

SISTEMAS ROBOTIZADOS

UNIVERSIDAD DE VALLADOLID

PROYECTO 2015 2016-GRUPO5

Estación compuesta por dos robots, una cinta transportadora -generadora de piezas, una mesa giratoria y una cinta transportadora - destructora de piezas.

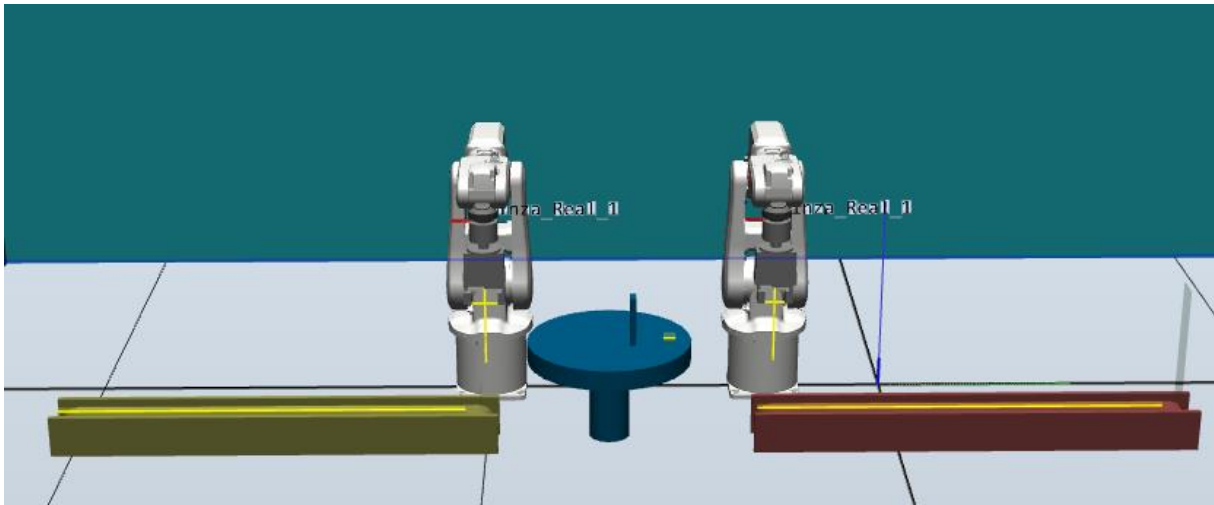
Pilar Estefanía Samaniego

pilar.samaniego@hotmail.com

Ken Vera Chan

kenveracha@yahoo.es

PROYECTO 2015 2016



Crea una estación compuesta por dos robots, una cinta transportadora - generadora de piezas, una mesa giratoria y una cinta transportadora - destructora de piezas. El sistema robótico debe realizar las siguientes operaciones:

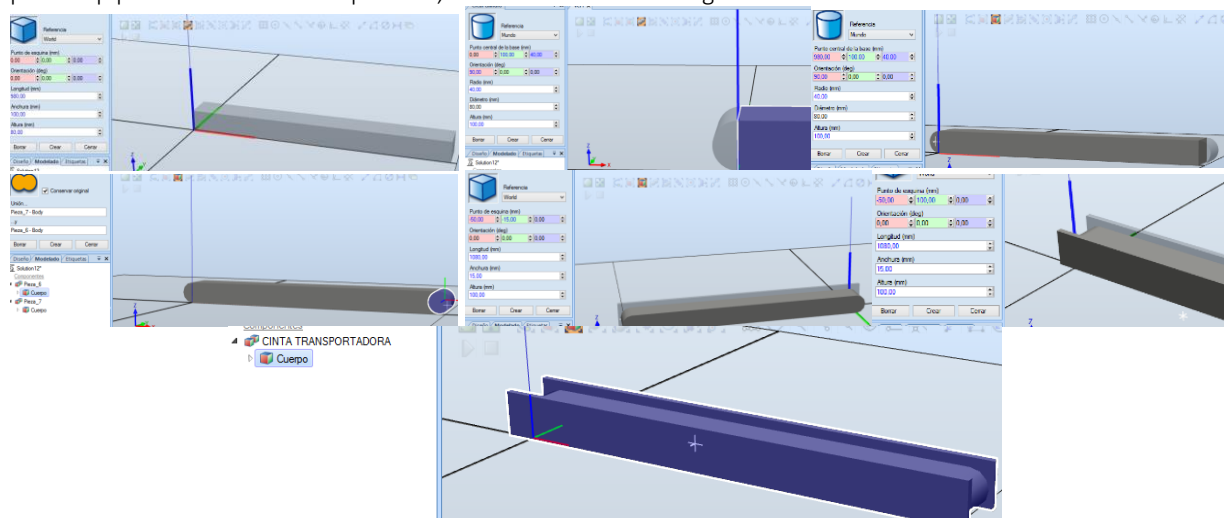
- Se genera una pieza en la cinta generadora que va hasta un determinado punto.
- El primer robot coge la pieza y la pone sobre la mesa cuando el operario se lo pida.
- La mesa gira 180°.
- Cuando el operario se lo pida, el segundo robot coge la pieza de la mesa y la coloca en la cinta destructora de piezas.
- La cinta transportadora detecta la pieza y la mueve hasta un punto en el que se destruye la pieza.

✓ REALIZACIÓN DE LOS EJERCICIOS PREVIOS:

➤ EJERCICIO PREVIO 1

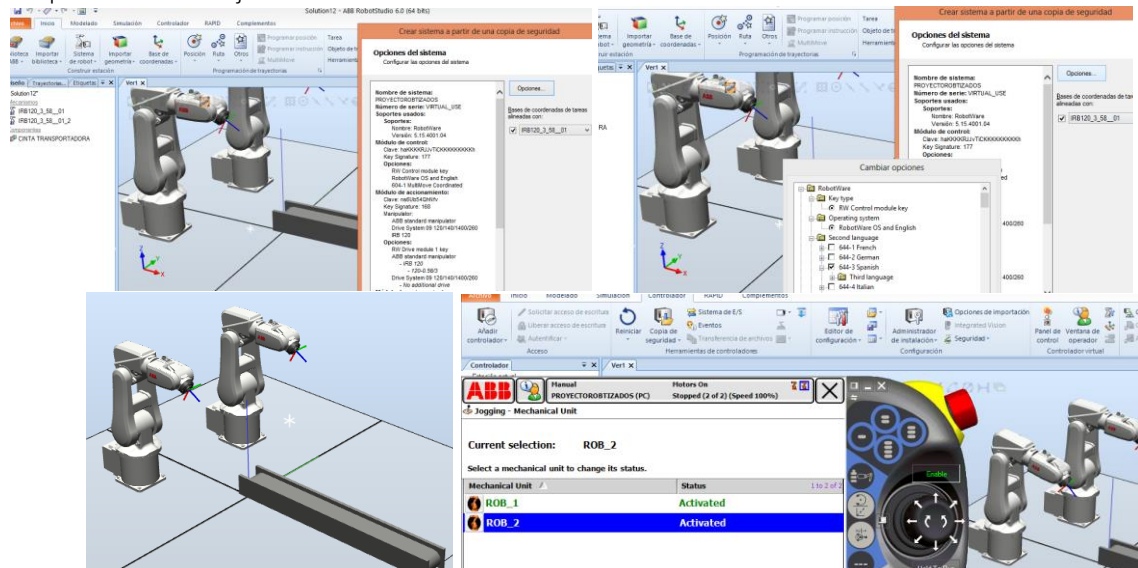
Crea un conjunto de sólidos que representen una cinta transportadora similar a la que existe en el laboratorio, de medidas 980x100x80 mm.

- Diseñar los elementos de la cinta como dos cilindros a los extremos, un paralelepípedo sólido como cinta y dos paralelepípedos laterales como paredes, de forma similar a la original.

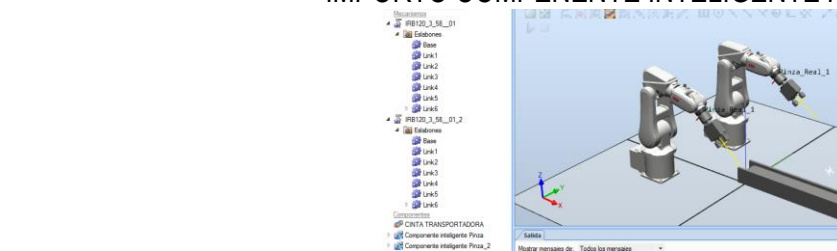


➤ EJERCICIO PREVIO 2

Crea una estación robótica con dos robots ABB IRB120. Los robots estarán en una posición tal que puedan cooperar en un trabajo común.



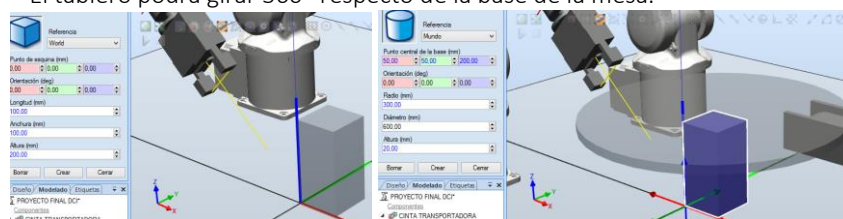
IMPORTO COMPONENTE INTELIGENTE PINZA



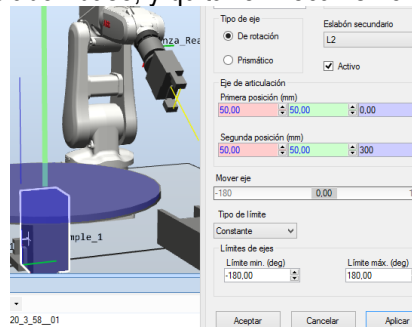
➤ EJERCICIO PREVIO 3

Crea un mecanismo que sea una mesa giratoria,

- La base de la mesa es un paralelepípedo de 100x100x200 mm (altura = 200 mm), y el tablero superior de la mesa es un cilindro de 600 mm de diámetro y 20 mm de altura.
- El tablero podrá girar 360° respecto de la base de la mesa.

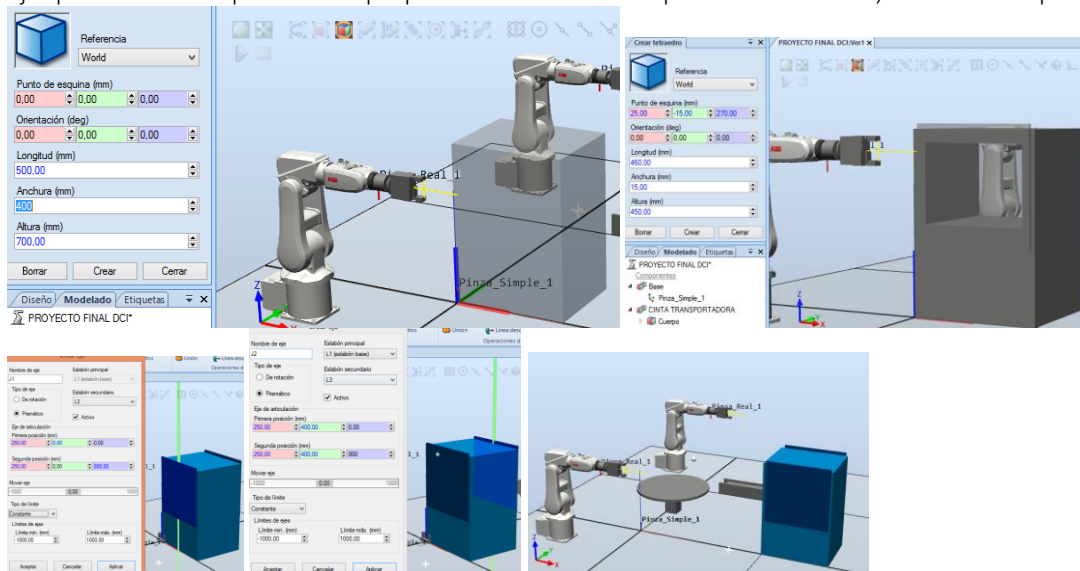


Marcar el paralelepípedo como eslabon base, y quitar el mecanismo de la biblioteca



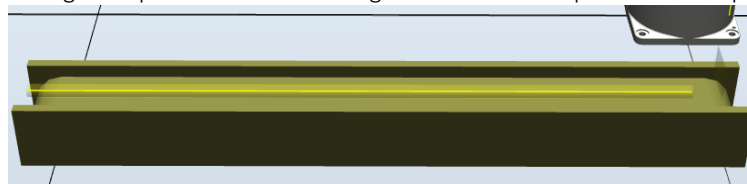
➤ EJERCICIO PREVIO 4

Crea un mecanismo que represente un horno de un proceso de producción. El horno debe tener la forma geométrica de un paralelepípedo. Dispondrá de dos puertas ubicadas en caras opuestas. Estas puertas se abrirán, de manera independiente, mediante desplazamientos verticales hacia arriba. Las dimensiones del horno las determinará el alumno, teniendo en cuenta el volumen de trabajo del robot IRB 120. Se debe modelar dos ejes prismáticos independientes que permitan mover las dos puertas hacia arriba, de forma independiente.



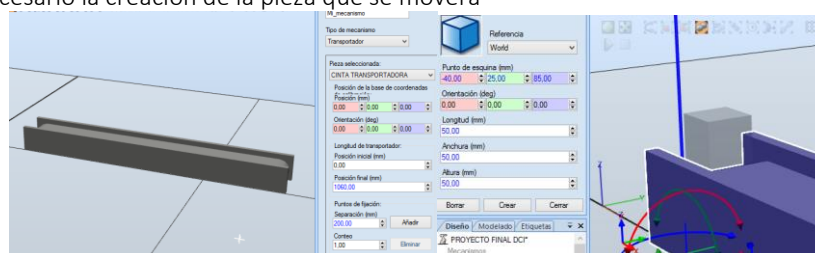
➤ EJERCICIO PREVIO 5

Crea un componente inteligente que simule una cinta “generadora – transportadora” de piezas:

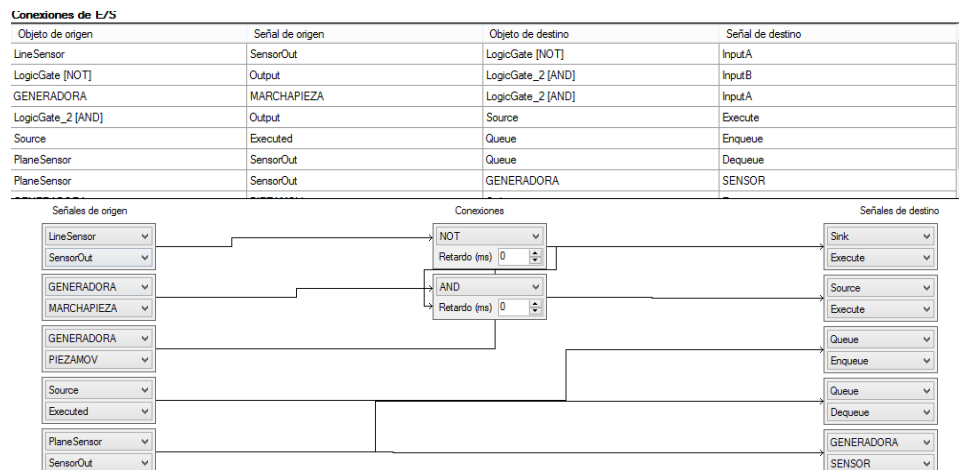


- La base del sistema será el conjunto de sólidos que representan la cinta del laboratorio (ver ejercicio RS 1)
- Se deberá añadir al sistema los siguiente elementos:
 - Generador de piezas cúbicas (*Source*).
 - Movimiento de la pieza sobre la cinta (*LinearMove*), que permita mover la pieza cúbica a lo largo de la cinta transportadora.
 - Sensor de detección de piezas (*LineSensor*), que detecte la llegada de la pieza al punto final de la cinta transportadora.
 - Puerta lógica (*LogicSRLatch*) que consiga parar el movimiento de la pieza cuando esta es detectada por el sensor. La pieza se parará y quedará en espera de ser recogida de la cinta transportadora.

Una vez creado los solidos de la cinta transportadora, se procede a crear el nuevo componente inteligente: Para lo cual es necesario la creación de la pieza que se movera

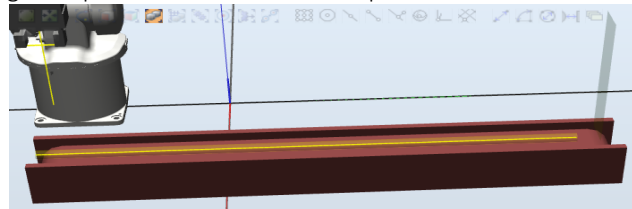


la cinta transportadora. Queue nos permite ir enlazando piezas creadas, aparentemente es algo innecesario pero una vez que se implementó el proyecto es útil junto con la función SINK.



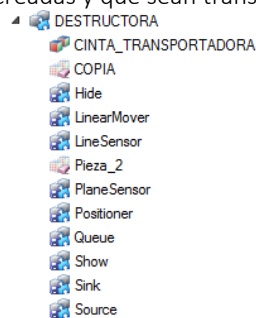
➤ EJERCICIO PREVIO 6

Crea un componente inteligente que simule una cinta “transportadora – destructora” de piezas:



- La base del sistema será el conjunto de sólidos que representan la cinta del laboratorio (ver ejercicio RS 1)
- Se deberá añadir al sistema los siguiente elementos:
 - Sensor de detección de la pieza que llega (*LineSensor*) que detecte la llegada de la pieza al punto inicial de la cinta.
 - Movimiento de la pieza sobre la cinta (*LinearMove*) que mueva a la pieza a lo largo de la cinta.
 - Sensor de detección de la pieza (*LineSensor*) que detecte la llegada de la pieza al punto final de la cinta.
 - Destrucción de la pieza (*Sink*) cuando se detecte que la pieza ha llegado al sensor final.

La cinta transportadora Destructora de piezas se la ha realizado de manera similar a la cinta Generadora, el elemento primordial para eliminar los elementos de la cinta destructora es el componente SINK mediante este será posible la eliminación de las piezas creadas y que sean transportadas en esta.

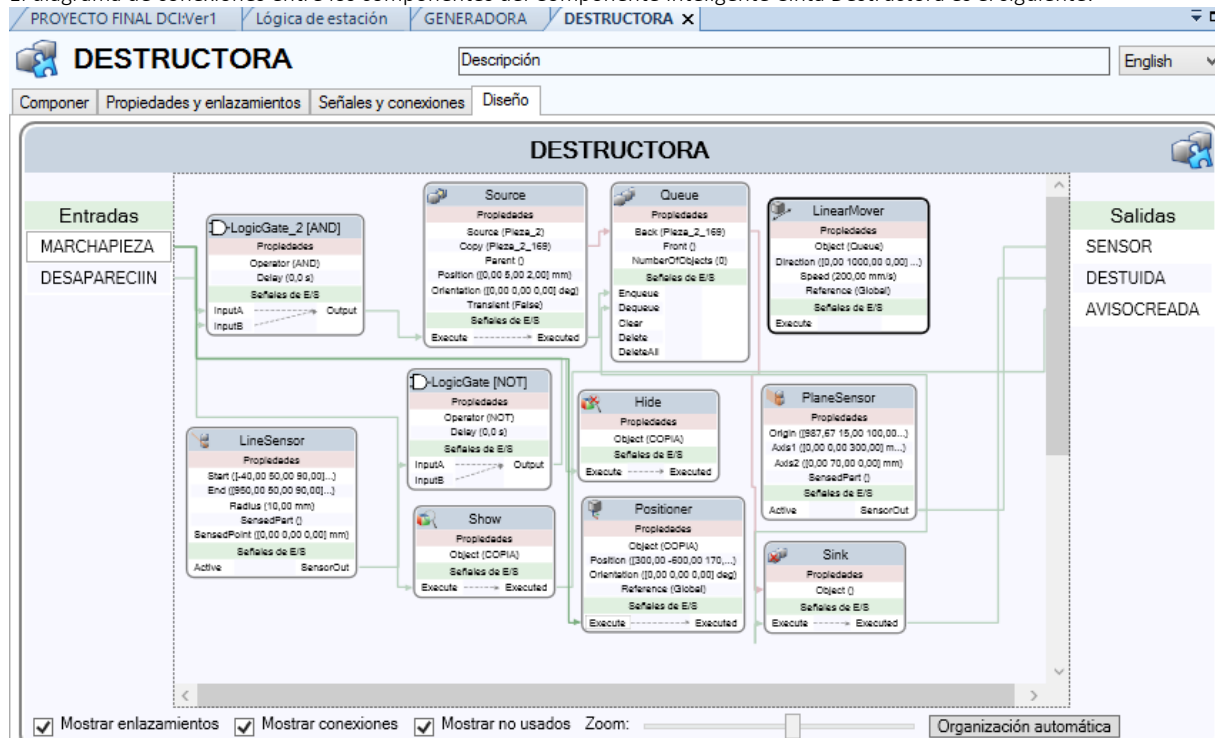


Adicionalmente puede observarse que se tienen añadidos varios componentes más como HIDE, SHOW, POSITIONER y como se explica anteriormente estos se añaden para poder lograr la simulación de manera lógica de lo que se pide en el enunciado del proyecto.

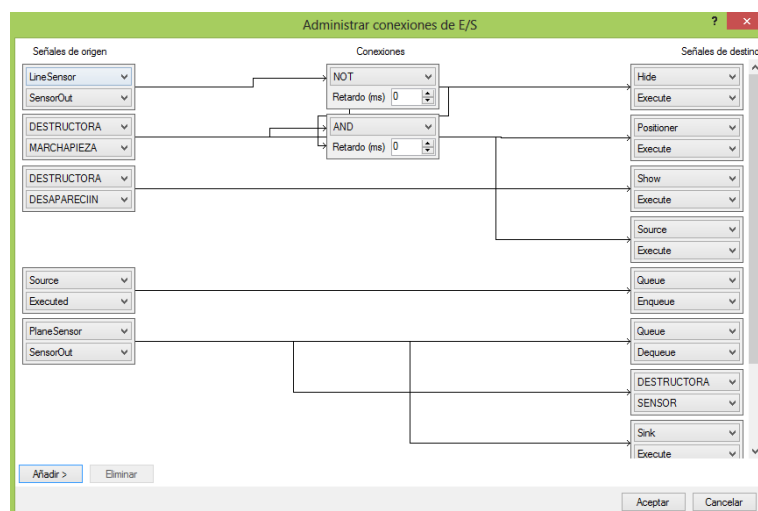
Las señales de entradas y salidas para este componente son las siguientes, la funcionalidad de cada una de ellas podrá notarse de mejor manera posteriormente en el diagrama de la lógica de la estación:

PROYECTO FINAL DCI:Ver1 / Lógica de estación / GENERADORA / DESTRUCTORA x		
DESTRUCTORA		
Descripción		English
Componer Propiedades y enlazamientos Señales y conexiones Diseño		
Señales de E/S		
Nombre	Tipo de señal	Valor
MARCHAPIEZA	DigitalInput	0
SENSOR	DigitalOutput	0
DESTUIDA	DigitalOutput	0
DESAPARECIIN	DigitalInput	0
AVISOCREADA	DigitalOutput	0

El diagrama de conexiones entre los componentes del Componente Inteligente Cinta Destructora es el siguiente:



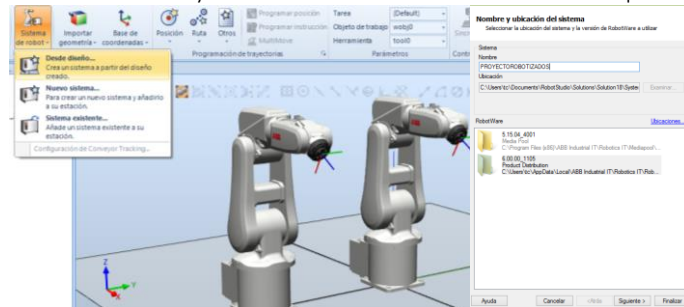
La conexión de todos estos componentes puede visualizarse de mejor manera en la siguiente figura:



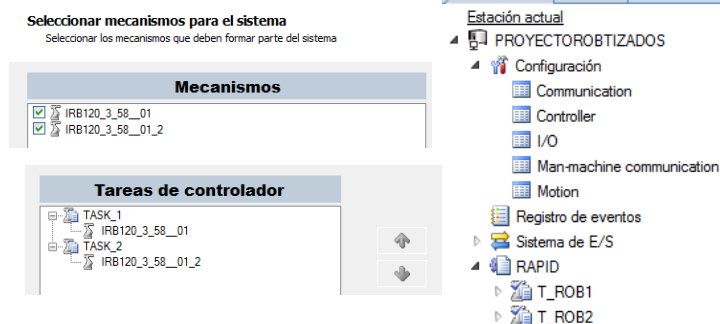
✓ CONFIGURACIÓN DEL SISTEMA DE ROBOTS

Hubo la necesidad de crear un sistema que permita insertar ambos robots, y que este sistema sea capaz de manipular los dos a la vez.

En primer lugar se insertan ambos robots y se crea un Nuevo Sistema de Robot a partir del diseño:



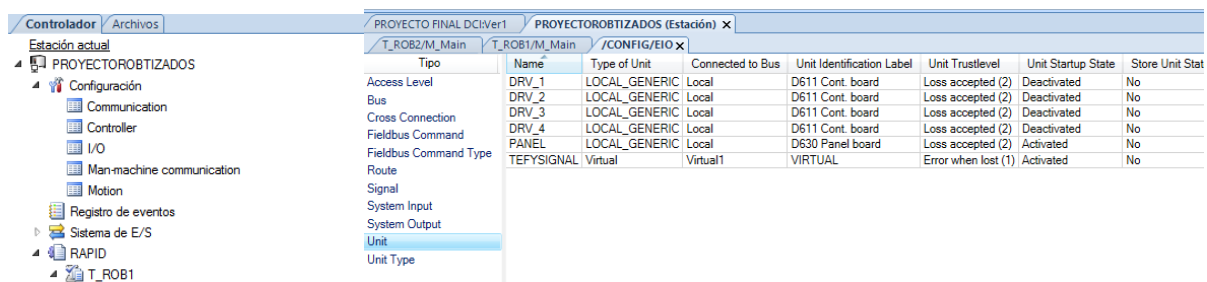
Se seleccionan ambos robots y luego aparecerán las dos tareas correspondientes para cada uno de ellos.



Con esta configuración ya se puede empezar la realización del proyecto.

✓ CONFIGURACIÓN DE ENTRADAS Y SALIDAS DEL SISTEMA

En la pestaña de Rapid dentro del Sistema anteriormente creado y dentro de la pestaña de configuración I/O se procederá a crear una nueva Unidad para poder asignar a esta unidad nuestras señales.



La unidad será de tipo virtual y se crea en la pestaña UNIT.

Luego en la pestaña Signal aparecerán todas las señales definidas por el sistema pero no las nuestras, así que se procede a crear de una en una tanto entradas como salidas, tantas como se necesiten para el programa.

PROYECTO FINAL DCI:Ver1 PROYECTOROBOTIZADOS (Estación) X									
T_ROB2/M_Main		T_ROB1/M_Main		/CONFIG/EIO X					
Tipo	Name	Type of Signal	Assigned to Unit	Signal Identification Label	Unit Mapping	Category	Access Level	Dr	
Access Level	AS1	Digital Input	PANEL	Automatic Stop chain(X5:11 to X5:6) and (X5:9 to X5:1)	13	safety	INTERNAL	0	
Bus	AS2	Digital Input	PANEL	Automatic Stop chain backup(X5:5 to X5:6) and (X5:3 to X5:1)	14	safety	INTERNAL	0	
Cross Connection	AUTO1	Digital Input	PANEL	Automatic Mode(X9:6)	5	safety	INTERNAL	0	
Fieldbus Command	AUTO2	Digital Input	PANEL	Automatic Mode backup(X9:2)	6	safety	INTERNAL	0	
Fieldbus Command Type	CH1	Digital Input	PANEL	Run Chain 1	22	safety	INTERNAL	0	
Route	CH2	Digital Input	PANEL	Run Chain 2	23	safety	INTERNAL	0	
Signal	di1	Digital Input	TEFY SIGNAL	di1	1		All	0	
System Input	di2	Digital Input	TEFY SIGNAL	di2	2		All	0	
System Output	di3	Digital Input	TEFY SIGNAL	di3	3		All	0	
Unit	di4	Digital Input	TEFY SIGNAL	di4	4		All	0	
Unit Type	di5	Digital Input	TEFY SIGNAL	di5	5		All	0	
	di6	Digital Input	TEFY SIGNAL	di6	6		All	0	
	di7	Digital Input	TEFY SIGNAL	di7	7		All	0	
	di8	Digital Input	TEFY SIGNAL	di8	8		All	0	
	di9	Digital Input	TEFY SIGNAL	di9	9		All	0	
	do1	Digital Output	TEFY SIGNAL	do1	1		All	0	
	do2	Digital Output	TEFY SIGNAL	do2	2		All	0	
	do3	Digital Output	TEFY SIGNAL	do3	3		All	0	
	do4	Digital Output	TEFY SIGNAL	do4	4		All	0	
	do5	Digital Output	TEFY SIGNAL	do5	5		All	0	
	do6	Digital Output	TEFY SIGNAL	do6	6		All	0	
	do7	Digital Output	TEFY SIGNAL	do7	7		All	0	
	do8	Digital Output	TEFY SIGNAL	do8	8		All	0	
	do9	Digital Output	TEFY SIGNAL	do9	9		All	0	
	DRV1BRAKE	Digital Output	DRV_1	Brake-release coil	2	safety	INTERNAL	0	
	DRV1BRAKEFB	Digital Input	DRV_1	Brake Feedback(X3:6) at Contactor Board	11	safety	INTERNAL	0	

En la ventana se observa las señales que se creó para el caso de este proyecto una vez creadas estas señales ya se podrán utilizar dentro del código de Rapid y se podrán hacer las conexiones en la lógica de la estación para poder lograr los objetivos.

✓ LOGICA DE LA ESTACIÓN:

La lógica de la estación nos permitirá hacer la conexión de las señales que se controlarán desde el programa en RAPID y la estación con las señales que son necesarias para que los componentes inteligentes entren en funcionamiento durante la simulación.

PROYECTO FINAL DCI:Ver1 Lógica de estación X GENERADORA DESTRUCTORA

PROYTEFY3

Componer Propiedades y enlazamientos Señales y conexiones Diseño

Componentes subordinados Añadir componente Editar objeto superior

Componentes inteligentes

- Componente inteligente Pinza
- Componente inteligente Pinza_2
- GENERADORA
- MESAGIRA
- LinearMover
Mueve un objeto en una trayectoria lineal
- DESTRUCTORA
- Otros
- IRB120_3_58_01
- IRB120_3_58_01_2

MESAGIRA

Editar objeto subordinado Eliminar Desconexión de una biblioteca

Entradas:
GIRAR (Digital)

Salidas:
LISTO (Digital)
DESAPARECI (Digital)

Estados guardados

Nombre	Fecha	Descripción
ESTADODINICIAL	08/01/2016 18:41:00	

Guardar estado actual Restaurar estado seleccionado Detalles Eliminar

En la parte izquierda se pueden ver todos los elementos que conforman nuestra estación y los cuales serán interconectados de manera similar a un componente inteligente.

Editar

Objeto de origen

PROYECTOROBOTIZADOS

Señal de origen

do7

Objeto de destino

MESAGIRA

Señal de destino

GIRAR

☐ Permitir conexión cíclica

Aceptar

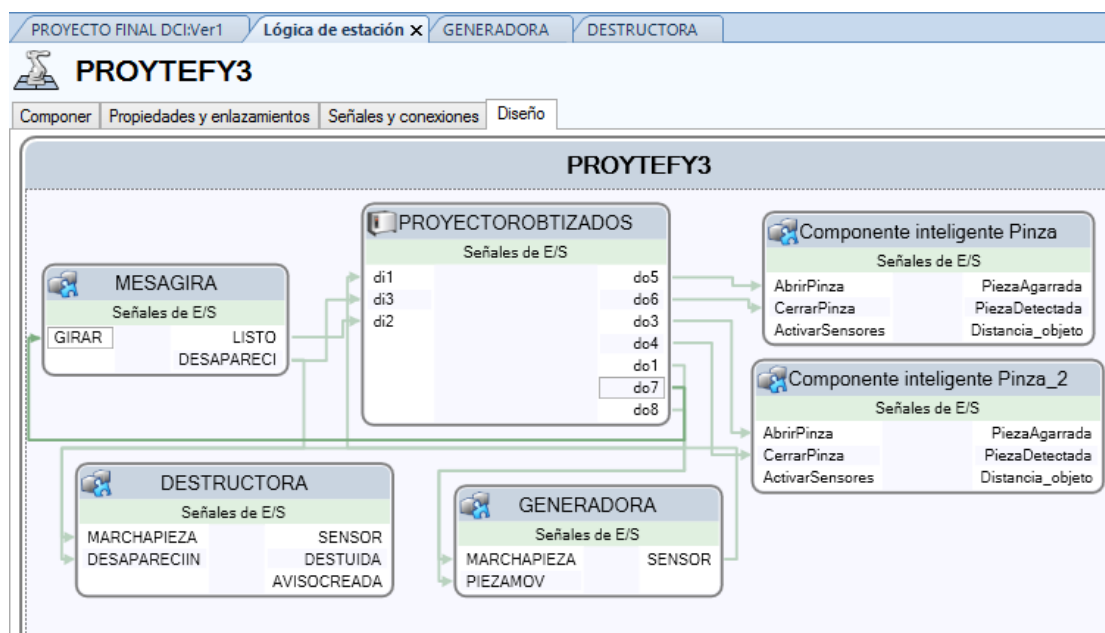
Cancelar

Objeto de origen	Señal de origen	Objeto de destino
PROYECTOROBOTIZADOS	do5	Componente inteligente
PROYECTOROBOTIZADOS	do6	Componente inteligente
PROYECTOROBOTIZADOS	do3	Componente inteligente
PROYECTOROBOTIZADOS	do4	Componente inteligente
PROYECTOROBOTIZADOS	do1	GENERADORA
PROYECTOROBOTIZADOS	do7	GENERADORA

Las conexiones de E/S se la hace tal como se indica en el cuadro anterior, se indica el objeto de origen y la señal perteneciente al mismo y luego se señala el objeto de destino y su respectiva señal, así ambas quedarán ya conectadas.

Por ejemplo para mover la cinta transportadora se necesita de una señal de entrada en el componente inteligente de la Cinta pero a su vez esta señal será gobernada por una salida digital que se la activará desde el código de RAPID, y para que su funcionamiento en simulación tenga validez se debe realizar estas conexión es de E/S.

DISEÑO DE LA LOGICA DE LA ESTACIÓN :



Se ven las entradas y salidas digitales asociadas al Sistema PROYECTOROBOTIZADOS con cada uno de los componentes inteligentes creados.

Los componentes pinza Inteligente son tomados de los archivos del Campus.

Cada salida digital se asocia a una entrada de alguno de los componentes y las señales para detectar sensores o algún tipo de estos se asocian a entradas digitales.

Conexiones de E/S			
Objeto de origen	Señal de origen	Objeto de destino	Señal de destino
PROYECTOROBOTIZADOS	do5	Componente inteligente Pinza	AbrirPinza
PROYECTOROBOTIZADOS	do6	Componente inteligente Pinza	CerrarPinza
PROYECTOROBOTIZADOS	do3	Componente inteligente Pinza_2	AbrirPinza
PROYECTOROBOTIZADOS	do4	Componente inteligente Pinza_2	CerrarPinza
PROYECTOROBOTIZADOS	do1	GENERADORA	MARCHAPIEZA
PROYECTOROBOTIZADOS	do7	GENERADORA	PIEZAMOV
PROYECTOROBOTIZADOS	do7	MESAGIRA	GIRAR

Administrar conexiones de t/S		
Señales de origen	Conexiones	Señales de destino
GENERADORA		PROYECTOROBTIZ
SENSOR		d1
MESAGIRA		PROYECTOROBTIZ
LISTO		d2
MESAGIRA		DESTRUCTORA
DESAPARECI		DESAPARECIIN
		PROYECTOROBTIZ
		d3
PROYECTOROBTIZ		Componente intelige
do5		AbrirPinza
PROYECTOROBTIZ		Componente intelige
do6		CerrarPinza
PROYECTOROBTIZ		Componente intelige
do3		AbrirPinza
PROYECTOROBTIZ		Componente intelige
do4		CerrarPinza

```

MoveJ TOMAR33,MaxVel,fine,tool0;
AbrirPinza;!ABRO PINZA AL DEJAR LA PIEZA SOBRE LA CINTA DESTRUCTORA
TOMAR33.trans.z:=TOMAR33.trans.z+28;
MoveJ TOMAR33,MaxVel,fine,tool0;
!INSTRUCCIONES PARA QUE LA CINTA DESTRUCTORA DE PIEZAS REALICE LOS MOVIMIENTOS RESPECTIVOS
!Y DESTRUYA LA PEIZA QUE SE LA HA DEJADO
SeTDO do8,1;
WaitTime 4;
SeTDO do8,0;
WaitTime 2;
! SE RETORNA A LA POSICION INICIAL
MoveAbsJ [[0,0,0,0,90,-90],[9E9,9E9,9E9,9E9,9E9,9E9]],MaxVel,fine,tool0;!Posición Inicial.
!SE DEJA LAS VARIABLES A SU VALOR INICIAL.
OPTION2:=0;
ENDWHILE
ENDPROC
!=====Procedimiento para ABRIR la pinza=====
LOCAL PROC AbrirPinza()
SetDO do5,1;

WaitTime 0.5;
SetDO do5,0;
WaitTime 0.5;

ENDPROC
!=====Procedimiento para CERRAR la pinza=====
LOCAL PROC CerrarPinza()
SetDO do6,1;
WaitTime 0.2;
SetDO do6,0;
WaitTime 1.3;
ENDPROC
ENDMODULE

```

➤ T ROB2(ROBOT QUE TOMA LA PIEZA DE LA CINTA GENERADORA Y LA DEJA EN LA MESA GIRATORIA)

```

!=====PROYECTO DE SISTEMAS ROBOTIZADOS=====
!*****UNIVERSIDAD DE VALLADOLID*****
!=====KEN VERA CHAN=====
!=====PILAR ESTFANÍA SAMANIEGO=====
!PROGRAMA ROBOT 1( EL ROBOT QUE TOMARÁ LAS PIEZAS DESPUES DE PASAR POR LA MESA GIRATORIA)

MODULE M_Main
LOCAL PERS tooldata pinza:=TRUE,[[0,0,206],[0.965926,0,0,0,0.258819]],[1.8,[2.7,0,84.4],[1,0,0,0],0.033,0.034,0.007]];
LOCAL CONST speeddata MinVel:=v50;!Velocidad mínima cuando se baja hacia una pieza o se la suelta muy cerca de las torres
LOCAL CONST speeddata MaxVel:=v600;!Velocidad para mover por encima de cada torre.
LOCAL CONST robtarget reposo:=[[380.51,-2.86,453.87],[0.00151261,-0.707302,0.0016418,-0.706909],[-1,-1,1,1],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
VAR num TOMARPIEZA;
VAR num DEJARPIEZA;
VAR num OPTION:=0;
VAR num OPTION2:=0;
VAR num useroption:=0;
VAR jointtarget POSE2:=[[29.9,13.8,-0.9,0,77.1,-60.1],[9E9,9E9,9E9,9E9,9E9,9E9]];
VAR robtarget POSE22;
VAR jointtarget POSE1:=[[-5.9,18.3,26.4,0,45.2,-95.3],[9E9,9E9,9E9,9E9,9E9,9E9]];
VAR num nu:=0;
PROC main()
!LAZO WHILE PARA REPETICIÓN CONTINUA DEL PROGRAMA
WHILE TRUE DO

AbrirPinza;! ABRIMOS PRIMERO LA PINZA
!MOVIMIENTO A LA POSICIÓN INICIAL
!LA MAYORIA DE MOVIMIENTOS ESTA MAYORMENTE BASADA EN MOVIMIENTO DE LOS EJES DEL ROBOT.
MoveAbsJ [[0,0,0,0,90,-90],[9E9,9E9,9E9,9E9,9E9,9E9]],MaxVel,fine,tool0;!Posición Inicial.
!ESPERA MIENTRAS EL USUARIO GENERA LA PRIMERA PIEZA
WHILE OPTION<>1 DO
TPReadFK OPTION,"SELECCIONE UNA OPCION","GENERAR PIEZA"," "," "," "," ";
ENDWHILE
!SI EL USUARIO YA PRESIONA EN GENERAR PIEZA CONTINUA Y LA GENERA
generarpieza;
! UNA VEZ QUE GNERA LA PEIZA SE LE PREGUNTA AL USUARIO SI DESEA RETIRAR LA PEIZA DE LA MESA
WHILE useroption<>1 DO
TPReadFK useroption,"RETIRAR PIEZA DE LA MESA","SI"," "," "," "," ";
ENDWHILE
! SI EL USUARIO DECIDE RETIRARLA SE TIENE OTRA ESPERA MIENTRAS LA PIEZA REALMENTE ALCANZA LA POSICION FINAL EN LA CINTA TRANSPORTADORA
WHILE di1=0 DO
ENDWHILE
!EJECUTA LAS ACCCIONES PARA RECOGER LA PIEZA CUANDO YA ESTA LISTA LA PIEZA EN LA CINTA TRANSPORTADORA

```

```

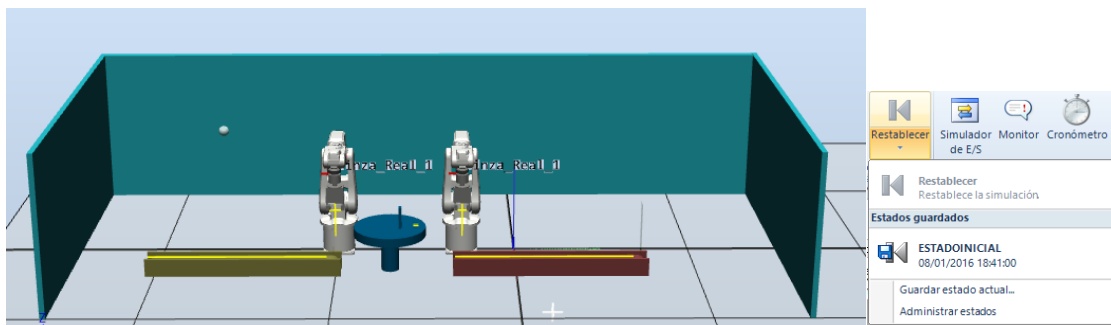
MoveAbsJ POSE1,MaxVel,fine,tool0;
CerrarPinza;
MoveAbsJ [[0,0,0,90,-90],[9E9,9E9,9E9,9E9,9E9,9E9]],MaxVel,fine,tool0;!Posición Inicial.
POSE22:=CalcRobT(POSE2,tool0);
POSE22.trans.z:=POSE22.trans.z+30;
POSE22.trans.Y:=POSE22.trans.Y+13;
MoveJ POSE22,MaxVel,fine,tool0;
POSE22.trans.z:=POSE22.trans.z-35;
MoveJ POSE22,MaxVel,fine,tool0;
POSE22.trans.z:=POSE22.trans.z+30;
POSE22.trans.Y:=POSE22.trans.Y-3;
ABRIRPinza;
MoveJ POSE22,MaxVel,fine,tool0;
!SUBE Y DEJA LA PIEZA EN LA MESA
!INSTRUCCIONES PARA HACER EL GIRO DE LA MESA EN 180 GRADOS.
SetDO do7,1;!PARA HACER GIRAR LA MESA 180 GRADOS
WAITTIME 1;
SetDO do7,0;
WAITTIME 1;
!REGESO A LA POSICION INICIAL
MoveAbsJ [[0,0,0,90,-90],[9E9,9E9,9E9,9E9,9E9,9E9]],MaxVel,fine,tool0;!Posición Inicial.
IF di2=1 THEN
    MoveAbsJ [[0,0,0,90,-90],[9E9,9E9,9E9,9E9,9E9,9E9]],MaxVel,fine,tool0;
ENDIF
!RETORNO EL VALOR DE LAS OPCIONES DEL USUARIO PARA CONTINUAR LA EJECUCIÓN SIN PROBLEMAS
OPTION:=0;
useroption:=0;
ENDWHILE
ENDPROC
!PROCEDIMIENTO PARA GENERAR LA PIEZA
LOCAL PROC generarpieza()
    SetDO do1,1;
    WaitTime 0.5;
    SetDO do1,0;
    WaitTime 0.5;
ENDPROC
!=====Procedimiento para ABRIR la pinza=====
LOCAL PROC AbrirPinza()
    SetDO do3,1;
    WaitTime 0.5;
    SetDO do3,0;
    WaitTime 0.5;
ENDPROC
!=====Procedimiento para CERRAR la pinza=====
LOCAL PROC CerrarPinza()
    SetDO do4,1;
    WaitTime 0.2;
    SetDO do4,0;
    WaitTime 1.3;
ENDPROC

¡NOTAAAAA: Estos programas no funcionan hay un error por los comentarios que se los agrego después, los que funcionan
están dentro de la estación empaquetada que se envió.
ENDMODULE

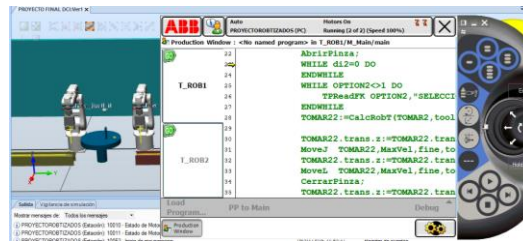
```

✓ FUNCIONAMIENTO DE LA ESTACIÓN

En primer lugar se tiene un estado inicial para poderlo restablecer en caso de algún problema con las piezas, ya que el componente inteligente Pinza posee sensores que suelen llevarse piezas que detecta. La estación inicialmente se encuentra de la manera siguiente:



En primer lugar se da al botón PLAY en la pestaña de simulación y aparecerá en la FlexPendant los mensajes para el Usuario, pero previo la apertura de la flexpendant:



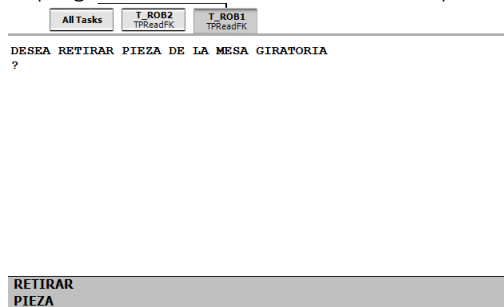
La primera de las opciones para el usuario es la generar la pieza:



Una vez generada la pieza se le pregunta si desea coger la pieza y ponerla sobre la mesa giratoria:



Luego la tarea 2 del segundo robot pregunta si el usuario desea retirar la pieza de la mesa giratoria:



Posteriormente el proceso se repite una y otra vez.

✓ CONCLUSIONES:

- El proyecto nos ha permitido conocer el entorno de trabajo de RobotStudio, nos ha permitido profundizar en el uso de componentes inteligentes y el manejo de conexiones de E/S dentro de la lógica de la estación para poder simular una estación como si fuera la realidad. Además de la ventaja de saber la programación en lenguaje Rapid ya que mediante esta se tiene la programación dentro de un robot real. Nos dimos cuenta además que los componentes inteligentes tienen un sin número de funciones que nos permiten realizar varias aplicaciones de cualquier tipo.
- Se debe tener en cuenta la lógica de programación o de conexión entre señales de robot estudio ya que esta es la que permitirá el adecuado o incorrecto funcionamiento durante la simulación.