SESIÓN 5 (PRÁCTICA ENTREGABLE 2)

En esta sesión, haremos uso de la herramienta 'Ventosa' que se creó y configuró en la sesión anterior. En este caso, el objetivo será, como indica el enunciado, trasladar torres de piezas de una ubicación original a una ubicación destino, de forma que el número de piezas se pueda especificar con libertad y que la altura de las piezas también sea configurable.

Se proponen dos versiones de esta práctica, la versión básica y la versión avanzada. La presente guía está particularizada para la versión básica, aunque serviría con pocas variaciones para la sesión avanzada. Se pide entregar, al menos la sesión básica. La versión avanzada, supone una entrega complementaria para obtener una calificación más alta en esta práctica. Para más detalles, ver la 'Actividad' correspondiente en Enseñanza Virtual.

Parte I:

1) Abriremos el archivo Pack&Go 'PracticaEntregable2EstacionDePartida.rspag', que contiene los elementos de partida necesarios, incluida la ventosa plenamente operativa.

2) En primer lugar, comprobamos que la ventosa, efectivamente, tiene asociado el componente inteligente que le da funcionalidad.

3) Verificamos las posiciones que están predefinidas, que, de acuerdo al enunciado, son: 'pReposo', 'pBaseTorre1', 'pBaseTorre2'.

4) Comprobamos que ya están definidas las dos señales de I/O para manejo de la ventosa. Esto lo haremos a través de la pestaña 'Controlador', haciendo doble click en la opción 'Sistema de E/S / DeviceNet / MiDeviceNet'.

5) Medimos el alto y el diámetro de la pieza muestra que aparece en el entorno.

6) Podemos hacer copias de la pieza muestra, para ir formando una primera torre.

7) Una vez que se tenga la(s) torre(s) preparadas en la(s) posición(es) de partida, se recomienda guardar el estado inicial del escenario, para que, cuando finalice el programa, puede fácilmente restablecerse todo al punto de partida y pueda hacerse una nueva simulación. Se recuerda que esto se hará seleccionando, en la pestaña 'Simulación', el botón 'Restablecer', el cual deberá desplegarse para poder acceder a la opción 'Guardar estado actual'. Como ya se indicó en alguna sesión anterior, se recomienda no guardar es estado actual de todos los elementos, sino únicamente la posición del manipulador, de las mesas de trabajo y de las piezas.

Parte II:

8) Pasamos a RAPID, para comprobar si las posiciones ya están definidas en el programa.

9) Preparamos el procedimiento 'main ()', definiendo las variables que almacenen las dimensiones de las piezas y haciendo las llamadas a un procedimiento que se encargue de trasladar la torre desde la posición original a la posición de destino, usando una determinada posición intermedia.

10) Las posiciones destino e intermedia se definirán, dentro del programa, de acuerdo al siguiente criterio: posición destino situada sobre la mesa, a una distancia igual a tres veces el diámetro de las piezas, en la dirección positiva del eje Y. La posición intermedia, situada a igual distancia de la posición inicial, pero en lado opuesto a lo largo del eje Y. Nota: Pese a que en el vídeo, finalmente, se ha usado un desplazamiento de dos veces el diámetro de la pieza, quizás sea preferible usar tres veces).

11) El traslado de la torre tiene dos fases: en primer lugar, se traslada la torre a la posición intermedia, donde quedará invertida, y a continuación, se traslada desde

la posición intermedia a la posición destino final, donde ya quedarán las piezas en el mismo orden que estaban inicialmente. Teniendo en cuenta que la secuencia de acciones para cada uno de estos dos pasos necesarios en el traslado es la misma, se puede optimizar y simplificar sustancialmente el código requerido.

12) Si se quiere que, tal y como ocurre en el vídeo de muestra, en una misma ejecución del programa, se haga el ensayo con dos torres ya preparadas de distinta altura, puede optarse por hacer sendas llamadas, una a continuación de otra, en el 'main ()'.

13) Nuevamente, se recomienda preparar un procedimiento 'Inicializa()' y otro 'Finaliza()', que se encarguen de garantizar que la ventosa se desactiva al comenzar y al terminar la ejecución del programa.

14) Debe usarse con buen criterio la precisión de los movimientos: precisión baja (por ejemplo, 'z100', precisión alta, teniendo muy en cuenta la diferencia entre las dos variantes de esta última, 'z0' y 'fine',que ya se explicó en alguna sesión anterior.

15) Debe tenerse especial cuidado con la sincronización de los movimientos y la activación/desactivación de la ventosa. Aunque no es imprescindible, puede recurrirse a una instrucción que unifica el movimiento con la activación/desactivación de una señal. Se trata de la instrucción 'MoveLDO' ó 'MoveJDO'. La sintaxis es la misma que la de las correspondientes 'MoveL' y 'MoveJ', pero se añaden un par de argumentos al final indicando la señal que se quiere poner a un cierto estado y si ese estado es 0 ó 1. Como digo, no es imprescindible y se puede seguir utilizando el par de instrucciones separadas, primero la orden de movimiento y, a continuación, la instrucción de activación/desactivación de señal.

16) A partir de estos consejos, cada alumno, de forma individual, deberá realizar el trabajo solicitado.

17) Recordemos que, para poder ver durante la simulación el estado de las señales de entrada/salida, podemos desplegar el tablero de señales, desde la pestaña 'Simulación', con el botón 'Simulador de E/S', y seleccionando en 'Dispositivo' 'MiDeviceNet'.