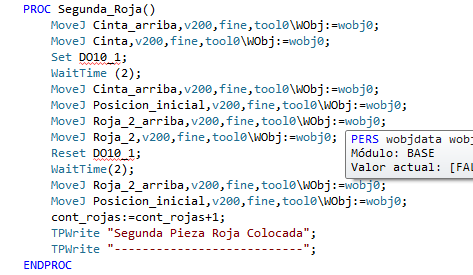
En líneas posteriores tenemos la subrutina Path\_10() , que contiene todos los movimientos que nuestro robot va ejecutar en el transcurso de nuestro programa. Solamente los contiene nunca se va ejecutar esta rutina.







Ahora vendrían las funciones que si se ejecutan. Son funciones que son llamadas cuando sea necesaria su ejecución, siempre desde **main().** Al poner en modo continuo main() se ejecuta cíclicamente. En su contenido están instrucciones de movimiento, de control de la herramienta, de bucle de espera y escritura en FlexPendant. Pondremos como ejemplo una determinada pieza de un determinado color. La función **Segunda\_Roja** coge la segunda pieza roja de la cinta y la lleva a su correspondiente sitio en el pallet.



En cada **instrucción de movimiento** vienen los siguientes datos:

* Tipo de movimiento.
  + MoveL: desplazamiento hasta el punto en línea recta.
  + MoveC: desplazamiento describiendo un círculo.
  + MoveJ: desplazamiento rápido, sin garantizar la trayectoria seguida (sin coordinación de velocidad entre los ejes del robot).
* El nombre del punto.
* Velocidad.
  + Dato tipo ‘SpeedData’: indica velocidad deseada de traslación y de rotación.
* Precisión.
  + Dato tipo ‘ZoneData’: indica la precisión con la que hay que alcanzar la posición pedida antes de continuar el movimiento hasta la posición siguiente. En nuestro caso ‘fine’ significa precisión absoluta.
* Herramienta.
  + Dato tipo ‘ToolData’: describe la herramienta acoplada al robot: dimensiones para calcular trayectorias más peso y momentos de inercia para calcular esfuerzos. Nosotros no tenemos insertada ninguna, es decir, ‘tool0’.
* Plano de trabajo.
  + Especifica el plano de trabajo sobre el que estamos trabajando. Nosotros trabajamos sobre el ‘wobj0’ , que es el plano de trabajo por defecto.

Las **instrucciones de control de herramienta** son en nuestro caso simplemente set y reset. Estas activan y desactivan respectivamente la salida del robot que corresponda.





Podemos sacar mensajes por pantalla en el FlexPendant mediante la instrucción “TPWrite”. Introducimos entre comillas el mensaje que queremos mostrar.



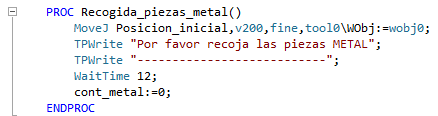
En este tipo de funciones tenemos también instrucciones de incremento de contadores. Cada vez que una pieza de un determinado color se coloca en su sitio se incrementa en uno el contador.



También tenemos instrucciones de espera necesarias para que aguante en un punto un determinado tiempo, y poder activar y desactivar nuestra pinza con el robot parado. Entre paréntesis introducimos el tiempo en segundos.

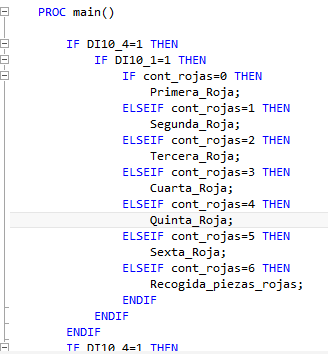


Ahora vamos a explicar la función **Recogida\_piezas\_rojas/negras/metal ().** Esta función se va ejecutar cuando hayamos completado un pallet de un determinado color y recibamos otra pieza de ese color. Entonces llevaremos el robot a la posición, mostraremos un mensaje en el FlexPendant, ejecutaremos una instrucción de espera necesaria para la retirada del pallet y pondremos el contador correspondiente a 0.



## 3.2.- Programa principal ( main() )

La estructura del programa es mediante condicionales anidados. En la primera comprobación ejecutaremos una sentencia **IF** para ver si hay una pieza a la espera de ser recogida por el robot en la cinta. Si la hay, pasamos a otro **IF** interior que comprueba de que color es la pieza.



Una vez que el programa ha encontrado de que color es la pieza, tenemos que ver en que estado se encuentra el contador de piezas que corresponda. Para no tener que anidar multitud de instrucciones IF, utilizaremos **ELSEIF ( sino si…)** para no hacer demasiado complejo el código.

## 3.3.- Programa completo

MODULE Module1

CONST robtarget Posicion\_inicial:=[[364.353829072,0,594],[0.000000001,0,1,0],[0,0,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Cinta\_arriba:=[[153.19,407.4,540],[0,0.707106782,-0.70710678,0],[0,-1,-1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Cinta:=[[153.19,407.4,472.46],[0,0.707106782,-0.70710678,0],[0,-1,-1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Roja\_1\_arriba:=[[500,-185.45,300],[0,-0.000000002,1,0],[-1,0,-1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Roja\_1:=[[500,-185.45,135.37],[0,-0.000000002,1,0],[-1,0,-1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Roja\_2\_arriba:=[[390,-185.688,300],[0,-0.000000002,1,0],[-1,0,-1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Roja\_2:=[[390,-185.688,135.37],[0,-0.000000002,1,0],[-1,0,-1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Negra\_1\_arriba:=[[500,-89.620201258,300],[0,-0.000000002,1,0],[-1,0,-1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Negra\_1:=[[500,-89.620201258,135.37],[0,-0.000000002,1,0],[-1,0,-1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Negra\_2\_arriba:=[[390,-89.65921987,300],[0,-0.000000002,1,0],[-1,0,-1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Negra\_2:=[[390,-89.65921987,135.37],[0,-0.000000002,1,0],[-1,0,-1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Metal\_1\_arriba:=[[500,179.325845344,300],[0,-0.000000002,1,0],[0,-1,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Metal\_1:=[[500,179.325845344,135.37],[0,-0.000000002,1,0],[0,-1,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Metal\_2\_arriba:=[[390,175.02788292,300],[0,-0.000000002,1,0],[0,-1,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Metal\_2:=[[390,175.02788292,135.37],[0,-0.000000002,1,0],[0,-1,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Caja:=[[230.207,-9.65921987,300],[0,-0.000000002,1,0],[-1,0,-1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];

VAR num cont\_rojas:=0;

VAR num cont\_negras:=0;

VAR num cont\_metal:=0;

CONST robtarget Roja\_3\_arriba:=[[500,-265.45,300],[0,-0.000000002,1,0],[-1,0,-1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Roja\_3:=[[500,-265.45,135.37],[0,-0.000000002,1,0],[-1,0,-1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Roja\_4\_arriba:=[[390,-265.688,300],[0,-0.000000002,1,0],[-1,0,-1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Roja\_4:=[[390,-265.688,135.37],[0,-0.000000002,1,0],[-1,0,-1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Negra\_3\_arriba:=[[500,-9.62,300],[0,-0.000000002,1,0],[-1,0,-1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Negra\_3:=[[500,-9.62,135.37],[0,-0.000000002,1,0],[-1,0,-1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Negra\_4\_arriba:=[[390,-9.659,300],[0,-0.000000002,1,0],[-1,0,-1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Negra\_4:=[[390,-9.659,135.37],[0,-0.000000002,1,0],[-1,0,-1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Metal\_3\_arriba:=[[500,99.326,300],[0,-0.000000002,1,0],[0,-1,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Metal\_3:=[[500,99.326,135.37],[0,-0.000000002,1,0],[0,-1,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Metal\_4\_arriba:=[[390,95.028,300],[0,-0.000000002,1,0],[0,-1,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Metal\_4:=[[390,95.028,135.37],[0,-0.000000002,1,0],[0,-1,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Roja\_6:=[[280,-185.688,135.37],[0,-0.000000002,1,0],[-1,0,-1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Roja\_6\_arriba:=[[280,-185.688,300],[0,-0.000000002,1,0],[-1,0,-1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Roja\_5:=[[280,-265.688,135.37],[0,-0.000000002,1,0],[-1,0,-1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Roja\_5\_arriba:=[[280,-265.688,300],[0,-0.000000002,1,0],[-1,0,-1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Negra\_6:=[[280,-9.659,135.37],[0,-0.000000002,1,0],[-1,0,-1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Negra\_6\_arriba:=[[280,-9.659,300],[0,-0.000000002,1,0],[-1,0,-1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Negra\_5:=[[280,-89.65921987,135.37],[0,-0.000000002,1,0],[-1,0,-1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Negra\_5\_arriba:=[[280,-89.65921987,300],[0,-0.000000002,1,0],[-1,0,-1,0],[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Metal\_6:=[[280,175.02788292,135.37],[0,-0.000000002,1,0],[0,-1,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Metal\_6\_arriba:=[[280,175.02788292,300],[0,-0.000000002,1,0],[0,-1,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Metal\_5:=[[280,95.028,135.37],[0,-0.000000002,1,0],[0,-1,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];

CONST robtarget Metal\_5\_arriba:=[[280,95.028,300],[0,-0.000000002,1,0],[0,-1,0,0],[9E9,9E9,9E9,9E9,9E9,9E9]];

PROC Path\_10()

MoveJ Posicion\_inicial,v200,fine,tool0\WObj:=wobj0;

MoveJ Cinta\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Cinta,v200,fine,tool0\WObj:=wobj0;

MoveJ Cinta\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Roja\_1\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Roja\_1,v200,fine,tool0\WObj:=wobj0;

MoveJ Roja\_1\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Posicion\_inicial,v200,fine,tool0\WObj:=wobj0;

MoveJ Cinta\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Cinta,v200,fine,tool0\WObj:=wobj0;

MoveJ Cinta\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Roja\_2\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Roja\_2,v200,fine,tool0\WObj:=wobj0;

MoveJ Roja\_2\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Posicion\_inicial,v200,fine,tool0\WObj:=wobj0;

MoveJ Cinta\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Cinta,v200,fine,tool0\WObj:=wobj0;

MoveJ Cinta\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Negra\_1\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Negra\_1,v200,fine,tool0\WObj:=wobj0;

MoveJ Negra\_1\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Posicion\_inicial,v200,fine,tool0\WObj:=wobj0;

MoveJ Cinta\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Cinta,v200,fine,tool0\WObj:=wobj0;

MoveJ Cinta\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Negra\_2\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Negra\_2,v200,fine,tool0\WObj:=wobj0;

MoveJ Negra\_2\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Posicion\_inicial,v200,fine,tool0\WObj:=wobj0;

MoveJ Cinta\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Cinta,v200,fine,tool0\WObj:=wobj0;

MoveJ Cinta\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Metal\_1\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Metal\_1,v200,fine,tool0\WObj:=wobj0;

MoveJ Metal\_1\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Cinta\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Cinta,v200,fine,tool0\WObj:=wobj0;

MoveJ Cinta\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Posicion\_inicial,v200,fine,tool0\WObj:=wobj0;

MoveJ Metal\_2\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Metal\_2,v200,fine,tool0\WObj:=wobj0;

MoveJ Metal\_2\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Caja,v200,fine,tool0\WObj:=wobj0;

MoveJ Roja\_3\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Roja\_3,v200,fine,tool0\WObj:=wobj0;

MoveJ Roja\_4\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Roja\_4,v200,fine,tool0\WObj:=wobj0;

MoveJ Negra\_3\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Negra\_3,v200,fine,tool0\WObj:=wobj0;

MoveJ Negra\_4\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Negra\_4,v200,fine,tool0\WObj:=wobj0;

MoveJ Metal\_3\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Metal\_3,v200,fine,tool0\WObj:=wobj0;

MoveJ Metal\_4\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Metal\_4,v200,fine,tool0\WObj:=wobj0;

MoveJ Roja\_6,v200,fine,tool0\WObj:=wobj0;

MoveJ Roja\_6\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Roja\_5,v200,fine,tool0\WObj:=wobj0;

MoveJ Roja\_5\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Negra\_6,v200,fine,tool0\WObj:=wobj0;

MoveJ Negra\_6\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Negra\_5,v200,fine,tool0\WObj:=wobj0;

MoveJ Negra\_5\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Metal\_6,v200,fine,tool0\WObj:=wobj0;

MoveJ Metal\_6\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Metal\_5,v200,fine,tool0\WObj:=wobj0;

MoveJ Metal\_5\_arriba,v200,fine,tool0\WObj:=wobj0;

ENDPROC

!A partir de aqui empiezan las funciones a realizar

!Primera\_Roja recoge la primera pieza roja de la cinta y la lleva a su posición

PROC Primera\_Roja()

Reset DO10\_1;

MoveL Cinta\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveL Cinta,v200,fine,tool0\WObj:=wobj0;

Set DO10\_1;

WaitTime (2); !Es necesario colocar una pequena espera tanto al cerrar como al abrir la pinza

MoveL Cinta\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Posicion\_inicial,v200,fine,tool0\WObj:=wobj0;

MoveL Roja\_1\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveL Roja\_1,v200,fine,tool0\WObj:=wobj0;

Reset DO10\_1;

WaitTime(2);

MoveL Roja\_1\_arriba,v200,fine,tool0\WObj:=wobj0;

!Al dejar la pieza en su sitio volvemos a la posicion inicial

MoveL Posicion\_inicial,v200,fine,tool0\WObj:=wobj0;

cont\_rojas:=cont\_rojas+1;

TPWrite "Primera Pieza Roja Colocada";

TPWrite "---------------------------";

ENDPROC

PROC Segunda\_Roja()

MoveJ Cinta\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Cinta,v200,fine,tool0\WObj:=wobj0;

Set DO10\_1;

WaitTime (2);

MoveJ Cinta\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Posicion\_inicial,v200,fine,tool0\WObj:=wobj0;

MoveJ Roja\_2\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Roja\_2,v200,fine,tool0\WObj:=wobj0;

Reset DO10\_1;

WaitTime(2);

MoveJ Roja\_2\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Posicion\_inicial,v200,fine,tool0\WObj:=wobj0;

cont\_rojas:=cont\_rojas+1;

TPWrite "Segunda Pieza Roja Colocada";

TPWrite "---------------------------";

ENDPROC

PROC Tercera\_Roja()

MoveJ Cinta\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Cinta,v200,fine,tool0\WObj:=wobj0;

Set DO10\_1;

WaitTime (2);

MoveJ Cinta\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Posicion\_inicial,v200,fine,tool0\WObj:=wobj0;

MoveJ Roja\_3\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Roja\_3,v200,fine,tool0\WObj:=wobj0;

Reset DO10\_1;

WaitTime(2);

MoveJ Roja\_3\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Posicion\_inicial,v200,fine,tool0\WObj:=wobj0;

cont\_rojas:=cont\_rojas+1;

TPWrite "Tercera Pieza Roja Colocada";

TPWrite "---------------------------";

ENDPROC

PROC Cuarta\_Roja()

MoveJ Cinta\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Cinta,v200,fine,tool0\WObj:=wobj0;

Set DO10\_1;

WaitTime (2);

MoveJ Cinta\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Posicion\_inicial,v200,fine,tool0\WObj:=wobj0;

MoveJ Roja\_4\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Roja\_4,v200,fine,tool0\WObj:=wobj0;

Reset DO10\_1;

WaitTime(2);

MoveJ Roja\_4\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Posicion\_inicial,v200,fine,tool0\WObj:=wobj0;

cont\_rojas:=cont\_rojas+1;

TPWrite "Cuarta Pieza Roja Colocada";

TPWrite "---------------------------";

ENDPROC

PROC Quinta\_Roja()

MoveJ Cinta\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Cinta,v200,fine,tool0\WObj:=wobj0;

Set DO10\_1;

WaitTime (2);

MoveJ Cinta\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Posicion\_inicial,v200,fine,tool0\WObj:=wobj0;

MoveJ Roja\_5\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Roja\_5,v200,fine,tool0\WObj:=wobj0;

Reset DO10\_1;

WaitTime(2);

MoveJ Roja\_5\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Posicion\_inicial,v200,fine,tool0\WObj:=wobj0;

cont\_rojas:=cont\_rojas+1;

TPWrite "Quinta Pieza Roja Colocada";

TPWrite "---------------------------";

ENDPROC

PROC Sexta\_Roja()

MoveJ Cinta\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Cinta,v200,fine,tool0\WObj:=wobj0;

Set DO10\_1;

WaitTime (2);

MoveJ Cinta\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Posicion\_inicial,v200,fine,tool0\WObj:=wobj0;

MoveJ Roja\_6\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Roja\_6,v200,fine,tool0\WObj:=wobj0;

Reset DO10\_1;

WaitTime(2);

MoveJ Roja\_6\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Posicion\_inicial,v200,fine,tool0\WObj:=wobj0;

cont\_rojas:=cont\_rojas+1;

TPWrite "Sexta Pieza Roja Colocada";

TPWrite "---------------------------";

ENDPROC

PROC Recogida\_piezas\_rojas()

MoveJ Posicion\_inicial,v200,fine,tool0\WObj:=wobj0;

TPWrite "Por favor recoja las piezas ROJAS";

TPWrite "---------------------------";

WaitTime 12;

cont\_rojas:=0;

ENDPROC

PROC Recogida\_piezas\_negras()

MoveJ Posicion\_inicial,v200,fine,tool0\WObj:=wobj0;

TPWrite "Por favor recoja las piezas NEGRAS";

TPWrite "---------------------------";

WaitTime 12;

cont\_negras:=0;

ENDPROC

PROC Recogida\_piezas\_metal()

MoveJ Posicion\_inicial,v200,fine,tool0\WObj:=wobj0;

TPWrite "Por favor recoja las piezas METAL";

TPWrite "---------------------------";

WaitTime 12;

cont\_metal:=0;

ENDPROC

!Primera\_Negra recoge la primera pieza negra de la cinta y la lleva a su posicion

PROC Primera\_Negra()

Reset DO10\_1;

MoveJ Cinta\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Cinta,v200,fine,tool0\WObj:=wobj0;

Set DO10\_1;

WaitTime (2);

MoveJ Cinta\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Posicion\_inicial,v200,fine,tool0\WObj:=wobj0;

MoveJ Negra\_1\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Negra\_1,v200,fine,tool0\WObj:=wobj0;

Reset DO10\_1;

WaitTime(2);

MoveJ Negra\_1\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Posicion\_inicial,v200,fine,tool0\WObj:=wobj0;

cont\_negras:=cont\_negras+1;

TPWrite "Primera Pieza Negra Colocada";

TPWrite "---------------------------";

ENDPROC

PROC Segunda\_Negra()

MoveJ Cinta\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Cinta,v200,fine,tool0\WObj:=wobj0;

Set DO10\_1;

WaitTime (2);

MoveJ Cinta\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Posicion\_inicial,v200,fine,tool0\WObj:=wobj0;

MoveJ Negra\_2\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Negra\_2,v200,fine,tool0\WObj:=wobj0;

Reset DO10\_1;

WaitTime(2);

MoveJ Negra\_2\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Posicion\_inicial,v200,fine,tool0\WObj:=wobj0;

cont\_negras:=cont\_negras+1;

TPWrite "Segunda Pieza Negra Colocada";

TPWrite "---------------------------";

ENDPROC

PROC Tercera\_Negra()

MoveJ Cinta\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Cinta,v200,fine,tool0\WObj:=wobj0;

Set DO10\_1;

WaitTime (2);

MoveJ Cinta\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Posicion\_inicial,v200,fine,tool0\WObj:=wobj0;

MoveJ Negra\_3\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Negra\_3,v200,fine,tool0\WObj:=wobj0;

Reset DO10\_1;

WaitTime(2);

MoveJ Negra\_3\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Posicion\_inicial,v200,fine,tool0\WObj:=wobj0;

cont\_negras:=cont\_negras+1;

TPWrite "Tercera Pieza Negra Colocada";

TPWrite "---------------------------";

ENDPROC

PROC Cuarta\_Negra()

MoveJ Cinta\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Cinta,v200,fine,tool0\WObj:=wobj0;

Set DO10\_1;

WaitTime (2);

MoveJ Cinta\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Posicion\_inicial,v200,fine,tool0\WObj:=wobj0;

MoveJ Negra\_4\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Negra\_4,v200,fine,tool0\WObj:=wobj0;

Reset DO10\_1;

WaitTime(2);

MoveJ Negra\_4\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Posicion\_inicial,v200,fine,tool0\WObj:=wobj0;

cont\_negras:=cont\_negras+1;

TPWrite "Cuarta Pieza Negra Colocada";

TPWrite "---------------------------";

ENDPROC

PROC Quinta\_Negra()

MoveJ Cinta\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Cinta,v200,fine,tool0\WObj:=wobj0;

Set DO10\_1;

WaitTime (2);

MoveJ Cinta\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Posicion\_inicial,v200,fine,tool0\WObj:=wobj0;

MoveJ Negra\_5\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Negra\_5,v200,fine,tool0\WObj:=wobj0;

Reset DO10\_1;

WaitTime(2);

MoveJ Negra\_5\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Posicion\_inicial,v200,fine,tool0\WObj:=wobj0;

cont\_negras:=cont\_negras+1;

TPWrite "Quinta Pieza Negra Colocada";

TPWrite "---------------------------";

ENDPROC

PROC Sexta\_Negra()

MoveJ Cinta\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Cinta,v200,fine,tool0\WObj:=wobj0;

Set DO10\_1;

WaitTime (2);

MoveJ Cinta\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Posicion\_inicial,v200,fine,tool0\WObj:=wobj0;

MoveJ Negra\_6\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Negra\_6,v200,fine,tool0\WObj:=wobj0;

Reset DO10\_1;

WaitTime(2);

MoveJ Negra\_6\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Posicion\_inicial,v200,fine,tool0\WObj:=wobj0;

cont\_negras:=cont\_negras+1;

TPWrite "Sexta Pieza Negra Colocada";

TPWrite "---------------------------";

ENDPROC

PROC Primera\_Metal()

Reset DO10\_1;

MoveJ Cinta\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Cinta,v200,fine,tool0\WObj:=wobj0;

Set DO10\_1;

WaitTime (2);

MoveJ Cinta\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Posicion\_inicial,v200,fine,tool0\WObj:=wobj0;

MoveJ Metal\_1\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Metal\_1,v200,fine,tool0\WObj:=wobj0;

Reset DO10\_1;

WaitTime(2);

MoveJ Metal\_1\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Posicion\_inicial,v200,fine,tool0\WObj:=wobj0;

cont\_metal:=cont\_metal+1;

TPWrite "Primera Pieza Metal Colocada";

TPWrite "---------------------------";

ENDPROC

PROC Segunda\_Metal()

MoveJ Cinta\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Cinta,v200,fine,tool0\WObj:=wobj0;

Set DO10\_1;

WaitTime (2);

MoveJ Cinta\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Posicion\_inicial,v200,fine,tool0\WObj:=wobj0;

MoveJ Metal\_2\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Metal\_2,v200,fine,tool0\WObj:=wobj0;

Reset DO10\_1;

WaitTime(2);

MoveJ Metal\_2\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Posicion\_inicial,v200,fine,tool0\WObj:=wobj0;

cont\_metal:=cont\_metal+1;

TPWrite "Segunda Pieza Metal Colocada";

TPWrite "---------------------------";

ENDPROC

PROC Tercera\_Metal()

MoveJ Cinta\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Cinta,v200,fine,tool0\WObj:=wobj0;

Set DO10\_1;

WaitTime (2);

MoveJ Cinta\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Posicion\_inicial,v200,fine,tool0\WObj:=wobj0;

MoveJ Metal\_3\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Metal\_3,v200,fine,tool0\WObj:=wobj0;

Reset DO10\_1;

WaitTime(2);

MoveJ Metal\_3\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Posicion\_inicial,v200,fine,tool0\WObj:=wobj0;

cont\_metal:=cont\_metal+1;

TPWrite "Tercera Pieza Metal Colocada";

TPWrite "---------------------------";

ENDPROC

PROC Cuarta\_Metal()

MoveJ Cinta\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Cinta,v200,fine,tool0\WObj:=wobj0;

Set DO10\_1;

WaitTime (2);

MoveJ Cinta\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Posicion\_inicial,v200,fine,tool0\WObj:=wobj0;

MoveJ Metal\_4\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Metal\_4,v200,fine,tool0\WObj:=wobj0;

Reset DO10\_1;

WaitTime(2);

MoveJ Metal\_4\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Posicion\_inicial,v200,fine,tool0\WObj:=wobj0;

cont\_metal:=cont\_metal+1;

TPWrite "Cuarta Pieza Metal Colocada";

TPWrite "---------------------------";

ENDPROC

PROC Quinta\_Metal()

MoveJ Cinta\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Cinta,v200,fine,tool0\WObj:=wobj0;

Set DO10\_1;

WaitTime (2);

MoveJ Cinta\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Posicion\_inicial,v200,fine,tool0\WObj:=wobj0;

MoveJ Metal\_5\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Metal\_5,v200,fine,tool0\WObj:=wobj0;

Reset DO10\_1;

WaitTime(2);

MoveJ Metal\_5\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Posicion\_inicial,v200,fine,tool0\WObj:=wobj0;

cont\_metal:=cont\_metal+1;

TPWrite "Quinta Pieza Metal Colocada";

TPWrite "---------------------------";

ENDPROC

PROC Sexta\_Metal()

MoveJ Cinta\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Cinta,v200,fine,tool0\WObj:=wobj0;

Set DO10\_1;

WaitTime (2);

MoveJ Cinta\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Posicion\_inicial,v200,fine,tool0\WObj:=wobj0;

MoveJ Metal\_6\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Metal\_6,v200,fine,tool0\WObj:=wobj0;

Reset DO10\_1;

WaitTime(2);

MoveJ Metal\_6\_arriba,v200,fine,tool0\WObj:=wobj0;

MoveJ Posicion\_inicial,v200,fine,tool0\WObj:=wobj0;

cont\_metal:=cont\_metal+1;

TPWrite "Sexta Pieza Metal Colocada";

TPWrite "---------------------------";

ENDPROC

PROC main()

IF DI10\_4=1 THEN

IF DI10\_1=1 THEN

IF cont\_rojas=0 THEN

Primera\_Roja;

ELSEIF cont\_rojas=1 THEN

Segunda\_Roja;

ELSEIF cont\_rojas=2 THEN

Tercera\_Roja;

ELSEIF cont\_rojas=3 THEN

Cuarta\_Roja;

ELSEIF cont\_rojas=4 THEN

Quinta\_Roja;

ELSEIF cont\_rojas=5 THEN

Sexta\_Roja;

ELSEIF cont\_rojas=6 THEN

Recogida\_piezas\_rojas;

ENDIF

ENDIF

ENDIF

IF DI10\_4=1 THEN

IF DI10\_3=1 THEN

IF cont\_negras=0 THEN

Primera\_Negra;

ELSEIF cont\_negras=1 THEN

Segunda\_Negra;

ELSEIF cont\_negras=2 THEN

Tercera\_Negra;

ELSEIF cont\_negras=3 THEN

Cuarta\_Negra;

ELSEIF cont\_negras=4 THEN

Quinta\_Negra;

ELSEIF cont\_negras=5 THEN

Sexta\_Negra;

ELSEIF cont\_negras=6 THEN

Recogida\_piezas\_negras;

ENDIF

ENDIF

ENDIF

IF DI10\_4=1 THEN

IF DI10\_2=1 THEN

IF cont\_metal=0 THEN

Primera\_Metal;

ELSEIF cont\_metal=1 THEN

Segunda\_Metal;

ELSEIF cont\_metal=2 THEN

Tercera\_Metal;

ELSEIF cont\_metal=3 THEN

Cuarta\_Metal;

ELSEIF cont\_metal=4 THEN

Quinta\_Metal;

ELSEIF cont\_metal=5 THEN

Sexta\_Metal;

ELSEIF cont\_metal=6 THEN

Recogida\_piezas\_metal;

ENDIF

ENDIF

ENDIF

ENDPROC

ENDMODULE

## 3.4.- Diagrama de Flujo del Programa

## 

INICIO

Main()

DI10\_4 = 1

(Sensor llegada = 1)

**¿?**

**NO**

**SI**

**SI SI**

Recogida\_piezas\_rojas()

Cont\_rojas=0

DI10\_1 = 1

Roja ¿?

Cont\_rojas=6

¿?

**NO** **NO**

Pieza\_n\_Roja()

Cont\_rojas ++

**SI SI**

Recogida\_piezas\_negras()

Cont\_negras=0

Cont\_negras=6

¿?

DI10\_3 = 1

Negra ¿?

**NO** **NO**

Pieza\_n\_Negra()

Cont\_negras ++

DI10\_2 = 1

Metal ¿?

Recogida\_piezas\_metal()

Cont\_metal=0

Pieza\_n\_Metal()

Cont\_metal ++

Cont\_metal=6

¿?

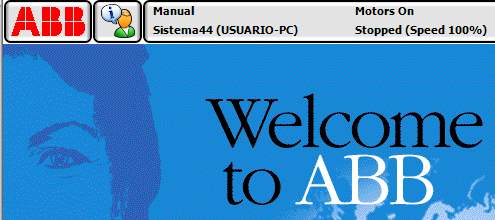
**SI** **SI**

**NO NO**

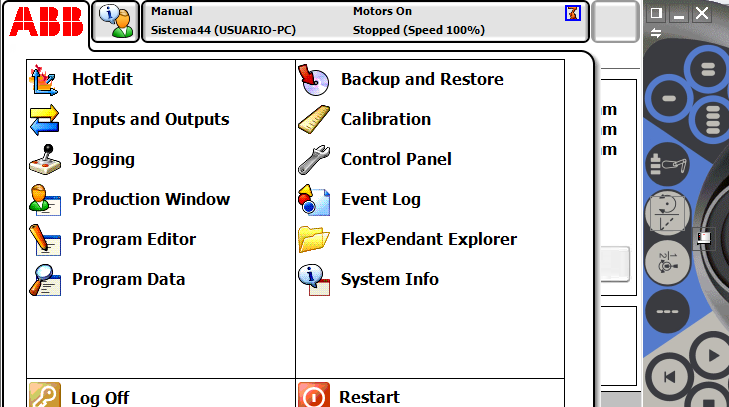
**NO POSIBLE**

# 4.- Como guardar un punto con FlexPendant

Primeramente nos vamos al menú principal pinchando en el logo de ABB en rojo que hay arriba en la página principal.

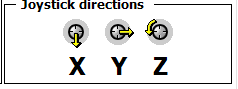


Nos lleva al menú principal y pinchamos en “**Jogging**” para mover el mover al punto que deseemos.

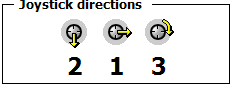


Podemos elegir entre dos tipos de movimientos:

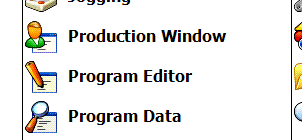
* **Lineal**, moviendo el robot a lo largo de los ejes de coordenadas.

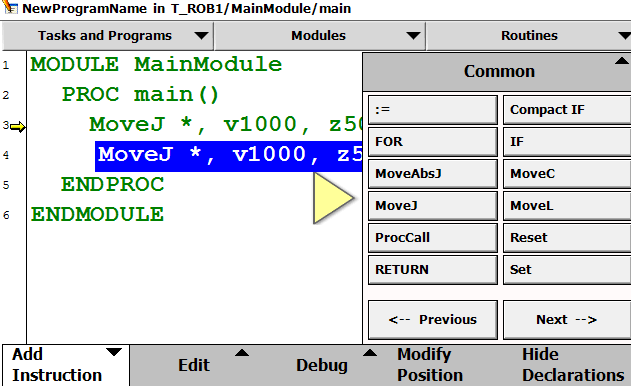
* O variando la posición de los **ejes.** Pulsaremos el botón para alternar entre los trios de ejes.

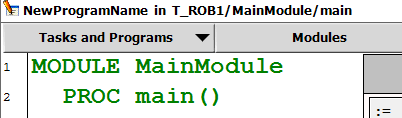
Una vez que hayamos desplazado el robot al punto deseado, nos debemos ir de nuevo al menú principal, pinchando en el logo de ABB de la esquina de arriba a la izquierda, y abrir el **editor de programas**.



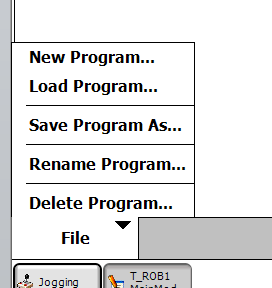
Ya tendríamos el primer movimiento de nuestro programa. Si quisiéramos encontrar las coordenadas de otro punto, moveríamos el robot de nuevo y pincharíamos en “**Add Instruction**”, seleccionamos el tipo de movimiento que queremos y seleccionamos el orden donde queremos que esté la instrucción, si encima o debajo de la anterior.



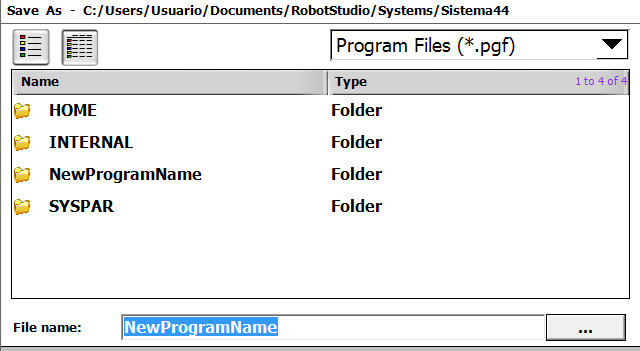
Supongamos que no queremos hallar las coordenadas de más puntos y queremos ver cuales son para utilizarlas en un programa hecho por nosotros. Pinchamos en “**Tasks and Programs**” (Tareas y programas si nuestro FlexPendant está en español).



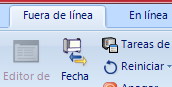
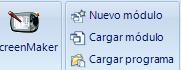
Nos vamos a “**File**” y “**Save Program As**”.



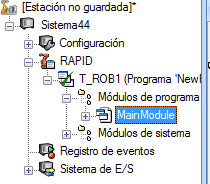
Escribimos el nombre de nuestro programa y el directorio donde queremos guardarlo.



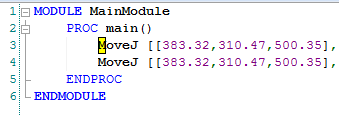
Ya tenemos el programa en nuestro ordenador. Para visualizar las coordenadas XYZ de cada punto, en RobotStudio nos iríamos a la pestaña “**Fuera de línea**” y pincharíamos en “**Cargar Programa**”.

Vamos al directorio donde guardamos anteriormente el programa desde FlexPendant y lo abrimos. Para ver el código nos vamos a la izquierda de nuestra pantalla y pinchamos en “**MainModule**”. Así se nos abrirá nuestro código en el editor de programas.

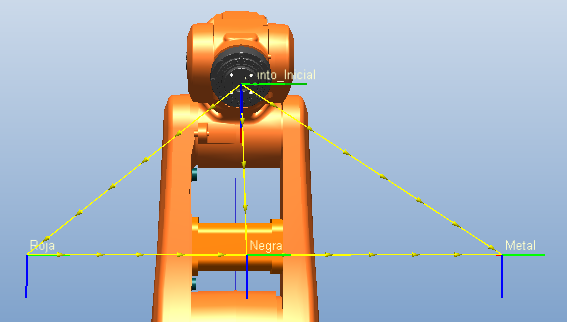


Ya sólo tenemos que mirar la primera lista de valores de cada instrucción de movimiento, que nos indicará las coordenadas XYZ de nuestro punto.

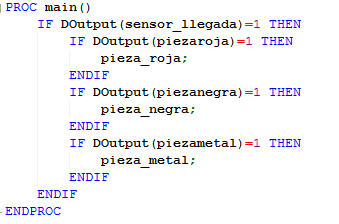


# 5.- Simulación de Entradas y Salidas en RobotStudio

Para la explicación de como se simulan entradas y salidas en RobotStudio, primero vamos a hacernos un sencillo programa con un punto de inicio y tres más que corresponderán a pieza roja, pieza negra y pieza metal.

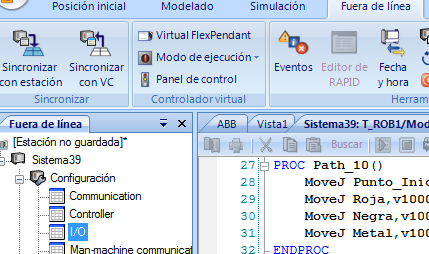


Si se produce la activación del sensor de llegada, nuestro robot se dirigirá a uno de los tres puntos según el color de pieza que esté activado. La parte principal de nuestro programa en código RAPID sería la siguiente:

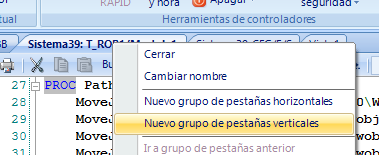


Son una serie de sentencias condicionales que comprueban en primer lugar si se activado “sensor\_llegada”. Si se produce esto, el programa analiza el estado de las entradas de color, es decir, “piezaroja”,”piezanegra”,”piezametal”. Si alguna esta a nivel 1, se ejecuta la función que corresponda, que no es más que el desplazamiento del robot hacia uno de los tres puntos.

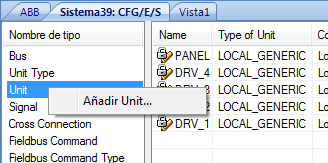
Ahora vamos a ver como se simulan estas entradas. En primer lugar estando en la pestaña “**Fuera de línea**” , abrimos la configuración de nuestro Sistema y hacemos doble click en “**I/O**”.



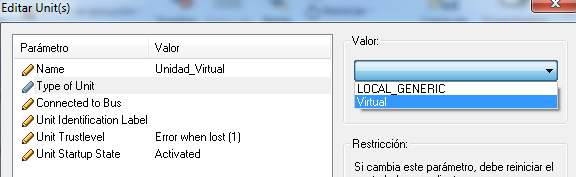
Recomendamos activar la visualización de varias pestañas simultáneas, para poder trabajar más cómodamente. Pinchamos con el botón derecho del ratón en cualquiera de las pestañas y seleccionamos “**Nuevo grupo de pestañas verticales**”.



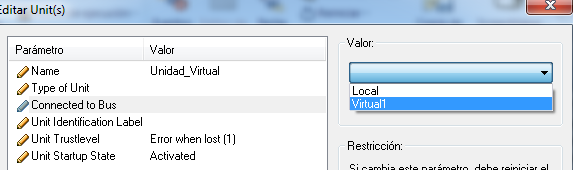
Una vez que tengamos abierta la configuración de I/O , vamos a crear una unidad virtual. Pinchamos con el botón derecho en “**Unit**” y añadimos unidad.



La ventana que se abre ahora es para configurar los parámetros de la unidad. Primeramente nos pedirá el nombre, ponemos el que queramos. En “**Type of Unit**” seleccionamos Virtual.



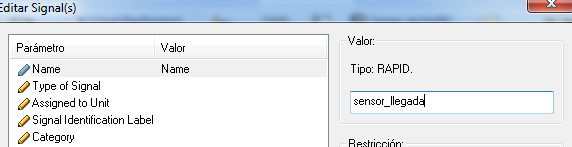
Igualmente en “**Connected to Bus**” seleccionamos Virtual1.



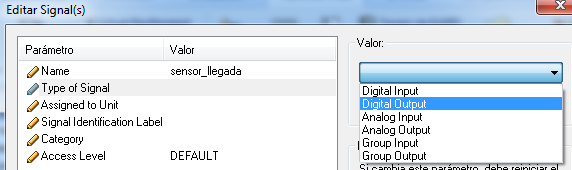
En “**Unit Identification Label**” escribimos la etiqueta que creamos correspondiente. Pulsamos Aceptar y nos indicará que será necesario un reinicio en caliente del sistema, cosa que haremos después.

Ya tenemos creada nuestra unidad, ahora vamos a crear las señales simuladas. Pinchamos con el botón derecho en “**Signal**” y seleccionamos Añadir señal.

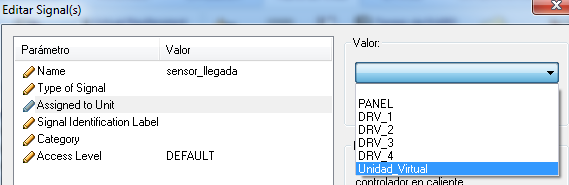
Nos saldrá una ventana similar a la anterior para editar los parámetros de la señal. Ponemos el nombre que queramos, pero con cuidado porque este es el que debemos poner en el código RAPID para comprobar las condiciones. En nuestro caso “sensor\_llegada” por ejemplo.



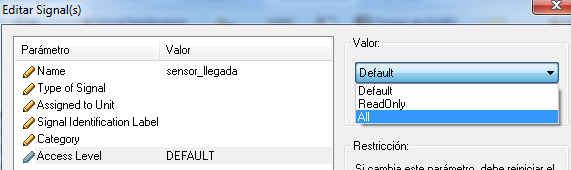
En “**Type of Signal**” ponemos el tipo de señal que sea, en nuestro caso Digital Output.



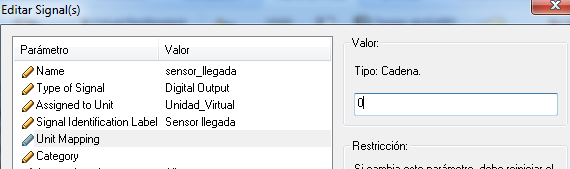
En “Assigned to Unit” debemos poner la unidad virtual que hemos creado anteriormente.



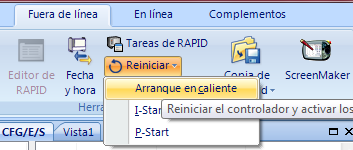
En “**Access Level**” debemos poner “All” sino no funcionará correctamente.



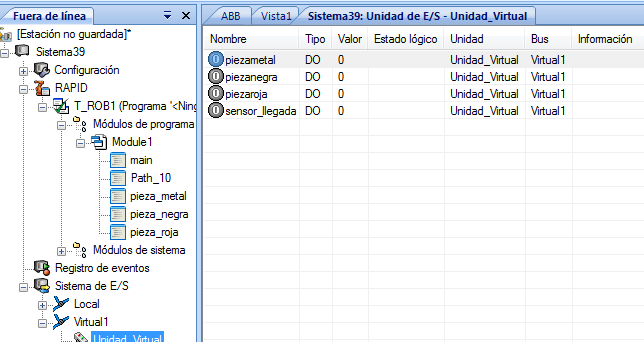
Por último en “**Unit Mapping**” debemos colocar el número de unidad que corresponda. Al ser la primera le designaremos la 0 y así sucesivamente en posteriores señales.



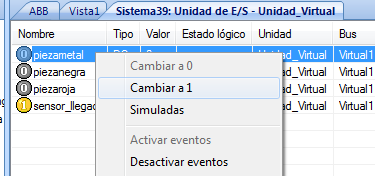
Pulsamos aceptar y nos dirá de nuevo que hay que reiniciar en caliente. Seguimos añadiendo las señales que creamos oportunas. Una vez que hayamos terminado debemos reiniciar para que los cambios se produzcan.



Cerramos la pestaña de configuración de I/O y abrimos nuestra unidad virtual en “**Sistema de E/S**”.



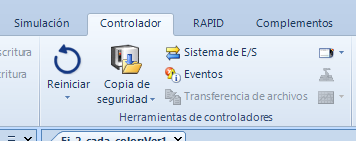
Vemos que como se nos muestra una tabla con las diferentes señales que hemos creado. Ahora solo nos falta poner el programa en ejecución, en modo continuo, y poner a 1 las señales que queramos. Para hacer esto pinchamos con el botón derecho en cualquiera de ellas y seleccionamos “**Cambiar a 1**”.



# 6.- Como cargar un programa desde RobotStudio

Para esta explicación, vamos a usar otra versión diferente de RobotStudio que la utilizada en otros apartados, asi que algunos iconos y nombres de las pestañas pueden ser diferentes, pero básicamente la estructura del programa es la misma.

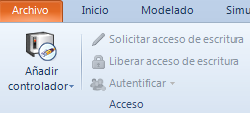
Vamos a suponer para esta explicación que nuestro Robot se encuentra en la dirección **IP** **192.168.125.1.** Con nuestro programa ya realizado, debemos irnos a la pestaña “**Controlador”.**



Si la IP de nuestro ordenador esta puesta fija , y en el rango del Robot, es decir, 192.168.125.X (siendo X menor de 255) , al clickar en “**Añadir Controlador**” se debe conectar correctamente con el controlador de nuestro Robot.

Suponiendo que nos hemos conectado correctamente vamos a describir los pasos a seguir para cargar nuestro programa.

El primer paso es “**Solicitar Acceso de Escritura**”. Pinchamos en el icono de al lado de “Añadir Controlador”. (En la imagen esta desactivado puesto que no tenemos conexión)



Puede ser que el controlador te la de automáticamente, o que tengamos que Concedernosla desde el FlexPendant. Simplemente sería pulsar “**Conceder**” en este dispositivo.

Una vez que tengamos el acceso de Escritura, debemos pinchar en “**Crear Relación**”, ponemos el nombre que queramos y seleccionamos en la pestaña de abajo el identificador de nuestro Controlador Remoto ( En nuestro caso sería 120.xxx.xx..xx.x ).

Pulsamos Aceptar y nos saldrá la ventana de transferencia del programa. Seleccionamos todas las casillas y pulsamos en ” **Transferir**”. Hay que decir que si un programa está ejecutándose en el Robot, es necesario pararlo pulsando Stop en el FlexPendant, sino no podremos transferir el nuestro.

Por último pulsamos “**Liberar Escritura**” , nos vamos al FlexPendant y colocamos el **Puntero de Programa en Main** (PP a Main). Le damos al Play y nuestro programa ya estará ejecutándose.

# 7.- Variar las coordenadas de un Punto en Ejecución

Habrá veces en que necesitemos crear multitud de puntos en nuestro programa, con el consiguiente incremento de tamaño del código de este. Vamos a explicar una forma con la que es posible variar las coordenadas XYZ de un punto, pudiéndolo utilizar por tanto para varios movimientos del Robot.

Primeramente nos creamos un punto. Este por defecto en el código Rapid nos vendrá en formato “**CONST**”, con lo cual sería imposible variar cualquiera de sus atributos ya que es **constante**. Debemos ponerlo en formato **“PERS”** para poder variar cualquiera de sus características.

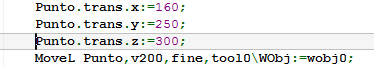




Una vez que hayamos hecho esto podremos variar las coordenadas del punto en cualquier momento del programa. Esto se realiza de la siguiente forma:

**Nombredelpunto**.**trans**.**ejecoordenadas:=NuevaCoordenada;**

Ejemplo:



En el ejemplo variamos las coordenas XYZ del punto llamado **“Punto”** y posteriormente movemos el Robot hacia la nueva posición.

# 7.- Plano de Conexiones del Robot

