

ANEXO A

Anexo que incluye el código desarrollado en lenguaje RAPID de RobotStudio

MODULE ModulePiano

!Herramientas

PERS tooldata

Servo:=[TRUE,[[0,0,114.2],[1,0,0,0]],0.215,[8.7,12.3,49.2],[1,0,0,0],0.00021,0.00024,0.00009];

PERS tooldata Pinza_Pulsador:=[TRUE,[[0,0,175],[1,0,0,0]],1,[0,0,1],[1,0,0,0],0,0,0];

CONST string GripOpen := "Open";

CONST string GripClose := "Close";

!Workobjects

TASK PERS wobjdata Teclado_1:=[FALSE,TRUE,"",[[340,230,60],[0.707107,0,0,-0.707107]],[[0,0,0],[1,0,0,0]]];

!Robtarjets

CONST robtarget

Target_30:=[[217.426750238,65.704591866,306.461059933],[0.055665079,-0.751661944,0.647438141,0.112825416],[-1,0,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

!Punto de partida de la primera escala

CONST robtarget Target_10:=[[30,100,0],[0,1,0,0],[0,0,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_10_2:=[[62.5,100,0],[0,1,0,0],[0,0,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

!Variables del main

PERS bool nosalir;

PERS num opcion;

PERS num notaPulsada;

LOCAL PERS num seed:=10426;

!Variable generales

PERS bool ActiveRobot1;

PERS bool ActiveRobot2;

PERS bool Dos_Notas := FALSE;

!Distancia entre notas

CONST num DistNota:=65;

PERS num Nota_a_tocar;

CONST num Nota_DO := 0;

CONST num Nota_RE := 1;

CONST num Nota_MI := 2;

CONST num Nota_FA := 3;

CONST num Nota_SOL := 4;

CONST num Nota_LA := 5;

```

CONST num Nota_SI := 6;
CONST num Nota_Do_Alto := 7;

LOCAL VAR bool saltodelinea := FALSE;

!Variables para la lectura de ficheros
VAR string lineRead;
VAR string filePath := "/home/cancion.txt";
VAR string note;
VAR num i;

!Variables server
VAR socketdev server_L;
VAR socketdev client_L;
VAR socketdev server_R;
VAR socketdev client_R;
VAR string nota:= " ";

!Variables de interrupción
VAR intnum Idi0; !DO
VAR intnum Idi1; !RE
VAR intnum Idi2; !MI
VAR intnum Idi3; !FA
VAR intnum Idi4; !SOL
VAR intnum Idi5; !LA
VAR intnum Idi6; !SI
VAR intnum Idi7; !Dance ON
VAR intnum Idi8; !ASCENDENTE
VAR intnum Idi9; !DESCENDENTE
VAR intnum Idi10; !CANCION
VAR intnum Idi11; !NOTAS ALEATORIAS
VAR intnum Idi12; !Una nota o dos
VAR intnum Idi13; !ROB1
VAR intnum Idi14; !ROB2
VAR intnum Idi15; !SALIR

LOCAL PROC Init()

    IDelete Idi0;
    IDelete Idi1;
    IDelete Idi2;
    IDelete Idi3;
    IDelete Idi4;
    IDelete Idi5;
    IDelete Idi6;
    IDelete Idi7;
    IDelete Idi8;

```

IDelete Idi9;
IDelete Idi10;
IDelete Idi11;
IDelete Idi12;
IDelete Idi13;
IDelete Idi14;
IDelete Idi15;

CONNECT Idi0 WITH trap_0_DO;
CONNECT Idi1 WITH trap_1_RE;
CONNECT Idi2 WITH trap_2_MI;
CONNECT Idi3 WITH trap_3_FA;
CONNECT Idi4 WITH trap_4_SOL;
CONNECT Idi5 WITH trap_5_LA;
CONNECT Idi6 WITH trap_6_SI;
CONNECT Idi7 WITH trap_7_DanceON;
CONNECT Idi8 WITH trap_8_EscalaAscendente;
CONNECT Idi9 WITH trap_9_EscalaDescendente;
CONNECT Idi10 WITH trap_10_Partitura;
CONNECT Idi11 WITH trap_11_Aleatorio;
CONNECT Idi12 WITH trap_12_UnaDosNotas;
CONNECT Idi13 WITH trap_13_ROB1;
CONNECT Idi14 WITH trap_14_ROB2;
CONNECT Idi15 WITH trap_15_NOSALIR;

ISignalDI DI_1_DO, 1, Idi0;
ISignalDI DI_2_RE, 1, Idi1;
ISignalDI DI_3_MI, 1, Idi2;
ISignalDI DI_4_FA, 1, Idi3;
ISignalDI DI_5_SOL, 1, Idi4;
ISignalDI DI_6_LA, 1, Idi5;
ISignalDI DI_7_SI, 1, Idi6;
ISignalDI DI_8_DanceON, 1, Idi7;
ISignalDI DI_9_Asc, 1, Idi8;
ISignalDI DI_10_Desc, 1, Idi9;
ISignalDI DI_11_Fich, 1, Idi10;
ISignalDI DI_12_Rand, 1, Idi11;
ISignalDI DI_13_NumN, 1, Idi12;
ISignalDI DI_14_Rob1, 1, Idi13;
ISignalDI DI_15_Rob2, 1, Idi14;
ISignalDI DI_16_Salir, 1, Idi15;

ENDPROC

TRAP trap_0_DO

Nota_a_tocar := Nota_DO;
opcion := 5;

ENDTRAP

TRAP trap_1_RE

Nota_a_tocar := Nota_RE;
opcion := 5;

ENDTRAP

TRAP trap_2_MI

Nota_a_tocar := Nota_MI;
opcion := 5;

ENDTRAP

TRAP trap_3_FA

Nota_a_tocar := Nota_FA;
opcion := 5;

ENDTRAP

TRAP trap_4_SOL

Nota_a_tocar := Nota_SOL;
opcion := 5;

ENDTRAP

TRAP trap_5_LA

Nota_a_tocar := Nota_LA;
opcion := 5;

ENDTRAP

TRAP trap_6_SI

Nota_a_tocar := Nota_SI;
opcion := 5;

ENDTRAP

TRAP trap_7_DanceON

IF DO_5_DanceON = 1 THEN

```

    TPWWrite "Activo DanceRobot -> Pasamos a desactivarlo";
    reset DO_5_DanceON;
ELSE
    TPWWrite "Incactivo DanceRobot -> Pasamos a Activarlo";
    set DO_5_DanceON;
ENDIF

ENDTRAP

TRAP trap_8_EscalaAscendente

    opcion := 1;

ENDTRAP

TRAP trap_9_EscalaDescendente

    opcion := 2;

ENDTRAP

TRAP trap_10_Partitura

    opcion := 3;

ENDTRAP

TRAP trap_11_Aleatorio

    opcion := 4;

ENDTRAP

TRAP trap_12_UnaDosNotas

    IF DO_1_OneNote1 = 1 THEN
        TPWWrite "Una nota, cambiamos a dos";
        reset DO_1_OneNote1;
        reset DO_3_OneNote2;
    ELSE
        TPWWrite "Dos notas, cambiamos a una";
        set DO_1_OneNote1;
        set DO_3_OneNote2;
    ENDIF

ENDTRAP

TRAP trap_13_ROB1

```

```
IF DO_2_ActiveRobot1 = 1 THEN
    TPWWrite "Activo Robot1 -> Pasamos a desactivarlo";
    reset DO_2_ActiveRobot1;
    ActiveRobot1 := FALSE;
ELSE
    TPWWrite "Incactivo Robot1 -> Pasamos a Activarlo";
    set DO_2_ActiveRobot1;
    ActiveRobot1 := TRUE;
ENDIF
```

ENDTRAP

TRAP trap_14_ROB2

```
IF DO_4_ActiveRobot2 = 1 THEN
    TPWWrite "Activo Robot2 -> Pasamos a desactivarlo";
    reset DO_4_ActiveRobot2;
    ActiveRobot2 := FALSE;
ELSE
    TPWWrite "Incactivo Robot2 -> Pasamos a Activarlo";
    set DO_4_ActiveRobot2;
    ActiveRobot2 := TRUE;
ENDIF
```

ENDTRAP

TRAP trap_15_NOSALIR

```
    nosalir:=FALSE;
```

ENDTRAP

PROC mainPiano()

```
    TPWWrite "Comenzando actividad";
```

```
    !Iniciamos variables e interrupciones
```

```
    Init;
```

```
    nosalir:=TRUE;
```

```
    opcion:=0;
```

```
    SocketCreate server_L;
```

```
    SocketBind server_L, "127.0.0.1", 5000;
```

```
    Socketlisten server_L;
```



```
SocketAccept server_L, client_L;
```

```
WHILE nosalir DO
```

```
    TPWrite "Entramos en el bucle while";
```

```
    IF ActiveRobot1 = TRUE THEN
```

```
        TEST opcion
```

```
        CASE 1:
```

```
            TPWrite "Tocar escala de manera ascendente";
```

```
            IF DO_1_OneNote1 = 1 THEN
```

```
                Tocar_Notas_Asc;
```

```
            ELSE
```

```
                Tocar_Notas_Asc_Intervalos;
```

```
            ENDIF
```

```
            opcion := 0;
```

```
        CASE 2:
```

```
            TPWrite "Tocar escala de manera descendente";
```

```
            IF DO_1_OneNote1 = 1 THEN
```

```
                Tocar_Notas_Desc;
```

```
            ELSE
```

```
                Tocar_Notas_Desc_Intervalos;
```

```
            ENDIF
```

```
            opcion := 0;
```

```
        CASE 3:
```

```
            TPWrite "Leemos partitura";
```

```
            LeerPartitura("Sonrisas y Lagrimas.txt");
```

```
            opcion:=0;
```

```
        CASE 4:
```

```
            NotasAleatorias;
```

```
        CASE 5:
```

```
            TocarNota(Nota_a_tocar);
```

```
            opcion := 0;
```

```
            ResetDO;
```

```

        DEFAULT:

        ENDTEST

    ENDIF

    WaitTime .5;

ENDWHILE

SocketClose client_L;

ENDPROC

LOCAL PROC ResetDO()

    Reset DO_6_DO;
    Reset DO_7_RE;
    Reset DO_8_MI;
    Reset DO_9_FA;
    Reset DO_10_SOL;
    Reset DO_11_LA;
    Reset DO_12_SI;

ENDPROC

! Tocar notas Aleatorias
PROC NotasAleatorias()

    TPWrite "Tocar nota aleatoria";
    Nota_a_tocar:=rand(6);
    TocarNota(Nota_a_tocar);

ENDPROC

! Generar un numero aleatorio
LOCAL FUNC num rand(num posibilidad)
    VAR num random;
    VAR num valor:=0;
    VAR num ej:=0;
    VAR num inicio:=0;
    VAR num fin:=0;

    seed:=(171*seed) MOD 30269;
    random:=seed/30269;
    valor:=random*posibilidad+1;
    valor:=Trunc(valor\Dec:=0);

```

RETURN valor;

ENDFUNC

! Leer partitura de un fichero .txt

PROC LeerPartitura(string nombre)

VAR string caracter;

VAR string linea;

VAR num caso;

VAR num fila :=0;

VAR iodev fic;

Close fic;

Open "HOME:"\File:=nombre,fic\Read;

WHILE linea <> EOF DO

TPWrite "Empezamos por la primera linea";

IF saltodelinea THEN

linea:=ReadStr(fic);

saltodelinea:=FALSE;

fila:=0;

GOTO init;

ENDIF

init: linea:=ReadStr(fic\Delim:="\2F");

TPWrite linea;

IF opcion = 0 OR StrLen(linea)=0 THEN

GOTO last;

ENDIF

TEST linea

CASE "C":

TPWrite "Tocamos el DO";

Nota_a_tocar := Nota_DO;

CASE "D":

TPWrite "Tocamos el RE";

Nota_a_tocar := Nota_RE;

CASE "E":

TPWrite "Tocamos el MI";

Nota_a_tocar := Nota_MI;

CASE "F":

```
TPWrite "Tocamos el FA";  
Nota_a_tocar := Nota_FA;
```

```
CASE "G":  
  TPWrite "Tocamos el SOL";  
  Nota_a_tocar := Nota_SOL;
```

```
CASE "A":  
  TPWrite "Tocamos el LA";  
  Nota_a_tocar := Nota_LA;
```

```
CASE "B":  
  TPWrite "Tocamos el SI";  
  Nota_a_tocar := Nota_SI;
```

```
DEFAULT:
```

```
ENDTEST
```

```
TocarNota(Nota_a_tocar);
```

```
ENDWHILE
```

```
last: ENDPROC
```

```
PROC TocarNota(num notaPulsada)
```

```
  MoveJ Offs (Target_10, DistNota*notaPulsada,  
0,0),v300,z0,Pinza_Pulsador\WObj:=Teclado_1;
```

```
IF notaPulsada = 0 THEN  
  nota := "Do";  
  Set DO_6_DO;  
ELSEIF notaPulsada = 1 THEN  
  nota := "Re";  
  Set DO_7_RE;  
ELSEIF notaPulsada = 2 THEN  
  nota := "Mi";  
  Set DO_8_MI;  
ELSEIF notaPulsada = 3 THEN  
  nota := "Fa";  
  Set DO_9_FA;  
ELSEIF notaPulsada = 4 THEN  
  nota := "Sol";  
  Set DO_10_SOL;  
ELSEIF notaPulsada = 5 THEN  
  nota := "La";  
  Set DO_11_LA;
```

```
ELSEIF notaPulsada = 6 THEN
```

```
    nota := "Si";
```

```
    Set DO_12_Sl;
```

```
ENDIF
```

```
!iniciamos el server
```

```
SocketSend client_L \Str:=nota;
```

```
MoveL Target_30,v500,z0,Pinza_Pulsador\WObj:=Teclado_1;
```

```
ENDPROC
```

```
LOCAL PROC Tocar_Notas_Asc()
```

```
    VAR intnum cont;
```

```
    TPWrite "Empezamos a tocar la escala ascendente";
```

```
    FOR cont FROM 0 TO 6 DO
```

```
        MoveJ Offs (Target_10, DistNota*cont,  
0,0),v300,z0,Pinza_Pulsador\WObj:=Teclado_1;
```

```
        MoveL Target_30,v500,z0,Pinza_Pulsador\WObj:=Teclado_1;
```

```
        !TocarNota(cont);
```

```
    ENDFOR
```

```
ENDPROC
```

```
PROC Tocar_Notas_Asc_Intervalos()
```

```
    VAR intnum cont;
```

```
    FOR cont FROM 0 TO 3 DO
```

```
        MoveJ Offs (Target_10_2, (DistNota*2)*cont,  
0,0),v300,z0,Pinza_Pulsador\WObj:=Teclado_1;
```

```
        MoveL Target_30,v300,z0,Pinza_Pulsador\WObj:=Teclado_1;
```

```
    ENDFOR
```

```
ENDPROC
```

```
LOCAL PROC Tocar_Notas_Desc()
```

```
    VAR intnum cont;
```

```
    TPWrite "Empezamos a tocar la escala descendente";
```

```
    FOR cont FROM 6 TO 0 DO
```

```
        TocarNota(cont);
```

ENDFOR

ENDPROC

LOCAL PROC Tocar_Notas_Desc_Intervalos()

VAR intnum cont;

FOR cont FROM 3 TO 0 DO

MoveJ Offs (Target_10_2, (DistNota*2)*cont,
0,0),v300,z0,Pinza_Pulsador\WObj:=Teclado_1;

MoveL Target_30,v300,z0,Pinza_Pulsador\WObj:=Teclado_1;

ENDFOR

ENDPROC

!Ir a posicion de do con

PROC Path_30()

MoveL Target_10_2,v1000,z100,Pinza_Pulsador\WObj:=Teclado_1;
ENDPROC

PROC Path_10()

MoveL Target_10,v1000,z100,Servo\WObj:=Teclado_1;
ENDPROC

PROC Path_30_()

MoveL Target_10_2,v1000,z100,Pinza_Pulsador\WObj:=Teclado_1;
ENDPROC

ENDMODULE

MODULE ModulePiano

CONST robtarget

Target_20:=[[30,100,0],[0,1,0,0],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget

Target_20_2:=[[62.5,100,0],[0,1,0,0],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget

Target_Init:=[[217.426750238,65.704591866,306.461059933],[0.055665079,-0.751661944,0.647438141,0.112825416],[-1,0,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

PERS tooldata

Pinza_Pulsador:=[TRUE,[[0,0,175],[1,0,0,0]],[1,[0,0,1],[1,0,0,0],0,0,0]];

TASK PERS wobjdata

Teclado_2:=[FALSE,TRUE,"",[340,230,60],[0.707107,0,0,-0.707107]],[[0,0,0],[1,0,0,0]]];

!Herramientas

PERS tooldata

Servo:=[TRUE,[[0,0,114.2],[1,0,0,0]],[0.215,[8.7,12.3,49.2],[1,0,0,0],0.00021,0.00024,0.00009]];

CONST string GripOpen := "Open";

CONST string GripClose := "Close";

CONST robtarget

Target_30:=[[217.426750238,65.704591866,306.461059933],[0.055665079,-0.751661944,0.647438141,0.112825416],[-1,0,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

!Punto de partida de la primera escala

CONST robtarget Target_10:=[[30,100,0],[0,1,0,0],[0,0,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

CONST robtarget Target_10_2:=[[62.5,100,0],[0,1,0,0],[0,0,-1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];

!Variables del main

PERS bool nosalir;

PERS num opcion;

PERS num notaPulsada;

LOCAL PERS num seed:=10426;

!Distancia entre notas

CONST num DistNota:=65;

PERS num Nota_a_tocar;

CONST num Nota_DO := 0;

```

CONST num Nota_RE := 1;
CONST num Nota_MI := 2;
CONST num Nota_FA := 3;
CONST num Nota_SOL := 4;
CONST num Nota_LA := 5;
CONST num Nota_SI := 6;
CONST num Nota_Do_Alto := 7;

LOCAL VAR bool saltodelinea := FALSE;

!Variable generales
PERS bool ActiveRobot1;
PERS bool ActiveRobot2;
PERS bool Dos_Notas;
VAR string NotesFile := "Notes.txt";

!Variables para la lectura de ficheros
VAR string lineRead;
VAR string filePath := "/home/cancion.txt";
VAR string note;
VAR num i;

!Variables server
VAR socketdev server;
VAR socketdev client;
VAR string nota:= " ";

!Variables de interrupción
VAR intnum Idi0; !DO
VAR intnum Idi1; !RE
VAR intnum Idi2; !MI
VAR intnum Idi3; !FA
VAR intnum Idi4; !SOL
VAR intnum Idi5; !LA
VAR intnum Idi6; !SI
VAR intnum Idi7; !DanceON
VAR intnum Idi8; !ASCENDENTE
VAR intnum Idi9; !DESCENDENTE
VAR intnum Idi10; !CANCION
VAR intnum Idi11; !NOTAS ALEATORIAS
VAR intnum Idi12; !Una nota o dos
VAR intnum Idi13; !ROB1
VAR intnum Idi14; !ROB2
VAR intnum Idi15; !SALIR

LOCAL PROC Init()

    IDelete Idi0;
    IDelete Idi1;

```



```
IDelete Idi2;  
IDelete Idi3;  
IDelete Idi4;  
IDelete Idi5;  
IDelete Idi6;  
IDelete Idi7;  
IDelete Idi8;  
IDelete Idi9;  
IDelete Idi10;  
IDelete Idi11;  
IDelete Idi15;
```

```
CONNECT Idi0 WITH trap_0_D0;  
CONNECT Idi1 WITH trap_1_RE;  
CONNECT Idi2 WITH trap_2_MI;  
CONNECT Idi3 WITH trap_3_FA;  
CONNECT Idi4 WITH trap_4_SOL;  
CONNECT Idi5 WITH trap_5_LA;  
CONNECT Idi6 WITH trap_6_SI;  
CONNECT Idi7 WITH trap_7_DanceON;  
CONNECT Idi8 WITH trap_8_EscalaAscendente;  
CONNECT Idi9 WITH trap_9_EscalaDescendente;  
CONNECT Idi10 WITH trap_10_Partitura;  
CONNECT Idi11 WITH trap_11_Aleatorio;  
CONNECT Idi15 WITH trap_15_NOSALIR;
```

```
ISignalDI DI_1_D0, 1, Idi0;  
ISignalDI DI_2_RE, 1, Idi1;  
ISignalDI DI_3_MI, 1, Idi2;  
ISignalDI DI_4_FA, 1, Idi3;  
ISignalDI DI_5_SOL, 1, Idi4;  
ISignalDI DI_6_LA, 1, Idi5;  
ISignalDI DI_7_SI, 1, Idi6;  
ISignalDI DI_8_DanceON, 1, Idi7;  
ISignalDI DI_9_Asc, 1, Idi8;  
ISignalDI DI_10_Desc, 1, Idi9;  
ISignalDI DI_11_Fich, 1, Idi10;  
ISignalDI DI_12_Rand, 1, Idi11;  
ISignalDI DI_16_Salir, 1, Idi15;
```

ENDPROC

TRAP trap_0_D0

```
Nota_a_tocar := Nota_D0;  
opcion := 5;
```

ENDTRAP

TRAP trap_1_RE

```

        Nota_a_tocar := Nota_RE;
        opcion := 5;

ENDTRAP

TRAP trap_2_MI

        Nota_a_tocar := Nota_MI;
        opcion := 5;

ENDTRAP

TRAP trap_3_FA

        Nota_a_tocar := Nota_FA;
        opcion := 5;

ENDTRAP

TRAP trap_4_SOL

        Nota_a_tocar := Nota_SOL;
        opcion := 5;

ENDTRAP

TRAP trap_5_LA

        Nota_a_tocar := Nota_LA;
        opcion := 5;

ENDTRAP

TRAP trap_6_SI

        Nota_a_tocar := Nota_SI;
        opcion := 5;

ENDTRAP

TRAP trap_7_DanceON

        IF DO_5_DanceON = 1 THEN
            TPWWrite "Activo DanceRobot -> Pasamos a desactivarlo";
            reset DO_5_DanceON;
            ActiveRobot1 := FALSE;
        ELSE
            TPWWrite "Incactivo DanceRobot -> Pasamos a Activarlo";

```

```

        set DO_5_DanceON;
        ActiveRobot1 := TRUE;
    ENDIF

ENDTRAP

TRAP trap_8_EscalaAscendente

    opcion := 1;

ENDTRAP

TRAP trap_9_EscalaDescendente

    opcion := 2;

ENDTRAP

TRAP trap_10_Partitura

    opcion := 3;

ENDTRAP

TRAP trap_11_Aleatorio

    opcion := 4;

ENDTRAP

TRAP trap_12_UnaDosNotas

    IF DO_3_OneNote2 = 1 THEN
        TPWRite "Una nota, cambiamos a dos";
        reset DO_3_OneNote2;
    ELSE
        TPWRite "Dos notas, cambiamos a una";
        set DO_3_OneNote2;
    ENDIF

ENDTRAP

TRAP trap_13_ROB1

    IF DO_2_ActiveRobot1 = 1 THEN
        TPWRite "Activo Robot1 -> Pasamos a desactivarlo";
        reset DO_2_ActiveRobot1;
        ActiveRobot1 := FALSE;
    ELSE

```

```

        TPWRite "Incactivo Robot1 -> Pasamos a Activarlo";
        set DO_2_ActiveRobot1;
        ActiveRobot1 := TRUE;
    ENDIF

ENDTRAP

TRAP trap_14_ROB2

    IF DO_4_ActiveRobot2 = 1 THEN
        TPWRite "Activo Robot2 -> Pasamos a desactivarlo";
        reset DO_4_ActiveRobot2;
        ActiveRobot2 := FALSE;
    ELSE
        TPWRite "Incactivo Robot2 -> Pasamos a Activarlo";
        set DO_4_ActiveRobot2;
        ActiveRobot2 := TRUE;
    ENDIF

ENDTRAP

TRAP trap_15_NOSALIR

    nosalir:=FALSE;

ENDTRAP

PROC mainPiano()

    TPWrite "Comenzando actividad";

    !Iniciamos variables e interrupciones
    Init;
    nosalir:=TRUE;
    opcion:=0;

    WHILE nosalir DO

        TPWrite "Entramos en el bucle while";

        IF ActiveRobot2 = TRUE THEN

            TEST opcion
            CASE 1:
                TPWrite "Tocar escala de manera ascendente";
                Tocar_Notas_Asc;
                opcion := 0;
            END
        END IF
    END WHILE
END PROC

```

```
CASE 2:
    TPWrite "Tocar escala de manera descendente";
    Tocar_Notas_Desc;
    opcion := 0;
```

```
CASE 3:
    TPWrite "Leemos partitura";
    LeerPartitura("Sonrisas y Lagrimas.txt");
    opcion:=0;
```

```
CASE 4:

    NotasAleatorias;
```

```
CASE 5:

    TocarNota(Nota_a_tocar);
    opcion := 0;
```

```
DEFAULT:
```

```
ENDTEST
```

```
ENDIF
```

```
WaitTime .5;
```

```
ENDWHILE
```

```
ENDPROC
```

```
! Tocar notas Aleatorias
PROC NotasAleatorias()
```

```
    TPWrite "Tocar nota aleatoria";
    Nota_a_tocar:=rand(6);
    TocarNota(Nota_a_tocar);
```

```
ENDPROC
```

```
! Generar un numero aleatorio
LOCAL FUNC num rand(num posibilidad)
    VAR num random;
    VAR num valor:=0;
    VAR num ej:=0;
    VAR num inicio:=0;
    VAR num fin:=0;
```

```

seed:=(171*seed) MOD 30269;
random:=seed/30269;
valor:=random*posibilidad+1;
valor:=Trunc(valor\Dec:=0);

RETURN valor;

ENDFUNC

! Leer partitura de un fichero .txt
PROC LeerPartitura(string nombre)
  VAR string caracter;
  VAR string linea;
  VAR num caso;
  VAR num fila :=0;
  VAR iodev fic;

  Close fic;
  Open "HOME:"\File:=nombre,fic\Read;

  WHILE linea <> EOF DO

    TPWrite "Empezamos por la primera linea";

    IF saltodelinea THEN
      linea:=ReadStr(fic);
      saltodelinea:=FALSE;
      fila:=0;
      GOTO init;
    ENDIF

    init: linea:=ReadStr(fic\Delim:="\2F");

    TPWrite linea;

    IF opcion = 0 OR StrLen(linea)=0 THEN
      GOTO last;
    ENDIF

    TEST linea
    CASE "C":
      TPWrite "Tocamos el DO";
      Nota_a_tocar := Nota_DO;

    CASE "D":
      TPWrite "Tocamos el RE";
      Nota_a_tocar := Nota_RE;

    CASE "E":

```

```

        TPWrite "Tocamos el MI";
        Nota_a_tocar := Nota_MI;

CASE "F":
    TPWrite "Tocamos el FA";
    Nota_a_tocar := Nota_FA;

CASE "G":
    TPWrite "Tocamos el SOL";
    Nota_a_tocar := Nota_SOL;

CASE "A":
    TPWrite "Tocamos el LA";
    Nota_a_tocar := Nota_LA;

CASE "B":
    TPWrite "Tocamos el SI";
    Nota_a_tocar := Nota_SI;

DEFAULT:

ENDTEST

TocarNota(Nota_a_tocar);

ENDWHILE

last: ENDPROC

PROC TocarNota(num notaPulsada)

    MoveJ Offs (Target_10, DistNota*notaPulsada,
0,0),v300,z0,Pinza_Pulsador\WObj:=Teclado_2;

    IF notaPulsada = 0 THEN
        nota := "Do";
    ELSEIF notaPulsada = 1 THEN
        nota := "Re";
    ELSEIF notaPulsada = 2 THEN
        nota := "Mi";
    ELSEIF notaPulsada = 3 THEN
        nota := "Fa";
    ELSEIF notaPulsada = 4 THEN
        nota := "Sol";
    ELSEIF notaPulsada = 5 THEN
        nota := "La";

```

```

ELSEIF notaPulsada = 6 THEN
    nota := "Si";
ENDIF

MoveL Target_30,v500,z0,Pinza_Pulsador\WObj:=Teclado_2;

ENDPROC

LOCAL PROC Tocar_Notas_Asc()

    VAR intnum cont;

    TPWrite "Empezamos a tocar la escala ascendentemente";
    FOR cont FROM 0 TO 6 DO
        TocarNota(cont);
    ENDFOR

ENDPROC

PROC Tocar_Notas_Asc_Intervalos()

    VAR intnum cont;

    FOR cont FROM 0 TO 3 DO
        MoveJ Offs (Target_10_2, (DistNota*2)*cont,
0,0),v300,z0,Pinza_Pulsador\WObj:=Teclado_2;
        MoveL Target_30,v300,z0,Pinza_Pulsador\WObj:=Teclado_2;
    ENDFOR

ENDPROC

LOCAL PROC Tocar_Notas_Desc()

    VAR intnum cont;

    TPWrite "Empezamos a tocar la escala descendentemente";

    FOR cont FROM 6 TO 0 DO
        TocarNota(cont);
    ENDFOR

ENDPROC

LOCAL PROC Tocar_Notas_Desc_Intervalos()

    VAR intnum cont;

    FOR cont FROM 3 TO 0 DO

```



```

        MoveJ Offs (Target_10_2, (DistNota*2)*cont,
0,0),v300,z0,Pinza_Pulsador\WObj:=Teclado_2;
        MoveL Target_30,v300,z0,Pinza_Pulsador\WObj:=Teclado_2;
    ENDFOR

```

```

ENDPROC

```

```

LOCAL PROC LeerPartitura_()

```

```

    VAR intnum cont;
    VAR string frag;

    FOR cont FROM 6 TO 0 DO
        MoveJ Offs (Target_10, DistNota*cont,
0,0),v500,z0,Pinza_Pulsador\WObj:=Teclado_2;
        MoveL Target_30,v500,z0,Pinza_Pulsador\WObj:=Teclado_2;
    ENDFOR

```

```

ENDPROC

```

```

PROC Path_20()

```

```

    MoveL Target_20,v1000,z100,Servo\WObj:=Teclado_2;

```

```

ENDPROC

```

```

PROC Path_40()

```

```

    MoveL Target_20_2,v1000,z100,Pinza_Pulsador\WObj:=Teclado_2;

```

```

ENDPROC

```

```

PROC Tocar_Notas_Asc_2()

```

```

    MoveJ Offs (Target_20, 65*Nota_a_tocar,
0,0),v300,z0,Pinza_Pulsador\WObj:=Teclado_2;
    MoveL Target_Init,v150,z0,Pinza_Pulsador\WObj:=Teclado_2;

```

```

ENDPROC

```

```

PROC Tocar_Notas_Desc_2()

```

```

    MoveJ Offs (Target_20, 65*Nota_a_tocar,
0,0),v300,z0,Pinza_Pulsador\WObj:=Teclado_2;
    MoveL Target_Init,v150,z0,Pinza_Pulsador\WObj:=Teclado_2;

```

```

ENDPROC

```

```

ENDMODULE

```

MODULE ModuleDanceL

PERS tooldata

Pinza_Yumi_L:= [TRUE, [[0,0,114.2],[1,0,0,0]], [0.2,[0,0,50],[1,0,0,0]], 0, 0,0]];

PERS tooldata

Pinza_Yumi_L_Prueba:= [TRUE, [[0,0,114.2],[1,0,0,0]], [1,[0,0,1],[1,0,0,0]], 0,0,0]];

LOCAL CONST jointtarget target_do := [[-133.471326165,-46.983870968,-89.218637993,11.433691756,16.494623656,0.82078853],[0,9E+09,9E+09,9E+09,9E+09,9E+09]];

LOCAL CONST jointtarget target_2 := [[-133.471326165,-50.335125448,-61.501792115,11.433691756,16.494623656,0.82078853],[0,9E+09,9E+09,9E+09,9E+09,9E+09]];

LOCAL CONST jointtarget Abajo_ := [[-82.740143369,-84.517921147,-8.985663082,13.512544803,44.035842294,0.82078853],[33.216845878,9E+09,9E+09,9E+09,9E+09,9E+09]];

LOCAL CONST jointtarget Adentro_ := [[-105.689964158,-73.123655914,-8.985663082,13.512544803,44.035842294,0.82078853],[33.216845878,9E+09,9E+09,9E+09,9E+09,9E+09]];

LOCAL CONST jointtarget Afuera_ := [[-41.672043011,-73.123656915,-8.985663683,13.512544018,44.035838089,0.820788475],[33.216845878,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR bool nosalir :=FALSE;

PERS bool dance_ON := FALSE;

VAR num opcion;

VAR string notaSocket:= " ";

VAR socketdev sock;

VAR string ip := "127.0.0.1";

PERS string nota := "Si";

VAR intnum Idi0; !Dance ON

VAR intnum Idi1; !DO

VAR intnum Idi2; !RE

VAR intnum Idi3; !MI

VAR intnum Idi4; !FA

VAR intnum Idi5; !SOL

VAR intnum Idi6; !LA

VAR intnum Idi7; !SI

LOCAL PROC Init()

 IDelete Idi0;

 IDelete Idi1;

 IDelete Idi2;

 IDelete Idi3;

 IDelete Idi4;

```
IDelete Idi5;  
IDelete Idi6;  
IDelete Idi7;
```

```
CONNECT Idi0 WITH trap_0_ON;  
CONNECT Idi1 WITH trap_1_do;  
CONNECT Idi2 WITH trap_2_re;  
CONNECT Idi3 WITH trap_3_mi;  
CONNECT Idi4 WITH trap_4_fa;  
CONNECT Idi5 WITH trap_5_sol;  
CONNECT Idi6 WITH trap_6_la;  
CONNECT Idi7 WITH trap_7_si;
```

```
ISignalDI DI_0_ON, 1, Idi0;  
ISignalDI DI_1_DO, 1, Idi1;  
ISignalDI DI_2_RE, 1, Idi2;  
ISignalDI DI_3_MI, 1, Idi3;  
ISignalDI DI_4_FA, 1, Idi4;  
ISignalDI DI_5_SOL, 1, Idi5;  
ISignalDI DI_6_LA, 1, Idi6;  
ISignalDI DI_7_SI, 1, Idi7;
```

```
ENDPROC
```

```
TRAP trap_0_ON
```

```
    opcion := 2;
```

```
ENDTRAP
```

```
TRAP trap_1_do
```

```
    opcion := 1;
```

```
ENDTRAP
```

```
TRAP trap_2_re
```

```
    opcion := 1;
```

```
ENDTRAP
```

```
TRAP trap_3_mi
```

```
    opcion := 1;
```

```
ENDTRAP
```

```
TRAP trap_4_fa
```

```

        opcion := 1;

ENDTRAP

TRAP trap_5_sol

        opcion := 1;

ENDTRAP

TRAP trap_6_la

        opcion := 1;

ENDTRAP

TRAP trap_7_si

        opcion := 1;

ENDTRAP

PROC mainL()

    nosalir := TRUE;

    Init;

    SocketCreate sock;
    SocketConnect sock, ip, 5000 \Time := 100;

    WHILE nosalir DO

        IF DI_0_ON = 1 OR dance_ON = TRUE THEN

            TPWrite "Que comience el baile: ";
            TPWrite nota;

            TEST opcion

            CASE 1:
                nota := LeerNotaDeIRB1200();
                BailarNota;
                opcion:= 0;

            CASE 2:
                BailarNormal;

```

```

        DEFAULT:

        ENDTEST

    ENDIF

        WaitTime 0.1;
    ENDWHILE

    SocketClose sock;

ENDPROC

FUNC string LeerNotaDeIRB1200()

    SocketReceive sock \Str:=notaSocket;

    RETURN notaSocket;

ENDFUNC

PROC BailarNota()

    ConfL\Off;
    ConfJ\Off;

    IF nota = "Do" THEN
        MoveAbsJ Abajo_, v100, fine, Pinza_Yumi_L;
    ELSEIF nota = "Re" THEN
        MoveAbsJ Adentro_, v100, fine, Pinza_Yumi_L;
    ELSEIF nota = "Mi" THEN
        MoveAbsJ Abajo_, v100, fine, Pinza_Yumi_L;
    ELSEIF nota = "Fa" THEN
        MoveAbsJ Adentro_, v100, fine, Pinza_Yumi_L;
    ELSEIF nota = "Sol" THEN
        MoveAbsJ Abajo_, v100, fine, Pinza_Yumi_L;
    ELSEIF nota = "La" THEN
        MoveAbsJ Adentro_, v100, fine, Pinza_Yumi_L;
    ELSEIF nota = "Si" THEN
        MoveAbsJ Abajo_, v100, fine, Pinza_Yumi_L;
    ENDIF

ENDPROC

LOCAL PROC BailarNormal()

    ConfL\Off ;
    ConfJ\Off;

```

```
MoveAbsJ Abajo_, v100, fine, Pinza_Yumi_L_Prueba;  
MoveAbsJ Adentro_, v100, fine, Pinza_Yumi_L_Prueba;  
!MoveAbsJ Afuera_, v100, fine, Pinza_Yumi_L_Prueba;  
MoveAbsJ Abajo_, v100, fine, Pinza_Yumi_L_Prueba;  
MoveAbsJ Adentro_, v100, fine, Pinza_Yumi_L_Prueba;  
!MoveAbsJ Afuera_, v100, fine, Pinza_Yumi_L_Prueba;
```

ENDPROC

ENDMODULE

MODULE ModuleDanceR

```
LOCAL VAR string notaSocket:= " ";
LOCAL VAR bool nosalir :=FALSE;
LOCAL VAR bool dance_ON := FALSE;
LOCAL VAR num opcion;
```

```
CONST jointtarget Arriba := [[0,-
123.104829751,26.552420163,0,26.20968127,0],[ -
41.915358049,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST jointtarget Arriba_L:= [[0,-
123.104829751,55.930107527,0,26.209679562,0],[ -
41.915354634,9E+09,9E+09,9E+09,9E+09,9E+09]];
CONST jointtarget Abajo := [[0,-
66.421146953,55.930107527,0,26.209679562,0],[ -
41.915354634,9E+09,9E+09,9E+09,9E+09,9E+09]];
```

PERS tooldata

```
Pinza_Yumi_R:=[TRUE,[[0,0,114.2],[1,0,0,0]], [0.215,[8.7,12.3,49.2],[1,
0,0,0],0.00021,0.00024,0.00009]];
```

```
LOCAL VAR intnum Idi0; !Dance ON
LOCAL VAR intnum Idi1; !DO
LOCAL VAR intnum Idi2; !RE
LOCAL VAR intnum Idi3; !MI
LOCAL VAR intnum Idi4; !FA
LOCAL VAR intnum Idi5; !SOL
LOCAL VAR intnum Idi6; !LA
LOCAL VAR intnum Idi7; !SI
```

LOCAL PROC Init()

```
  IDelete Idi0;
  IDelete Idi1;
  IDelete Idi2;
  IDelete Idi3;
  IDelete Idi4;
  IDelete Idi5;
  IDelete Idi6;
  IDelete Idi7;
```

```
  CONNECT Idi0 WITH trap_0_ON;
  CONNECT Idi1 WITH trap_1_do;
  CONNECT Idi2 WITH trap_2_re;
  CONNECT Idi3 WITH trap_3_mi;
  CONNECT Idi4 WITH trap_4_fa;
  CONNECT Idi5 WITH trap_5_sol;
  CONNECT Idi6 WITH trap_6_la;
  CONNECT Idi7 WITH trap_7_si;
```

```
ISignalDI DI_0_ON, 1, Idi0;  
ISignalDI DI_1_DO, 1, Idi1;  
ISignalDI DI_2_RE, 1, Idi2;  
ISignalDI DI_3_MI, 1, Idi3;  
ISignalDI DI_4_FA, 1, Idi4;  
ISignalDI DI_5_SOL, 1, Idi5;  
ISignalDI DI_6_LA, 1, Idi6;  
ISignalDI DI_7_SI, 1, Idi7;
```

ENDPROC

LOCAL TRAP trap_0_ON

opcion := 2;

ENDTRAP

LOCAL TRAP trap_1_do

opcion := 1;

ENDTRAP

LOCAL TRAP trap_2_re

opcion := 1;

ENDTRAP

LOCAL TRAP trap_3_mi

opcion := 1;

ENDTRAP

LOCAL TRAP trap_4_fa

opcion := 1;

ENDTRAP

LOCAL TRAP trap_5_sol

opcion := 1;

ENDTRAP

LOCAL TRAP trap_6_la


```

        opcion := 1;

ENDTRAP

LOCAL TRAP trap_7_si

        opcion := 1;

ENDTRAP

PROC mainR()

    nosalir := TRUE;

    Init;

    WHILE nosalir DO

        IF DI_0_ON = 1 THEN

            TEST opcion

            CASE 1:
                notaSocket := nota;
                BailarNota;
                opcion:= 0;

            CASE 2:
                BailarNormal;

            DEFAULT:

            ENDTEST

            ENDIF

            WaitTime 0.1;
        ENDWHILE

    ENDPROC

LOCAL PROC BailarNota()

    Confl\Off;
    ConfJ\Off;

    IF notaSocket = "Do" THEN
        MoveAbsJ Arriba, v100, fine, Pinza_Yumi_R;
    ELSEIF notaSocket = "Re" THEN

```

```
        MoveAbsJ Arriba_L, v100, fine, Pinza_Yumi_R;
ELSEIF notaSocket = "Mi" THEN
    MoveAbsJ Abajo, v100, fine, Pinza_Yumi_R;
ELSEIF notaSocket = "Fa" THEN
    MoveAbsJ Arriba, v100, fine, Pinza_Yumi_R;
ELSEIF notaSocket = "Sol" THEN
    MoveAbsJ Arriba_L, v100, fine, Pinza_Yumi_R;
ELSEIF notaSocket = "La" THEN
    MoveAbsJ Abajo, v100, fine, Pinza_Yumi_R;
ELSEIF notaSocket = "Si" THEN
    MoveAbsJ Arriba, v100, fine, Pinza_Yumi_R;
ENDIF
```

ENDPROC

LOCAL PROC BailarNormal()

```
    Confl\Off ;
    ConfJ\Off;
```

```
    MoveAbsJ Arriba, v100, fine, Pinza_Yumi_R;
    MoveAbsJ Arriba_L, v100, fine, Pinza_Yumi_R;
    MoveAbsJ Abajo, v100, fine, Pinza_Yumi_R;
    MoveAbsJ Arriba, v100, fine, Pinza_Yumi_R;
```

ENDPROC

ENDMODULE

MODULE ModuleDanceL

PERS tooldata

Pinza_Yumi_L:= [TRUE, [[0,0,114.2],[1,0,0,0]], [0.2,[0,0,50],[1,0,0,0]], 0, 0,0]];

PERS tooldata

Pinza_Yumi_L_Prueba:= [TRUE, [[0,0,114.2],[1,0,0,0]], [1,[0,0,1],[1,0,0,0]], 0,0,0]];

LOCAL CONST jointtarget target_do := [[-133.471326165,-46.983870968,-89.218637993,11.433691756,16.494623656,0.82078853],[0,9E+09,9E+09,9E+09,9E+09,9E+09]];

LOCAL CONST jointtarget target_2 := [[-133.471326165,-50.335125448,-61.501792115,11.433691756,16.494623656,0.82078853],[0,9E+09,9E+09,9E+09,9E+09,9E+09]];

LOCAL CONST jointtarget Abajo_ := [[-82.740143369,-84.517921147,-8.985663082,13.512544803,44.035842294,0.82078853],[33.216845878,9E+09,9E+09,9E+09,9E+09,9E+09]];

LOCAL CONST jointtarget Adentro_ := [[-105.689964158,-73.123655914,-8.985663082,13.512544803,44.035842294,0.82078853],[33.216845878,9E+09,9E+09,9E+09,9E+09,9E+09]];

LOCAL CONST jointtarget Afuera_ := [[-41.672043011,-73.123656915,-8.985663683,13.512544018,44.035838089,0.820788475],[33.216845878,9E+09,9E+09,9E+09,9E+09,9E+09]];

VAR bool nosalir :=FALSE;

PERS bool dance_ON := FALSE;

VAR num opcion;

VAR string notaSocket:= " ";

VAR socketdev sock;

VAR string ip := "127.0.0.1";

PERS string nota := "Si";

VAR intnum Idi0; !Dance ON

VAR intnum Idi1; !DO

VAR intnum Idi2; !RE

VAR intnum Idi3; !MI

VAR intnum Idi4; !FA

VAR intnum Idi5; !SOL

VAR intnum Idi6; !LA

VAR intnum Idi7; !SI

LOCAL PROC Init()

 IDelete Idi0;

 IDelete Idi1;

 IDelete Idi2;

 IDelete Idi3;

 IDelete Idi4;

```
IDelete Idi5;  
IDelete Idi6;  
IDelete Idi7;
```

```
CONNECT Idi0 WITH trap_0_ON;  
CONNECT Idi1 WITH trap_1_do;  
CONNECT Idi2 WITH trap_2_re;  
CONNECT Idi3 WITH trap_3_mi;  
CONNECT Idi4 WITH trap_4_fa;  
CONNECT Idi5 WITH trap_5_sol;  
CONNECT Idi6 WITH trap_6_la;  
CONNECT Idi7 WITH trap_7_si;
```

```
ISignalDI DI_0_ON, 1, Idi0;  
ISignalDI DI_1_DO, 1, Idi1;  
ISignalDI DI_2_RE, 1, Idi2;  
ISignalDI DI_3_MI, 1, Idi3;  
ISignalDI DI_4_FA, 1, Idi4;  
ISignalDI DI_5_SOL, 1, Idi5;  
ISignalDI DI_6_LA, 1, Idi6;  
ISignalDI DI_7_SI, 1, Idi7;
```

```
ENDPROC
```

```
TRAP trap_0_ON
```

```
    opcion := 2;
```

```
ENDTRAP
```

```
TRAP trap_1_do
```

```
    opcion := 1;
```

```
ENDTRAP
```

```
TRAP trap_2_re
```

```
    opcion := 1;
```

```
ENDTRAP
```

```
TRAP trap_3_mi
```

```
    opcion := 1;
```

```
ENDTRAP
```

```
TRAP trap_4_fa
```

```

        opcion := 1;

ENDTRAP

TRAP trap_5_sol

        opcion := 1;

ENDTRAP

TRAP trap_6_la

        opcion := 1;

ENDTRAP

TRAP trap_7_si

        opcion := 1;

ENDTRAP

PROC mainL()

    nosalir := TRUE;

    Init;

    SocketCreate sock;
    SocketConnect sock, ip, 5000 \Time := 100;

    WHILE nosalir DO

        IF DI_0_ON = 1 OR dance_ON = TRUE THEN

            TPWrite "Que comience el baile: ";
            TPWrite nota;

            TEST opcion

            CASE 1:
                nota := LeerNotaDeIRB1200();
                BailarNota;
                opcion:= 0;

            CASE 2:
                BailarNormal;

```

```

        DEFAULT:

        ENDTEST

    ENDIF

        WaitTime 0.1;
    ENDWHILE

    SocketClose sock;

ENDPROC

FUNC string LeerNotaDeIRB1200()

    SocketReceive sock \Str:=notaSocket;

    RETURN notaSocket;

ENDFUNC

PROC BailarNota()

    ConfL\Off;
    ConfJ\Off;

    IF nota = "Do" THEN
        MoveAbsJ Abajo_, v100, fine, Pinza_Yumi_L;
    ELSEIF nota = "Re" THEN
        MoveAbsJ Adentro_, v100, fine, Pinza_Yumi_L;
    ELSEIF nota = "Mi" THEN
        MoveAbsJ Abajo_, v100, fine, Pinza_Yumi_L;
    ELSEIF nota = "Fa" THEN
        MoveAbsJ Adentro_, v100, fine, Pinza_Yumi_L;
    ELSEIF nota = "Sol" THEN
        MoveAbsJ Abajo_, v100, fine, Pinza_Yumi_L;
    ELSEIF nota = "La" THEN
        MoveAbsJ Adentro_, v100, fine, Pinza_Yumi_L;
    ELSEIF nota = "Si" THEN
        MoveAbsJ Abajo_, v100, fine, Pinza_Yumi_L;
    ENDIF

ENDPROC

LOCAL PROC BailarNormal()

    ConfL\Off ;
    ConfJ\Off;

```

```
MoveAbsJ Abajo_, v100, fine, Pinza_Yumi_L_Prueba;  
MoveAbsJ Adentro_, v100, fine, Pinza_Yumi_L_Prueba;  
MoveAbsJ Abajo_, v100, fine, Pinza_Yumi_L_Prueba;  
MoveAbsJ Adentro_, v100, fine, Pinza_Yumi_L_Prueba;
```

ENDPROC

ENDMODULE