```
1
2  // Controlling a servo position using a potentiometer (variable resistor)
3  // by Michal Rinott <http://people.interaction-ivrea.it/m.rinott>
4
5  #include <Servo.h>
6  #include <Wire.h>
7  #include <LiquidCrystal_I2C.h>
8
9  LiquidCrystal_I2C lcd(0x27,16,2);  // set the LCD address to 0x27 for a 16 chars and 2 line display
10 const int buttonOK=8;
11 const int buttonPLUS=7;  // the number of the pushbutton pin
12 const int buttonMINUS=4;
13 const int buttonINTERRUPT=2;
14 volatile int a; // flag for start a function
15 boolean changeData; //disable timer2 interrupt if false
16 int choiceProgram;  //number of program
17
18 Servo servo0; // create servo object to control a servo 0 is for thumb
19 Servo servo1; // 1 is for the index
20 Servo servo2; // 2 is for the middle finger
21 Servo servo3; // 3 is for the ring
22 Servo servo4; // 4 is for the little
23
24 int potpin_servo[5]; // analog entry pin connected to the flex sensor
25 int tmp;    // working variable
26 int time; //wanted delay in ms
27 int in[5];  // memorize the position in vector positions where we write the measured value for each servo
28 int out[5]; // memorize the position in vector positions where we read the measured value for each servo
29 int high[7]; // memorize for each finger the highest value of resistance (when hand is closed)
30 int low[7];  // memorize for each finger the lowest value of resistance (when hand is open)
31 int positions[5][100]; // delay max = 1000ms and interrupt occur every 10ms => 100 positions max
32
33
34
35 void setup()
36 {
37   servo0.attach(3);  // attaches the servo on pin 3 to the servo object
38   servo1.attach(5);
39   servo2.attach(6);
40   servo3.attach(9);
41   servo4.attach(10);
42   lcd.init();                      // initialize the lcd
43   lcd.backlight(); // set the back light on
44   time=0; //initial delay
45   choiceProgram=1; // intial program
46   a=1; // flag for the change data function
47    for (int i=0; i<5; i++)
48    {
49       in[i]=0;  // intitialization of vector containing position of end and beginning
50       out[i]=0;
51    }
52
53   pinMode(buttonOK, INPUT_PULLUP);
54   pinMode(buttonPLUS, INPUT_PULLUP);
55   pinMode(buttonMINUS, INPUT_PULLUP);
56   pinMode(buttonINTERRUPT, INPUT_PULLUP);
57   ServoInit(); //Set an initial position for servo
58   Init_data (); // Communication with user for choice of parameter
59
60   interrupts();
61   attachInterrupt(0,InterruptChange, LOW); //interrupt linked with button to change the data when program
is running
62   changeData=false;
63   Timer2init();
64 } ;
65
```

```
66  void ServoInit()
67  {
68    servo0.write(90);
69    servo1.write(90);
70    servo2.write(90);
71    servo3.write(90);
72    servo4.write(90);
73
74  }
75  void Init_data ()
76  {
77    lcd.clear();
78    lcd.print("Welcome");
79    delay (3000);
80    lcd.clear();
81    lcd.print("Open your hand");
82    lcd.setCursor(0,1);
83    lcd.print("Press OK");
84    while (digitalRead(buttonOK) == HIGH)
85    {
86       //wait for pushbutton is pressed
87    }
88    lcd.clear();
89    for (int i=0; i<=6; i++)
90       {
91           high[i]=0;
92           if(i!=4 && i!=5) //no sensor on pine 4 and 5
93           {
94               high[i]=analogRead(i); // memorize highest values  of resistance for each flex sensor (in
order to use it to scale movement and fit it better)
95           }
96
97       }
98
99    delay(1000); //wait a second : debouncing
100   lcd.print("Close your hand");
101   lcd.setCursor(0,1);
102   lcd.print("Press OK");
103   while (digitalRead(buttonOK) == HIGH)
104   {
105      // wait for the pushbutton is pressed.
106   }
107   lcd.clear();
108   for(int i=0; i<=6; i++)
109      {
110        low[i]=0;
111          if(i!=4 && i!=5)
112          {
113              low[i]=analogRead(i);// memorize lowest values  of resistance for each flex sensor
114          }
115
116       }
117
118   delay(1000);
119   lcd.print ("Delay = ");
120   lcd.setCursor(0,1); // begin to write on row 0, line 1
121   lcd.print(time);
122   lcd.print( " ms");
123
124   while (digitalRead(buttonOK) == HIGH)
125   {
126     delay(300);
127
128     if (digitalRead(buttonPLUS) == LOW) // button pressed = LOW because 1k in pull up
129       {
130           time+=10;
```

```
            lcd.clear(); // Refresh screen with actual value
            lcd.print ("Delay = ");
            lcd.setCursor(0,1); // begin to write on row 0, line 1
            lcd.print(time);
            lcd.print( " ms");
           if(time> 1000) // delay max
            { time=1000;}
         }
        if (digitalRead(buttonMINUS) == LOW) // button pressed = LOW because 1k in pull up
        {
            time-=10;
            lcd.clear(); // Refresh screen with actual value
            lcd.print ("Delay = ");
            lcd.setCursor(0,1); // begin to write on row 0, line 1
            lcd.print(time);
            lcd.print( " ms");
             if(time<0) // mininum delay
            { time=0;}
         }
      }

      lcd.clear();
      delay(1000);
      lcd.print ("Program : ");
      lcd.setCursor(0,1); // begin to write on row 0, line 1
      lcd.print(choiceProgram);
      while (digitalRead(buttonOK) == HIGH)
    {
      delay(300);

       if (digitalRead(buttonPLUS) == LOW) // button pressed = LOW because 1k in pull up
           {
                 choiceProgram+=1;
                  lcd.clear();
                  lcd.print ("Program : ");
                  lcd.setCursor(0,1); // begin to write on row 0, line 1
                  lcd.print(choiceProgram);
                 if (choiceProgram > 3) // Program max
                 { choiceProgram=3;}
           }
        if (digitalRead(buttonMINUS) == LOW) // button pressed = LOW because 1k in pull up
             {
                 choiceProgram-=1;
                  lcd.clear();
                  lcd.print ("Program : ");
                  lcd.setCursor(0,1); // begin to write on row 0, line 1
                  lcd.print(choiceProgram);
                 if(choiceProgram < 1)
                 { choiceProgram=1;}
             }
      }

      lcd.clear();        // resume of parameters for the user
      lcd.print ("Delay = ");
      lcd.print(time);
      lcd.print( " ms");
      lcd.setCursor(0,1); // begin to write on row 0, line 1
      lcd.print ("Program : ");
      lcd.print(choiceProgram);
      Program(); // Set parameter for selected program
      a=1; // put flag back to normal value
}

void Change_data ()
{
  changeData=true; // avoid reseting the interrupt=> stop it
```

```
197    lcd.clear();
198    lcd.print ("Delay = ");
199    lcd.setCursor(0,1); // begin to write on row 0, line 1
200    lcd.print(time);
201    lcd.print( " ms");
202
203   while (digitalRead(buttonOK) == HIGH)
204   {
205     delay(300);
206
207     if (digitalRead(buttonPLUS) == LOW) // button pressed = LOW because 1k in pull up
208       {
209           time+=10;
210           lcd.clear(); // Refresh screen with actual value
211           lcd.print ("Delay = ");
212           lcd.setCursor(0,1); // begin to write on row 0, line 1
213           lcd.print(time);
214           lcd.print( " ms");
215          if(time> 1000) // delay max
216           { time=1000;}
217        }
218       if (digitalRead(buttonMINUS) == LOW) // button pressed = LOW because 1k in pull up
219       {
220           time-=10;
221           lcd.clear(); // Refresh screen with actual value
222           lcd.print ("Delay = ");
223           lcd.setCursor(0,1); // begin to write on row 0, line 1
224           lcd.print(time);
225           lcd.print( " ms");
226            if(time<0) // mininum delay
227           { time=0;}
228        }
229     }
230
231     lcd.clear();
232     delay(1000);
233     lcd.print ("Program : ");
234     lcd.setCursor(0,1); // begin to write on row 0, line 1
235     lcd.print(choiceProgram);
236     while (digitalRead(buttonOK) == HIGH)
237   {
238     delay(300);
239
240      if (digitalRead(buttonPLUS) == LOW) // button pressed = LOW because 1k in pull up
241          {
242                choiceProgram+=1;
243                 lcd.clear();
244                 lcd.print ("Program : ");
245                 lcd.setCursor(0,1); // begin to write on row 0, line 1
246                 lcd.print(choiceProgram);
247                if (choiceProgram > 3) // Program max
248                { choiceProgram=3;}
249           }
250       if (digitalRead(buttonMINUS) == LOW) // button pressed = LOW because 1k in pull up
251            {
252                choiceProgram-=1;
253                 lcd.clear();
254                 lcd.print ("Program : ");
255                 lcd.setCursor(0,1); // begin to write on row 0, line 1
256                 lcd.print(choiceProgram);
257                if(choiceProgram < 1)
258                { choiceProgram=1;}
259           }
260     }
261
262     lcd.clear();        // resume of parameters for the user
```

```
    lcd.print ("Delay = ");
    lcd.print(time);
    lcd.print( " ms");
    lcd.setCursor(0,1); // begin to write on row 0, line 1
    lcd.print ("Program : ");
    lcd.print(choiceProgram);
    Program(); // Set of
    changeData=false, //a
    TCNT2 = 99;     // reset timer ct to 99 out of 255
    TIFR2 = 0x00;
    a=1; //  put flag back to normal value
}

void InterruptChange ()
{
a=2; // change the flag

}

void Timer2init()
{
    // Setup Timer2 overflow to fire every 8ms (125Hz)
    //   period [sec] = (1 / f_clock [sec]) * prescale * (255-count)
    //                  (1/16000000)  * 1024 * (255-99) = .00998 sec = 10ms

    TCCR2B = 0x00;        // Disable Timer2 while we set it up
    TCNT2  = 99;         // Reset Timer Count  (255-99) = execute ev 125-th T/C clock
    TIFR2  = 0x00;        // Timer2 INT Flag Reg: Clear Timer Overflow Flag
    TIMSK2 = 0x01;        // Timer2 INT Reg: Timer2 Overflow Interrupt Enable
    TCCR2A = 0x00;        // Timer2 Control Reg A: Wave Gen Mode normal
    TCCR2B = 0x07;        // Timer2 Control Reg B: Timer Prescaler set to 1024
};
void Program()
{
    if (choiceProgram==1)// 1st program : normal mode;
    {
        for (int i=0; i<5;i++)
        {
            potpin_servo[i]=i;
            if(i==4)
            {
                potpin_servo[i]=6;
            }
        }
    }
     if (choiceProgram==2)// 2nd program : only the index mode
    {
         for (int i=0; i<5;i++)
        {
            potpin_servo[i]=9; //disable movement
            if(i==1)
            {
                potpin_servo[i]=1;
            }
        }
    }
     if (choiceProgram==3)// 3rd program : random mode
    {
        int j=random(5); // Random from 0 to 4
        potpin_servo[0]=j;
         for (int i=1; i<5;i++)
        {
            potpin_servo[i]=((j+i)%5);
            if(potpin_servo[i]==4)
            {
                potpin_servo[i]=6;
```

```
            }
          }
      }
}

 void write_position_on_servo(int number_of_servo, Servo servo)
{
   tmp=positions[number_of_servo][out[number_of_servo]]; //read first value in vector containing data from flex sensor
   if(number_of_servo==0 || number_of_servo==2 || number_of_servo==4 )
      {tmp= map(tmp, 30, 160, 160, 30);} // invert movement of servo
   servo.write(tmp);
   out[number_of_servo]=out[number_of_servo]+1; //increment number that memorize where we must read
    if (out[number_of_servo]>=100)  //if end of array reached, go to the beginning
      {
        out[number_of_servo]=0;
      }
};

void write_position_from_finger(int number_of_servo, int potpin_finger)
{
   if(potpin_finger==9) // If 9, servo is disabled
   {
     tmp=30;
   }
   else
   {
      tmp=analogRead(potpin_finger); // read the value of flex
      tmp>>1; // Bit shift to neglect last value (error of measure)
      tmp= map(tmp, low[potpin_finger], high[potpin_finger], 30, 160);
   }
    if (tmp<30) // protection of the servo if analog input signal defective
    {
    tmp=30;
    }
    if(tmp>160)
    {
      tmp=160;
    }
   positions[number_of_servo][in[number_of_servo]]=tmp;
   in[number_of_servo]=in[number_of_servo]+1; //increment number that memorize where we must stock data
   if (in[number_of_servo]>=100)  //if end of array reached, go to the beginning
      {in[number_of_servo]=0;}

};



ISR(TIMER2_OVF_vect)
{

  SREG=SREG | B10000000; // set bit 7 to 1 (global interrupt disabled)

  write_position_from_finger(0,potpin_servo[0]);
  write_position_from_finger(1,potpin_servo[1]);
  write_position_from_finger(2,potpin_servo[2]);
  write_position_from_finger(3,potpin_servo[3]);
  write_position_from_finger(4,potpin_servo[4]);

  if((in[0]-out[0])>=(time/10) || (out[0]-in[0])>=(99-(time/10)-1)) // If we wait enougth
  {

    write_position_on_servo(0, servo0);
    write_position_on_servo(1, servo1);
    write_position_on_servo(2, servo2);
    write_position_on_servo(3, servo3);
```

```
394       write_position_on_servo(4, servo4);
395     }
396
397     if (!changeData) // if the user is changing variables, stop the interrupt (it will be started at the end
of the aquisition of data from user)
398      {
399        TCNT2 = 99;      // reset timer ct to 99 out of 255
400        TIFR2 = 0x00;     // timer2 int flag reg: clear timer overflow flag
401      }
402      else // reinitialise postion of writing and reading in vector (in time is smaller after changing data
for example)
403      {
404         for (int i=0; i<5; i++)
405          {
406             in[i]=0;
407             out[i]=0;
408          }
409      }
410  };
411
412
413
414
415
416  void loop()
417  {
418
419   if(a==2) // if the user press the interrupt button, we change data
420    {
421
422    Change_data ();
423    Program();
424
425
426    }
427
428    }
429
430
```