1 EJEMPLO DE CREACIÓN DE UN PROGRAMA EN EL SIMULADOR

Aquí se incluye una pequeña guía para crear un primer programa con posiciones y objetos para mostrar algunas de las capacidades del simulador.

Los pasos importantes serán indicados con el signo \checkmark . Estos pasos irán acompañados de una posterior explicación.

Enunciado del problema

El escenario que vamos a crear consistirá en un objeto que recogeremos de una posición llamada POS1 y lo dejaremos en la POS2. Esta acción será realizada cuando se active la entrada IN[1], la cual nos indicará cuando existe una objeto en POS1. Debemos recoger el objeto desde una posición situada 10cm por encima de POS1, la cual nos servirá como paso previo a la recogida.

Una vez depositado el objeto en POS2 pasando por una posición previa situada 10cm por encima de POS2, debemos volver a la posición inicial.

Pasos a seguir:

✓ cargar el simulador con el archivo JSCORBOT3D.BAT

Existen dos modos para arrancar el simulador, el que usaremos aquí es el que nos permite tener la visualización del modelo en 3D. Una vez arrancado nos aparecerá el logotipo del programa con una barra situada en la parte inferior que nos indicará el porcentaje de carga del programa.

 ✓ posicionar las ventanas de manera que tengamos en pantalla las más importantes: la de edición, la de representación, la de entradas y salidas digitales, la ventana de diálogo y la pistola de programación.

Debemos tener una buena forma de posicionar las ventanas ya que esto nos ahorrará mucho tiempo a la hora de acceder a las acciones más comunes. Esta acción sólo debemos realizarla una vez ya que el simulador guardará la posición y visibilidad de todas estas ventanas cuando sea finalizado. De esta forma la próxima vez que el simulador sea arrancado tendremos la misma disposición en la pantalla.

Una buena disposición, y la que se carga por defecto en el simulador, es la mostrada por la siguiente captura de pantalla:



✓ crear un nuevo programa vacío

Para realizar esta acción debemos pulsar sobre el botón \Box . Será creado un programa vacío con el nombre NUEVOO. Este programa podremos grabarlo con cualquier otro nombre después, este nombre es solamente temporal.

✓ crear las posiciones POS1, POS2 y las situadas 10cm por encima de aquellas.

Las posiciones POS1 y POS2 podemos situarlas donde queramos en el espacio coordenado, pero lo normal es actuar de la siguiente manera:

- mover el robot mediante la pistola de programación hasta una posición alcanzable, es decir, que esté dentro del volumen de trabajo del robot.
- alcanzar esa posición con el pitch y roll adecuado.
- definir la posición
- igualar la posición actual a la posición creada

Para crear la posición usaremos en modo directo el comando "DEFP POS1". Para mover el robot hasta la posición que hayamos decidido usaremos los controles proporcionados por la pistola de programación.

Una vez alcanzada dicha posición usaremos el comando "HERE POS1" para almacenar la posición actual en la posición recientemente creada. Actuaremos de igual manera con la posición POS2. La posición POS1 viene dada por los siguientes atributos:

🗎 Salida	de texto			_ 🗆	×
>listpv	pos1				
Posicio	n POS1				
1:3831	2:4618	3:5650	4:-5046	5:0	
X:2000	Y:2000	Z:2000	P:1800	R:0	
Ok					333
					•
				88888 🔹 🕨	

La posición POS2 viene dada por los siguientes atributos:

🗎 Salida	de texto			<u> ×</u>
>listpv	/ pos2			
Posicio	on POS2			
1:-169	2:4618	3:5650	4:-5046	5:0
X:-98	Y:2000	Z:2827	P:1799 P	२:०
Ok				265
				-
4 33333333				

Esta posición fue creada a partir de POS1 con el siguiente comando "SETP POS2=POS1; SHIFT POS2 BY 1 -4000". Es decir, es una rotación sobre el eje y de la posición POS1.

Para la creación de las posiciones situadas 10cm por encima de las creadas anteriormente usaremos posiciones relativas a aquellas. También pueden ser creadas sin más que realizar una copia de aquellas y realizar un desplazamiento de +1000 unidades en la coordenada y de cada una de ellas. Estas posiciones serán llamadas POSR1 y POSR2.

Para la definición de estas posiciones usaremos el comando "TEACHR POSR1 POS1", una vez las hayamos creado. Actuaremos de igual manera con la posición POSR2.

TEACH	R Dialog 🛛 🗙	TEACHR	Dialog 🗙	
TEACHR POSR1 POS1		TEACHR POSR2 POS2		
×	0	х	0	
Y	1000	Y	1000	
Z	0	Z	0	
Р	0	Р	0	
R	0	R	0	
ОК			ОК	

✓ crearemos un objeto en POS1

Para ello deberemos mover el robot hasta la posición POS1 con el comando "MOVE POS1", y una vez situado allí obtendremos la cuadrícula correspondiente a la posición POS1 con la terna proporcionada por la pistola de programación.

Ahora debemos acceder a la ventana de creación de objetos y crear un nuevo objeto en la posición dada por la terna (7,1,7).

😹 Lista de Objetos		_ D ×
nuevo0	color	0
	posx	7
	posy	1
	posz	7
	Ok	No
🖗 🗅 🖬 🚘 🗌		

Ya que tenemos creadas las posiciones y el objeto sólo resta escribir el texto del programa.

✓ escribir el texto del programa

El texto del programa es el que se muestra en la figura siguiente. No es necesario incluir los comentarios ya que estos han sido introducidos para separar las distintas partes del pequeño programa.

🗞 JSCORBOT	
Archivo Intérprete Ventanas Ayuda	
home	
speed 50	8 X
label 1	
wait in[1]=1 //esperamos hasta que llegue	
set in[1]=0	S 94
println "objeto en pos1"	
open	@
moved posr1 //nos movemos a pos1	
moveld pos1	
delay 150	0000
close //recogemos el objeto	500 C
delay 150	
moveld posr1	2000
moved posr2 //nos movemos a pos2	0000
moveld pos2	2000
delay 150	
open //depositamos el objeto	
delay 150	
moveld posr2	
goto 1 //volvemos a empezar	-
fuerza el refresco de las ventanas activas	

Hemos supuesto que el usuario que introduce este programa tiene algunos conocimientos del lenguaje ACL, por tanto los comentarios que aquí haremos serán sólo sobre los comandos que pueden llevar a mayor confusión.

Para los comandos de movimiento han sido usadas las versiones de aquellos con espera hasta finalizar la ejecución del movimiento, es decir el comando MOVED en lugar de MOVE. Esto es así debido a que, en el caso de usar MOVE, los comandos posteriores pueden ejecutarse en cualquier lugar de la trayectoria entre la posición inicial y la final.

De igual manera han sido incluidos unos tiempos muertos antes y después de cerrar y abrir la pinza del efector final.

✓ primera ejecución del programa

Para ello podemos optar entre ejecutarlo desde la ventana principal o desde el modo directo mediante el comando "RUN NUEVOO". Desde la ventana sólo tenemos que pulsar sobre el icono . También podemos optar por utilizar el modo debug de ejecución, pulsando sobre el botón ", en el cual se ejecutarán los comandos uno a uno. Para finalizar mostraremos algunas capturas de la pantalla de representación del modelo durante la ejecución de nuestro programa.

