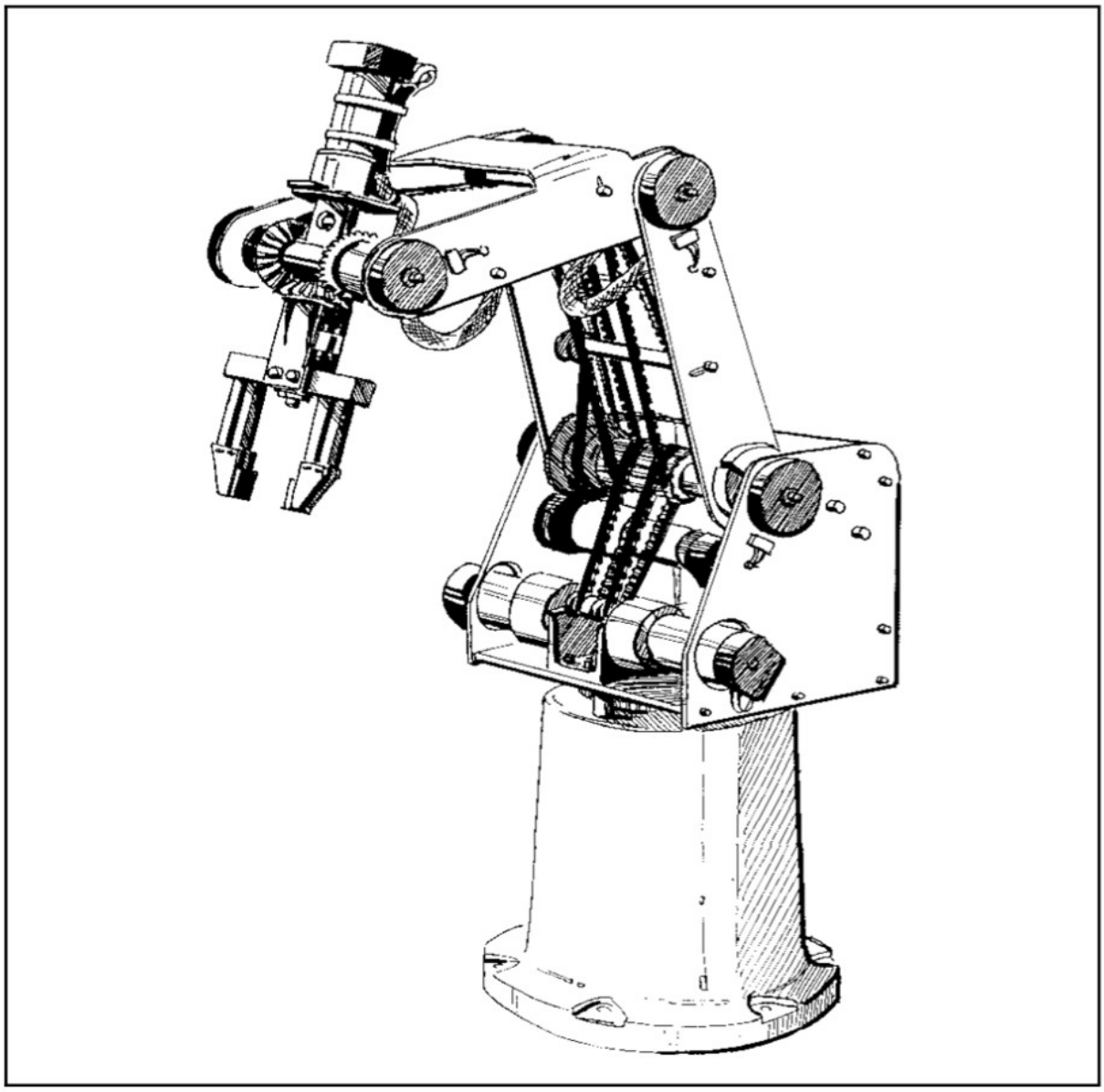
PRÁCTICA ACL:



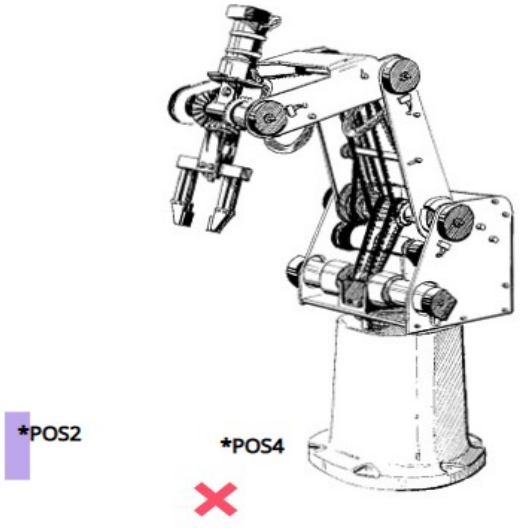
Alumna: Yanierkis Justina Armario Aguirre.
Grado: GITI

ÍNDICE

Ejercicio 1: Programa sencillo. Aproximación a posiciones. Piezas apiladas. Construcción.

1a) Realizar un programa simple en ACL. El robot tendrá que coger una pieza situada en una posición determinada, levantarla y llevarla a otra posición predeterminada.
- Marcar las posiciones inicial y final, y grabarlas.
- Poner la velocidad al 20%.
- Realizar el programa e introducirlo en el SCORBOT con el editor.
- Probar el funcionamiento del programa.

En la siguiente imagen se muestra una localización aproximada de las posiciones:



Primero mostraremos un pseudocódigo para su mejor comprensión:

1) Partimos de la posición del HOME.
2) Nos movemos a la posición donde se encuentra la pieza y la cogemos.
3) Volvemos, con la pieza, a la posición de HOME.
4) Colocamos la pieza en la posición final y la soltamos.
5) El brazo robótico volverá a su posición de inicio (en el HOME).

A continuación, se muestra el código empleado:

Antes de realizar los programas será necesario declarar fuera de este las posiciones (con el comando *DEFP*), además de guardarlas de forma manual (con el comando *HERE*). Para guardarla de modo manual tendremos que seleccionar ctrl+m y usar las coordenadas cartesianas o movernos a través de las articulaciones para colocarnos en la posición que queramos guardar, una vez colocados en ese punto, nos salimos y usamos el comando *HERE*.
También, hay que señalar que no es conveniente usar una velocidad de 100, sino una más lenta, por ejemplo, 20 (nosotros hemos tomado 25). Existe una posición en el mismo lugar donde se queda al hacer el *HOME*, irá a esta al hacer *MOVE 0*.

```
DEFP  POS2                    //Definimos la posición
HERE  POS2          // Una vez la tengamos manualmente, la grabamos
DEFP  POS4                    //Def. Posición
HERE  POS4
```

```
EDIT EJ1                                    //Creamos el programa
*******************************************************************************
SPEED  25                                   //Seleccionamos una velocidad de 25
MOVED  0    //La orden MOVED impide realizar otra acción hasta que se termine esta
OPEN                                        //Abrimos garra
MOVED  POS2                                 //Nos movemos a la pos donde está la pieza
CLOSE                                       //Cerramos garra y cogemos pieza
MOVED  0                                    //Volvemos a la pos de HOME
MOVED  POS4                                 //Nos movemos a la pos donde queremos colocarla
OPEN                                        //Abrimos garra
MOVED  0                                    //Nos movemos a la pos de HOME
CLOSE                                       //Cerramos garra
EXIT                                        //Salimos del programa, lo hemos acabado
*******************************************************************************
```

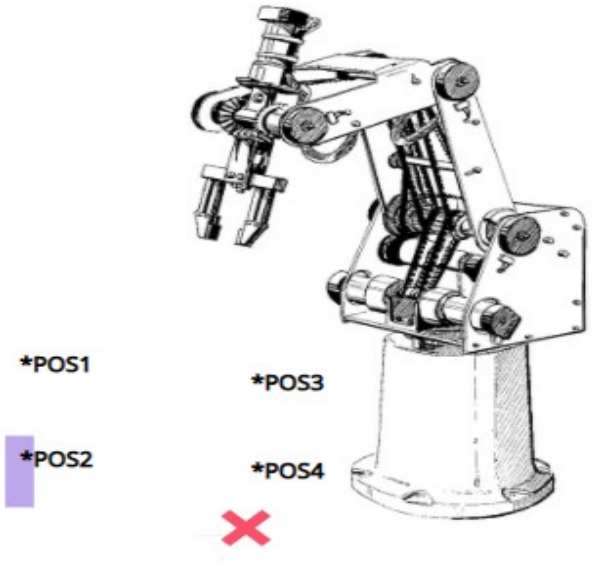
Para ejecutar el programa habría que utilizar el comando *RUN EJ1*.

1b) Al aproximarse la pinza del robot a las piezas que quiere coger o a otros objetos del entorno, existe el riesgo de que se produzca una colisión, debido a las pequeñas imprecisiones en los codificadores y las articulaciones, o que la pinza choque con la pieza. Por ello es conveniente bajar la velocidad cuando estamos cerca de algún objeto, y aproximarnos con movimiento lineal. Este ejercicio consistirá en programar el ejercicio anterior, pero disminuyendo la velocidad en las zonas en las que haya posibilidad de colisión, y aproximándonos con movimiento lineal.

- Grabar posiciones de aproximación situadas unos 10 cm. por encima de las posiciones inicial y final.
- Cambiar el programa anterior de forma que el robot vaya a velocidad normal cuando se desplaza en zona libre, y al 25% de la velocidad normal cuando va desde la posición de aproximación hasta la posición final o viceversa (movimiento lineal).

Aquí nos piden guardar otras posiciones de aproximación para evitar colisiones y disminuir un poco la velocidad a la hora de coger y dejar la pieza (se nos pide el 25% de la velocidad normal, pero nosotros hemos tomado 10).

Vamos a seguir utilizando las posiciones anteriores y definiremos las de aproximación que estarán a 10cm por encima como se muestra en la siguiente imagen:



Primero mostraremos un pseudocódigo para su mejor comprensión:

1) Partimos de la posición del HOME.
2) Nos movemos a la posición donde se encuentra la pieza y la cogemos reduciendo la velocidad en la posición de aproximación.
3) Volvemos, con la pieza, a la posición de HOME reduciendo la velocidad hasta llegar a la posición de aproximación y luego poniéndola a su valor normal hasta que llega al HOME.
4) Colocamos la pieza en la posición final y la soltamos reduciendo la velocidad cuando sea conveniente.
5) El brazo robótico volverá a su posición de inicio (en el HOME) reduciendo la velocidad en el tramo citado anteriormente.

A continuación, se muestra el código empleado:

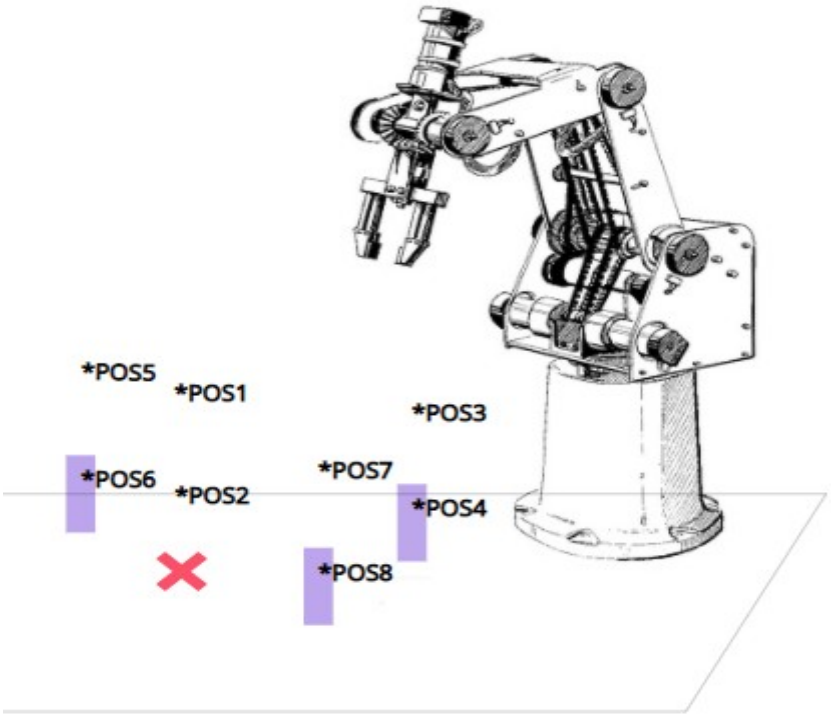
```
DEFP  POS1                    //Definimos la posición de aproximación a POS2
HERE  POS1          //La grabamos una vez la hallamos conseguido de forma manual
DEFP   POS3                   //Definimos la posición de aproximación a POS4
HERE  POS3                    //Grabamos la posición

EDIT EJ2                                  //Creamos el programa
*************************************************************************************************
SPEED  25
MOVED  0
MOVED  POS1                    //Se mueve a la pos de aproximación
OPEN                           //Es aquí donde abre la garra
SPEED 10                       //Reducimos la velocidad
MOVED POS2
CLOSE                          //Cerramos y cogemos la pieza
MOVED POS1                     //Volvemos a la pos de aprox
SPEED  25           //Volvemos a la velocidad habitual fuera de la zona de peligro
MOVED 0
MOVED POS3
SPEED  10
MOVED POS4
OPEN                           //Depositamos la pieza
MOVED POS3
CLOSE
SPEED  25
MOVED 0
EXIT
*************************************************************************************************
```

Para ejecutar el programa habría que utilizar el comando *RUN EJ2*.

1c) Utilizando como base los programas realizados anteriormente, habrá que realizar un programa que coja 3 piezas de posiciones predeterminadas y las apile en otra posición, empleando unos subprogramas que realicen las funciones de acercamiento/alejamiento (usar la orden SHIFT/SHIFTC).

La situación es la mostrada en esta imagen:



A continuación mostramos el pseudocódigo:

(1) Partimos de una posición inicial HOME.
(2) Cogemos la primera pieza y volvemos a la posición HOME.
(3) Colocamos la primera pieza en su posición final, actualizamos la posición de aprox para que ahora esté 10cm por encima y volvemos a la pos HOME.
(4) Cogemos la segunda pieza y realizamos un proceso análogo al paso anterior.
(5) Cogemos la tercera pieza y la colocamos en la pos final, quedando las tres piezas alineadas verticalmente.
(6) Volvemos a la pos del HOME y finalizamos el programa.

En este ejercicio, nos piden el uso de *SHIFTC* para realizar un programa que apile las 3 piezas de una determinada manera. Para ello, nos valdremos de las posiciones auxiliares: POS2A y POS1A ya que la orden *SHIFTC* nos machacaría el valor anterior.
Las 3 piezas estarán inicialmente en las posiciones POS6, POS4 y POS8. Procedemos a definir y guardar las nuevas posiciones y también las de aproximación de estas. El código es el siguiente:

```
DEFP  POS5
HERE  POS5
DEFP  POS6
HERE  POS6
DEFP  POS7
HERE  POS7
DEFP  POS8
HERE  POS8
```

```
DEFP POS1A
DEFP POS2A

EDIT EJ3
*********************************************************************************************
SPEED  25
MOVED  0
MOVED  POS5
OPEN
SPEED  10
MOVED  POS6
CLOSE                                   //Cogemos la primera pieza
MOVED  POS5
SPEED  25
MOVED  0                                 //Para evitar colisiones
MOVED  POS1
SETP  POS1A=POS1                                 //Trabajaremos con esta variables
SPEED  10
MOVED  POS2
SETP  POS2A=POS2
OPEN                                    //Colocamos la primera pieza
MOVED  POS1
CLOSE
SPEED 25
MOVED  0
SHIFTC  POS1A  BY  Z 440//Sobrescribimos el valor para colocarlo  más arriba en el eje z
SHIFTC  POS2A  BY  Z 440
MOVED  POS7
OPEN
SPEED 10
MOVED  POS8
CLOSE                                   //Cogemos la segunda pieza
MOVED POS7
SPEED  25
MOVED  0
MOVED  POS1A
SPEED  10
MOVED  POS2A
OPEN                                    //Colocamos la segunda pieza
MOVED  POS1A
CLOSE
SPEED  25
MOVED  0
SHIFTC  POS1A  BY  Z 440//Sobrescribimos el valor para colocarlo  más arriba en el eje z
SHIFTC  POS2A  BY  Z 440
MOVED  POS3
OPEN
SPEED 10
MOVED  POS4
CLOSE                                   //Cogemos la tercera pieza
MOVED  POS3
SPEED  25
MOVED  0
MOVED  POS1A
SPEED  10
```

```
MOVED  POS2A
OPEN                                    //Colocamos la tercera pieza
MOVED POS1A
SPEED 25
CLOSE
MOVED  0
EXIT
*****************************************************************************************
```

Para ejecutar el programa habría que utilizar el comando *RUN EJ3*.

Consideraciones del programa anterior:

Se podría haber evitado el uso de *CLOSE* en algunos momentos, ya que la garra puede permanecer abierta desde que deposita una pieza en la posición final hasta que va a coger la siguiente, pero para evitar fallos y que se nos olvide abrir o cerrar la garra en algún momento, se ha decidido que se abrirá o cerrará la garra solo en la posición de aproximación. Cuando no se encuentre allí permanecerá cerrada.
También se puede comprobar que se podría haber usado un subprograma para subir las posiciones auxiliares en lugar de realizarlo varias veces en el mismo código. Sería el siguiente:

```
EDIT SUBE
*****************************************************************************************

SHIFTC  POS1A  BY  Z 440//Sobrescribimos el valor para colocarlo  más arriba en el eje z
SHIFTC  POS2A  BY  Z 440
EXIT
*****************************************************************************************
```

En lugar de repetir varias veces estas líneas de código se usaría *RUN SUBE.* El único inconveniente sería que ahora las variables POS1A y POS2A habrían que definirse como variables globales. También habría que tener cuidado conque no se ejecuten más de un programa a la vez, o que el programa principal vaya por delante del subprograma, habría que usar una variable bandera.

1d) Cada grupo ideará una construcción con las piezas que hay disponibles y realizará un programa que coja las piezas, y las vaya colocando hasta terminar la construcción. Después la desmontará dejando cada pieza en su lugar.

Nos piden a continuación realizar un programa que apile una serie de piezas de una determinada manera y luego las vuelva a colocar cada una en su sitio. Debemos de tener en cuenta que las posiciones usadas ya están definidas en los ejercicios anteriores.

El pseudocódigo es similar al del ejercicio anterior solo que esta vez, también desapilará las piezas:

1. Partimos de una posición inicial HOME.
2. Cogemos la primera pieza y volvemos a la posición HOME.
3. Colocamos la primera pieza en su posición final, actualizamos la posición de aprox para que ahora esté 10cm por encima y volvemos a la pos HOME.
4. Cogemos la segunda pieza y realizamos un proceso análogo al paso anterior.
5. Cogemos la tercera pieza y la colocamos en la pos final, quedando las tres piezas alineadas verticalmente, también volvemos a actualizar la posición de aproximación.

6. Volvemos a la pos del HOME.
7. Cogemos la tercera pieza y la colocamos en su posición inicial, esta vez disminuimos en 10cm la pos de aproximación y volvemos a la pos HOME.
8. Hacemos lo mismo con la segunda y primera pieza.
9. Volvemos a la pos HOME y finalizamos el programa.

El código es el siguiente:

```
EDIT EJ4
*********************************************************************************************
SPEED  25
MOVED  0
MOVED  POS5
OPEN
SPEED  10
MOVED  POS6
CLOSE                                   //Cogemos la primera pieza
MOVED  POS5
SPEED  25
MOVED  0                                //Para evitar colisiones
MOVED  POS1
SETP  POS1A=POS1                        //Trabajaremos con esta variables
SPEED  10
MOVED  POS2
SETP  POS2A=POS2
OPEN                                    //Colocamos la primera pieza
MOVED  POS1
CLOSE
SPEED 25
MOVED  0
SHIFTC  POS1A  BY  Z 440//Sobrescribimos el valor para colocarlo  más arriba en el eje z
SHIFTC  POS2A  BY  Z 440
MOVED  POS7
OPEN
SPEED 10
MOVED  POS8
CLOSE                                   //Cogemos la segunda pieza
MOVED POS7
SPEED  25
MOVED  0
MOVED  POS1A
SPEED  10
MOVED  POS2A
OPEN                                    //Colocamos la segunda pieza
MOVED  POS1A
CLOSE
SPEED  25
MOVED  0
SHIFTC  POS1A  BY  Z 440//Sobrescribimos el valor para colocarlo  más arriba en el eje z
SHIFTC  POS2A  BY  Z 440
MOVED  POS3
OPEN
SPEED 10
MOVED  POS4
```

```
CLOSE                                   //Cogemos la tercera pieza
MOVED  POS3
SPEED  25
MOVED  0
MOVED  POS1A
SPEED  10
MOVED  POS2A
OPEN                                    //Colocamos la tercera pieza
MOVED POS1A
SPEED 25
CLOSE
MOVED  0
DELAY  500
SHIFTC  POS1A  BY  Z 440//Sobrescribimos el valor para colocarlo  más arriba en el eje z
SHIFTC  POS2A  BY  Z 440
MOVED  POS1A
OPEN
SPEED 10
MOVED POS2A
CLOSE                                   //Volvemos a coger la 3º pieza
MOVED POS1A
SPEED 25
MOVED  0
SHIFTC  POS1A  BY  Z -440//Sobrescribimos el valor para colocarlo  más abajo en el eje z
SHIFTC  POS2A  BY  Z -440
MOVED  POS3
SPEED  10
MOVED  POS4
OPEN                                    //Volvemos a depositar en su sitio la 3º pieza
MOVED  POS3
SPEED  25
CLOSE
MOVED  0
MOVED  POS1A
OPEN
SPEED 10
MOVED POS2A
CLOSE                                   //Volvemos a coger la 2º pieza
MOVED POS1A
SPEED 25
MOVED  0
SHIFTC  POS1A  BY  Z -440//Sobrescribimos el valor para colocarlo  más abajo en el eje z
SHIFTC  POS2A  BY  Z -440
MOVED  POS7
SPEED  10
MOVED  POS8
OPEN                                    //Volvemos a depositar la 2º pieza en su sitio
MOVED  POS7
SPEED  25
CLOSE
MOVED  0
MOVED  POS1A
OPEN
SPEED 10
MOVED POS2A
```

```
CLOSE                          //Volvemos a coger la 1º pieza
MOVED POS1A
SPEED 25
MOVED  0
MOVED  POS5
SPEED  10
MOVED  POS6
OPEN                       //Volvemos a colocar la pieza 1 en su lugar
MOVED POS5
SPEED  25
CLOSE
MOVED 0

EXIT
*****************************************************************************************************
```

Para ejecutar el programa habría que utilizar el comando *RUN EJ4*.

El anterior fue el realizado durante la práctica. Un programa alternativo más eficiente sería:

```
DEFP P1A               //Definimos globalmente las posiciones auxiliares
HERE P1A               //Inicialmente le damos el mismo valor de POS1
DEFP P2A
HERE  P2A              //Inicialmente le damos el mismo valor de POS2
DEFP   F1                    //Def de variables bandera
DEFP  F2

EDIT SUB
*****************************************************************************************************
SPEED  25
MOVED  P1A
SPEED 10
MOVED  P2A
OPEN
MOVED P1A
SHIFTC  P1A  BY  Z 440//Sobrescribimos el valor para colocarlo  más arriba en el eje z
SHIFTC  P2A  BY  Z 440
SPEED 25
CLOSE
MOVED  0
SET F1=1                   //Ponemos a 1 nuestra variable bandera
EXIT
*****************************************************************************************************


EDIT BAJ
*****************************************************************************************************
SPEED 25
SHIFTC  P1A  BY  Z -440//Sobrescribimos el valor para colocarlo  más abajo en el eje z
```

```
SHIFTC  P2A  BY  Z -440
MOVED  P1A
OPEN
SPEED 10
MOVED P2A
CLOSE
MOVED P1A
SPEED 25
MOVED  0
SET  F2=1                    //Ponemos a 1 nuestra variable bandera
EXIT
*****************************************************************************************************

EDIT EJ5
*****************************************************************************************************
SET  F1=0                    //Inicializamos las variables bandera
SET  F2=0
SPEED  25
MOVED  0
MOVED  POS5
OPEN
SPEED  10
MOVED  POS6
CLOSE                              //Cojo la 1º pieza
MOVED POS5
SPEED 25
MOVED  0
RUN SUB
WAIT F1=1                    //Nos aseguramos de que se ha finalizado el subprograma
MOVED  POS7
SET F1=0                   //Volvemos a darle el valor 0 para cuando la volvamos a utilizar
OPEN
SPEED 10
MOVED  POS8
CLOSE                                //Cojo la 2º pieza
MOVED POS7
SPEED  25
MOVED 0
RUN SUB
WAIT  F1=1                   //Nos aseguramos de que se ha finalizado el subprograma
MOVED  POS3
SET  F1=0                  //Volvemos a darle el valor 0 para cuando la volvamos a utilizar
OPEN
SPEED 10
MOVED  POS4
CLOSE                                //Cojo la 3º pieza
MOVED  POS3
SPEED  25
MOVED 0
RUN SUB
WAIT  F1=1
DELAY  500
RUN BAJ
WAIT  F2=1
MOVED  POS3
```

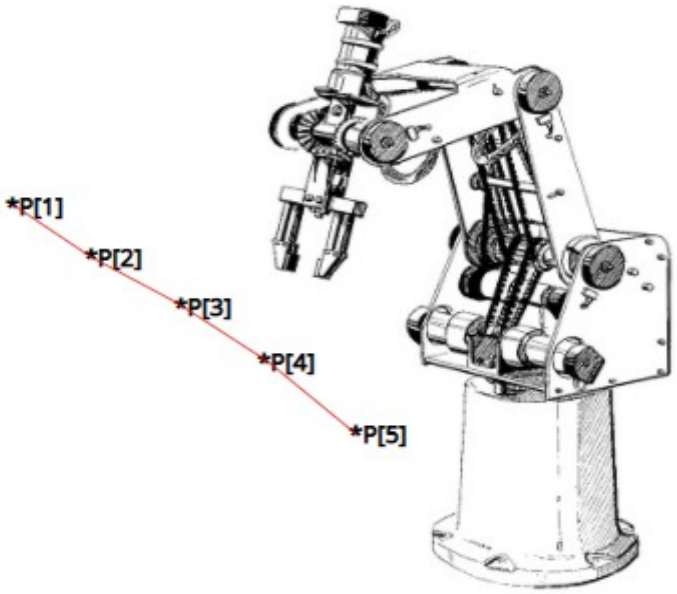
```
SET  F1=0
SET F2=0
SPEED 10
MOVED POS4
OPEN                              //Deposito la 3º pieza en su sitio
MOVED POS3
CLOSE
SPEED 25
MOVED  0
RUN BAJ
WAIT  F2=1
MOVED POS7
SET  F2=0
SPEED  10
MOVED POS8
OPEN                              //Deposito la 2º pieza en su sitio
MOVED  POS7
SPEED 25
CLOSE
MOVED  0
RUN BAJ              //Aquí P1A y P2A vuelven a tener los valores de POS1 y POS2
WAIT  F2=1
MOVED  POS5
SET  F2=0
SPEED  10
MOVED  POS6
OPEN                              //Deposito la 1º pieza en su sitio
MOVED  POS5
SPEED  25
CLOSE
MOVED  0
EXIT
*************************************************************************************************************
```

Para ejecutar el programa habría que utilizar el comando *RUN EJ5*.

Ejercicio 2: Seguimiento de trayectorias.

Definir un vector de posiciones de al menos 5 posiciones que constituya una trayectoria suave (utilizar la orden DIMP) y realizar un programa que permita al robot recorrer la trayectoria tanto en sentido directo como inverso (utilizar el comando MOVESD). La separación entre puntos consecutivos debe ser similar, para evitar cambios bruscos en la velocidad.

Nos piden realizar un programa que siga una trayectoria suave usando la orden *DIMP* y *MOVESD*. Para ello definiremos fuera 5 posiciones. Las posiciones elegidas son las siguientes:



Primero mostraremos el pseudocódigo para su mejor comprensión:

1. Partimos de la posición HOME.
2. Ordenamos que se mueva ordenadamente de la pos 1 a la 5 que hemos guardado previamente.
3. Esperamos 3s.
4. Nos volvemos a mover esta vez de la pos 5 a la 1.
5. Nos movemos a la pos HOME y finalizamos el programa.

El código empleado es el siguiente:

```
DIMP P[5]                          //Definimos las posiciones de la trayectoria
HERE P[1]                          //Hallamos manualmente las posiciones
HERE P[2]
HERE P[3]
HERE P[4]
HERE P[5]

EDIT EJ22
*************************************************************************************************
SPEED 25                            //Le damos esa velocidad por seguridad
MOVED 0                             //Partimos de la pos HOME
MOVESD P  1   5                     //Nos movemos de la P[1] a la P[5]
DELAY  300  //Hacemos una pausa pequeña para que se aprecie mejor el siguiente movto
```

```
MOVESD P  5   1                    //Nos movemos ahora de P[5] a P[1]
MOVED 0                            //Volvemos a la posición HOME
EXIT
**************************************************************************************************
```

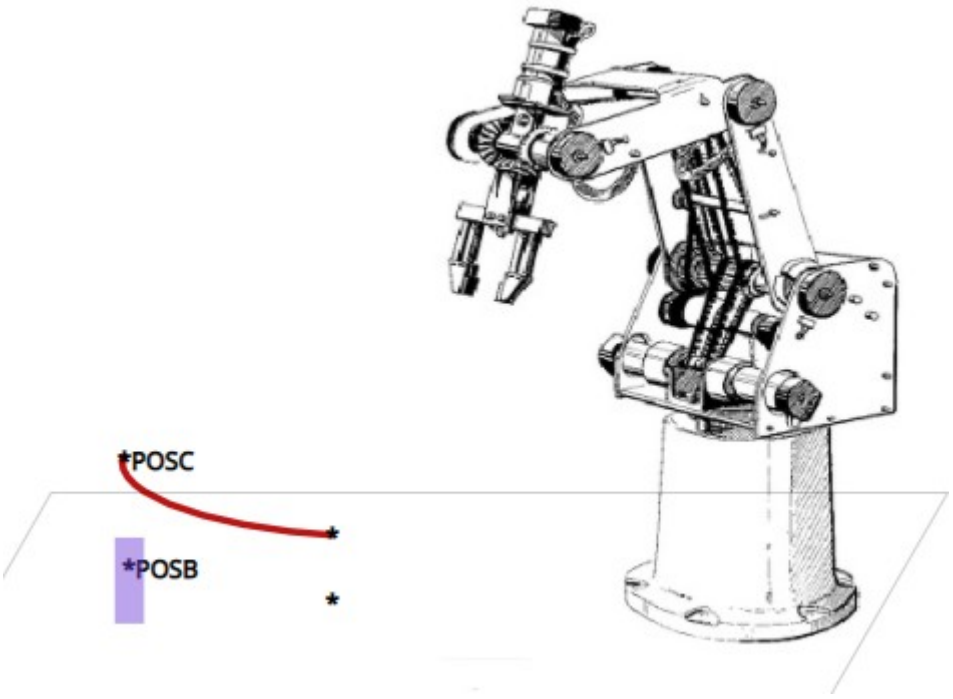
Para ejecutar el programa habría que utilizar el comando *RUN EJ22.**

***Nota**: A la hora de nombrar programas o variables, solo se guardan los 5 primeros caracteres, es decir, si guardamos EJEM10 Y EJEM12 se nos guardará solo EJEM1.

Ejercicio 3: Movimientos relativos.

Familiarizarse con las instrucciones de posiciones relativas HERER y TEACHR, utilizando uno y dos argumentos. Para ello, se realizará un programa que:
• Recoja una pieza de una posición predeterminada. Para ello se deberá acceder a una posición de aproximación, abrir garra, acercarse a la pieza con trayectoria lineal, cerrar la garra y acceder de nuevo a la posición de aproximación linealmente. Dicha posición se definirá 10 cm. por encima de la posición donde se debe recoger la pieza y de forma relativa a ésta.
• Realice repetidamente un movimiento relativo girando la base 10 grados en cada movimiento hasta que alcance un lugar adecuado donde depositar la pieza.
• Suelte la pieza en el lugar elegido siguiendo una rutina similar a la del primer punto.

En este ejercicio nos piden realizar un programa que recoja una pieza de una posición predeterminada, la lleve a una posición 10 cm por encima de esta y luego realice un movimiento relativo girando la base 10º. Nos piden usar las ordenes *TEACHR* y *HERER,* asociadas con las traslaciones y rotaciones de movimientos relativos, respectivamente. La situación inicial es esta:



A continuación mostramos un pseudocódigo para su mayor comprensión:

1. Partimos de una pos HOME.
2. Cogemos la pieza de su pos inicial.
3. Con la ayuda de un contador procedemos a girar la pieza 10º y así vamos actualizando nuestra POSC.
4. Como POSB está asociada a POSC (esta primera siempre estará 10cm por debajo de la primera).
5. Cuando termina de moverse colocamos la pieza en su posición final.
6. Volvemos a la pos HOME y finalizamos el programa.

Primero habrá que definir fuera del programa las posiciones POSB y POSC.

```
DEFP  POSB                          //Definimos la posición inicial de la pieza
DEFP  POSC                          //Def. Pos de aproximación a la pieza
HERE  POSC                    //Le damos valor manualmente, a 10cm de la pieza
DEFP  CC                            //Variable encargada del giro
HERER  CC                    //Indicamos qué articulaciones queremos que giren y cuánto
1—[.]>1000                          //Solo queremos que gire la base
2--[.]>0
3--[.]>0
4--[.]>0
5--[.]>0
TEACHR  POSB  POSC                  //Asociamos el valor de POSB a POSC
X--[.]>0
Y--[.]>0
Z--[.]>-1000                  //POSB siempre va a estar 10cm por debajo de POSC
P--[.]>0
R--[.]>0


EDIT EJ33                           //Creamos nuestro programa
*************************************************************************************************************
SPEED  25                           //Velocidad de seguridad
MOVED  0
MOVED  POSC                   //Nos vamos a la pos de aproximación de la pieza
OPEN
SPEED  10
MOVED  POSB                   //El valor de POSB ya está definido
CLOSE                               //Cogemos pieza
MOVED POSC
SPEED 25
DEFINE  CONT                        //Creamos un contador
SET  CONT=1                         //Lo inicializamos a 1
FOR  CONT=1  TO  10           //Creamos un for que vaya entre 1 y 10
MOVED  CC
HERE  POSC                    //Vamos actualizando POSC
ENDFOR
SPEED 10
MOVED  POSB             //POSB al estar asociada a POSC se ha actualizado también
OPEN                                //Soltamos la pieza en su nueva posición
MOVED  POSC
CLOSE
SPEED  25
```

```
MOVED  0
EXIT
**********************************************************************************************************************
```

Para ejecutar el programa habría que utilizar el comando *RUN EJ33.**

***Nota:** Hay que tener en cuenta varias cosas en este código, se van a sobrescribir los valores iniciales que tuviesen POSB y POSC para evitar esto se podrían usar variables auxiliares como en el ejercicio 1d). También, a la hora de relacionar dos variables, debemos tener en cuenta que el movimiento que queremos va a ser posible.

Ejercicio 4: Manejo de cinta transportadora.

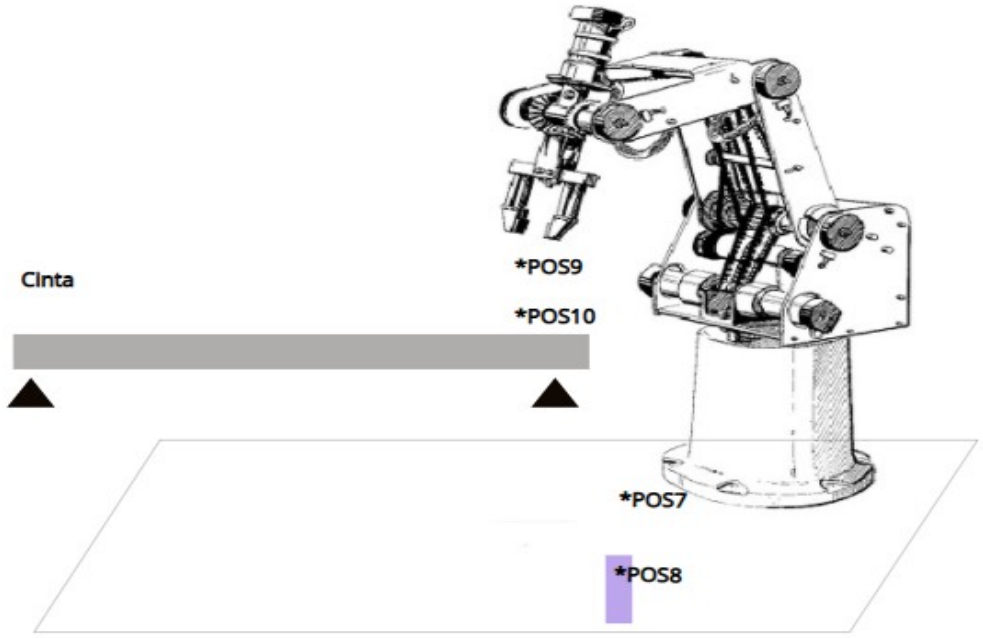
4a) Usar la instrucción SET ANOUT para manejar la cinta transportadora (ver eje en etiqueta en controlador). Se realizará un programa con las siguientes especificaciones:
• El robot cogerá una pieza que estará colocada en una posición A predeterminada.
• Dejará dicha pieza sobre la cinta transportadora (utilizando posiciones de aproximación) y regresará a su posición de aproximación.
• Se accionará la cinta transportadora suavemente para que la pieza no se caiga, y después se aumentará la velocidad.
• Al cabo de un cierto tiempo (antes de que la pieza llegue al final de la cinta), parar la cinta transportadora de forma suave, e invertir el sentido de giro del motor de la cinta.
• Esperar que la pieza regrese a la posición donde se dejó en la cinta, recogerla y llevarla de nuevo a la posición A. Nota: Es probable que se tenga que hacer un ajuste con los tiempos de espera.

A partir de aquí no me dio tiempo a realizar los programas durante la práctica, por lo que pueden presentar algunos errores.

En el ejercicio se nos pide colocar una pieza desde una determinada posición en la mesa a otra de la cinta, una vez depositado en la cinta la garra debe esperar a que llegue al final de la cinta y vuelva a la posición inicial, es entonces cuando el brazo robótico coge la pieza y la vuelve a poner en la mesa. Para ello, nos pide utilizar la instrucción *SET ANOUT*.

También hay que definir y darle valor a las posiciones de aproximación a la cinta y a la de colocación de la cinta. Crearemos 2 programas, uno principal para quitar y poner la pieza, y otro secundario, para mover la cinta. Mientras se realiza el subprograma el programa principal deberá esperar con la orden *WAIT*.

La situación inicial es la siguiente:



Primero, mostraremos un pseudocódigo:

1. Partimos de la posición HOME.
2. Cogemos la pieza de su posición inicial y nos movemos a la pos HOME.
3. Colocamos la pieza en la cinta y esperamos a que nuestra variable bandera se ponga a ‘1’.
   3.a Movemos la cinta hasta el final.
   3.b Esperamos 3s.
   3.c Movemos la cinta a su pos inicial.
   3.d Ponemos la variable bandera a ‘1’.
4. Una vez esté la variable a ‘1’ procedemos a coger la pieza y moverse a la pos HOME.
5. Depositamos la pieza en la mesa, en su pos inicial.
6. Volvemos a la pos HOME y terminamos el programa.

El código es el siguiente:

```
DEFP  POS9                    //Def pos de aproximación a la cinta
HERE  POS9                    //Le damos valores manualmente
DEFP  POS10
HERE  POS10
DEFP  FLAG                    //Definimos una variable bandera

EDIT  EJ44                    //Creamos nuestro programa
*****************************************************************************************************************
SPEED  25
SET FLAG=0                  //Le damos inicialmente valor 0 a nuestra FLAG
MOVED  0
MOVED  POS7
OPEN
SPEED  10
MOVED  POS8
CLOSE                         //Cogemos pieza de la mesa
MOVED  POS7
SPEED  25
MOVED  0
MOVED  POS9
SPEED  10
MOVED  POS10
OPEN                          //Colocamos la pieza en la cinta
MOVED  POS9                   //Nos alejamos a la pos de aproximación para evitar colisiones
RUN  CINTA                    //Ejecutamos nuestro programa
WAIT FLAG=1                 //Esperamos hasta que nuestra variable bandera cambie de valor
MOVED  POS10
CLOSE                         //Cogemos la pieza de la cinta
MOVED  POS9
SPEED  25
MOVED  0
MOVED  POS7
SPEED  10
MOVED  POS8
OPEN                          //Volvemos a colocar la pieza en la mesa
MOVED  POS7
SPEED  25
CLOSE
MOVED  0
EXIT
```

```
*****************************************************************************************************************
```

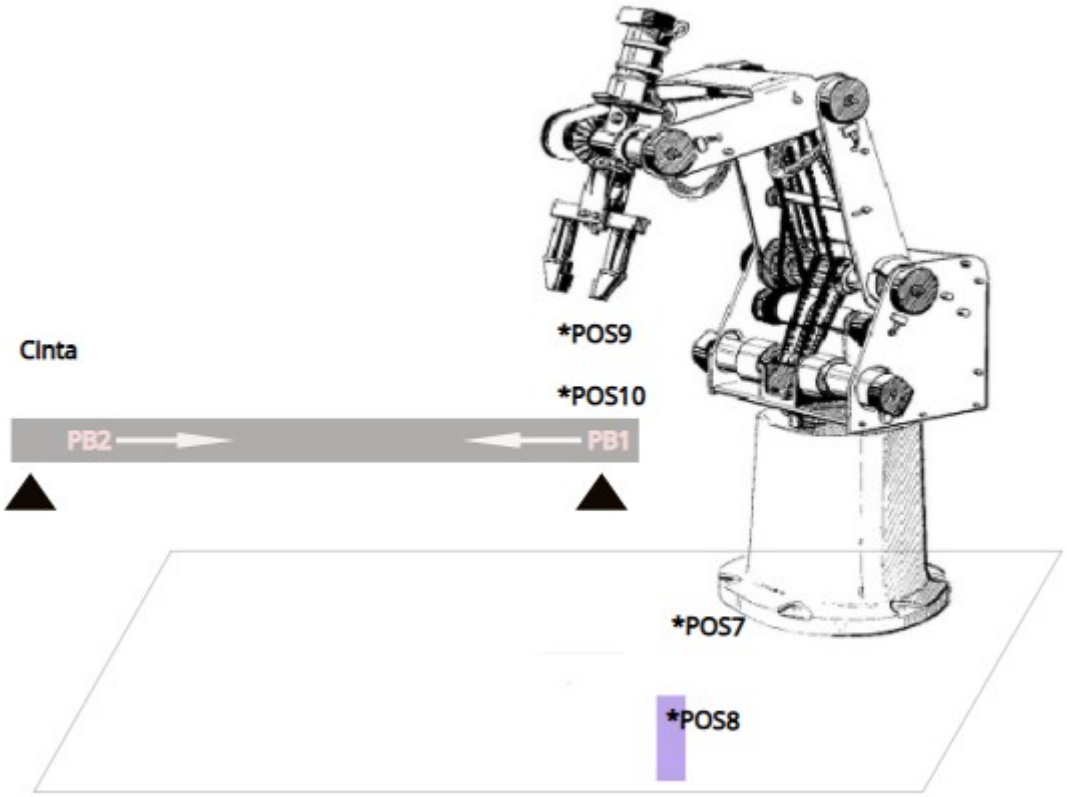
Para ejecutar el programa habría que utilizar el comando *RUN EJ44*.

```
EDIT  CINTA                          //creamos el programa secundario CINTA
*****************************************************************************************************************
DEFINE  CONT                                        //Definimos un contador
DEFINE  F                                  //Definimos la función del movto de la cinta
DEFINE  I                                           //Segundo contador
DEFINE  V                                           //Función decreciente
SET I=0                                           //Inicializamos los valores
SET  CONT=0
SET  FLAG=0
FOR  CONT=0 TO  100                                 //Entramos en un for
SET F= CONT*20                                        //f=20*cont
SET  ANOUT[8]= F                                  //Le damos la posición a la cinta
DELAY 300                                        //Nos esperamos un corto periodo de tiempo
ENDFOR                                              //Salimos del for
FOR CONT=100  TO  0                             //Empezamos el segundo for
SET V= I*20
SET F= 2000- V                                        //f=f(anterior)- i*20
SET ANOUT[8]= F                                 //Le damos la posición a la cinta
SET I= I+1                                      //Vamos actualizando el valor de i
DELAY 5
ENDFOR                                            //Salimos del segundo for
SET  FLAG=1                                       //La bandera ahora vale 1
EXIT
*****************************************************************************************************************
```

Para ejecutar el programa habría que utilizar el comando *CINTA*.

4b) Realizar el mismo utilizando las posiciones del grupo B (DEFPB) para manejar la cinta transportadora (eje 7/8).

Ahora nos piden el mismo ejercicio pero utilizando las posiciones del grupo B (*DEFPB*) para manejar la cinta transportadora.
Si no elegimos ningún grupo DEFP asume que estás en el grupo A, por lo tanto, todas las variables definidas anteriormente se han encontrado en el grupo A.
El comando MOVED también nos permite movernos durante un periodo de tiempo determinado, tiene esta construcción:
MOVED POSX <TIME>
El tiempo esta en centésimas de segundos.
La situación inicial será la misma que en el apartado anterior, salvo que tenemos que definir 2 posiciones más en la cinta.



Nuestro psedocódigo es el siguiente:

1. Partimos de la pos HOME.
2. Cogemos la pieza de su pos inicial y la colocamos en la cinta, el brazo robótico se mueve a la pos de aproximación para evitar colisiones.
3. Movemos la cinta a la izq durante 6s.
4. Nos esperamos 3s.
5. Movemos la cinta hacía la derecha durante 6s.
6. Cogemos la pieza y vamos a la pos HOME.
7. Depositamos la pieza en la mesa, en su pos inicial.
8. Volvemos a pos HOME y finalizamos nuestro programa.

A continuación mostramos nuestro código:

```
DEFPB  PB1                          //Def pos de inicio en la cinta
HERE  PB1
DEFPB  PB2                          //Def pos de fin en la cinta
HERE  PB2

EDIT  EJ4B                          //Creamos programa
*****************************************************************************************************************
SPEED  25
MOVED  0
MOVED  POS7
OPEN
SPEED  10
MOVED  POS8
CLOSE                               //Cogemos pieza
MOVED  POS7
SPEED  25
MOVED  0
MOVED  POS9
SPEED  10
MOVED  POS10
OPEN                                //Depositamos pieza en la cinta
MOVED  POS9
MOVED  PB1  600                     //Movemos la cinta durante 6 segundos
DELAY 300                           //Esperamos 3 segundos
MOVED  PB2  600                     //Movemos hacia la otra dirección la cinta durante 6 segundos
MOVED  POS10
CLOSE                               //Cogemos la pieza de la cinta
MOVED  POS9
SPEED  25
MOVED  0
MOVED  POS7
SPEED  10
MOVED  POS8
OPEN                                //Depositamos la pieza nuevamente en la mesa
MOVED  POS7
SPEED  25
CLOSE
MOVED  0
EXIT
*****************************************************************************************************************
```

Para ejecutar el programa habría que utilizar el comando *RUN EJ4B*.