

Memoria Práctica 5 Robot Studio

Realizada por:

Jesús Carrascosa Carro, Ing. Computadores, Robótica y Automatización.

Versión Básica

1. Se ha creado y ubicado las torres(6 y 4 piezas) guardando el estado original en un estado llamado prueba(idem en la versión avanzada).

Estructura del código.

Se ha desarrollado el código rapid siguiendo la estructura a continuación descrita:

1. Procedimiento Main que contiene la llamada a los metodos **trasladatorre** para ambas torres , junto con el set a 0 de la ventosa al inicio y al final.
2. **TrasladaTorre**, Se encarga de hacer el traslado de la posición origen a la intermedia y de la posición intermedia a la origen.También averigua las posiciones de las torres intermedias.
3. **TrasladaTorreAux**, Se encarga de mover una torre desde un punto hacia otro punto, Esto lo hace llamando iterativamente a la función **muevepieza** para cada una de las piezas que tiene la torre.
4. **muevePieza**, Mueve una pieza de un punto a otro, pasando por puntos de aproximación.
El vídeo demostrativo se encuentra en la carpeta versionBasica/RobotStudio 5-21_4.mp4 junto con el código utilizado (codigo.txt)

```
MODULE Module1
CONST robtarget pBaseTorre1:=[1500,500,750],[0,0,1,0],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09]
CONST robtarget pBaseTorre2:=[1500,-500,750],[0,0,1,0],[-1,0,-1,0],[9E+09,9E+09,9E+09,9E+09]
CONST robtarget pReposo:=[1808.910161514,0,1855],[0.5,0,0.866025404,0],[0,0,0,0]
PROC main()
VAR robtarget pbt:= pBaseTorre1;
Inicializa;
trasladaTorre pbt,100,100,4;
WaitTime 3;
trasladaTorre pBaseTorre2,100,100,6;
Finaliza;
ENDPROC
PROC Inicializa()
SetDO SD_ActivaVentosa, 0;
ENDPROC
PROC Finaliza()
SetDO SD_ActivaVentosa, 0;
ENDPROC
PROC trasladaTorre(robtarget pBaseTorre,num ap, num dp,num np)
VAR robtarget pIntermedia;
VAR robtarget pDestino;
pIntermedia:= pBaseTorre;
pDestino:= pBaseTorre;
pIntermedia.trans.y:= pBaseTorre.trans.y - 3*dp;
pDestino.trans.y:= pBaseTorre.trans.y + 3*dp;
```

```

        trasladaTorreAux pBaseTorre, pIntermedia,ap, dp,np;
        WaitTime 3;
        trasladaTorreAux pIntermedia,pDestino,ap, dp,np;
    ENDPROC
    PROC trasladaTorreAux(robtarget pBaseOrigen,robTarget pBaseDestino, num ap,
        VAR num i;
        VAR robtarget pBOri;
        VAR robtarget PBDesti;
        FOR i FROM np TO 1 STEP -1 DO
            pBOri:= pBaseOrigen;
            !toco por la superficie superior del cilindro
            pBOri.trans.z:= pBaseOrigen.trans.z + (i)*ap;!es i o i-1?
            pBDesti:= pBaseDestino;
            !toco la superficie de la mesa
            pBDesti.trans.z:= pBaseDestino.trans.z + (np-i+1)*ap;
            muevePieza pBOri,pBDesti;
        ENDFOR
    ENDPROC
    PROC muevePieza(robtarget pOrigen,robTarget pDestino)
        VAR robtarget pOrigenAprox;
        VAR robtarget pDestinoAprox;
        VAR num alturaAprox:= 150;
        pOrigenAprox := pOrigen;
        pDestinoAprox := pDestino;
        pOrigenAprox.trans.z := pOrigen.trans.z + alturaAprox;
        pDestinoAprox.trans.z := pDestino.trans.z + alturaAprox;
        MoveJ pOrigenAprox,v2000,z100,Ventosa\WObj:=wobj0;
        MoveL pOrigen,v300,fine,Ventosa\WObj:=wobj0;
        SetDO SD_ActivaVentosa, 1;! se puede combinar con mode LDO
        WaitDI ED_PiezaCogida, 1;
        MoveL pOrigenAprox,v300,z0,Ventosa\WObj:=wobj0;
        MoveJ pDestinoAprox,v2000,z100,Ventosa\WObj:=wobj0;
        MoveL pDestino,v300,fine,Ventosa\WObj:=wobj0;
        SetDO SD_ActivaVentosa, 0;
        WaitDI ED_PiezaCogida, 0;
        MoveL pDestinoAprox,v300,fine,Ventosa\WObj:=wobj0;
    ENDPROC
ENDMODULE

```

Versión Avanzada

Se ha realizado para torres de 16,10 y 6 piezas con radios 400, 300 y 500 respectivamente. El estado inicial de la última torre(6,500), se encuentra guardado como prueba.

Mi solución propuesta combina técnicas vistas en otras prácticas como el empleo de un array de posiciones junto con el uso del operador MOD para la asignación de piezas de la torre origen a la torre destino correspondiente.

Estructura del código.

Además del uso de variables globales para el guardado de las últimas posiciones de las torresDestino (**posicionesDestino{nTorresDestino}**), destacan los siguientes procedimientos:

1. **TrasladaTorre**, Por cada pieza de la torre origen, asigna empleando la aritmética modular la torre destino, moviéndola a su nueva posición con el procedimiento **muevePieza** y posteriormente actualizando el array **posicionesDestino{n Torre actual}** .

2. **CalcularPosicionesDestino**, Calcula e incorpora al array **posicionesDestino** las posiciones de las bases de las torres.

Los videos demostrativos se encuentran en la carpeta versionAvanzada siguiendo la estructura RobotStudioFecha_Radio-Npiezas.mp4 junto con el código utilizado (codigo.txt)

```
MODULE Module1
  CONST robtarget pBaseOrigen:=[[1500,-500,750],[0,0,1,0],[0,0,0,0],[9E+09,9E+09,9E+09,0]]
  CONST robtarget pCentroDestino:=[[1700,1000,750],[0,0,1,0],[0,0,0,0],[9E+09,9E+09,9E+09,0]]
  CONST robtarget pReposo:=[[1808.910161514,0,1855],[0.5,0,0.866025404,0],[0,0,0,0]]
  VAR num altoPieza := 50;
  var num nPiezasTorreOrigen := 10;
  CONST num nTorresDestino := 3;
  var num radioCircunf := 300;
  VAR robtarget posicionesDestino{nTorresDestino};
  PROC MAIN()
    SetDO SD_ActivaVentosa, 0;
    MoveJ pReposo, v1000,z50,Ventosa;
    TrasladaTorre pBaseOrigen,pCentroDestino,pReposo,altoPieza,nPiezasTorreOrigen;
    MoveJ pReposo, v1000,z50,Ventosa;
    SetDO SD_ActivaVentosa, 0;
  ENDPROC
  PROC TrasladaTorre( robtarget pBaseOrigen,robtarget pCentroDestino,robtarget pReposo,
    num altoPieza,num nPiezasTorreOrigen)
    VAR num x;
    VAR robtarget pOrigen;
    VAR robtarget pDestino;

    CalcularPosicionesDestino radioCircunf,nTorresDestino, pCentroDestino;
    FOR i FROM 1 TO nPiezasTorreOrigen DO
      x := ((i-1)MOD nTorresDestino) +1;
      pOrigen:= pBaseOrigen;
      pOrigen.trans.z := pBaseOrigen.trans.z +((nPiezasTorreOrigen-i+1)*a;
      pDestino:= posicionesDestino{x};
      pDestino.trans.z := pDestino.trans.z +altoPieza;
      muevePieza pOrigen,pDestino;
      posicionesDestino{x}:= pDestino;
    ENDFOR
  ENDPROC

  PROC CalcularPosicionesDestino(num radioCircunf,num nTorresDestino,robtarget pCentroDestino)
    VAR num angulo;
    VAR robtarget pAux;

    FOR i FROM 1 TO nTorresDestino DO
      pAux:= pCentroDestino;
      angulo := (360 / nTorresDestino) * (i - 1);
      pAux.trans.x := pCentroDestino.trans.x + radioCircunf * Cos(angulo)
      pAux.trans.y := pCentroDestino.trans.y + radioCircunf * Sin(angulo)
      pAux.trans.z:= pCentroDestino.trans.z;
      posicionesDestino{i} := pAux;
    ENDFOR
  ENDPROC

  PROC muevePieza(robtarget pOrigen,robTarget pDestino)
    VAR robtarget pOrigenAprox;
    VAR robtarget pDestinoAprox;
    VAR num alturaAprox:= 150;
    pOrigenAprox := pOrigen;
```

```
    pDestinoAprox := pDestino;
    pOrigenAprox.trans.z := pOrigen.trans.z + alturaAprox;
    pDestinoAprox.trans.z := pDestino.trans.z + alturaAprox;
    MoveJ pOrigenAprox,v2000,z100,Ventosa\WObj:=wobj0;
    MoveL pOrigen,v300,fine,Ventosa\WObj:=wobj0;
    SetDO SD_ActivaVentosa, 1; ! se puede combinar con mode LDO
    WaitDI ED_PiezaCogida, 1;
    MoveL pOrigenAprox,v300,z0,Ventosa\WObj:=wobj0;
    MoveJ pDestinoAprox,v2000,z100,Ventosa\WObj:=wobj0;
    MoveL pDestino,v300,fine,Ventosa\WObj:=wobj0;
    SetDO SD_ActivaVentosa, 0;
    WaitDI ED_PiezaCogida, 0;
    MoveL pDestinoAprox,v300,fine,Ventosa\WObj:=wobj0;
ENDPROC
ENDMODULE
```