

Práctica 4.1. Paletizado 3x3 (1)

Proyectos de innovación: CÉLULA ROBOTIZADA

El contenido difundido o publicado compromete exclusivamente a los autores.

INDICE

1. Bloque de ejercicios para familiarizarse con el código de Nivel 4. PALETIZACIÓN 3

| | |
|--|---|
| 1.1.- ENUNCIADO | 3 |
| 1.2.- Conocimientos previos..... | 3 |
| 1.2.1.- Sintaxis de la instrucción PLT | 3 |
| 1.2.2.- Sintaxis de la instrucción DEF PLT..... | 4 |
| 1.3.- Descripción de las entradas, salidas y instrucciones a utilizar..... | 6 |
| 1.3.1.- Vista inicial para la simulación. | 6 |
| 1.3.2.- Vista final para la simulación. | 7 |
| 1.4.- Programa a realizar. | 7 |
| 1.4.1.- Realización de un pallet de una sola altura. Método de ZIG-ZAG | 7 |
| 1.4.2.- Realización de un pallet de una sola altura. Método de MISMA DIRECCIÓN. | 9 |



1. BLOQUE DE EJERCICIOS PARA FAMILIARIZARSE CON EL CÓDIGO DE NIVEL 4.

1.1.- ENUNCIADO

Se desea formar un pallet de 3x3 mediante suministro de piezas que nos llegan de la rampa de aporte a la estación de robot.

La rampa de aporte de piezas llevará a las mismas a la peana donde está situado el sensor denominado PART_AV. Este será el encargado de trasladar a las piezas hasta la posición final donde se conformará el pallet.

Para esta primera práctica realizaremos un pallet de 3x3 de una sola altura, por lo tanto, entran en juego 9 piezas.

1.2.- Conocimientos previos.

En esta práctica ya debemos controlar operaciones como OVDR, SPD de control de velocidad del equipo; operaciones de MOV, MVS, etc., como movimientos del brazo robot; bucles de control con las instrucciones WHILE / WEND; y definición de variables como serán INTE, POS, IO, y la que nos ocupa en esta práctica como es PLT.

1.2.1.- Sintaxis de la instrucción PLT.

PLT <Pallet No.> <Expresión numérica de la expresión>

Parámetros:

<Pallet No.> Primero debemos tener definido el pallet entre 1 y 8,
mediante DEF PLT

<Expresión numérica de la expresión> Número de la red del pallet seleccionado.

Explicaciones:

- Se debe definir mediante DEF PLT el tipo de pallet y la forma de realizarlo.
- El número pallet va de 1 a 8, y hasta 8 se puede definir de una vez.
- La posición de la cuadrícula del pallet dependerá de la conformación del pallet ya definido.

- Tendremos un error durante la ejecución si se ha definido un valor superior al que tenemos definido en el pallet.
- Cuando se utiliza el punto de la cuadrícula como la paleta de posición de destino de la orden de movimiento, se producirá un error si el punto no se encuentra definido.

1.2.2.- Sintaxis de la instrucción DEF PLT.

DEF PLT <Pallet N°.>, <Start point>, <End point A>, <End point B>, [<Diagonal point>], <Quantity A>, <Quantity B>, <Assignment direction>

Parámetros:

| | |
|------------------------|--|
| <Pallet No.> | Definición del número del pallet. Debe ser entre 1 y 8. |
| <Start Point> | Punto inicial del pallet. |
| <End Point A> | Uno de los puntos definidos como final. Punto de tránsito de arco para pallets de arco. Puede ser descrito como posición P#. |
| <End Point B> | Otro de los puntos definidos como final. Punto de tránsito de arco para pallets de arco. Puede ser descrito como posición P#. |
| <Diagonal Point> | Punto diagonal referido a la posición inicial. No es utilizado para el pallet de arco. Puede ser descrito como posición P#. |
| <Quantity A> | Número de piezas desde la posición inicial hasta END POINT A. Es también el nº de piezas desde inicio al final en el pallet de arco. Se puede definir mediante expresiones numéricas. |
| <Quantity B> | Número de piezas desde la posición inicial hasta END POINT B. No es utilizado para un pallet de arco. (lo designamos por 0, etc). Se puede definir mediante expresiones numéricas. |
| <Assignment Direction> | Describimos con 1, 2, ó 3 la forma de rellenar la rejilla del pallet, cuando este está numéricamente dividido. Se puede definir mediante expresiones numéricas. (Explicaremos la dirección de esta asignación) |

Explicaciones:

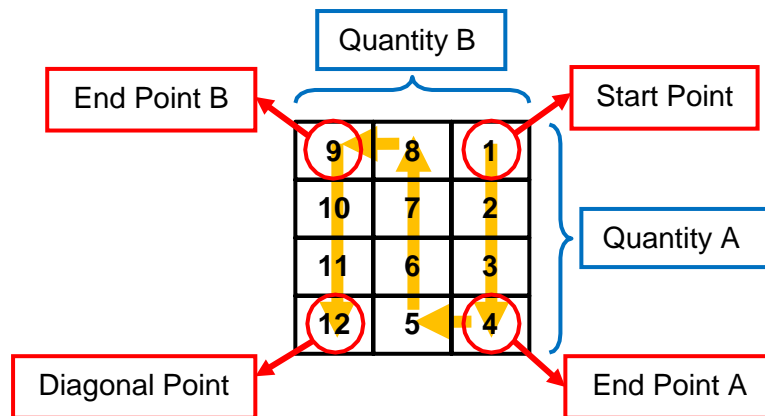
- Un pallet podemos seleccionarlo mediante 3 ó 4 puntos. La definición de 3 puntos es un caso simplificado, mientras que una definición por medio de 4 puntos es más eficaz.
- La dirección de asignación (<Assignment Direction>) es 1 = ZigZag; 2 = Misma dirección; 3 = Arco.
- Comando solo válido dentro del programa que se ejecuta. Hay que redefinir so se ejecuta desde otro programa.
- Cantidades A y B deben ser positivos y distintos de 0. Tendremos un aviso de error en el caso de asignar un valor negativo ó 0.
- El producto de cantidades A x B no debe exceder de 32767. En caso de exceder tendremos un error en el inicio de la operación.

- Para un pallet de arco el valor de la Cantidad B es indiferente, pero no lo podemos omitir. El punto en diagonal será indiferente, incluso si ha sido definido.

Ejemplos:

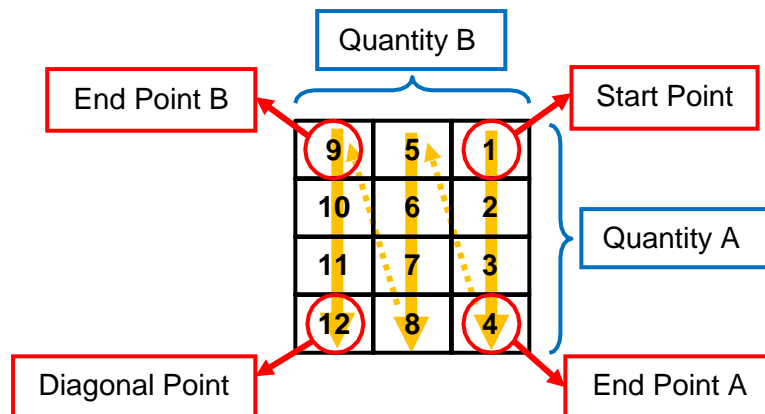
Definición mediante 3 puntos un pallet de 3x3. Llenado mediante ZigZag

DEF PLT 1, P1,P2,P3, ,3,3,1



Definición mediante 4 puntos un pallet de 4x3. Llenado mediante misma dirección.

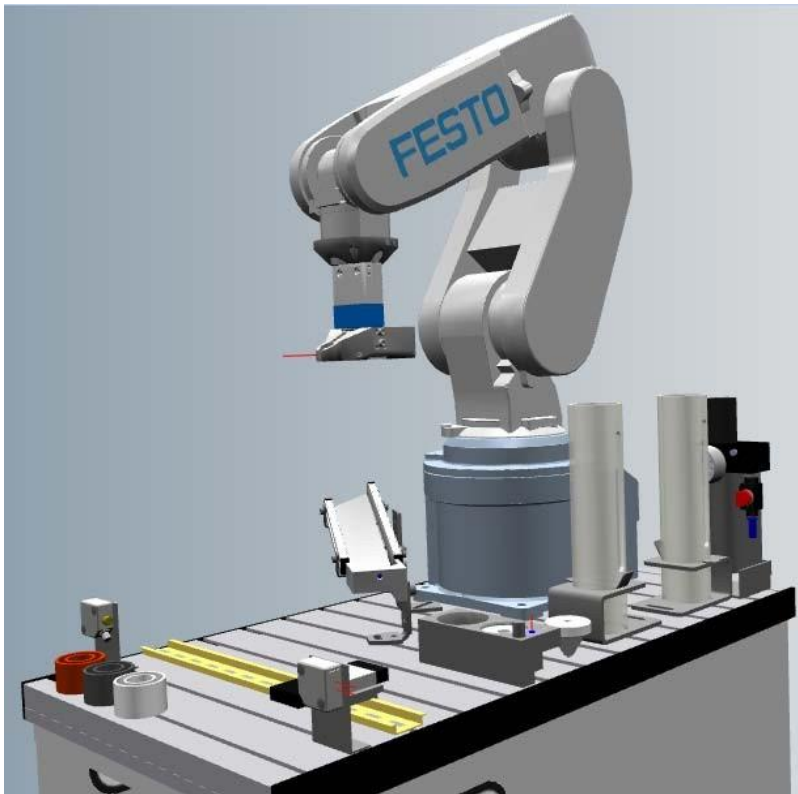
DEF PLT 2, P1,P2,P3,P4,4,3,2



1.3.- Descripción de las entradas, salidas y instrucciones a utilizar.

| Nombre | Descripción | Dirección | Función |
|---------|------------------------|-----------|---------|
| PART_AV | Detector pieza entrada | Bit 6 | Entrada |

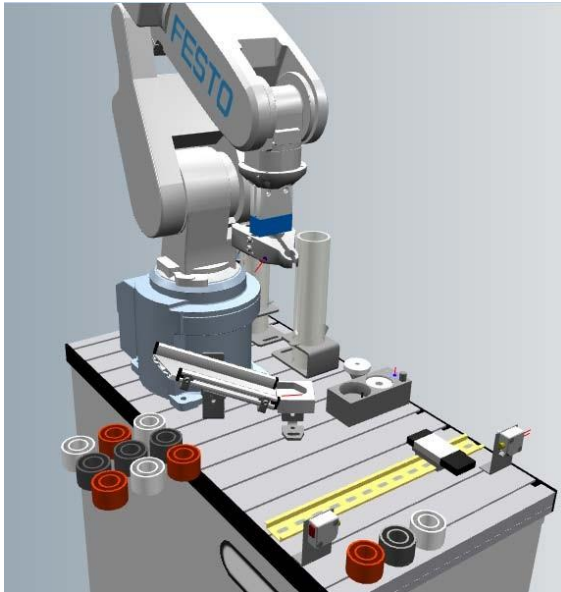
1.3.1.- Vista inicial para la simulación.



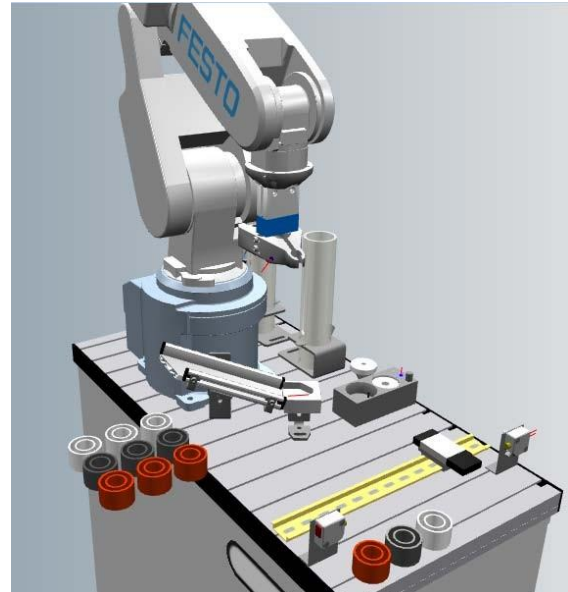
Partimos de la posición de reposo y asegurándonos que la pinza esté abierta. Será cuando las piezas sean detectadas al final de la rampa de entrada, en el denominado PART_AV, cuando se iniciará el movimiento del robot para llevar las piezas al paletizado.

1.3.2.- Vista final para la simulación.

Modo de paletizado ZIG-ZAG



Modo de paletizado en MISMA DIRECCIÓN



1.4.- Programa a realizar.

Son dos formas diferentes de paletizar y la única diferencia está en el último carácter de la instrucción. Tan y como mostramos a continuación.

```
DEF PLT 1, P100,P101,P102,P103,3,3,1
```

```
DEF PLT 1, P100,P101,P102,P103,3,3,2
```

1.4.1.- Realización de un pallet de una sola altura. Método de ZIG-ZAG.

```
'-----'
'Definimos variables de posicionamiento para los puntos del palet'
'como son las cuatro esquinas del mismo '
'-----'
DEF POS P101
DEF POS P102
DEF POS P103
'-----'
'Definimos una variable de posición para una distancia '
'de seguridad, desde la cual realizaremos las operaciones MVS '
'-----'
DEF POS VecZ25
'-----'
'Definimos variables enteras para poder realizar el bucle '
'while / wend para el total de las piezas a paletizar. '
'-----'
DEF INTE M
'-----'
'Definimos la variable para usarla en el bucle para while/wend '
'asignando el palet y el contador de piezas '
```

```
'-----'  
DEF POS P70
```

```
'-----'  
'Asignamos valores a cada una de las posiciones de las esquinas '  
'del palet y a la distancia de seguridad '  
'-----'
```

```
P101 = P100 + (100,0,0,0,0,0)  
P102 = P100 + (0,100,0,0,0,0)  
P103 = P100 + (100,-100,0,0,0,0)  
VecZ25 = (0,0,+25,0,0,0)
```

```
'-----'  
'Definimos la entrada de control del programa como detector de '  
'pieza al final de rampa. Dado por el bit 6 '  
'-----'
```

```
DEF IO PART_AV = BIT,6
```

```
'-----'  
'Definimos el palet. '  
'-----'
```

```
DEF PLT 1, P100,P101,P102,P103,3,3,1
```

```
MOV p99 'Posición de inicio.
```

```
HOPEN 1 'Nos aseguramos que esté la pinza abierta.
```

```
M=1 'Asignamos el primer valor del palet de 3x3. Número 1
```

```
WHILE (M<=9) 'Inicio de bucle hasta que lleguemos a los 9 del 3x3
```

```
  *L0 'Permanecemos en este lazo hasta que se tenga pieza al
```

```
  IF PART_AV=0 THEN *L0
```

```
    HOPEN 1 'Nos aseguramos que esté la pinza abierta.
```

```
    MOV P1 + VecZ25
```

```
    MVS P1
```

```
    HClose 1 'Nos aseguramos que esté la pinza abierta.
```

```
    Dly 0.5
```

```
    MVS P1 + VecZ25 'Desplazamos hasta P1 + distancia de seguridad
```

```
    P70 = (PLT 1,M) 'Asignamos a una variable que controle al palet
```

```
    MOV P70 + VecZ25 'y número de piezas. Controlamos la distancia de seguridad
```

```
    MVS P70 'y nos movemos linealmente hasta posición que indique
```

```
    'según el control del palet y número de pieza a colocar.
```

```
    HOPEN 1
```

```
    Dly 0.5
```

```
    MVS P70 + VecZ25 'Volvemos a posición segura después de dejar pieza
```

```
    M = M + 1 'Contamos una pieza más y mientras sea inferior a 9
```

```
    MOV P99 'volvemos a la posición de inicio y repetimos proceso.
```

```
WEND
```

```
End
```


1.4.2.- Realización de un pallet de una sola altura. Método de MISMA DIRECCIÓN.

```

'-----'
'Definimos variables de posicionamiento para los puntos del palet '
'como son las cuatro esquinas del mismo '
'-----'
DEF POS P101
DEF POS P102
DEF POS P103
'-----'
'Definimos una variable de posición para una distancia '
'de seguridad, desde la cual realizaremos las operaciones MVS '
'-----'
DEF POS VecZ25
'-----'
'Definimos variables enteras para poder realizar el bucle '
'while / wend para el total de las piezas a paletizar. '
'-----'
DEF INTE M
'-----'
'Definimos la variable para usarla en el bucle para while/wend '
'asignando el palet y el contador de piezas '
'-----'
DEF POS P70
'-----'
'Asignamos valores a cada una de las posiciones de las esquinas '
'del palet y a la distancia de seguridad '
'-----'
P101 = P100 + (100,0,0,0,0,0)
P102 = P100 + (0,100,0,0,0,0)
P103 = P100 + (100,-100,0,0,0,0)
VecZ25 = (0,0,+25,0,0,0)
'-----'
'Definimos la entrada de control del programa como detector de '
'pieza al final de rampa. Dado por el bit 6 '
'-----'
DEF IO PART_AV = BIT,6
'-----'
'Definimos el palet. '
'-----'
DEF PLT 1, P100,P101,P102,P103,3,3,2
MOV p99 'Posición de inicio.
HOPEN 1 'Nos aseguramos que esté la pinza abierta.
M=1 'Asignamos el primer valor del palet de 3x3. Número 1
WHILE (M<=9) 'Inicio de bucle hasta que lleguemos a los 9 del 3x3
  *L0 'Permanecemos en este lazo hasta que se tenga pieza al
  IF PART_AV=0 THEN *L0
    HOPEN 1 'Nos aseguramos que esté la pinza abierta.
    MOV P1 + VecZ25
    MVS P1
    HClose 1 'Nos aseguramos que esté la pinza abierta.
    Dly 0.5
    MVS P1 + VecZ25 'Desplazamos hasta P1 + distancia de seguridad
    P70 = (PLT 1,M) 'Asignamos a una variable que controle al palet
    MOV P70 + VecZ25 'y número de piezas. Controlamos la distancia de seguridad
    MVS P70 'y nos movemos linealmente hasta posición que indique
    'según el control del palet y número de pieza a colocar.
    HOPEN 1
    Dly 0.5
    MVS P70 + VecZ25 'Volvemos a posición segura después de dejar pieza

```

```
M = M + 1 'Contamos una pieza más y mientras sea inferior a 9  
MOV P99 'volvemos a la posición de inicio y repetimos proceso.
```

WEND

End