

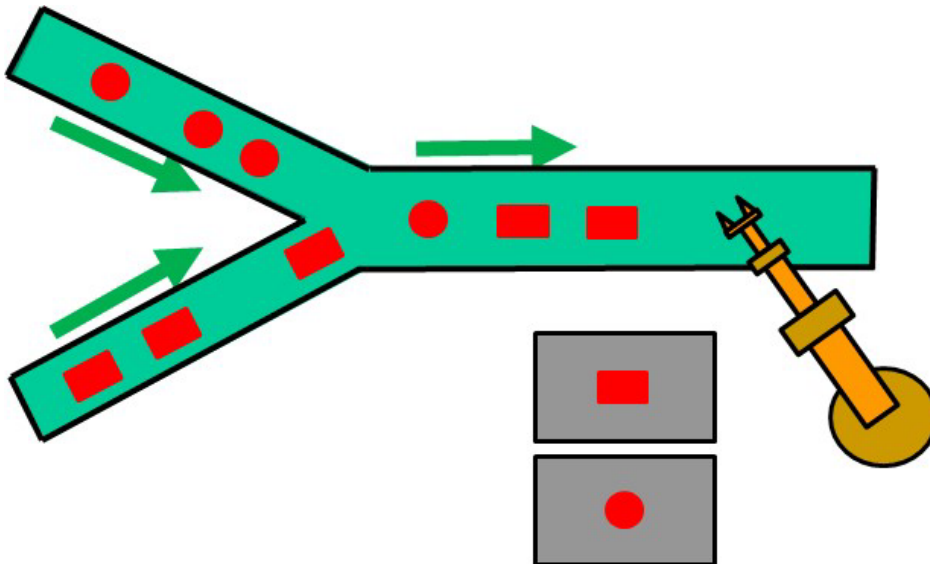
PRÁCTICA 2: PROGRAMACIÓN DE APLICACIONES DE PICK & PLACE CON UN ROBOT INDUSTRIAL

ROBÓTICA INDUSTRIAL
MÁSTER U. INGENIERÍA INDUSTRIAL
ESCUELA TÉCNICA SUPERIOR INGENIEROS INDUSTRIALES

ÁNGEL VALERA FERNÁNDEZ
CATEDRÁTICO DE UNIVERSIDAD
DPTO. INGENIERÍA DE SISTEMAS Y AUTOMÁTICA
INSTITUTO U. DE AUTOMÁTICA E INFORMÁTICA INDUSTRIAL
DESPACHO 7, ED. 8G (CPI), ACCESO D, 1ª PLANTA, ALA OESTE
GIUPROG@ISA.UPV.ES

PLANTEAMIENTO DEL PROBLEMA

- Objetivo: programar una aplicación industrial pick&place





PLANTEAMIENTO DEL PROBLEMA

- **Objetivo: programar una aplicación industrial pick&place**
 - Empaquetar objetos con formas y/o orientaciones diferentes
 - Formación de pallets
- **Célula de inspección y empaquetado**
 - Robot industrial ABB IRB140
 - Unidad mecánica IRB140
 - Controladora IRC5 (+ tarjeta E/S)
 - Actuador neumático
 - Cinta transportadora con sensor de presencia de objetos

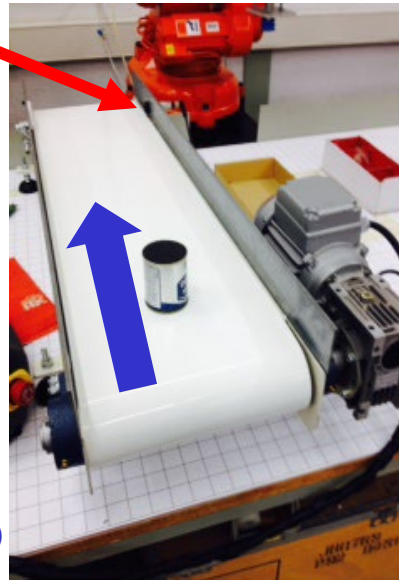
PLANTEAMIENTO DEL PROBLEMA

- **Objetivo: programar una aplicación industrial pick@place**
 - Empaquetar objetos con formas y/o orientaciones diferentes
 - Formación de pallets
- **Etapas**
 - Ir a la posición de origen
 - Poner cinta en marcha
 - Esperar que el objeto llegue a la pos de agarre y parar la cinta
 - Si es una lata, la entrada digital **objeto_Lata** = 1
 - Si es un brick, la entrada digital **objeto_Brick** = 1
 - Maniobra de pick (ir a la posición correspondiente y coger el objeto)
 - Maniobra de place (llevar objeto a posición de empaquetado y dejarlo)
 - Volver a la 1ª etapa

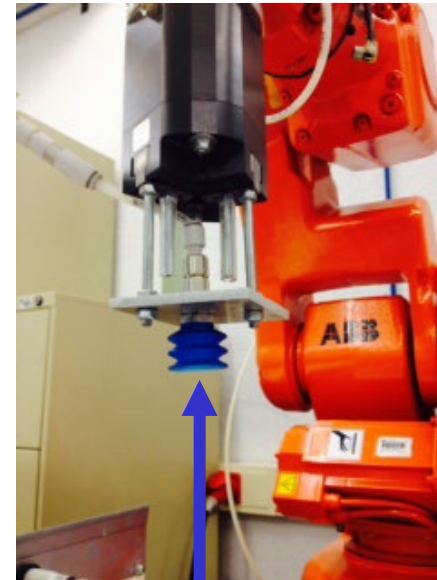
1.1 CÉLULA INSPECCIÓN Y EMPAQUETADO

- Señales de la estación: DeviceNet DSCQ-651

Sensor de presencia
(e.d. CONVEYOR_OBJ_SEN =1
si objeto en cinta)



Avance la cinta (si s.d.
CONVEYOR_FWD=1)



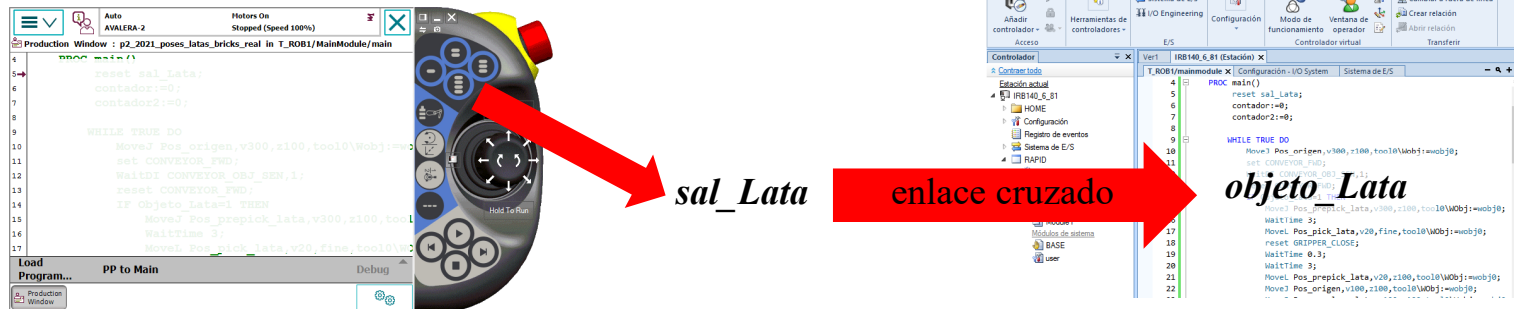
Ventosa neumática
(activa si s.d. GRIPPER_CLOSE =0)

1.1 CÉLULA INSPECCIÓN Y EMPAQUETADO

- **Señales de la estación: DeviceNet DSCQ-651**
 - Salidas digitales DeviceNet DSCQ-651
 - *CONVEYOR_FWD*
 - *GRIPPER_CLOSE*
 - Entradas digitales DeviceNet DSCQ-651
 - *CONVEYOR_OBJ_SEN*
 - Estas entradas existen físicamente en el robot real del laboratorio, pero para poder trabajar con el robot simulado en RobotStudio, se tendrán que **crear con el mismo nombre** en el entorno de RS (menú *Controlador* ⇒ *Editor de configuración* ⇒ *I/O System*)
 - Recordad que el *Access Level* deber ser ALL

1.1 CÉLULA INSPECCIÓN Y EMPAQUETADO

• Señales de la estación: virtuales



Creación de señales digitales

- Entradas digitales (virtuales)
 - *objeto_Lata*
 - *objeto_Brick*
 - *objeto_Brick45* (para la ampliación 1 y 2)
 - *objeto_Brick90* (para la ampliación 1 y 2)
- Salidas digitales (virtuales)
 - *sal_Lata*
 - *sal_Brick*
 - *sal_Brick45* (para la ampliación 1 y 2)
 - *sal_Brick90* (para la ampliación 1 y 2)

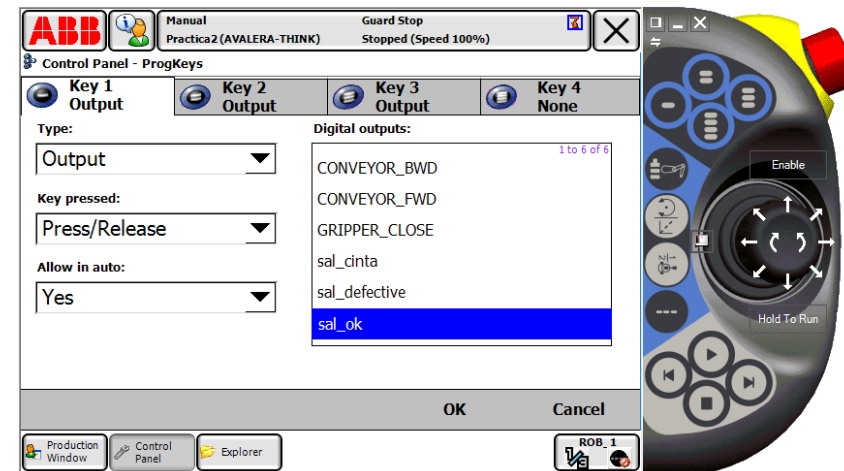
Conexiones cruzadas de señales

- Conexión 1:
 - *Resultante: objeto_Lata*
 - *Actor: sal_Lata*
- Conexión 2:
 - *Resultante: objeto_Brick*
 - *Actor: sal_Brick*

- Robot real: ya existen en la controladora IRC5 del robot
- RobotStudio: deben crearse para poder hacer la simulación

1.1 CÉLULA INSPECCIÓN Y EMPAQUETADO

- **Señales de la estación: virtuales**
 - Programación TeachPendant (ABB ⇒ Control Panel ⇒ ProgKeys)
 - *Controlador: modo Manual*
 - *ProgKey 1:*
 - *Type: Output*
 - *Key pressed: Set to 1*
 - *Allow in auto: Yes*
 - *Digital output: sal_Lata*
 - *ProgKey 2:*
 - *Type: Output*
 - *Key pressed: Set to 1*
 - *Allow in auto: Yes*
 - *Digital output: sal_Brick*
 - Programa RAPID: deberá de encargarse de poner a 0 estas señales después de haber hecho el pick para el empaquetado siguiente

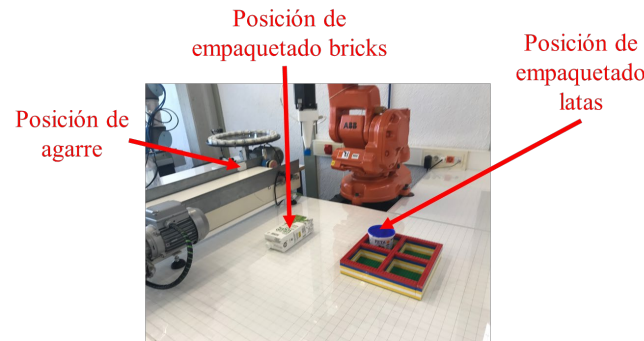


1.1 CÉLULA INSPECCIÓN Y EMPAQUETADO

- Comandos necesarios (además de *MoveL* y *MoveJ*):
 - Activar señal digital
 - *Set saldigital, SetDO saldigital, 1*
 - Desactiva señal digital
 - *Reset saldigital, SetDO saldigital, 0*
 - Esperar a que una entrada digital tenga un valor
 - *WaitDI entdigital, 1; WaitDI entdigital, 0;*
 - Esperar un determinado tiempo
 - *WaitTime (tiempo); !tiempo expresado en segundos*
 - Leer entrada digital
 - *valor=DInput (entdigital);*
 - *IF (Dinput (entdigital)=0) THEN ...*

1.2 PROBLEMA A RESOLVER

- Posiciones (previamente definidas para simulación y real)

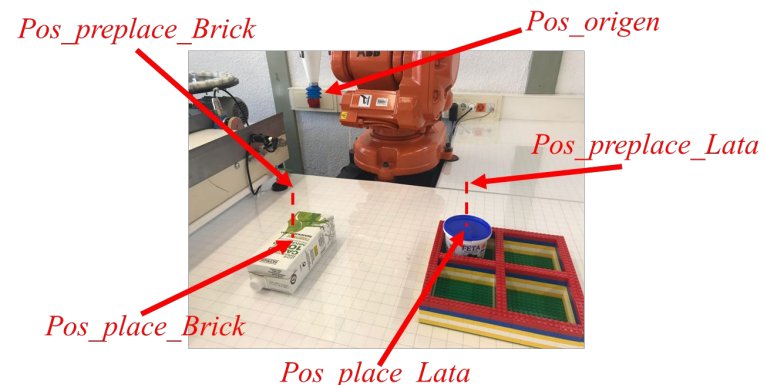
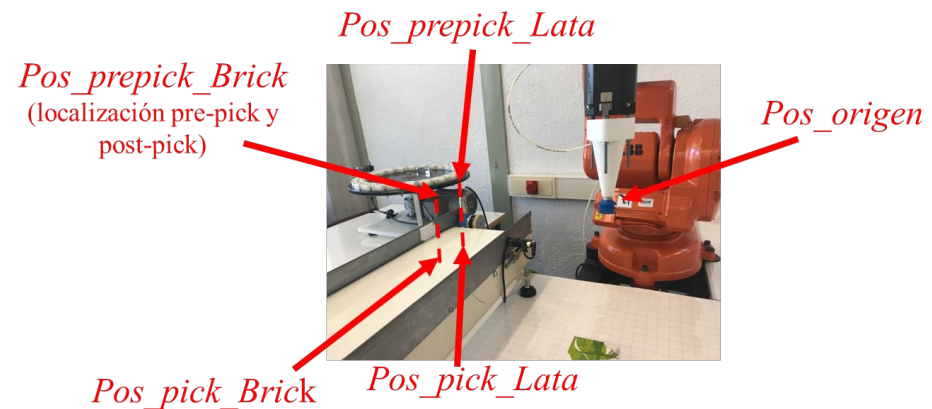
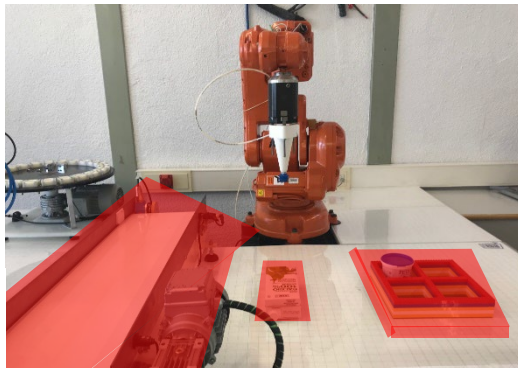
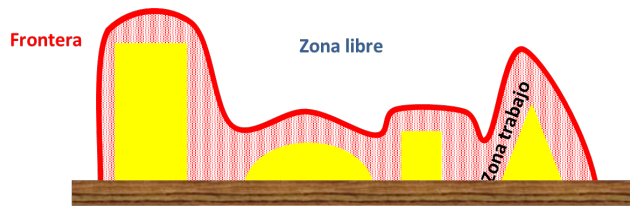


- p2_rapid_simulada.zip
 - *p2_2023simulada.pgf*
 - *MainModule.mod*
 - *Module2023_simulada.mod*
- p2_rapid_real.zip
 - *p2_2023_real.pgf*
 - *Module2023_real.mod*

```
T_ROB1/Module1 x
1  MODULE Module1
2  CONST robtarget Pos_origen:=[[494.97491565;
3  CONST robtarget Pos_prepick_Brick:=[[494.9;
4  CONST robtarget Pos_pick_Brick:=[[462.9749;
5  CONST robtarget Pos_preplace_Brick:=[[539.2;
6  CONST robtarget Pos_place_Brick:=[[539.2516;
7  CONST robtarget Pos_prepick_lata:=[[440.456;
8  CONST robtarget Pos_pick_lata:=[[440.45637;
9  CONST robtarget Pos_preplace_lata:=[[606.2;
10 CONST robtarget Pos_place_lata:=[[606.2156;
11  VAR robtarget prepick_var;
12  VAR robtarget pick_var;
13  ENDMODULE
```

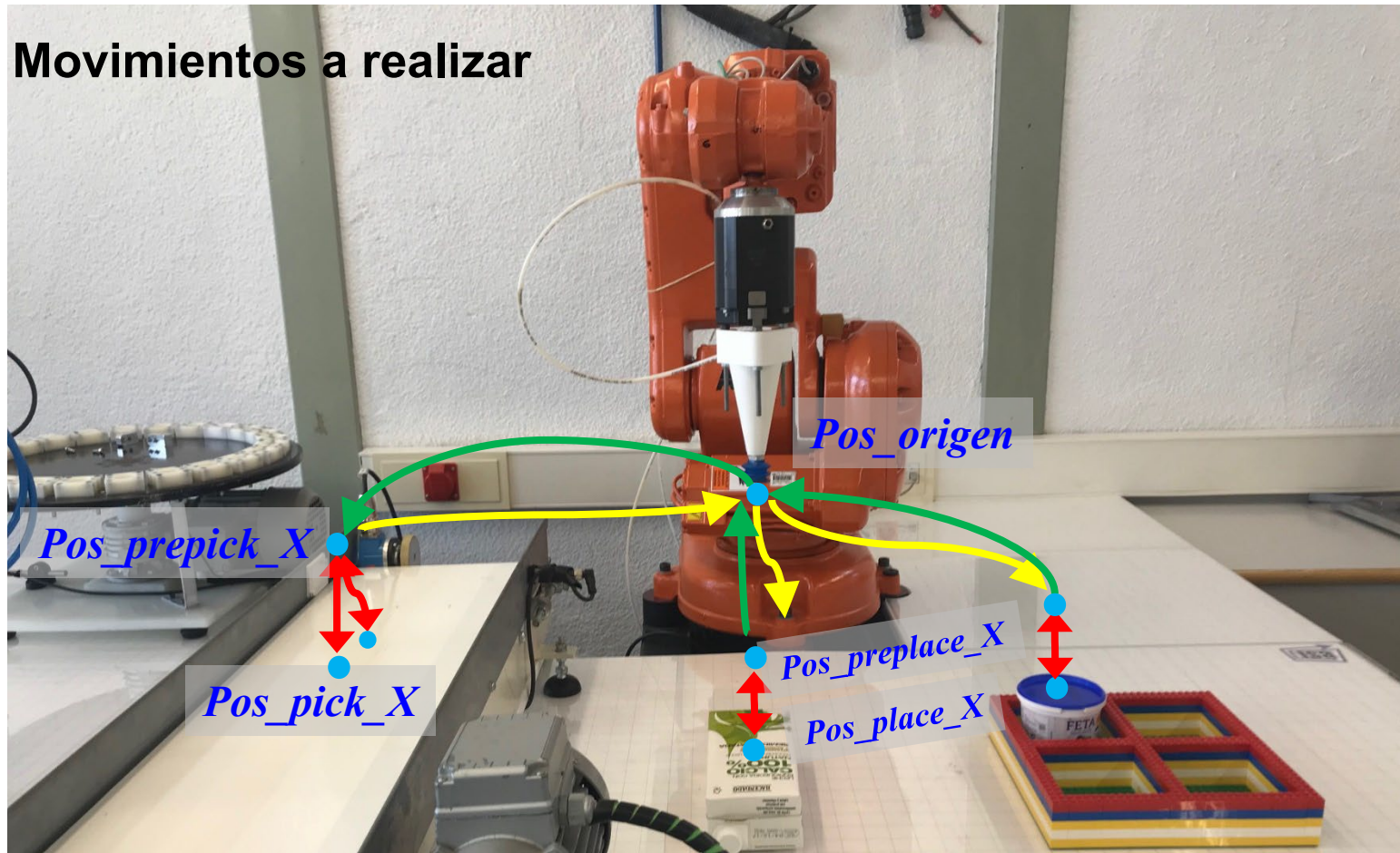
1.2 PROBLEMA A RESOLVER

- Posiciones (previamente definidas para simulación y real)



1.2 PROBLEMA A RESOLVER

- Movimientos a realizar



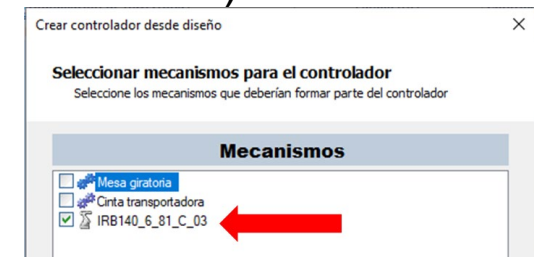
1.2 PROBLEMA A RESOLVER

- **Velocidades:**

- Las velocidades cuando se vaya a coger y dejar los objetos, deben ser bajas **v20** ($Pos_prepick_X \Rightarrow Pos_pick_X$; $Pos_pick_X \Rightarrow Pos_prepick_X$; $Pos_preplace_X \Rightarrow Pos_place_X...$).
- Las velocidades para ir a los puntos de aproximación con robot cargado: velocidades medias **v100** ($Pos_prepick_X \Rightarrow Pos_origen$).
- Cuando el robot no lleve carga y se mueva al punto de aproximación: velocidad alta (**v300**).

1.3 ROBOT VIRTUAL

- **Realización de la práctica: entorno simulado con RobotStudio**
 1. Descargar del PoliformaT y descomprimir **p2RobotStudio.zip**.
 2. Abrir la estación de RobotStudio **p2_celula_pick&place_R2022.rsstn**.
 3. En *Biblioteca ABB*, añadir un robot IRB140.
 4. Posicionar el robot en X: -250; Y:750; Z: 760 (*Posición* \Rightarrow *Fijar posición*)
 5. Arrastrar la ventosa hasta el robot (Actualizar posición: SI)
 6. En menú *Pos Inicial*: botón *Controlador virtual*
 - *Desde diseño*
 - Seleccionar sólo el mecanismo del robot
 7. Crear las señales digitales (*GRIPPER_CLOSE*, *CONVEYOR_FWD*, *sal_Lata*, *objeto_Brick*...)
 8. Realizar las conexiones cruzadas (reiniciar el controlador de nuevo)
 9. Realizar la programación de los botones del FlexPendant



1.3 ROBOT VIRTUAL

- **Realización de la práctica: entorno simulado con RobotStudio**

En la estación tenemos 3 dispositivos

- El robot IRB140
- La ventosa
- La cinta

Para poder funcionar, cada uno de ellos tienen un conjunto de entradas y salidas. Se necesita establecer la conexión entre ellas:

- En la célula real: conectar los cables de las señales eléctricas
- En simulación: conexiones lógicas virtuales

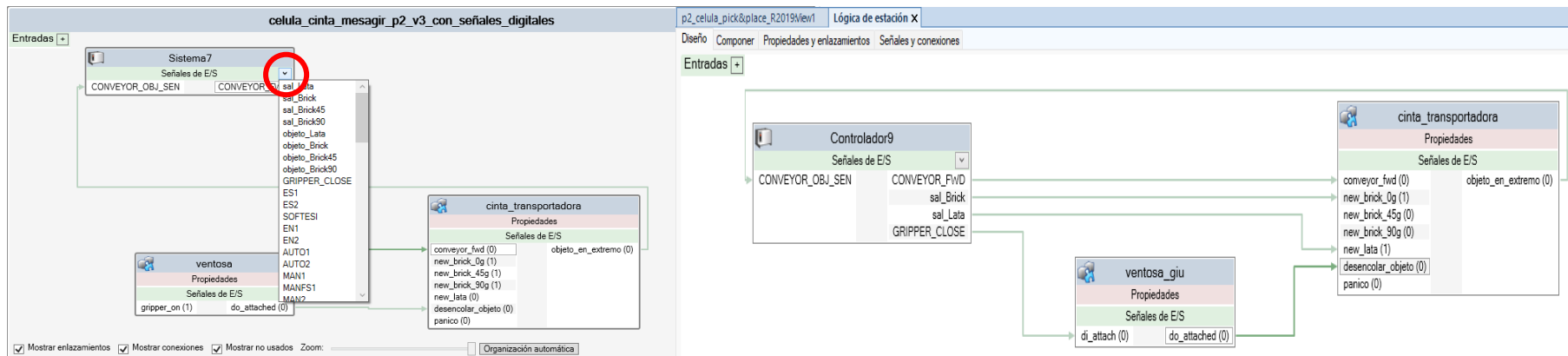
1.3 ROBOT VIRTUAL

• Realización de la práctica: entorno simulado con RobotStudio

10. En *M. Simulación* ⇒ Lógica Estación ⇒ Diseño

➤ haced las conexiones:

- Desde *controlador robot* a la *ventosa*: **GRIPPER_CLOSE** → **di_attach (ventosa_giu)**
- Desde *cinta_trans.* al *controlador robot*: **objeto_en_extremo** → **CONVEYOR_OBJ_SEN**
- Desde *controlador robot* a *cinta_trans.*
 - **CONVEYOR_FWD** → **conveyor_fwd**
 - **sal_Lata** → **new_lata**
 - **sal_Brick** → **new_brick_0g**
 - **sal_Brick45** → **new_brick_45g** (para las ampliaciones 1 y 2)
 - **sal_Brick90** → **new_brick_90g** (para las ampliaciones 1 y 2)



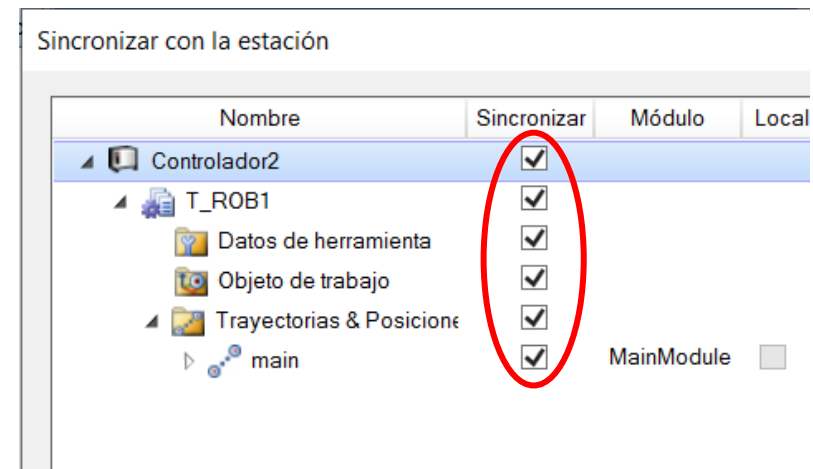
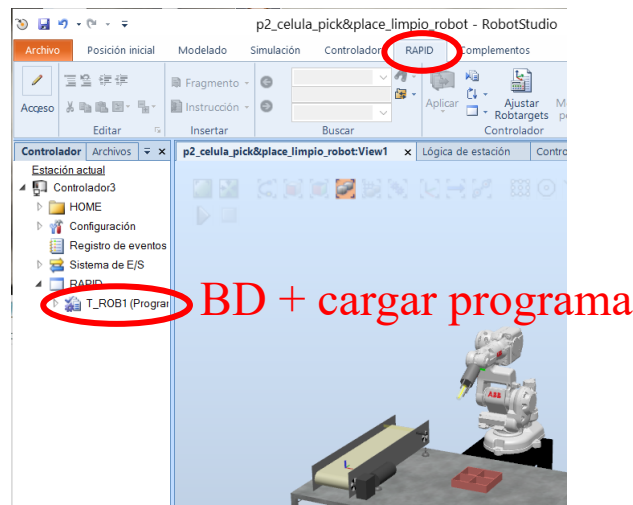
1.3 ROBOT VIRTUAL

• Realización de la práctica: entorno simulado con RobotStudio

11. Desde PoliformaT, cargar y descomprimir fichero de *p2_rapid_simulada.zip*

12. Cargar (sincronizando **todos** los componentes) el programa del robot simulado: *p2_2023_simulada.pgf*

- *Module2023_simulada.mod* (definición de las posiciones para el robot simulado)
- *MainModule.mod* (módulo con la rutina *main* a completar)



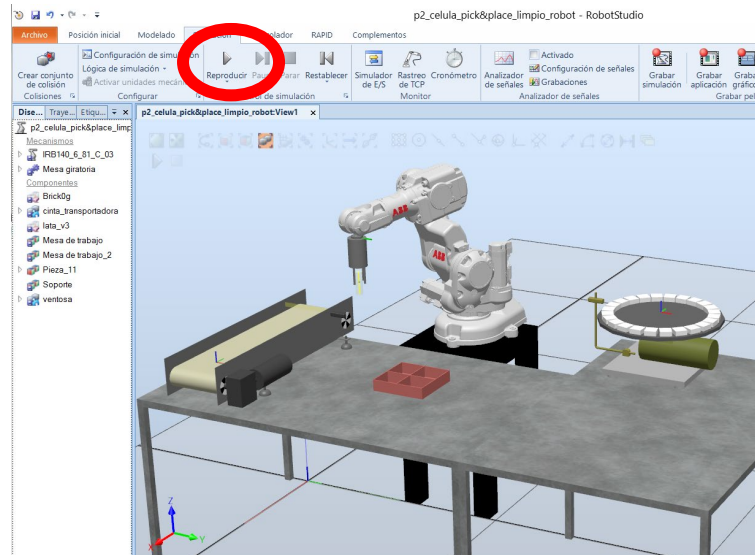
1.3 ROBOT VIRTUAL

- **Realización de la práctica: entorno simulado con RobotStudio**
 - 11. Desde PoliformaT, cargar y descomprimir fichero de *p2_rapid_simulada.zip*
 - 12. Cargar (sincronizando **todos** los componentes) el programa del robot simulado: *p2_2023_simulada.pgf*
 - *Module2023_simulada.mod* (definición de las posiciones para el robot simulado)
 - *MainModule.mod* (módulo con la rutina *main* a completar)
 - 13. Programar la aplicación de pick&place en MainModule.
 - En la parte de inicialización, se deberá llevar el robot con un movimiento punto a punto a **pos_origen**, y se tendrá que asegurar que todas las entradas (**objeto_Lata**, **objeto_Brick**) están a nivel bajo. Para ello se resetearán las salidas **sal_Lata** y **sal_Brick**.
 - Recordad que la ventosa (**GRIPPER_CLOSE**) se activa con un nivel bajo, y se desactiva con un nivel alto.

1.3 ROBOT VIRTUAL

- Realización de la práctica: entorno simulado con RobotStudio

14. Verificar el correcto funcionamiento del programa iniciando la simulación y pulsando los botones (I) y (II) del TeachPendant

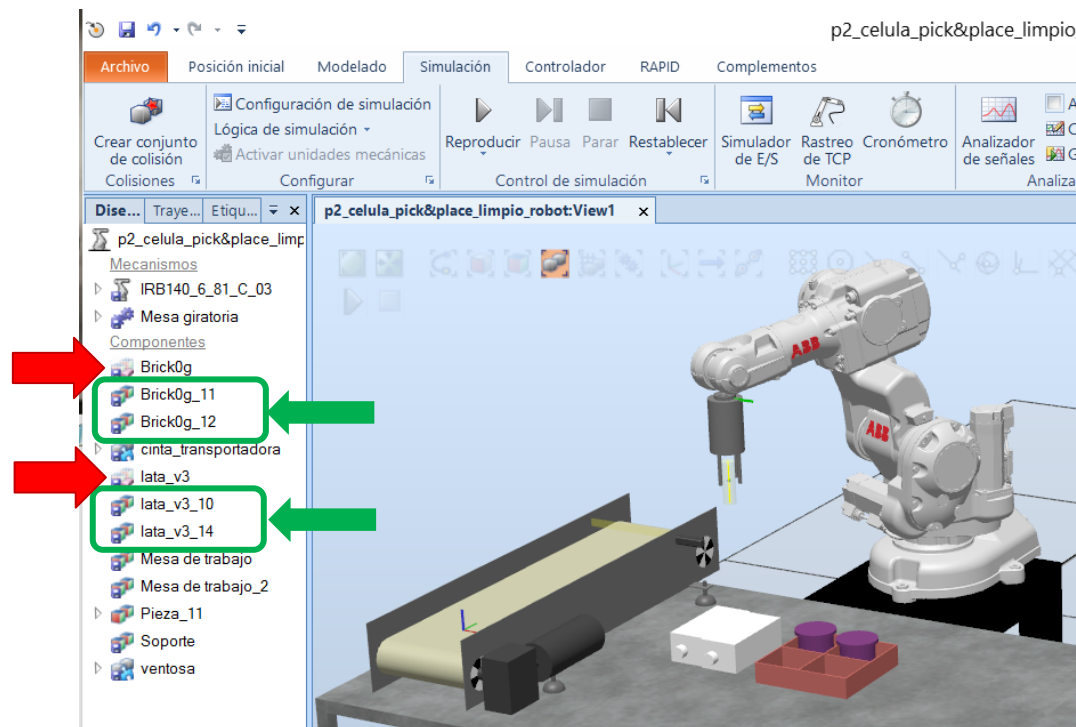


Para arrancar la simulación de la estación hay que pulsar sobre el botón de **Reproducir** del menú **Simulación**

1.3 ROBOT VIRTUAL

- Realización de la práctica: entorno simulado con RobotStudio

Si se hacen varias simulaciones, antes de arrancar una nueva, borrar las copias de los bricks (*Brick0g_XX*) y latas (*lata_v3_XX*), pero dejar las plantillas (*Brick0g*, *lata_v3*)

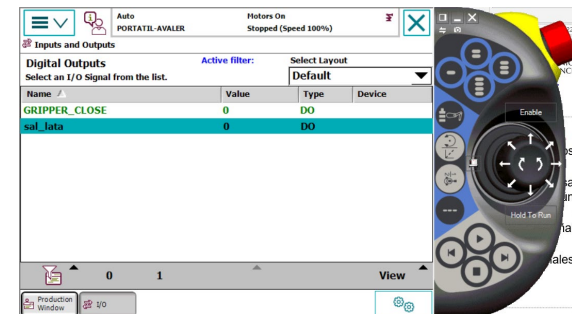
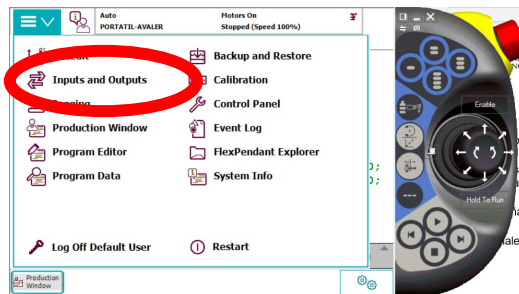


1.3 ROBOT VIRTUAL

- **Realización de la práctica: entorno simulado con RobotStudio**

En los componentes inteligentes las señales se activan mediante flancos de subida. Por ello:

- Para generar las latas y bricks hay que asegurarse que la señal pasa de 0 a 1. Si ya está a nivel alto, y se vuelve a poner a 1, no se generará ningún elemento nuevo puesto que no se tiene el flanco de subida.
- Debe ser el programa Rapid el que se encargue de poner a 0 la señal (**sal_Lata** o **sal_Brick**) después de hacer el pick del objeto.
- Para hacer pruebas en manual, se pueden poner a 0 esas señales mediante la opción el TP

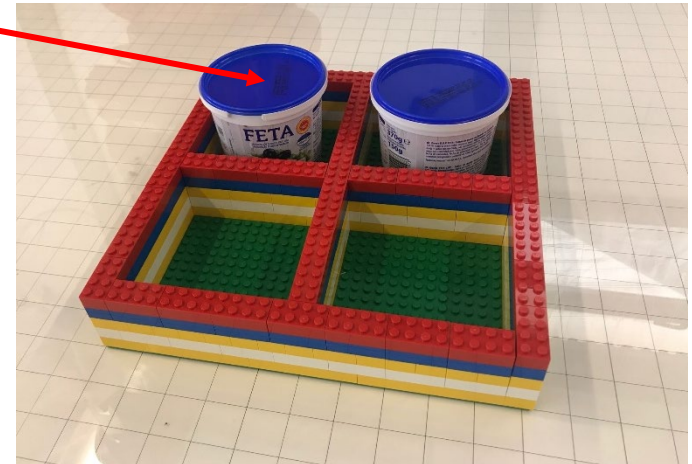
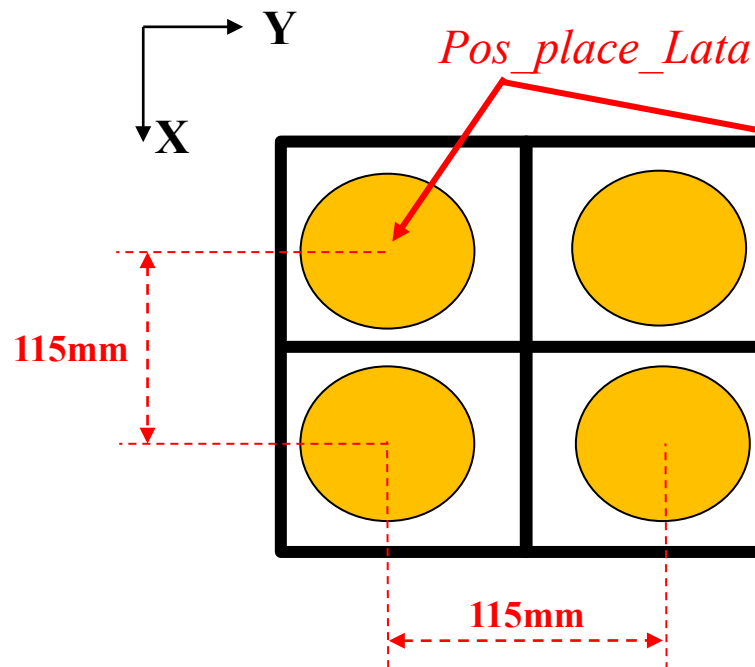


1.4 ROBOT REAL

- **Realización de la práctica: robot real del laboratorio**
 1. Descargar y descomprimir el fichero del Poliformat ***p2_rapid_real.zip*** en un directorio nuevo en el PC donde se esté realizando la práctica.
 2. En este directorio nuevo que se acaba de crear, copiar el fichero ***MainModule.mod*** con el programa de la aplicación de pick&place el apartado 1.3 anterior.
 - El ***Module2023_real.mod*** NO se debe modificar porque tiene posiciones reales del robot
 3. Transferir el programa al controlador del robot real (con un USB o con RobotStudio).
 4. Cargar el programa ***p2_2023_real.pgf*** en el controlador
 5. Ejecutar en modo manual y verificar que se cumplen las especificaciones de diseño
 6. Ejecutar el programa en modo automático

1.5 ENVASADO MULTIPACK

- Realización de la práctica: envasado paquetes 4 objetos:



- Hay que asegurar la repetitividad de la aplicación: se pueden hacer varios paquetes de 4 latas (no sólo 1 único paquete) de forma continua.

1.5 ENVASADO MULTIPACK

- Realización de la práctica: envasado paquetes 8 objetos:



- Hay que asegurar la repetitividad de la aplicación: se pueden hacer varios paquetes de 4 latas (no sólo 1 único paquete) de forma continua.

1.6 AMPLIACIÓN TRABAJO PRÁCTICO

- **Ampliación 1:**

- Realización de pallets de bricks de 2x2: la orientación del 2º nivel deberán estar girados 90º



- Tamaño de los bricks: 195x90x60: se deberá dejar una distancia de seguridad entre un brick y el contiguo de 15mm
- Hay que asegurarse que se pueden hacer varios pallets de 4 bricks

1.6 AMPLIACIÓN TRABAJO PRÁCTICO

- **Ampliación 2:**

- Hasta ahora, los bricks llegaban con una orientación de 0°
- En esta ampliación se considera que los bricks pueden llegar además con un ángulo de 45° o de 90° .



1.6 AMPLIACIÓN TRABAJO PRÁCTICO

- **Ampliación 2:**
 - Se deberán generar 2 entradas y 2 salidas digitales virtuales:
 - objeto_Brick45; objeto_Brick90;
 - sal_Brick45; sal_Brick90;
 - Se deberán generar 2 enlaces cruzados
 - **Conexión 3:**
 - **Resultante:** *objeto_Brick45*
 - **Actor:** *sal_Brick45*
 - **Conexión 4:**
 - **Resultante:** *objeto_Brick90*
 - **Actor:** *sal_Brick90*

1.6 AMPLIACIÓN TRABAJO PRÁCTICO

- **Ampliación 2:**
 - Programar las teclas 3ª y 4ª del FlexPendant
 - *ProgKey 3:*
 - *Type: Output*
 - *Key pressed: Set to 1*
 - *Allow in auto: Yes*
 - *Digital output: sal_Brick45*
 - *ProgKey 4:*
 - *Type: Output*
 - *Key pressed: Set to 1*
 - *Allow in auto: Yes*
 - *Digital output: sal_Brick90*
 - Hay que realizar las conexiones entre las salidas del robot y las entradas del componente inteligente de la cinta
 - *sal_Brick45* → *new_brick_45g* (para las ampliaciones 1 y 2)
 - *sal_Brick90* → *new_brick_90g* (para las ampliaciones 1 y 2)

1.6 AMPLIACIÓN TRABAJO PRÁCTICO

- **Ampliación 2:**

- Se deberá tener en cuenta la orientación de llegada (α) de los objetos para que, independientemente de ésta, los objetos se empaqueten adecuadamente: la ventosa deberá girar en función de dicha orientación
 - Si el brick tiene un ángulo de 45° , el ángulo de agarre α deberá ser -45°
 - Si el brick tiene un ángulo de 90° , el ángulo de agarre α deberá ser -90°
 - Se debe calcular los valores de los cuaterniones considerando que se ha realizado un giro de 180 alrededor del eje Y, y un ángulo α alrededor del eje Z

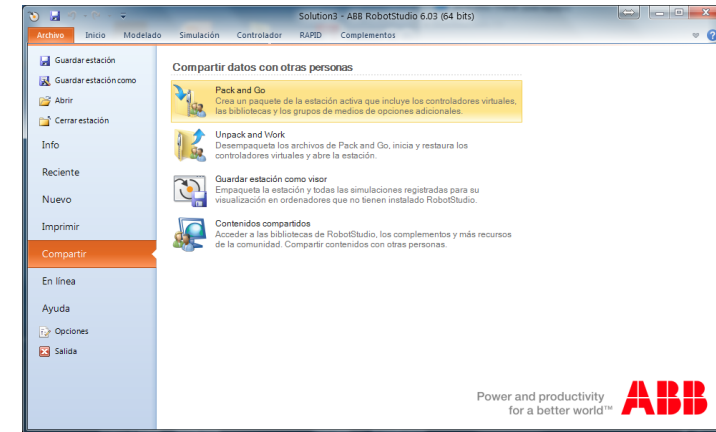
$$\left[\cos \frac{\alpha}{2}, 0, 0, \sin \frac{\alpha}{2} \right] \cdot [0, 0, 1, 0] = \left[0, \sin \frac{\alpha}{2}, \cos \frac{\alpha}{2}, 0 \right] = [q_0, q_1, q_2, q_3]$$

- Una vez calculados los cuaterniones, se tendrán que programar en una variable tipo robtarget: (VAR robtarget pos_pick_var)

1.7 REALIZACIÓN DE LA PRÁCTICA

Al finalizar la práctica:

1. Generar en la carpeta **PRACTICAS** del espacio compartido del PoliformaT de un miembro del grupo la carpeta **PRACTICA2**
2. Subir en **PRACTICA2**:
 - Ficheros de RobotStudio
 - Menú Fichero => Compartir => Pack and go
 - Programa RAPID generado
 - Vídeo ejecución del programa robot virtual
3. Generar la memoria de la práctica
 - 1 memoria por grupo de prácticas
 - PDF + Word/LaTex
 - Explicación trabajo realizado
 - Lengua: valenciano, castellano, ingles
 - Fecha límite: Grupo A: 24 de noviembre 2023
Grupo B: 25 de noviembre 2023



PRÁCTICA 2: PROGRAMACIÓN DE APLICACIONES DE PICK & PLACE CON UN ROBOT INDUSTRIAL

ROBÓTICA INDUSTRIAL
MÁSTER U. INGENIERÍA INDUSTRIAL
ESCUELA TÉCNICA SUPERIOR INGENIEROS INDUSTRIALES

ÁNGEL VALERA FERNÁNDEZ
CATEDRÁTICO DE UNIVERSIDAD
DPTO. INGENIERÍA DE SISTEMAS Y AUTOMÁTICA
INSTITUTO U. DE AUTOMÁTICA E INFORMÁTICA INDUSTRIAL
DESPACHO 7, ED. 8G (CPI), ACCESO D, 1ª PLANTA, ALA OESTE
GIUPROG@ISA.UPV.ES