

Práctica 4. Ensamblado virtual.

Objetivos

Se pretende que el alumno reconozca la importancia de los bucles, por medio de instrucciones de control de flujo de programa y de las vistas en la práctica anterior. Esta actividad ha de servir para comprender el funcionamiento de muchas actividades relacionadas con robots y ser capaz de desarrollarlas.

Información previa

Es necesario contar con las siguientes instrucciones relacionadas con el manejo de bucles, se trabajaran las instrucciones MoveAbsJ, While e IF.

- **MoveAbsJ:** Esta instrucción es utilizada para mover el robot hacia una posición de ejes absoluta. En este movimiento los ejes del robot se mueven simultáneos hasta llegar al punto de destino.

MoveAbsJ : p50,v100, z20, tool1

El robot tiene montada la herramienta deseada, y se mueve en una trayectoria de ejes no lineal hasta la zona z20 con la velocidad dada. En nuestro caso se realizará de la siguiente forma.

MoveAbsJ jpos20\NoEOffs, v100, z50, tool0;

En el programa principal ubicado actualmente el robot, se realiza un movimiento con esta instrucción antes y después de ejecutar el PROC programa1().

- **While:** Esta es una instrucción muy conocida en la programación, en este caso permitirá realizar los bucles cuantas veces sea necesario, la instrucción evalúa una condición y mientras esta se mantenga ejecuta el código afectado. Esta instrucción va acompañada de DO al final de la condición.

While reg1 < reg2 DO

.....

Reg1:=reg1+1

Como es de suponer se pueden anidar las instrucciones While de tal manera que se creen bucles dentro de otros bucles, en el caso del ejercicio propuesto se deberá hacer de esa forma.

- **IF :** Esta instrucción como ya sabemos es un condicional, usado cuando es necesario ejecutar instrucciones diferentes en función de si se cumple o no una condición. Debe aparecer, por lo general, el IF con condiciones como THEN, o ELSE y al final el ENDIF. Su estructura es la siguiente:

IF reg1>5 then

Set do1;

Set do2;

```

ELSE
  Reset do1;
  Reset do2;
ENDIF
    
```

Las señales do1 y do2 se activan o desactivan dependiendo del reg1.

Ejercicio 4.

Se supone el caso de una operación de ensamblaje y atornillado de una pieza. La operación se ha de realizar de la siguiente forma, se cogerá la pieza a ensamblar en la cinta de llegada ubicada en el sensor 10, y se ha de llevar a la zona de trabajo sensor 5, posteriormente es necesario fijar los tornillos en los puntos marcados con los sensores 3, 4, 7 y 6, en ese mismo orden, el robot tendrá que ir al sensor 2 a buscar cada uno de los tornillos. Una vez atornillada se toma por el sensor 5 y se deja en la cinta de salida ubicada en nuestro caso en el sensor 8. Veamos el plano de sensores.

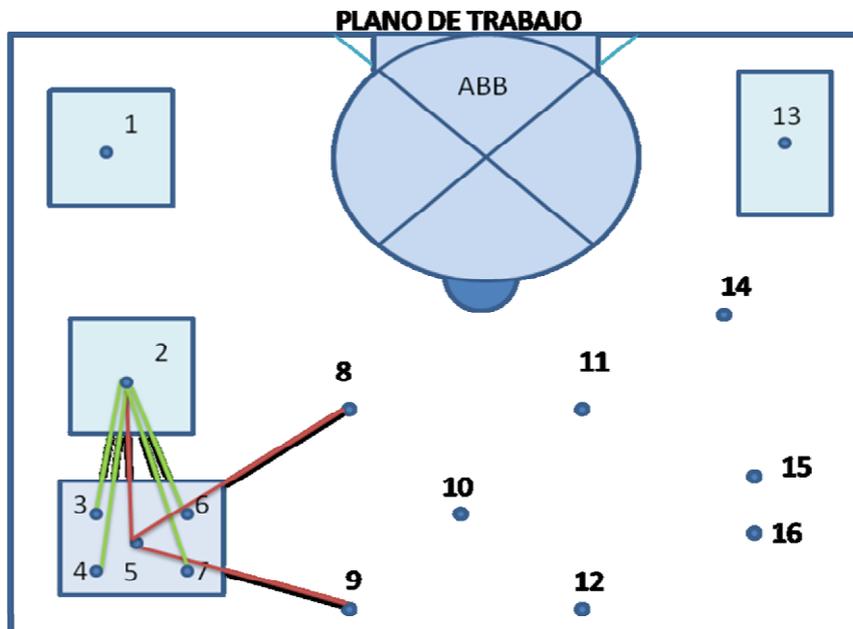


Figura 25. Plano de sensores

Sitio	Puntos	Coordenada X	Coordenada Y	Coordenada Z
Cinta llegada	Punto 9	0	20	20
Banco trabajo	Punto 5	-110	-190	20
Tornillo 1	Punto 4	-80	-250	20
Tornillo 2	Punto 3	-140	-250	20
Tornillo 3	Punto 7	-80	-130	20
Tornillo 4	Punto 6	-140	-130	20
Cinta salida	Punto 8	-250	20	20
Lugar tornillo	Punto 2	-295	-220	20

Tabla de puntos a recorrer

NOTA: Recordar que en esta práctica todos los puntos están referenciados respecto a p30 y que p30 está sobre el sensor 9, a nivel de tierra, por lo tanto, será necesario cada vez que se active un sensor, subir (+) unos 20 cm en vertical para evitar colisiones con otros elementos. De tal forma que los movimientos tendrán la siguiente forma.

```
Movej (p30,xi,yi,20),v100,z30,tool1;
```

Esta instrucción servirá para mover el brazo y evitar colisiones.

```
Movej (p30,xi,yi,0),v100,z30,tool1;
```

De esta forma se activará el sensor.

Para esta práctica es necesario incluir la siguiente línea de código en la primera línea del programa:

```
MODULE modulo1  
!Aquí declaramos todas la variables que necesitamos  
PROC programa1()  
!Vamos al punto de inicio  
MoveAbsJ jpos20\NoEOffs, v100, z50, tool0;  
! A partir de aquí código y al final se cierra programa y modulo ENDPROC  
ENDMODULE
```