

TITULO	
TIPO	Guía Práctica EPSON RC + 7.0 - Básica
EQUIPO	VT6L
FECHA ELABORACIÓN	05-03-2024
ING ENCARGADO CDM	Ing. Diego Castro

#### Recursos:

- Software EPSON RC + 7.0 – Versión 7.5.2
- Robot VT6L

**Objetivo General:** lograr consolidar en el estudiante, los conocimientos básicos para la puesta en marcha de un Robot VT6L de EPSON, por medio del software EPSON RC + 7.0.

#### Objetivos Específicos:

- Consolidar como realizar la instalación física y eléctrica del robot, y los requisitos para energizarlo.
- Conexión al software EPSON RC + 7.0.
- Afianzar manejo del Robot Manager.
- Consolidar el uso de los comandos Go, Move, Pallet y Pallet Outside, en el simulador y con el Robot.

Tomar como referencia para la realización de esta practica la presentación :

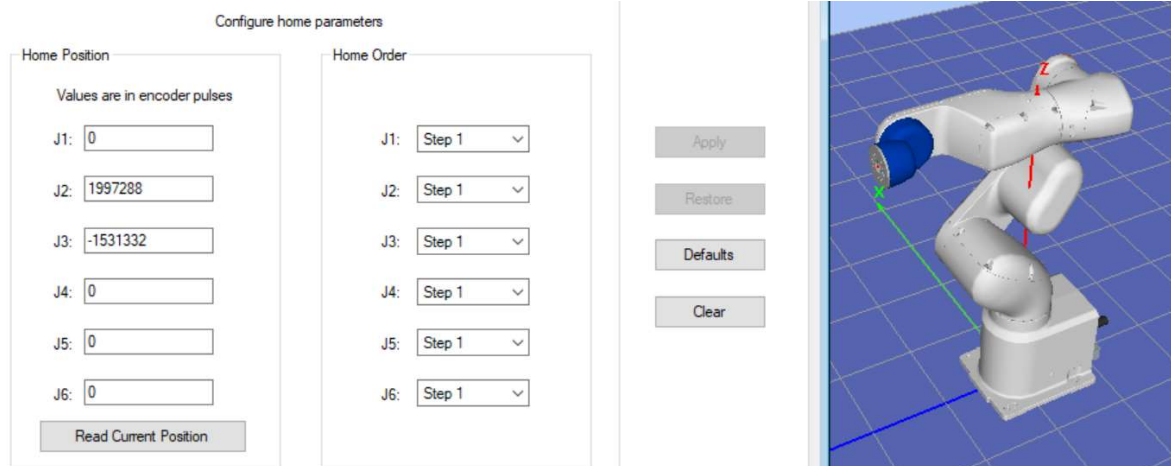
#### Capacitación básica EPSON RC + 7.0 SCARA SERIE T 131023

#### Practica Simulador:

1. Abrir EPSON RC + 7.0, crear una conexión virtual y configurar un robot VT6L en la **Configuración del Sistema** (Presentación: Pág. 62-74).



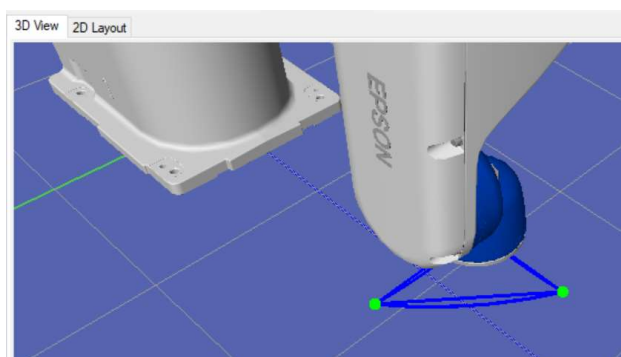
- Abra un proyecto nuevo y luego con el Robot Manager configure una posición de home como en la siguiente imagen:



- Crear tres puntos por medio de las herramientas de **JOG&TEACH**, con las siguientes coordenadas o aproximadas (*Presentación: Pág. 87*).

Number	Label	X	Y	Z	U	V	W	Local
0	Origen	0.000	460.263	122.920	90.000	0.000	180.000	0
1	EjeX	120.100	460.263	122.920	90.000	0.000	180.000	0
2	EjeY	0.000	565.263	122.920	90.000	0.000	180.000	0

- Realizar un programa en SPEL+ con la estructura básica mencionada en la presentación, que haga navegar al robot entre los tres puntos creados, con un retardo entre movimientos de 500 ms, por medio del comando Go, seguidamente con Move, dentro de un bucle infinito. Luego con Go agregue la modificación de la altura de Z en 200mm, antes y después de la ejecución de cada punto.



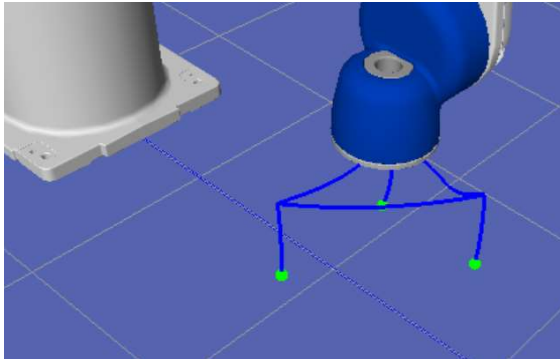
```

Do
  Go Origen
  Wait 0.5
  Go EjeX
  Wait 0.5
  Go EjeY
  Wait 0.5

  Move Origen
  Wait 0.5
  Move EjeX
  Wait 0.5
  Move EjeY
  Wait 0.5

```

Loop



```
Do
  Go Origen :Z(200)
  Go Origen
  Go Origen :Z(200)
  Wait 0.5
  Go EjeX :Z(200)
  Go EjeX
  Go EjeX :Z(200)
  Wait 0.5
  Go EjeY :Z(200)
  Go EjeY
  Go EjeY :Z(200)
  Wait 0.5
```

Loop

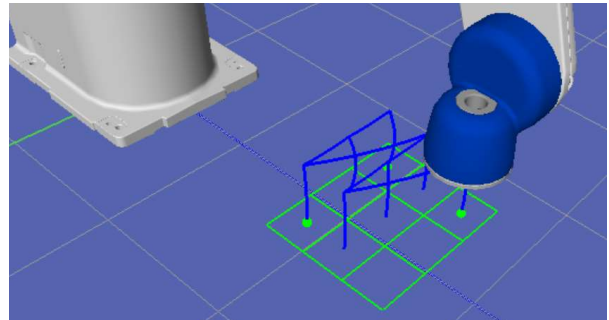
5. Genere una función llamada **“paletizado\_z”**, configure un Pallet con el comando del mismo nombre, de 3 x 3 posiciones, y luego de terminar de ejecutar los movimientos del punto anterior, llame esta función de **“paletizado\_z”** y ejecute un bucle FOR que navegue por las nueve posiciones con el comando Go en orden ascendente (*Presentación: Pág. 120-121*).

```
Function paletizado_z

  #define estado_paletizado_z 11

  Pallet 1, Origen, EjeX, EjeY, 3, 3

  On estado_paletizado_z
  For i = 1 To 9
    Go Pallet(1, i) :Z(200)
    Go Pallet(1, i)
    Go Pallet(1, i) :Z(200)
  Next
  Off estado_paletizado_z
Fend
```



6. Programe una segunda función llamada “**paletizado\_s**”, en el que recorra las nueve posiciones generando un Track Render en forma de “S”, es decir, recorrer las posiciones de la siguiente manera: 1,2,3,6,5,4,7,8,9. Esta función será ejecutada después de “**paletizado\_z**”.

```
Function paletizado_s

    #define estado_paletizado_s 12

    Pallet 1, Origen, EjeX, EjeY, 3, 3

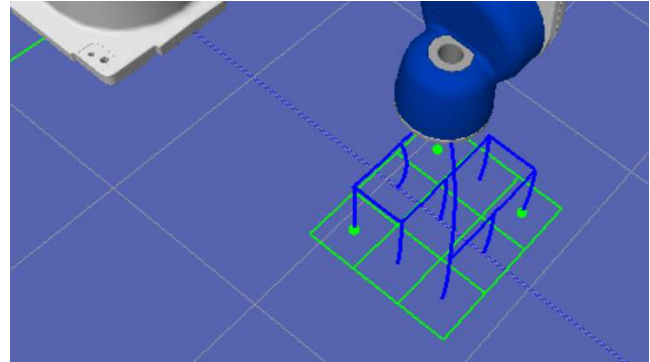
    On estado_paletizado_s

    For i = 1 To 3
        Go Pallet(1, i) :Z(200)
        Go Pallet(1, i)
        Go Pallet(1, i) :Z(200)
    Next

    For i = 6 To 4 Step -1
        Go Pallet(1, i) :Z(200)
        Go Pallet(1, i)
        Go Pallet(1, i) :Z(200)
    Next

    For i = 7 To 9
        Go Pallet(1, i) :Z(200)
        Go Pallet(1, i)
        Go Pallet(1, i) :Z(200)
    Next

    Off estado_paletizado_s
Fend
```



7. Codifique un condicional que ejecute el “**paletizado\_z**” solo si la entrada digital número 9 esta en ON, y otro que ejecute “**paletizado\_s**” solo si la entrada digital número 10 está en ON. Además programe dos salidas digitales, 11 y 12, para cuando se esté ejecutando “**paletizado\_z**” y “**paletizado\_s**” respectivamente.

```
If Sw(9) Then
    Call paletizado_z
EndIf
```

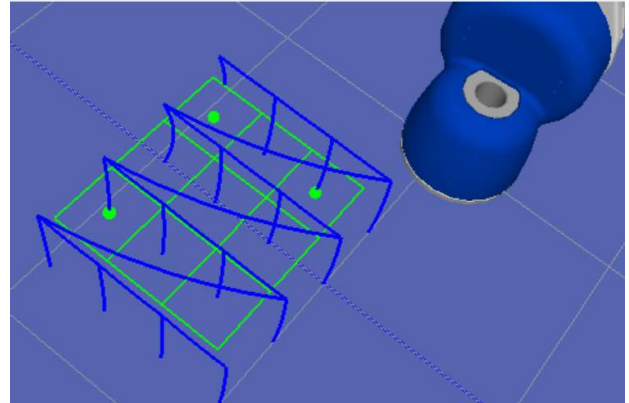
8. Genere una tercera función llamada “**paletizado\_externo**”, en el cual se utilizará el comando **Pallet Outside**, para configurar con los mismos 3 puntos creados al inicio, un pallet que agregue una cuarta fila y columna que vaya más allá del espacio circunscrito entre los puntos. Genere una rutina dentro de esta función que permita que el robot navegue por los 16 puntos del pallet. Esta función se ejecutará solo si la entrada digital 11 está activada.

```
Function paletizado_externo

    #define estado_paletizado_externo 12

    Pallet Outside, 2, Origen, EjeX, EjeY, 3, 3

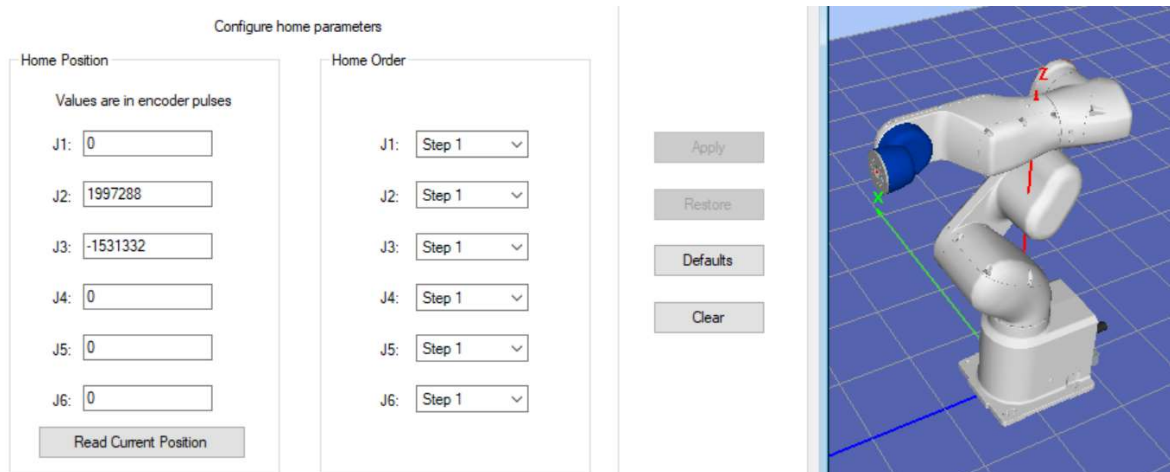
    On estado_paletizado_externo
    For i = 1 To 4
        For j = 1 To 4
            Go Pallet(2, i, j) :Z(200)
            Go Pallet(2, i, j)
            Go Pallet(2, i, j) :Z(200)
        Next
    Next
    Off estado_paletizado_externo
Fend
```





### Practica Robot VT6L:

9. Verifique la instalación mecánica y eléctrica de ROBOT VT6L, verifique que el pulsador de emergencia este activado y energice el robot. (*Presentación: Pág. 36-38*).
10. Abra el software EPSON RC + 7.0, cierre cualquier proyecto abierto, y conéctese por medio del puerto USB al robot. (*Presentación: Pág. 126*).
11. Abra el proyecto que creo en la práctica con el simulador, luego con el Robot Manager configure una posición de home como en la siguiente imagen:



12. En el código que desarrollo anteriormente, comente las líneas concernientes a velocidad, aceleración y desaceleración, y repítalas, pero limitadas a un 30% para Accel y Speed, y 100 para Accel y Speed. Ejecute **"Run Windows"**, y active el checkbox de **"Low Power"**, a continuación ejecute el programa.

