Apéndice D

Código para el aumento de velocidad en los motores.

```
///// para controlar el Nunchuk
#include <Wire.h>

uint8_t outbuf[6];           // array to store arduino output
int cnt = 0;
//int ledPin = 13;

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 t = 0;  // when it gets to 25, read Nunchuk

////

/////// para mover motores
int pulso = 4;
int act1 = 3;
int act2 = 5;
int act3 = 6;
int act4 = 11;

// valores de PWM
int e = 150;
int v= 0;

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

int xx  = 0;
```

```
int xx2 = 0;
int yy  = 0;
int yy2 = 0;

int xa = 0;
int xb = 0;
int xc = 0;
int xd = 0;

///////


void setup()
{
  beginSerial (19200);
  Serial.print ("Finished setup\n");
  Wire.begin ();        // join i2c bus with address 0x52
  Nunchuk_init ();             // send the initilization
handshake
  //pinMode(servoPin, OUTPUT);    // Set servo pin as an output
pin
  pulseWidth = minPulse;        // Set the motor position to the
minimum

}



void loop()
{
  // when it gets to 25, read Nunchuk
  t++;
  if( t == 25 )
  {

    inclinacion();
    aumentoVelocidad();
    Serial.println();
    Serial.print("xa: ");  Serial.print(xa);
    Serial.print("\txb: ");  Serial.print(xb);
    Serial.print("\txc: ");  Serial.print(xc);
    Serial.print("\txd: ");  Serial.println(xd);
    rstInclinacion();

  }

  delay(20);

}


void inclinacion()
{
   t = 0;
        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("\tx%: ");   Serial.print(xx2);
```

```
}

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("\n");
}

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;
}
```