PRÁCTICA 5:

Paletizado.

El objetivo de esta práctica consiste en seguir avanzando en la programación en RAPID, definiendo posición y orientación de puntos objetivos.

PARTE 1. NUEVO CONTENIDO SOBRE RAPID Y ROBOTSTUDIO

En esta sección se van a incluir nuevas funciones RAPID, además se van a declarar funciones que devuelven un parámetro de salida.

- Funciones y control de flujo en RAPID:

Algunas funciones de utilidad:

- Control de flujo con TEST, permite seleccionar la ejecución de un conjunto de operaciones para un *valor* concreto de una variable *nombre_vble*:

```
TEST nombre_vble
CASE valor1, valor2:
    …..
CASE valor3:
    …..
CASE valor4, valor5:
    …..
DEFAULT
    …..
ENDTEST
```

- La función DIV realiza la división entera de dos números.

Ejemplo:

a:= 10 DIV 3; La variable *a* tomaría el valor 3.

- La función MOD devuelve el resto de una división entera entre dos números.

Ejemplo:

b:= 10 MOD 3; La variable *b* tomaría el valor 1.

- La función TPReadNum pide y espera que se introduzca un dato numérico desde la consola (FlexPendant). En RobotStudio, para poder emplear esta función debe estar abierto el FlexPendant o bien una ventana de operador (pestaña *controlador → ventana del operador*).

Ejemplo:

TPReadNum altura_torre, “Indica la altura de la torre”; Tal y como muestra la Figura 1, en la pantalla de usuario del FlexPendant aparecería la frase “Indica la altura de la torre”, y quedaría a la espera de que el usuario introdujese un valor para dicha altura.

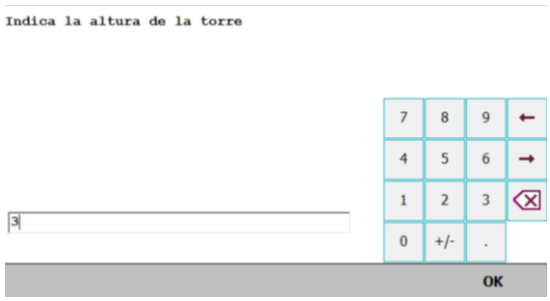


Figura 1. Ejemplo de empleo de *TPReadNum*. Visualización de la pantalla de FlexPendant.

- La función TPErase borra el texto mostrado en el FlexPendant.
- La función OrientZYX obtiene el cuaternio correspondiente a una rotación definida en formato *EulerZYX*, respecto a los ejes de coordenadas de la tarea.

Ejemplo:

Punto.rot:= OrientZYX(90,45,180); Esta instrucción asigna a la orientación de la variable *Punto*, el cuaternio correspondiente a realizar una rotación de 90º sobre el eje *Z*, seguida de una rotación de 45º sobre el eje *Y* y, finalmente, una rotación de 180º sobre el eje *X* (rotación sobre ejes móviles).

- La función RelTool se utiliza para añadir un desplazamiento y/o una rotación, **expresada en el sistema de coordenadas de la herramienta activa**, a una posición concreta del robot. Sus argumentos son: RelTool (Point, Dx, Dy, Dz \Rx \Ry \Rz), donde las rotaciones son opcionales.

Ejemplo:

Punto2:= RelTool(Punto1,0,0,100 \Rx:=90); Esta instrucción realiza un desplazamiento del Punto1 de 100 mm sobre el eje *Z* de la herramienta, así como una rotación del mismo de 90º sobre el eje *X* de la herramienta. El resultado se lo asigna al Punto2.

– Declaración de una función con parámetro de salida.

En la Figura 2 se muestra un ejemplo de declaración de función (*my_function*) que tiene como argumento una variable de tipo *num* (numero) y como salida un dato también de tipo *num*. Para devolver el dato de salida se debe emplear la instrucción RETURN. Finalmente, en el procedimiento *main* del mismo ejemplo se detalla como se realizaría la llamada a dicha función. El parámetro de entrada (numero2) tiene valor 3, por lo que la salida de la función, que se asigna a la variable numero1, tomaría el valor 6.

loffs es con respecto a la tarea con el z hacia arriba en cambio en reltool es con respecto a la herramienta

```
PROC main ()
VAR num numero1;
VAR num numero2;
numero2:=3;
numero1:=My_function (numero2);
ENPROC
FUNC num my_function (num numero)
   RETURN numero*2
ENDFUNC
```

Figura 2. Ejemplo de procedimiento *main* con llamada a *my_function*.

PARTE 2. EJERCICIOS DE LA PRÁCTICA

Los ejercicios de esta práctica están divididos en dos secciones. La primera (A) es obligatoria y puntúa hasta un máximo de 6 puntos sobre el total de la práctica 5. La segunda parte (B) es opcional y permitirá obtener una calificación superior.

A. CREACIÓN DE LA TORRE

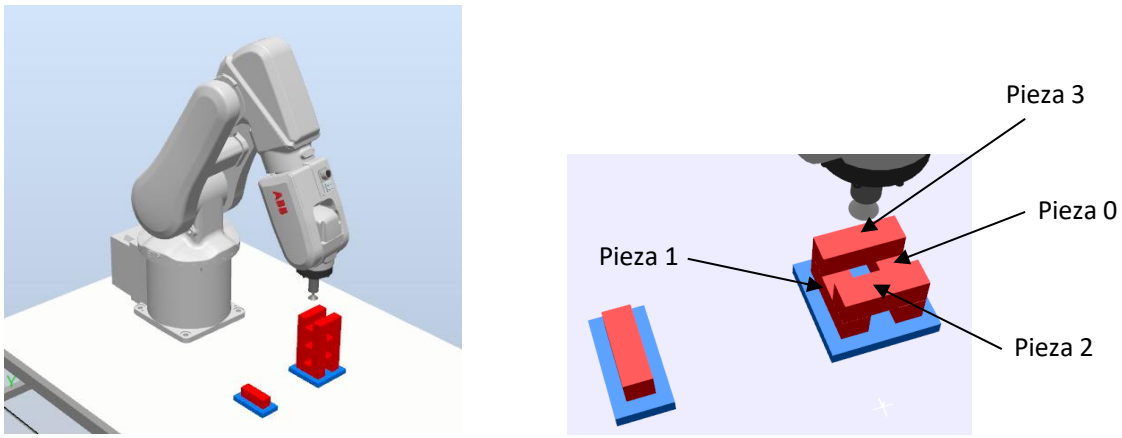
Este ejercicio parte de la estación de trabajo con nombre Práctica5_inicio que se puede descargar del campus virtual de la asignatura, en ésta se crea una torre situando una pieza sobre otra, si está activa la entrada Dpieza, hasta que se active la entrada DFin. El objetivo que se plantea ahora es modificar dicha estación (se deberá modificar **únicamente** el programa RAPID, no es necesaria la generación de ningún elemento ni punto adicional en la estación) para crear una torre tal y como se muestra en la Figura 3.a y se puede visualizar en el video *Práctica 5_A* que podéis encontrar en el campus virtual.

El funcionamiento general de la aplicación se describe a continuación:

1. Al comenzar y al concluir la aplicación el manipulador se debe situar en la posición de reposo.
2. Se le solicitará al usuario que introduzca por la consola el número de plantas de las que va a disponer la torre (ver función TPReadNum en PARTE 1). Cada una de las plantas estará formada por 4 piezas, tal y como muestra la Figura 3.b que se colocarán en el orden indicado en dicha figura.
3. Mientras que la torre no se haya completado, o bien no se active la señal *Dfin* se realizarán las siguientes acciones:
   - ❑ Se recoge una pieza (procedimiento Pick).
   - ❑ Se calcula la posición y la orientación donde se debe colocar la siguiente pieza de la torre. Para ello se diseñará una función con la siguiente cabecera (ver declaración de funciones en PARTE 1):

     FUNC *robtarget* calcula_posicion(*num* num_pieza)

     donde num_pieza es una variable numérica que indica el número de pieza que se colocará y por tanto de la que se quiere calcular la posición y orientación. Por otro lado, la salida, de tipo *robtarget*, es el punto calculado en el que se colocará dicha pieza.
   - ❑ Se sitúa la pieza en la torre (procedimiento Place).



(a) (b)

Figura 3. Construcción de una torre con 4 piezas por planta.

B. AMPLIACIÓN DE FUNCIONALIDAD

Modificar el programa RAPID para que no concluya la aplicación cuando haya terminado la torre, sino que vuelva a solicitar el número de plantas, de tal forma que la torre construida se adapte a la nueva consigna. Por ejemplo, si la torre tiene actualmente 2 plantas, y se indica que se desean 4, colocar dos plantas adicionales. Si, por el contrario, se indica que se desea 1 planta, se deberá quitar la última planta construida, y soltará las piezas de la torre en otra ubicación. Un ejemplo de dicha funcionalidad se puede visualizar en el video *Práctica 5_B* que podéis encontrar en el campus virtual.

Puede ser de utilidad emplear una variable que indique si el manipulador está en estado de espera a que se le introduzca un valor de altura de la torre, si está construyendo alguna planta de la torre o si está desmontándola.

Para que la simulación de estas funcionalidades funcione correctamente, se deberá, además cambiar la lógica de la estación.