

SESIÓN 6

=====

Podemos echarle un vistazo al vídeo demostrativo facilitado 'videoDemostrativo.wmv' para tener una idea de lo que queremos hacer en esta sesión. Se facilita también una estación de partida, 'sesion8_EstacionDePartida.rspag', con la que ya han completado los pasos de la Parte I y de la Parte II (excepto el apartado 1.3).

PARTE I. Creación de entorno inicial:

- 1) Abrir paquete: 'Sesion8_EstacionDePartida.rspag'. Dicho paquete contiene un robot de tipo IRB-6640 con un componente inteligente de tipo ventosa, como el usado en alguna sesión anterior, disponiendo, además, de un controlador con módulo de comunicaciones de tipo 'DeviceNet' (incluyendo las señales 'SD_ActivaVentosa' y 'ED_PiezaCogida').
- 2) Se creará una mesa de trabajo ('mesa de paletizado') con dos partes:
 - 2.1) Sólido tetraedro en la posición [0,-2000,0] de dimensiones (1000x1000x750) .
 - 2.2) Sólido tetraedro en la posición [-1000,-2000,0] de dimensiones (1000x1000x750).
- 3) Se creará un sólido que simule 'cinta transportadora':
Sólido tetraedro en la posición [1000,-1000,720] de dimensiones (1000x3000x30), de manera que la superficie quede a la misma altura de la mesa.
- 4) Se definirá la posición de reposo del robot: 'pReposo', coincidente con la posición inicial del brazo, mediante el botón 'Programar posición' en la pestaña 'Posicion Inicial'. Dicha posición quedará asociada al 'wobj0'.
- 5) Se definirán sendos objetos de trabajo asociados a los dos elementos anteriores:
 - 5.1) Objeto de trabajo 'W0_Paletizado', en la esquina superior izquierda de la mesa de paletizado (superior izquierda más alejada del robot; izquierda, desde el punto de vista del robot mirando a dicha mesa).
 - 5.2) Se definirán dos posiciones para la herramienta 'Ventosa', en el centro de cada una de las dos partes de la mesa de paletizado:
 - Posición izquierda se denominará 'pBasePalePiezasCortas'.
 - Posición derecha se denominará 'pBasePalePiezasLargas'.Ambos puntos de girarán (Z hacia abajo, X hacia base robot), se verificará su alcanzabilidad, visualizando el robot en posición.
 - 5.3) Objeto de trabajo 'W0_CintaTransportadora', en la esquina superior derecha más próxima al robot.
- 6) Se crearán dos piezas modelo de las piezas que luego se procesarán. Se colocarán en el extremo de la cinta para ayudar a la ubicación de los sensores:
 - 6.1) 'PiezaLarga': de dimensiones (500x1000x30), colocada de forma que su arista inferior derecha esté sobrepuesta a la arista superior derecha de la cinta y centrada en ésta (posición [1250,-1000,750]).
 - 6.2) Definiremos la posición base para la cogida adecuada de las piezas largas: 'pBaseCogidaPiezaLarga': En el centro de la cara superior de dicha pieza, con la orientación habitual (Z hacia abajo, X hacia base robot). Esta posición debe estar asociada al 'W0_CintaTransportadora'. Ojo: restarle 30 a la Z de la posición enseñada, para que quede a ras de la cinta (es posición base).
 - 6.3) 'PiezaCorta': de dimensiones (500x500x30), colocada sobre la pieza larga, también al límite derecho de la cinta (posición [1250,-1000,750+30]).

- 6.4) Definir posición de cogida para este tipo de pieza: 'pBaseCogidaPiezaCorta': En el centro de la cara superior de dicha pieza, con la orientación habitual. También asociada al 'W0_CintaTransportadora'. Ojo: restarle 60 a la Z de la posición enseñada con la bolita, porque es posición base y se quiere a ras de la cinta.
- 7) Crear trayectoria de prueba para que el robot recorra la siguiente secuencia: 'pReposo' -> 'pBaseCogidaPiezaLarga' -> 'pBaseCogidaPiezaCorta' -> 'pBasePalePiezasCortas' -> 'pBasePalePiezasLargas'. Validar configuración automática.

PARTE II. Creación de sensores para detección de pieza en cinta transportadora:

- 1) Crear un componente inteligente 'CI_SensoresCinta', que contendrá dos detectores de tipo 'LineSensor':
- 1.1) 'LineSensorFinalCinta': Rayo de 2mm de radio barriendo la línea final de la cinta transportadora, a la mitad de la altura de una pieza por encima de cinta:
- ```
Start: [1000,-1000,750+15]
End: [2000,-1000,750+15]
Radius: 2
```
- 1.2) 'LineSensorPiezaLarga': Rayo de 2mm de radio paralelo al anterior, algo más atrás de lo que alcanza una pieza corta y a la misma altura del otro:
- ```
Start: [1000,-1000+500+50,750+15]
End: [2000,-1000+500+50,750+15]
Radius: 2
```
- 1.3) Componer diseño 'CI_SensoresCinta' como aparece en imagen 'CI_SensoresCinta.jpg':
- Añadir entrada: 'ActivaSensores'.
 - Añadir salida: 'FinalCintaDetectado'.
 - Añadir salida: 'PiezaLargaDetectada'.

PARTE III. Simulación de suministro de piezas y movimiento de cinta:

En primer lugar, se simulará suministro únicamente de piezas cortas, a petición mediante señal.

- 1) Montar componente inteligente 'CI_CintaTransportadora' de forma preliminar diseño de la imagen: 'CI_CintaTransportadoraSoloPiezasCortas.jpg'.
Intervienen:
- 1.1) Componente de tipo 'Source': lo denominaremos: 'GeneradorPiezaCorta', que genera una copia de la pieza corta matriz en la posición que queramos. En particular, en el principio de la cinta:
- ```
Source: PiezaCorta
Position: [1250,1500,750]
```
- 1.2) Componente de tipo 'Queue', para aglutinar en forma de cola todas las piezas "copia" que se van creando.
- 1.3) Componente de tipo 'LinearMover' para desplazar las piezas en la cola:
- ```
Object: El objeto a desplazar es el componente cola anterior.
Direction: Vector dirección del movimiento: [0,-1000,0].
Speed: 200mm/s.
```
- 1.4) Crear entradas (no habrá salidas en este caso) y montar diseño mostrado en: 'CI_CintaTransportadoraSoloPiezasCortas.jpg'.
Como se ve: se genera una nueva pieza por petición a través de una señal. Se desencolará una pieza, cuando es detectada por el sensor de final de cinta.
- 2) Llevar las piezas matriz (corta y larga) a un sitio donde no estorben, por ejemplo, al

origen absoluto y hacer invisibles.

- 3) Crear I/O digitales en el controlador mostradas en:
'PanelEntradasSalidasDigitales.jpg'
- 4) Montar 'Lógica de estación' mostrada en: 'LogicaEstacion.jpg'.
- 5) Simular funcionamiento. El robot no se moverá, puesto que no se ha programada aún, pero
las piezas se generarán y se desplazarán en la cinta. Añadirle al principal un
"WaitTime 60", para que no termine la ejecución inmediatamente. No olvidarse se
activar manualmente 'SD_ActivaSensores'.

PARTE IV: Programación del robot en RAPID:

El robot deberá programarse para apilar ordenadamente las piezas que llegan por la cinta. Puede ser recomendable copiar los trozos de código que se facilitan a continuación, desde el archivo de 'Sesion8_Pasos.txt' en lugar de desde su versión en PDF.

- 1) Completar los procedimientos 'Inicializa()' y 'Finaliza()':

```
PROC Inicializa ()
  MoveJ pReposo, v3000, fine, Ventosa, \WObj:=wobj0;
  Reset SD_ActivaVentosa;
  SetDO SD_ActivaSensores, 1;
  SetDO SD_MueveCinta, 1;
  Reset SD_PideNuevaPieza;
ENDPROC
```

```
PROC Finaliza ()
  MoveJ pReposo, v3000, fine, Ventosa, \WObj:=wobj0;
  Reset SD_ActivaVentosa;
  Reset SD_PideNuevaPieza;
  Reset SD_MueveCinta;
  Reset SD_ActivaSensores;
ENDPROC
```

- 2) Completar el 'Main()':

```
PROC Main ()
  CONST num altoPieza := 30;
  VAR num nPiezas;

  Inicializa;

  FOR nPiezas FROM 1 TO 20 DO
    WaitDI ED_SensorFinalCinta, 1;
    LlevaPiezaAPale nPiezas, altoPieza;
  ENDFOR

  Finaliza;
ENDPROC
```

- 3) Completar el procedimiento 'LlevaPiezaAPale()':

```
PROC LlevaPiezaAPale (num nPiezas, num altoPieza)
  VAR robtarget pCoger;
  VAR robtarget pSoltar;
  VAR robtarget pCogerAprox;
  VAR robtarget pSoltarAprox;
  VAR num distSeguridad := 300;

  pCoger := Offs (pBaseCogidaPiezaCorta, 0, 0, altoPieza);
  pSoltar := Offs (pBasePalePiezasCortas, 0, 0, altoPieza*nPiezas);

  pCogerAprox := Offs (pCoger, 0, 0, distSeguridad);
```

```

    pSoltarAprox := Offs (pSoltar, 0, 0, distSeguridad);

    MoveJ pCogerAprox, v3000, z100, Ventosa, \WObj:=W0_CintaTransportadora;
    MoveLD0 pCoger, v1000, fine, Ventosa, \WObj:=W0_CintaTransportadora,
SD_ActivaVentosa, 1;
    WaitDI ED_PiezaCogida, 1;
    MoveL pCogerAprox, v1000, z100, Ventosa, \WObj:=W0_CintaTransportadora;
    MoveJ pSoltarAprox, v3000, z100, Ventosa, \WObj:=W0_Paletizado;
    MoveLD0 pSoltar, v1000, fine, Ventosa, \WObj:=W0_Paletizado, SD_ActivaVentosa,
0;
    WaitDI ED_PiezaCogida, 0;
    MoveL pSoltarAprox, v1000, z100, Ventosa, \WObj:=W0_Paletizado;
    MoveJ pCogerAprox, v3000, z100, Ventosa, \WObj:=W0_CintaTransportadora;
ENDPROC

```

PARTE V. Introducción de segundo robot y coordinación:

Se introducirá un segundo robot del mismo tipo que accederá a cada una de las piezas que se van colocando. Al existente se le denominará 'Robot1', al nuevo se le denominará 'Robot2'.

- 1) Introducir un segundo robot IRB-6640, con herramienta 'MyTool', cargar su controlador, también con módulo de comunicaciones DeviceNet.
 Posición: [0,-3000,0]
 Orientación: [0,0,90]
- 2) Crearle:
 - Posición de reposo, con 'Inicio/Programar posición' asociándola a 'wobj0'.
 - Copia del objeto de trabajo de mesa de paletizado, diciendo que "no actualice posición".
 - Los puntos del objeto de trabajo se habrán copiado también. Rotamos ambos puntos base, tanto el de apilamiento de piezas cortas como el de apilamiento de piezas largas, 180º respecto al eje Z local, para que 'Robot2' acceda a esas posiciones de forma más natural.
- 3) Crear trayectoria "dummy" para el 'Robot2' que contenga la posición de reposo y el nuevo punto. Sincronizar controlador de 'Robot2'.
- 4) Crear E/S de coordinación entre ambos robots:
 - 4.1) Crear E/S adicionales en 'Robot1' ("Access Level: Default", para que no se pueda cambiar el estado de estas señales mediante click en la pantalla de simulación, sino que sólo las pueda modificar el robot correspondiente desde programa:
 - SD_PiezaDisponible
 - ED_PiezaDisponibleAck
 - 4.2) Crear dispositivo DeviceNet para 'Robot2' y añadir las siguientes E/S, de nuevo con "Access Level" a "Default":
 - ED_PiezaDisponible
 - SD_PiezaDisponibleAck
 - 4.3) Conectar a través de 'Lógica de estación':
 - Robot1/SD_PiezaDisponible -> Robot2/ED_PiezaDisponible
 - Robot2/SD_PiezaDisponibleAck -> Robot1/ED_PiezaDisponibleAck
- 5) Modificar programa de 'Robot1' para que al depositar pieza active 'SD_PiezaDisponible' y espere la respuesta a través de 'ED_PiezaDisponibleAck'.
 - 5.1) Actualizar 'Inicializa()' y 'Finaliza()', para incluir:
 - Reset SD_PiezaDisponible;
 - 5.2) Actualizar 'Main()' con el siguiente contenido:


```

PROC Main ()
  CONST num altoPieza := 30;
  VAR num nPiezas;

```

```

Inicializa;

FOR nPiezas FROM 1 TO 20 DO
    WaitDI ED_SensorFinalCinta, 1;
    LlevaPiezaAPale nPiezas, altoPieza;
    SetDO SD_PiezaDisponible, 1;
    WaitDI ED_PiezaDisponibleAck, 1;
    Reset SD_PiezaDisponible;
ENDFOR

Finaliza;
ENDPROC

```

- 6) Preparar programa de 'Robot2' que espere activación de 'ED_PiezaDisponible' y, a continuación, active 'SD_PiezaDisponibleAck'. El Robot2 accederá a la pieza depositada, simulando que se le hace un taladro con herramienta en el centro de pieza.

6.1) Preparar procedimiento 'Inicializa()':

```

PROC Inicializa ()
    MoveJ pReposo, v3000, fine, MyTool;
    Reset SD_PiezaDisponibleAck;
ENDPROC

```

6.2) Preparar procedimiento 'Finaliza()':

```

PROC Finaliza ()
    MoveJ pReposo, v3000, fine, MyTool;
    Reset SD_PiezaDisponibleAck;
ENDPROC

```

6.3) Prepara procedimiento 'Main()':

```

PROC Main ()
    VAR num nPiezas;

    Inicializa;

    FOR nPiezas FROM 1 TO 20 DO
        WaitDI ED_PiezaDisponible, 1;
        SetDO SD_PiezaDisponibleAck, 1;
        WaitTime 0.3;
        Reset SD_PiezaDisponibleAck;
        TaladraPieza pBasePalePiezasCortas, nPiezas, 30;
    ENDFOR

    Finaliza;
ENDPROC

```

6.4) Preparar procedimiento que simula taladrado de pieza:

```

PROC TaladraPieza (robtargt pBase, num nPiezas, num altoPieza)
    VAR robtarget pTaladrado;
    VAR robtarget pAprox;
    VAR robtarget pEspera;
    VAR num distSeguridadZ := 300;
    VAR num distSeguridadEsperaX := -400;

    pTaladrado := Offs (pBasePalePiezasCortas, 0, 0, altoPieza*nPiezas);
    pAprox := Offs (pTaladrado, 0, 0, distSeguridadZ);
    pEspera := Offs (pTaladrado, distSeguridadEsperaX, 0, distSeguridadZ);

    MoveJ pAprox, v3000, z100, MyTool, \WObj:=W0_Paletizado;
    MoveL pTaladrado, v1000, fine, MyTool, \WObj:=W0_Paletizado;
    ! Aquí accionaríamos taladro. Simulamos con espera de 0.3 segundos:
    WaitTime 0.3;

```

```
MoveL pAprox, v1000, z100, MyTool, \WObj:=W0_Paletizado;  
MoveJ pEspera, v3000, z100, MyTool, \WObj:=W0_Paletizado;  
ENDPROC
```

7) Simular funcionamiento.

PARTE VI. Generación de ambos tipos de piezas y con temporización:

Se producirán piezas largas o cortas aleatoriamente en el inicio de la cinta. En esta parte, simplemente se dan algunas indicaciones, a modo de primeros pasos, para que los alumnos puedan completarla de forma autónoma.

1) Se requerirá una ampliación del componente inteligente: 'CI_CintaTransportadora', añadiendo los siguientes nuevos elementos:

- 1.1) 'Source': que denominaremos: 'GeneradorPiezaLarga', que genera una copia de la
pieza larga matriz al principio de la cinta:
Source: PiezaLarga
Position: [1250,1500,750]
- 1.1) 'Random': Generador de nro. aleatorio entre 0 y 1.
- 1.2) 'Comparer': Comparador para ver si el nro. aleatorio generado es ≤ 0.5 .
- 1.3) 'LogicGate[AND]': Para activar creación de pieza corta cada vez que se genere un
nro. aleatorio y éste sea menor o igual de 0.5.
- 1.4) 'LogicGate[AND]': Para activar creación de pieza larga cada vez que se genere un
nro. aleatorio y éste sea mayor de 0.5.
- 1.5) 'LogicGate[NOT]': Para obtener el negado de la señal de salida del comparador.
- 1.6) 'Timer': Para generar pieza automáticamente cada 8 segundos, sin necesidad de activar señal: 'SD_PideNuevaPieza'.

2) Conectar todos los componentes como se muestra en:

'CI_CintaTransportadoraFinal.jpg'.

La 'lógica de la estación' no requiere ser cambiada.

3) Deberá definirse una nueva señal de comunicación en cada uno de los robots, de forma que el primer robot pueda informar al segundo de si la pieza que ha depositado es corta o larga y, por tanto, si la ha colocado en la pila de piezas cortas o largas. Estas señales deberán interconectarse a través de lógica de estación:

Robot1/SD_EsPiezaLarga -> Robot2/ED_EsPiezaLarga

4) Habrá que modificar los programas de los robots, para que en función del tipo de pieza adecúen su funcionamiento.