Apéndice E

Código para operar los 2 sensores y manipular los 4 motores

```
// 4 motores

int pulso  = 4;
int act1 = 3;
int act2 = 5;
int act3 = 6;
int act4 = 9;

// valores de PWM
int e = 240;
int v= 0;
int vin=130;

// 4 motores a controlar
int a = 0;
int b = 0;
int c = 0;
int d = 0;

int ts = 10;
//--------------------------

// ultrasonico

int ultrasonico = 7;
long duracion, pulgadas, cm;      // se utilizan datos long debido a
la respuesta del ultrasonido
//--------------------------

// Nunchuk
#include <Wire.h>
uint8_t outbuf[6];           // array to store arduino output
int cnt = 0;

int pulseWidth = 0;    // Amount to pulse the servo
long lastPulse = 0;    // the time in millisecs of the last pulse
int refreshTime = 20;  // the time in millisecs needed in between
pulses
int minPulse = 700;   // minimum pulse width
int pulseWidth2 = 0;

#define pwbuffsize 4
int pwbuff[pwbuffsize]; // buffer for smoothing pulseWidths
int pwbuffpos = 0;      // position in pwbuff
int pwbuff2[pwbuffsize];
int pwbuffpos2 = 0;

int xx2 = 0;
int yy2 = 0;

//--------------------------
```

```
// inclinación y velocidad de motores
int xa = 0;
int xb = 0;
int xc = 0;
int xd = 0;

//--------------------------


void setup()
{

  pinMode(13, OUTPUT);  //utilizaremos el LED para saber que el
programa funciona
  beginSerial (19200);
  pinMode(pulso, OUTPUT);
  //Serial.print ("Inicio terminado\n");

  //////Nunchuk

  Wire.begin ();        // join i2c bus with address 0x52
  Nunchuk_init ();             // send the initilization handshake
  pulseWidth = minPulse;        // Set the motor position to the
minimum
  //////


}

void loop()
{
  digitalWrite(13, LOW);  //El Led está apagado
  digitalWrite(pulso,HIGH);


  // For de subida de velocidad
   for(v = vin; v <= e; v+=ts)

  {

    inclinacion();        //para obtener la inclinacion actual del
dispositivo
    aumentoVelocidad();


    a=v + xa;
    b=v + xb;
    c=v + xc;
    d=v + xd;

    analogWrite(act1, a);   // se le asigna un valor determinado a
cada motor
    analogWrite(act2, b);
    analogWrite(act3, c);
    analogWrite(act4, d);
    delay(500);                    // se espera un segundo para pasar
a una nueva velocidad

    char buf2[4];
```

```
    Serial.print (itoa(((a*100)/255), buf2, 10));
    Serial.print ("a");
    Serial.print (xa);
    Serial.print ("h");
    Serial.print (itoa(((b*100)/255), buf2, 10));
    Serial.print ("b");
    Serial.print (xb);
    Serial.print ("i");
    Serial.print (itoa(((c*100)/255), buf2, 10));
    Serial.print ("c");
    Serial.print (xc);
    Serial.print ("j");
    Serial.print (itoa(((d*100)/255), buf2, 10));
    Serial.print ("d");
    Serial.print (xd);
    Serial.print ("k");


     rstInclinacion();
     medirDistancia();
     delay(500);

  }


  ////////// for para estar en la velocidad máxima
  for(int q=1;q<=20;q++)
  {
    inclinacion();        //para obtener la inclinacion actual del
dispositivo
    aumentoVelocidad();

    a=v + xa;
    b=v + xb;
    c=v + xc;
    d=v + xd;

    analogWrite(act1, a);   // se le asigna un valor determinado a
cada motor
    analogWrite(act2, b);
    analogWrite(act3, c);
    analogWrite(act4, d);

    char buf2[4];

    Serial.print (itoa(((a*100)/255), buf2, 10));
    Serial.print ("a");
    Serial.print (xa);
    Serial.print ("h");
    Serial.print (itoa(((b*100)/255), buf2, 10));
    Serial.print ("b");
    Serial.print (xb);
    Serial.print ("i");
    Serial.print (itoa(((c*100)/255), buf2, 10));
    Serial.print ("c");
    Serial.print (xc);
    Serial.print ("j");
    Serial.print (itoa(((d*100)/255), buf2, 10));
    Serial.print ("d");
    Serial.print (xd);
    Serial.print ("k");
```

```
     rstInclinacion();
     medirDistancia();
    q++;
    delay(500);
  }

    /*

    digitalWrite(13, HIGH); //se enciende el  LED para indicar que
estamos en el máximo
    medirDistancia();
    inclinacion();
    delay(700);            //se espera 5 segundos a su velocidad
máxima
    medirDistancia();
    inclinacion();
    delay(700);            //se espera 5 segundos a su velocidad
máxima
    medirDistancia();
    inclinacion();
    delay(700);            //se espera 5 segundos a su velocidad
máxima
    medirDistancia();
    inclinacion();
    delay(700);            //se espera 5 segundos a su velocidad
máxima
    medirDistancia();
    inclinacion();
    delay(700);            //se espera 5 segundos a su velocidad
máxima
    medirDistancia();
    inclinacion();
    delay(700);            //se espera 5 segundos a su velocidad
máxima
    medirDistancia();
    inclinacion();
    delay(700);            //se espera 5 segundos a su velocidad
máxima
    medirDistancia();
    inclinacion();
    digitalWrite(13, LOW);  //apagar LED

    */

  // For de bajada de velocidad
  for(v = e; v >=vin; v-=ts)
  {
    inclinacion();        //para obtener la inclinacion actual del
dispositivo
    aumentoVelocidad();

    a=v + xa;
    b=v + xb;
    c=v + xc;
    d=v + xd;

    analogWrite(act1, a);
    analogWrite(act2, b);
    analogWrite(act3, c);
    analogWrite(act4, d);
```

```
    delay(500);

    char buf2[4];

    Serial.print (itoa(((a*100)/255), buf2, 10));
    Serial.print ("a");
    Serial.print (xa);
    Serial.print ("h");
    Serial.print (itoa(((b*100)/255), buf2, 10));
    Serial.print ("b");
    Serial.print (xb);
    Serial.print ("i");
    Serial.print (itoa(((c*100)/255), buf2, 10));
    Serial.print ("c");
    Serial.print (xc);
    Serial.print ("j");
    Serial.print (itoa(((d*100)/255), buf2, 10));
    Serial.print ("d");
    Serial.print (xd);
    Serial.print ("k");

    rstInclinacion();
    medirDistancia();
    delay(500);

  }

}



//_____________________________________________________________________________
_____________________________________________


long msPulgadas(long microsegundos)
{
  //De acuerdo a las hojas de datos se recorren 73.746 microsegundos
por pulgada
  //El sonido viaja a 1130 pies por segundo, así que para obtener una
distancia en
  //pulgadas solo es necesario dividir la señal que nos da entre 74
para obtener la
  //distancia total y finalmente dividir entre 2 para obtener la
distancia del sensor
  //al objeto
  return microsegundos / 74 / 2;
}

long msCentimetros(long microsegundos)
{
  //la velocidad del sonido es 340 m/s o 29 microsegundos por
centímetro.
  //El sonido llega hasta el objeto, rebota y regresa como eco por lo
que
  //solo es necesario tomar la distancia así que primero dividimos
entre 29
  //y posteriormente entre 2
```

```
  return microsegundos / 29 / 2;
}

void medirDistancia()
{
   //Se manda un pulso bajo de 5 microsegundos para asegurar que
siempre se inicie en bajo
  //después se manda un pulso alto de 15 microsegundos que sirve para
iniciar las mediciones

  pinMode(ultrasonico, OUTPUT);
  digitalWrite(ultrasonico, LOW);
  delayMicroseconds(5);
  digitalWrite(ultrasonico, HIGH);
  delayMicroseconds(15);
  digitalWrite(ultrasonico, LOW);


  //Se utiliza el mismo pin para recibir el eco así que lo cambiamos
de salida a entrada
  //y utilizamos la variable duracion para recibir el valor que nos da
el ultrasonico

  pinMode(ultrasonico, INPUT);
  duracion = pulseIn(ultrasonico, HIGH);

  // convertimos el tiempo que nos da el pulse in en distancia

  pulgadas = msPulgadas(duracion);
  cm = msCentimetros(duracion);

  //Serial.print(pulgadas);
  //Serial.print("in, ");
  Serial.print(cm);
  Serial.print("e");
  //Serial.println();

  delay(100);
}

////////////////////////////////
////////////////////////////////
// Nunchuk----------------------

void inclinacion()
{

        Wire.requestFrom (0x52, 6); // request data from Nunchuk
        while (Wire.available ())
          {
            // receive byte as an integer
             outbuf[cnt] = nunchuk_decode_byte (Wire.receive ());
             //digitalWrite (ledPin, HIGH);    // sets the LED on
             cnt++;
          }

        if (cnt >= 5)
        {
            ObtenerX();
        }
```

```
        if (cnt >= 5)
        {
          ObtenerY();
        }

        //moverMotor();
      cnt = 0;
      send_zero(); // send the request for next bytes

}

void moverMotor()
{
   a=xx2;
   Serial.print("Motor 1: ");  Serial.print(a);
   Serial.print("  \t");
   analogWrite(act1, a);

   b=yy2;
   Serial.print("Motor 2: ");  Serial.println(b);
   analogWrite(act2, b);

}

void ObtenerX()
{
   float tilt = outbuf[2];                    // z-axis, in this case
ranges from ~75 - ~185
          tilt = (tilt - 70) * 1.5;                  // convert to
degrees angle, approximately
          pulseWidth = (tilt * 9) + minPulse;        // convert angle
to microseconds
          pwbuff[pwbuffpos] = pulseWidth;            // save for
averaging
          if( ++pwbuffpos == pwbuffsize ) pwbuffpos = 0;
            pulseWidth=0;                              // reset so we
can
              for( int p=0; p<pwbuffsize; p++ )        // do the
smoothing
                  pulseWidth += pwbuff[p];             // sum up them
all
                  pulseWidth /= pwbuffsize;            // divide to
get average


            // aquí se asignan los datos del eje X
            // tilt2 es el dato que contiene los valores de
inclinación que da el
            //acelerómetro
            xx2=((tilt*100)/165);
            Serial.print(xx2);
            Serial.print("f");
}

void ObtenerY()
{


       float tilt2 = outbuf[3];            // z-axis, in this case
ranges from ~75 - ~185
```

```
          tilt2 = (tilt2 - 70) * 1.5;          // convert to degrees
angle, approximately
          pulseWidth2 = (tilt2 * 9) + minPulse; // convert angle to
microseconds
          pwbuff2[pwbuffpos2] = pulseWidth2;    // save for averaging

           if( ++pwbuffpos2 == pwbuffsize ) pwbuffpos2 = 0;
               pulseWidth2=0;                       // reset so we can
                 for( int p2=0; p2<pwbuffsize; p2++ )  // do the
smoothing
                     pulseWidth2 += pwbuff2[p2];         // sum up
them all
                     pulseWidth2 /= pwbuffsize;          // divide to
get average

      // aquí se asignan los datos del eje Y
      // tilt2 es el dato que contiene los valores de inclinación que
da el
      //acelerómetro
      yy2=((tilt2*100)/165);
      //Serial.print("\ty%: ");
      Serial.print(yy2);
      Serial.print("g");
}

void Nunchuk_init()
{
  Wire.beginTransmission (0x52);   // transmit to device 0x52
  Wire.send (0x40);           // sends memory address
  Wire.send (0x00);           // sends sent a zero.
  Wire.endTransmission ();    // stop transmitting
}

void send_zero()
{
  Wire.beginTransmission (0x52);   // transmit to device 0x52
  Wire.send (0x00);           // sends one byte
  Wire.endTransmission ();    // stop transmitting
}

char nunchuk_decode_byte (char x)
{
  x = (x ^ 0x17) + 0x17;
  return x;
}

void aumentoVelocidad()
{
  int x,y;

  x=xx2;
  y=yy2;

  if (x>53)
  {
    if(x==54)
        xa=1;
    if(x==55)
        xa=3;
    if(x==56)
        xa=4;
```

```
  if(x==57)
      xa=6;
  if(x==58)
      xa=7;
  if(x==59)
      xa=8;
  if(x==60)
      xa=10;
  if(x==61)
      xa=11;
  if(x==62)
      xa=12;
  if(x==63)
      xa=14;
  if(x>64)
      xa=0;
 }

 if (x<47)
 {
  if(x==46)
      xb=1;
  if(x==45)
      xb=3;
  if(x==44)
      xb=4;
  if(x==43)
      xb=6;
  if(x==42)
      xb=7;
  if(x==41)
      xb=8;
  if(x==40)
      xb=10;
  if(x==39)
      xb=11;
  if(x==38)
      xb=12;
  if(x==37)
      xb=14;
  if(x==36)
      xb=15;
  if(x<36)
      xb=0;
 }

if (y>53)
 {
  if(y==54)
      xc=1;
  if(y==55)
      xc=3;
  if(y==56)
      xc=4;
  if(y==57)
      xc=6;
  if(y==58)
      xc=7;
  if(y==59)
      xc=8;
  if(y==60)
```

```
        xc=10;
    if(y==61)
        xc=11;
    if(y==62)
        xc=12;
    if(y==63)
        xc=14;
    if(y==64)
        xc=15;
    if(y>64)
        xc=0;
  }

  if (y<47)
  {
    if(y==46)
        xd=1;
    if(y==45)
        xd=3;
    if(y==44)
        xd=4;
    if(y==43)
        xd=6;
    if(y==42)
        xd=7;
    if(y==41)
        xd=8;
    if(y==40)
        xd=10;
    if(y==39)
        xd=11;
    if(y==38)
        xd=12;
    if(y==37)
        xd=14;
    if(y==36)
        xd=15;
    if(y<36)
        xd=0;
  }

}

void rstInclinacion()
{
  xa=0;
  xb=0;
  xc=0;
  xd=0;
}
```