

Laboratorio No. 02 - Robótica Industrial - Trayectorias, Entradas y Salidas Digitales.



1. Resultados de aprendizaje

- Conocer los elementos de un robot industrial.
- Realizar calibración de herramientas en el robot real, así como en RobotStudio.
- Identificar los tipos de movimientos en el espacio de la herramienta útiles para trabajos de manipulación.
- Ampliar el manejo de funciones proporcionadas por RobotStudio.
- Utilizar diversas funciones de RAPID.
- Utilizar el módulo de entradas y salidas digitales dispuesto en el controlador IRC5.

2. Requisitos

- RobotStudio versión 5 o superior INSTALADO y activo.
- Hoja de datos del manipulador industrial IRB 140.
- Hoja de datos del módulo **3HAC025917-001/00 DSQC 652**.
- Software CAD que permita exportar en formato .STL o .SAT.
- Manipuladores industriales ABB IRB 140.
- Memoria USB por equipo de trabajo.
- Cable Ethernet RJ-45.

3. Descripción

En esta práctica se desarrollarán 5 ejes temáticos:

1. **Trayectorias:** El primer componente es la comprensión del flujo de programa en RAPID, comprender los tipos de movimientos MOVJ y MOVL, así como sus restricciones.
2. **Herramienta:** El segundo es el diseño y construcción de una herramienta a usar con el robot. Con esta herramienta se podrá comprender el concepto de MTH de Tool y se podrá realizar el proceso de calibración tanto en software como con el robot real (TCP).
3. **Calibración de Herramienta:** Comparación del proceso de calibración de la herramienta usando el robot real y empleando Robotstudio.
4. **Programación intermedia en RAPID:** Manejo de bucles, condicionales.

5. Manejo del módulo de Entradas y Salidas Digitales: Comunicaciones por señales digitales.

4. Trabajo a desarrollar

A continuación se describe la actividad que debe ser desarrollada en el laboratorio.



4.1. Introducción

En la industria de alimentos, en especial panadería, quieren mejorar su proceso de producción. Ellos consideran que la decoración de las tortas puede ser un punto a ser ejecutado por robots. En los siguientes links se puede observar diferentes automatización de decoración de tortas con robots [Video](#) [Video](#), [Video](#).

Inspirados en los vídeos anteriores, y considerando las limitaciones del laboratorio se propone hacer la decoración de una torta virtual. Es decir. Sobre una superficie plana bien sea redonda o rectangular, escribir los nombres de cada uno de los integrantes del grupo y una decoración a su gusto. En este caso se desea generar los *paths* o los movimientos de robot necesarios para representar las letras y la decoración. Se deben tener en cuenta las siguientes restricciones:

- El tamaño de la torta es para 20 personas
- Las trayectorias a desarrollar deberán realizarse en un rango de velocidades entre 100 y 1000.
- La zona tolerable de errores máxima debe ser de ± 10 .
- El movimiento debe partir de una posición home especificada (puede ser el home del robot) y realizar la trayectoria de cada palabra y decoración con un trazo continuo. El movimiento debe finalizar en la misma posición de home en la que se inició.
- La decoración de la torta debe ser realizada sobre una torta virtual.
- Los nombres deben estar separados.

4.2. Herramienta

Diseñar y construir una herramienta pro grupo que permita fijar un marcador o plumón al flanche del robot.

Realizar el proceso de calibración de la herramienta construida empleando la técnica más apropiada para crear el objeto *Tooldata*.

Realizar la importación del modelo CAD de la herramienta diseñada a *RobotStudio*, comparar y analizar las diferencias de los *tooldata* creados.

4.3. Workobject:

Se debe replicar la tarea de decorado sobre el cuadrante $x(+)$, $y(-)$, utilizando un nuevo *workobject* y reutilizando las trayectorias previamente definidas para el cuadrante $x(+)$, $y(+)$. Esta operación debe realizarse sobre una de las bandas transportadoras, de modo que el sistema mantenga la misma secuencia de movimientos adaptada a la nueva ubicación del pastel.

4.4. Entradas y Salidas Digitales

A partir del trabajo realizado en la práctica se debe recibir dos entradas digitales y controlar dos salidas digitales,

- Revisar con el docente en el laboratorio el funcionamiento de las señales digitales de entrada y salida, así como caracterizar y en caso de que sea necesario ajustar los paneles de mando.
- Se deben crear dos entradas y dos salidas digitales configurandolas correctamente en el código de RAPID.
- La primera señal de entrada debe iniciar una rutina de decorado sobre cualquier superficie y encender una luz de indicación. Al final de la rutina el brazo debe regresar a su posición de HOME donde todos los ángulos articulares son 0 grados.
- La segunda señal de entrada debe posicionar el brazo en una pose de mantenimiento donde se pueda instalar o desinstalar la herramienta y se debe apagar la luz de indicación.
- Los botones en el mando de señales digitales deben controlar la transición entre las rutinas.

Las velocidades, zonas de aproximación y otros parámetros de movimiento quedan a discreción de cada equipo de trabajo, según lo visto en clase y las indicaciones del docente.

4.5. Control del Transporte del Pastel

Para controlar el encendido y apagado de una de las bandas transportadoras, se debe conectar una de las salidas digitales del sistema de control a una de las entradas del variador de frecuencia que regula el motor de dicha banda. Esta conexión permitirá activar o detener el movimiento de la banda de forma programada.

Es importante que la torta se coloque previamente sobre la banda transportadora. De este modo, una vez el robot haya finalizado el proceso de decoración, se activará la señal digital correspondiente, encendiendo la banda transportadora y permitiendo que el pastel sea trasladado automáticamente hasta el otro extremo.

4.6. Requerimientos

Se debe presentar

- Descripción detallada de la solución planteada.
- Diagrama de flujo de acciones del robot.
- Plano de planta de la ubicación de cada uno de los elementos.
- Descripción de las funciones utilizadas.
- Diseño de la herramienta detallado.
- Código en RAPID del módulo utilizado para el desarrollo de la práctica.
- Vídeo que contenga la simulación en *RobotStudio* así como la implementación de la práctica con los robots reales.
- Los videos debe comenzar con la introducción oficial del laboratorio LabSIR [Intro LabSIR](#).



5. Entrega

1. **Forma de trabajo:** Grupal de 2 personas.
2. **Importante:** Cada integrante deberá poner la URL del repositorio creado.
3. **Entregables:** Se deberá crear un repositorio en GitHub con lo expresado en el punto requerimientos
4. **Fecha de entrega:** Según actividad en Moodle.

6. Ejemplos

6.1. Ejemplo 1

Uso de funciones para escribir en la pantalla del flexpendat , y leer información desde el flex.

- TPWrite
- TPErase
- CRobT
- MoveAbsJ

Ver el siguiente ejemplo

```
MODULE Module1
  PERS robtarget Target_x:[[-631.504,-0.631705,500],[0.000300926,0.798752,0.000399502,-0.601661],[-2,0,0],
  CONST jointtarget home:=[[0,0,0,0,-45,0],[9E9,9E9,9E9,9E9,9E9,9E9]];

  PROC main()
    TPErase;
    TPWrite "EN EL MAIN";
    Homming;
  ENDPROC

  PROC Homming()
    TPWrite "LLendo a casita";
    Target_x:=CRobT(\Tool:=tool0 \WObj:=wobj0);
    Target_x.trans.z:=500;
    MoveJ Target_x,v500,fine,tool0\WObj:=wobj0;
    MoveAbsJ home, v500, fine, tool0;
  ENDPROC
ENDMODULE
```

6.2. Ejemplo 2

Instrucciones utilizadas.

- CalcJointT
- IF ELSE ENDIF

```

PROC Llegable()
  VAR jointtarget jointpos; !la posicion que se evalua si es alcanzable
  VAR bool reach;!booleano para indicador
  reach:=TRUE;
  !calcular los angulos articulares para esa posicion
  jointpos:=CalcJointT(Target_x,tool0\WObj:=wobj0);
  !si no es alcanzable, arroja un error, el error se trata en el "catch" de abajo
!y sigue con el flujo del programa
  IF reach=TRUE THEN
    TPWrite "Alcanzable";
    Path_x;!ejecutar trayectoria en caso de que si sea alcanzable
  ELSE
    TPWrite "NO ALCANZABLE MANO";
  ENDIF

  RETURN ;

ERROR
  !handler del error
!los errores tienen su "id" o "nombre" especifico, en ERRNO se guarda el error presente
  IF ERRNO=ERR_ROBLIMIT or ERRNO= ERR_OUTSIDE_REACH THEN
!si se da el error, no es alcanzable, asi que el booleano es falso
!y no se ejecuta el path
    reach:=FALSE;
    TRYNEXT;
  ENDIF
ENDPROC

```

6.3. Ejemplo 3

Un procedimiento que ejecute dos trayectorias 4 veces mediante un bucle (para ello utilizar una variable que se incrementa en una unidad en cada ciclo a la que se denomina contador, la sentencia GOTO y una etiqueta denominada superior.

Utilizando if

```

PROC INICIO( )
  VAR num valor := 0;
  INIC:
    Path_RECTANGULO;
    Path_CIRCULO;
    valor:=valor+1;
    IF valor<4 THEN
      GOTO INIC;
    ENDIF
ENDPROC

```

Utilizando for

```

PROC INICIO( )
  VAR num valor := 0;
  FOR valor FROM 1 TO 4 DO
    Path_ RECTANGULO;
    Path_ CIRCULO;
  ENDFOR
ENDPROC

```

6.4. Ejemplo 4

Funciones de home, varios WOs, y trayectorias automaticas.

```

MODULE Module1
PERS tooldata Tregaskiss22deg:=[TRUE,[[148.62,0.25,326.31],[0.833900724,0,0.551914471,0]], [1,[0,0,100]]
PERS wobjdata WoCuadrado:=[FALSE,TRUE,"",[614.276136575,138.113342602,400],[0,0.382681799,0.923880209]]
PERS wobjdata WoCirculo:=[FALSE,TRUE,"",[538.541,-320.769,400],[0,0,1,0]], [[0,0,0],[1,0,0,0]]];
CONST robtarget Home:=[31.00391808,-21.153764968,-155.339057477],[0.681388409,0.049591309,-0.26259679]
CONST robtarget Target_10:=[0.000755226,-0.000224895,0],[0.707105531,0,0,-0.707108031],[0,0,0,0],[9E+09]
CONST robtarget Target_20:=[-0.000305522,299.999775103,0],[0.707105531,0,0,-0.707108031],[0,0,0,0],[9E+09]
CONST robtarget Target_30:=[399.999694475,300.001189434,0],[0.707105531,0,0,-0.707108031],[0,0,0,0],[9E+09]
CONST robtarget Target_40:=[400.000755224,0.001189436,0],[0.707105531,0,0,-0.707108031],[0,0,0,0],[9E+09]
CONST robtarget Target_50:=[0.000755226,-0.000224895,0],[0.707105531,0,0,-0.707108031],[0,0,0,0],[9E+09]
CONST robtarget homeC:=[111.923419362,27.246194358,-236.66299703],[0.96321759,0.096302747,-0.04691324]
CONST robtarget Target_60:=[-100.000420593,0.000077939,0],[0.923879533,0,0,0.382683432],[0,0,0,0],[9E+09]
CONST robtarget Target_70:=[10.452425733,99.452267475,0],[0.923879533,0,0,0.382683432],[0,0,0,0],[9E+09]
CONST robtarget Target_80:=[97.81433948,-20.791091143,0],[0.923879533,0,0,0.382683432],[0,0,0,0],[9E+09]
CONST robtarget Target_90:=[-10.45326692,-99.452111598,0],[0.923879533,0,0,0.382683432],[0,0,0,0],[9E+09]
CONST robtarget Target_100:=[-100.000420593,0.000077939,0],[0.923879533,0,0,0.382683432],[0,0,0,0],[9E+09]

```

! Variables asociadas al Homing

```

VAR num nAxle1;
VAR num nAxle2;
VAR num nAxle3;
VAR num nAxle4;
VAR num nAxle5;
VAR num nAxle6;

```

PROC main()

```

Path_10;
Path_20;
Path_30;

```

ENDPROC

PROC Path_10()

```

nAxle1:=0;
nAxle2:=0;
nAxle3:=0;
nAxle4:=0;
nAxle5:=30;
nAxle6:=0;
MoveAbsJ [[nAxle1,nAxle2,nAxle3,nAxle4,nAxle5,nAxle6],
[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]],v1000,Fine,Tregaskiss22deg\WObj:=WObj0;

```

ENDPROC

PROC Path_20()

```

MoveL Home,v1000,z100,Tregaskiss22deg\WObj:=WoCuadrado;
MoveL Target_10,v1000,z100,Tregaskiss22deg\WObj:=WoCuadrado;
MoveL Target_20,v1000,z100,Tregaskiss22deg\WObj:=WoCuadrado;
MoveL Target_30,v1000,z100,Tregaskiss22deg\WObj:=WoCuadrado;
MoveL Target_40,v1000,z100,Tregaskiss22deg\WObj:=WoCuadrado;
MoveL Target_50,v1000,z100,Tregaskiss22deg\WObj:=WoCuadrado;
MoveL Home,v1000,z100,Tregaskiss22deg\WObj:=WoCuadrado;

```

ENDPROC

PROC Path_30()

```

    MoveL homeC,v1000,z100,Tregaskiss22deg\WObj:=WoCirculo;
    MoveL Target_60,v1000,fine,Tregaskiss22deg\WObj:=WoCirculo;
    MoveC Target_70,Target_80,v1000,z100,Tregaskiss22deg\WObj:=WoCirculo;
    MoveC Target_90,Target_100,v1000,z100,Tregaskiss22deg\WObj:=WoCirculo;
    MoveL Target_100,v1000,z100,Tregaskiss22deg\WObj:=WoCirculo;
    MoveL homeC,v1000,fine,Tregaskiss22deg\WObj:=WoCirculo;
  ENDPROC
PROC Path_40()
  MoveJ homeC,v1000,z100,Tregaskiss22deg\WObj:=WoCirculo;
  MoveJ Home,v1000,z100,Tregaskiss22deg\WObj:=WoCuadrado;
  MoveJ homeC,v1000,z100,Tregaskiss22deg\WObj:=WoCirculo;
ENDPROC
ENDMODULE

```

6.5. Ejemplo

Manejo de entradas y salidas

```

VARnum n;
VARnumcont;
!*****TABULAR SIEMPRE!!!!*****
PROC main()
  WHILE TRUE DO
    SetDO Salida1,0;
    IF Entrada1=1 THEN
      TPreadFK n, "Selecciona el número de ciclos a realizar","Uno","Dos","Tres","Cuatro","Cinco";
      FOR cont FROM 1 TO n DO
        Path_10;
      ENDFOR
    ENDIF
  ENDWHILE
  WaitTime 0.1;
ENDPROC
PROC Path_10()
  MoveLINICIO,v1000,z10,MyTool\WObj:=wobj0;
  MoveL Target_10,v1000,fine,MyTool\WObj:=Workobject_CAJA;
  MoveL Target_20,v1000,fine,MyTool\WObj:=Workobject_CAJA;
  SetDO Salida2,1;
  MoveL Target_30,v1000,fine,MyTool\WObj:=Workobject_CAJA;
  MoveL Target_40,v1000,fine,MyTool\WObj:=Workobject_CAJA;
  MoveL Target_10,v1000,fine,MyTool\WObj:=Workobject_CAJA;
  SetDO Salida2,0;
  MoveL INICIO,v1000,z10,MyTool\WObj:=wobj0;
  Set Salida1; !otra forma de poner a uno una salida digital
  WaitTime 1;
  Reset Salida1; !otra forma de poner a cero una salida digital
  RETURN;
ENDPROC
ENDMODULE

```