

## 10 APÉNDICES

### 10.1 APÉNDICE A. CÓDIGO PARA SIMULACIÓN MEDIANTE PIC

#### 10.1.1 CÓDIGO PIC MAESTRO

```
#include<16f876A.h>
#use delay(clock=20000000)
#FUSES XT,NOWDT,NOPROTECT,NOLVP,PUT,BROWNOUT
#use rs232(baud=9600, xmit=PIN_C6, rcv=PIN_C7,STREAM=COM_PC)
#use I2C(master, sda=PIN_C4, scl=PIN_C3)

int dato=0;
int valor=0;
int valor1=0;
int direccion=0;
int sentido;
int cuenta=1;
int Vel=0;
int valor2=0;
char pic;
int PICS;

void leer(dirección){
do{
    i2c_start();
    i2c_write(dirección);
    dato=i2c_read(0);
    i2c_stop();
    putc(dato);
    valor1=1;
}while(valor1!=1);
}

void escribir(dato,direccion){
    delay_ms(50);
    i2c_start();
    delay_ms(50);
    i2c_write(dirección);
    delay_ms(50);
    i2c_write(dato);
    delay_ms(50);
    delay_ms(50);
    i2c_stop();
    delay_ms(50);
    valor1=1;
}

void Slave(PICS)
{

```

```

do{ //seleccionado.
  delay_ms(250);
  escribir(Vel,PICS); //Uso de función escribir para enviar Velocidad
  if(sentido=='L') //comparación de variable sentido para
  { //no enviar dato incorrecto de sentido.
    escribir(sentido,PICS); //Función escribir para enviar sentido en este
  } //caso "L"
  else if(sentido=='R') //compara sentido con letra R
  {
    escribir(sentido,PICS); //Función escribir para enviar sentido en este
  } //caso "R"
  else //Si la variable sentido no recibió una L o R
  { //el sentido será S para detener al motor.
    sentido='S'; //Sentido se establece como S
    escribir(sentido,PICS); //Función escribir para enviar sentido en este
  } //caso "S"
  escribir(cuenta,PICS); //Función escribir para envío de cuentas a moverse
  valor=1; //variable para terminar ciclo while
}while(valor!=1);
}
void main(){ //Función principal la cual tiene un ciclo
do{ //infinito

  if(kbhit()) //espera hasta que reciba datos
  { //por puerto serie
    valor=fgetc(COM_PC); //recepción de dato de PIC esclavo
    sentido=fgetc(COM_PC); //recepción dato de sentido de giro
    Vel=fgetc(COM_PC); //recepción de dato de velocidad

    if(valor=='A'){ //comparación de valor con PIC A
      Slave(0xA2); //escritura de datos a esclavo A2
      fprintf(COM_PC,"PIC A"); //enviar por puerto serie mensaje del PIC
    } //esclavo usado
    else if(valor=='B'){ //comparación de valor con PIC B
      Slave(0xA4); //escritura de datos a esclavo A4
      fprintf (COM_PC,"PIC B"); //mensaje por puerto serie del PIC usado
    }
    else if(valor=='C'){ //comparación de valor con PIC C
      Slave(0xA6); //escritura de datos a esclavo A6
      fprintf (COM_PC,"PIC C"); //mensaje por puerto serie del PIC usado
    }
    else if(valor=='D'){ //comparación de valor con PIC
      Slave(0xA8); //escritura de datos a esclavo A6
      fprintf (COM_PC,"PIC D"); //mensaje por puerto serie del PIC usado
    }
    else { //en caso de que no sea ningún valor

```

### 10.1.2 CÓDIGO PIC ESCLAVO

```
#include<16f876A.h>
#use delay(clock=2000000)
#FUSES XT,NOWDT,NOPROTECT,NOLVP,PUT,BROWNOUT
#use rs232(baud=9600, xmit=PIN_C6, rcv=PIN_C7)
#use I2C(slave, sda=PIN_C4, scl=PIN_C3,address=0xA2,FORCE_HW)
#byte porta=5
#byte portb=6
#byte portc=7

int dato=0;
int dato1=0;
int valor2=0;
int state=0;
int Vel=0;
int Sen=0;
int x=0;
int y=0;
int valor=0;
int Cuenta=0;

void leer(void){                                     //función para leer datos por I2C como esclavo
do{
    valor2=0;                                       //variable para mantener ciclo while hasta tener datos
    if(i2c_poll()==TRUE){                          //espera hasta recibir datos
        dato=i2c_read();                           //leer datos recibidos
        if(dato==0xA2){                            //comparar datos con dirección del PIC
            dato=i2c_read();                       //leer dato y guardarlo en variable
            valor2=1;                              //variable para terminar ciclo
        }
    }
}while(valor2!=1);
}

void escribir(dato1){                               //función para escribir datos a PIC maestro
do{
    valor2=0;                                       //variable para mantener ciclo while hasta escribir datos
    i2c_write(dato1);                              //escribir dato
    i2c_write(dato1);                              //reescribir dato
    fputc(dato1);                                  //mandar por puerto serie el dato enviado por I2C
    valor2=1;                                       //variable para terminar ciclo while
}while(valor2!=1);
}

void MoverDer()                                    //función para giro a la derecha
{
bit_set(portb,6);                                  //poner en 1 pin 6 del puerto B
bit_clear(portb,7);                                //poner en 0 pin 7 del puerto B
}
```

```

void MoverIzq() //función para giro a la izquierda
{
bit_clear(portb,6); //poner en 0 pin 6 del puerto B
bit_set(portb,7); //poner en 1 pin 7 del puerto B
}

void Stop() //función para detener giro con frenado rápido.
{
bit_set(portb,6); //poner en 1 pin 6 del puerto B
bit_set(portb,7); //poner en 1 pin 7 del puerto B
}

void Contar() //función para contar pulsos de encoder
{
x=get_timer0(); //obtener numero de cuentas del timer0
y=get_timer1(); //obtener numero de cuentas del timer1
if(y==128) //comparación para limpiar el timer1 una
{ //vez que ha llegado a 128 pulsos
set_timer1(0); //pues el timer0 ya llevara una cuenta mas
}
write_eeprom(0x02,x); //escritura de dato de timer0 en dir 02 de EEPROM
write_eeprom(0x03,y); //escritura de dato de timer1 en dir 03 de EEPROM
}

void Guardar() //función para guardar datos en EEPROM
{
write_eeprom(0x00,Vel); //escritura de dato de velocidad en dir 00 de EEPROM
write_eeprom(0x01,Sen); //escritura de dato de sentido en dir 01 de EEPROM
}

void PWM() //función para crear PWM
{
setup_timer_2(T2_DIV_BY_16,255,1); //configura timer2
setup_ccp1(CCP_PWM); //configura modulo CCP como PWM
set_pwm1_duty(Vel); //configura ciclo de trabajo
} //con variable Vel

void main(){

SET_TRIS_A( 0xFF ); //Puerto A como entradas
SET_TRIS_B( 0x00 ); //Puerto B como salidas
//SET_TRIS_C( 0xBB ); //Puerto C como salidas
setup_timer_0 (RTCC_DIV_256|RTCC_EXT_L_TO_H); //configurar timer0
setup_timer_1 ( T1_EXTERNAL | T1_DIV_BY_2 ); //configurar timer1
set_timer0(0); //resetear timer0
set_timer1(0); //resetear timer1
portb=0x00; //limpiar puerto B
delay_ms(050); //retraso de 50 ms

do
{
do{
delay_ms(050);

```

```

leer(); //leer dato de velocidad
write_eeprom(0x00,dato); //escribir en EEPROM
leer(); //leer dato de sentido
write_eeprom(0x01,dato); //escribir en EEPROM
leer(); //leer dato de cuenta
write_eeprom(0x02,dato); //escribir dato en EEPROM
}while(valor2!=1); //hacer ciclo mientras valor2 sea diferente de 1
valor2=0; //reseteo de variable para mantener ciclo
Vel=read_eeprom(0x00); //leer velocidad de la EEPROM
Sen=read_eeprom(0x01); //leer sentido de la EEPROM
Cuenta=read_eeprom(0x02); //leer cuenta de la EEPROM

if(Sen=='R') //comparación del sentido de giro
{ //si se recibe R
MoverDer(); //habilitar función de giro a la derecha
}
else if(Sen=='L') //si se recibe L
{ //habilitar giro a la izquierda
MoverIzq();
}
else if(Sen=='S') //si se recibe S
{ //habilitar frenado rápido del motor
Stop();
}
else //en caso de no recibir alguna de estas se detendrá
{ //al motor
Stop(); //habilitar frenado rápido del motor
}
PWM(); //habilitar función para salida de PWM
Contar(); //llamado de la función contar
x=read_eeprom(0x02); //lectura de la EEPROM para timer0
y=read_eeprom(0x03); //lectura de la EEPROM para timer1
}while (1); //ciclo infinito
}

```

### 10.1.3 CÓDIGO DE PANEL EN VISUAL BASIC

```
Private Sub Cad_Der_Click(Index As Integer)
Dim Letra As String
Dim V As String
Letra = "A"
MSComm1.Output = Letra
Letra = "R"
MSComm1.Output = Letra
V = Chr(HScroll1.Value)
MSComm1.Output = V
End Sub
```

'Función del Botón Cadera Derecha al hacer clic  
'Variable "Letra" como cadena  
'Variable "V" como cadena  
'Variable letra se le asigna valor de A  
'Envía por puerto de comunicación la variable Letra  
'Variable letra se le asigna valor de R  
'Envía por puerto de comunicación la variable Letra  
'variable V con valor del scroll convertido a cadena  
'Envía por puerto de comunicación la variable V

```
Private Sub Cad_Izq_Click(Index As Integer)
Dim Letra As String
Dim V As String
Letra = "A"
MSComm1.Output = Letra
Letra = "L"
MSComm1.Output = Letra
V = Chr(HScroll1.Value)
MSComm1.Output = V
End Sub
```

Función del Botón Cadera Izquierda al hacer clic  
'Variable "Letra" como cadena  
'Variable "V" como cadena  
'Variable letra se le asigna valor de A  
'Envía por puerto de comunicación la variable Letra  
'Variable letra se le asigna valor de L  
'Envía por puerto de comunicación la variable Letra  
'variable V con valor del scroll convertido a cadena  
'Envía por puerto de comunicación la variable V

```
Private Sub Cod_Der_Click(Index As Integer)
Dim Letra As String
Dim V As String
Letra = "C"
MSComm1.Output = Letra
Letra = "R"
MSComm1.Output = Letra
V = Chr(HScroll1.Value)
MSComm1.Output = V
End Sub
```

'Función del Botón Codo Derecha al hacer clic  
'Variable "Letra" como cadena  
'Variable "V" como cadena  
'Variable letra se le asigna valor de C  
'Envía por puerto de comunicación la variable Letra  
'Variable letra se le asigna valor de R  
'Envía por puerto de comunicación la variable Letra  
'variable V con valor del scroll convertido a cadena  
'Envía por puerto de comunicación la variable V

```
Private Sub Cod_Izq_Click(Index As Integer)
Dim Letra As String
Dim V As String
Letra = "C"
MSComm1.Output = Letra
Letra = "L"
MSComm1.Output = Letra
V = Chr(HScroll1.Value)
MSComm1.Output = V
End Sub
```

Función del Botón Codo Izquierda al hacer clic  
'Variable "Letra" como cadena  
'Variable "V" como cadena  
'Variable letra se le asigna valor de C  
'Envía por puerto de comunicación la variable Letra  
'Variable letra se le asigna valor de L  
'Envía por puerto de comunicación la variable Letra  
'variable V con valor del scroll convertido a cadena  
'Envía por puerto de comunicación la variable V

```

Private Sub Hom_Izq_Click(Index As Integer)
Dim Letra As String
Dim V As String
Letra = "B"
MSComm1.Output = Letra
Letra = "L"
MSComm1.Output = Letra
V = Chr(HScroll1.Value)
MSComm1.Output = V
End Sub

```

```

'Función del Botón Hombro Izquierda al hacer clic
'Variable "Letra" como cadena
'Variable "V" como cadena
'Variable letra se le asigna valor de B
'Envía por puerto de comunicación la variable Letra
'Variable letra se le asigna valor de L
'Envía por puerto de comunicación la variable Letra
'variable V con valor del scroll convertido a cadena
'Envía por puerto de comunicación la variable V

```

```

Private Sub Hom_Der_Click(Index As Integer)
Dim Letra As String
Dim V As String
Letra = "B"
MSComm1.Output = Letra
Letra = "R"
MSComm1.Output = Letra
V = Chr(HScroll1.Value)
MSComm1.Output = V
End Sub

```

```

'Función del Botón Hombro Derecha al hacer clic
'Variable "Letra" como cadena
'Variable "V" como cadena
'Variable letra se le asigna valor de B
'Envía por puerto de comunicación la variable Letra
'Variable letra se le asigna valor de R
'Envía por puerto de comunicación la variable Letra
'variable V con valor del scroll convertido a cadena
'Envía por puerto de comunicación la variable V

```

```

Private Sub Mun_Izq_Click(Index As Integer)
Dim Letra As String
Dim V As String
Letra = "D"
MSComm1.Output = Letra
Letra = "L"
MSComm1.Output = Letra
V = Chr(HScroll1.Value)
MSComm1.Output = V
End Sub

```

```

'Función del Botón Muñeca Izquierda al hacer clic
'Variable "Letra" como cadena
'Variable "V" como cadena
'Variable letra se le asigna valor de D
'Envía por puerto de comunicación la variable Letra
'Variable letra se le asigna valor de L
'Envía por puerto de comunicación la variable Letra
'variable V con valor del scroll convertido a cadena
'Envía por puerto de comunicación la variable V

```

```

Private Sub Mun_Der_Click(Index As Integer)
Dim Letra As String
Dim V As String
Letra = "D"
MSComm1.Output = Letra
Letra = "R"
MSComm1.Output = Letra
V = Chr(HScroll1.Value)
MSComm1.Output = V
End Sub

```

```

'Función del Botón Muñeca Derecha al hacer clic
'Variable "Letra" como cadena
'Variable "V" como cadena
'Variable letra se le asigna valor de D
'Envía por puerto de comunicación la variable Letra
'Variable letra se le asigna valor de R
'Envía por puerto de comunicación la variable Letra
'variable V con valor del scroll convertido a cadena
'Envía por puerto de comunicación la variable V

```

```

Private Sub Stop_Cad_Click()
Dim Letra As String
Dim V As String
Letra = "A"
MSComm1.Output = Letra
Letra = "S"
MSComm1.Output = Letra
V = Chr(HScroll1.Value)
MSComm1.Output = V
End Sub

```

```

'Función de Stop Cadera
'Variable "Letra" como cadena
'Variable "V" como cadena
'Variable letra se le asigna valor de A
'Envía por puerto de comunicación la variable Letra
'Variable letra se le asigna valor de S
'Envía por puerto de comunicación la variable Letra
'variable V con valor del scroll convertido a cadena
'Envía por puerto de comunicación la variable V

```

```

Private Sub Stop_Hombro_Click(Index As Integer)
Dim Letra As String
Dim V As String
Letra = "B"
MSComm1.Output = Letra
Letra = "S"
MSComm1.Output = Letra
V = Chr(HScroll1.Value)
MSComm1.Output = V
End Sub

```

```

'Función de Stop Hombro
'Variable "Letra" como cadena
'Variable "V" como cadena
'Variable letra se le asigna valor de B
'Envía por puerto de comunicación la variable Letra
'Variable letra se le asigna valor de S
'Envía por puerto de comunicación la variable Letra
'variable V con valor del scroll convertido a cadena
'Envía por puerto de comunicación la variable V

```

```

Private Sub Stop_Codo_Click()
Dim Letra As String
Dim V As String
Letra = "C"
MSComm1.Output = Letra
Letra = "S"
MSComm1.Output = Letra
V = Chr(HScroll1.Value)
MSComm1.Output = V
End Sub

```

```

'Función de Stop Codo
'Variable "Letra" como cadena
'Variable "V" como cadena
'Variable letra se le asigna valor de C
'Envía por puerto de comunicación la variable Letra
'Variable letra se le asigna valor de S
'Envía por puerto de comunicación la variable Letra
'variable V con valor del scroll convertido a cadena
'Envía por puerto de comunicación la variable V

```

```

Private Sub Stop_Muñeca_Click()
Dim Letra As String
Dim V As String
Letra = "D"
MSComm1.Output = Letra
Letra = "S"
MSComm1.Output = Letra
V = Chr(HScroll1.Value)
MSComm1.Output = V
End Sub

```

```

'Función de Stop Muñeca
'Variable "Letra" como cadena
'Variable "V" como cadena
'Variable letra se le asigna valor de D
'Envía por puerto de comunicación la variable Letra
'Variable letra se le asigna valor de S
'Envía por puerto de comunicación la variable Letra
'variable V con valor del scroll convertido a cadena
'Envía por puerto de comunicación la variable V

```

<pre> Private Sub Form_Load() With MSComm1  .CommPort = 3 .RThreshold = 1 .RTSEnable = True .Settings = "9600,n,8,1" .SThreshold = 1 .PortOpen = True End With End Sub  Private Sub Form_Unload(Cancel As Integer) MSComm1.PortOpen = False End Sub </pre>	<pre> 'Configuración del puerto de comunicaciones como MsComm1 'puerto serie com3 'espera 1 dato en buffer de entrada para analizarlo 'petición de envío habilitada 'velocidad, paridad, bits de información, bit de parada 'espera 1 dato en buffer de salida para analizarlo 'Habilitar puerto  'Cierra puerto de comunicación </pre>
--	---