ROBT 206 Microcontrollers with Lab

ROBOHAND

Adambek Aishabibi

Khuzyakhmetova Assiya
Sabyrov Arman
Suleimenov Aidarbek
Tuleuov Adilet

Project aims

- Building *the simulation* of a basic motion of a human hand
- Reach the basic functionality: *contraction of the fingers* and *the movement of a thumb in a plane around the wrist*;
- Implementation of a combinational logic through Arduino IDE

Design

Specifications - Formulation - Optimization - Circuit Design - Verification

Following the recommendations proposed by Mano and Kime in "Logic and Computer Design Fundamentals" (2014) we structure the design process in **the five steps** and also integrate **the details on a process of building a hand**

SPECIFICATIONS

RoboHand - a mechanical programmable arm that functions similar to a human arm.

Human arm motion is detected with help of *flex sensors* attached to the glove worn by an a tester (the glove provided by a Lab Instructor).

RoboHand components

- Foam;

- Fishline;

- Wires;

- Flex sensors;

- Servo motors;

- 12k ohms resistors.

- Breadboard.

The glove with flex sensors that detects the motion of a human hand

FORMULATION

- Boolean Equations and State Tables - *not feasible* to derive as wide range of analogous input from flex sensors;
- The alternative technique: *mapping* the analogous inputs to the ranges of input values for servo motors.

A servo motor is a device that turns an electrical signal into a rotary movement that can be precisely controlled.

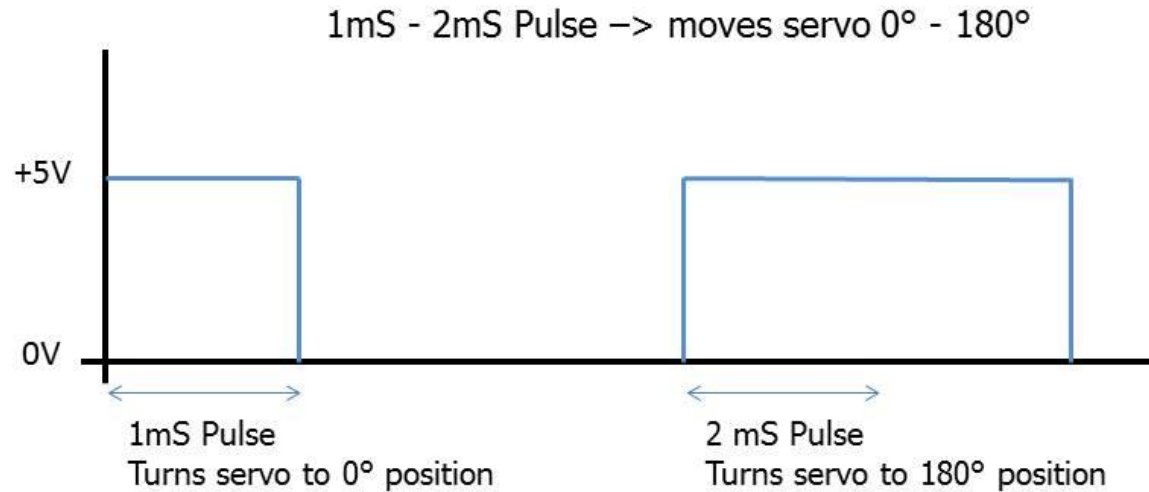**The challenge**: to write the correctly functioning software.

Figure 1. PWM for Servo Motors. Source: "How To Control A Servo Using Pulse Width Modulation (PWM) - Smartmicrocontroller.Com" 2018.

Optimization

Encountered issues:
- Noisiness of flex sensor readings
- Lack of flex sensors

Solutions:
- Median filtering of data readings
- 3 fingers

```cpp
1   #include <newFilter.h>
2   #include <iirFilter.h>
3   #include <firFilter.h>
4   #include <medianFilter.h>
5   #include <Servo.h>
6
7   medianFilter Filter1, Filter2, Filter3;
8
9   // Constants:
10  const int flexPin1 = A0; //pin A0 to read analog inputasd
11  const int flexPin2 = A1; //pin A1 to read analog inputasd
12  const int flexPin3 = A2; //pin A2 to read analog inputasd
13  Servo myservo1, myservo2,myservo3;  // create servo objects to control a servo
14  // Variables:
15  int value1 = 0, value2 = 0, value3 = 0; //save analog value
16
17  void setup(){
18      Serial.begin(9600);//Begin serial communication
19      // Start filtering instances
20      Filter1.begin();
21      Filter2.begin();
22      Filter3.begin();
23      myservo1.attach(7);
24      myservo2.attach(8);
25      myservo3.attach(9);
26  }
27  void loop(){
28      /* Read and save analog values from flex sensors */
29      value1 = analogRead(flexPin1);
30      value2 = analogRead(flexPin2);
31      value3 = analogRead(flexPin3);
32      /* Process values through filters */
33      value1 = Filter1.run(value1);
34      value2 = Filter2.run(value2);
35      value3 = Filter3.run(value3);
36      /* Map sensor values so that they could move servos */
37      int val1 = map(value1, 670, 800, 0, 180);
38      val1 = constrain(val1, 0, 180);
39      int val2 =  map(value2, 500, 1023, 180, 0);
40      val2 = constrain(val2, 0, 180);
41      int val3 =  map(value3, 880, 920, 0, 180);
42      val3 = constrain(val3, 0, 180);
43      /*
44        Print data to the serial in the proper format,
45        so that we could read data to Blender
46      */
47      Serial.print(val1);
48      Serial.print(" ");
49      Serial.print(val2);
50      Serial.print(" ");
51      Serial.print(val3);
52      Serial.print("\n");
53      /* Tell servos to move */
54      myservo3.write(val3);
55      myservo2.write(val2);
56      myservo1.write(val1);
57      delay(100); //Small delay
58  }
```
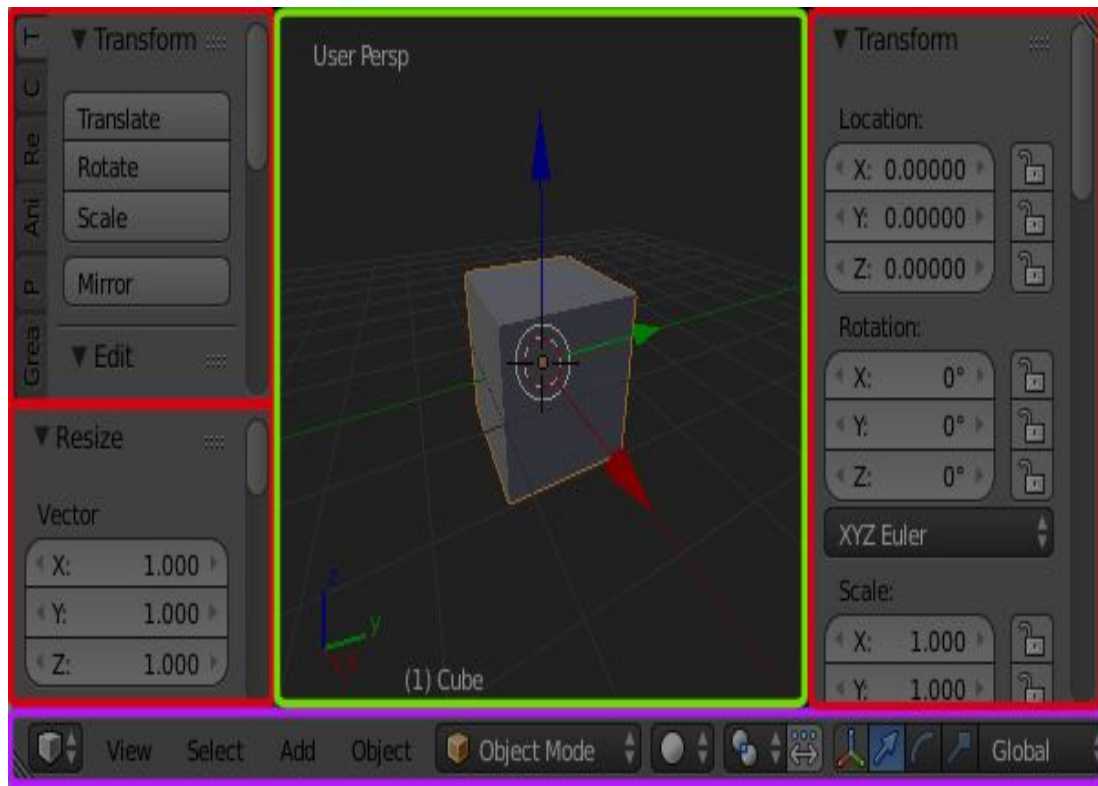
Circuit Design
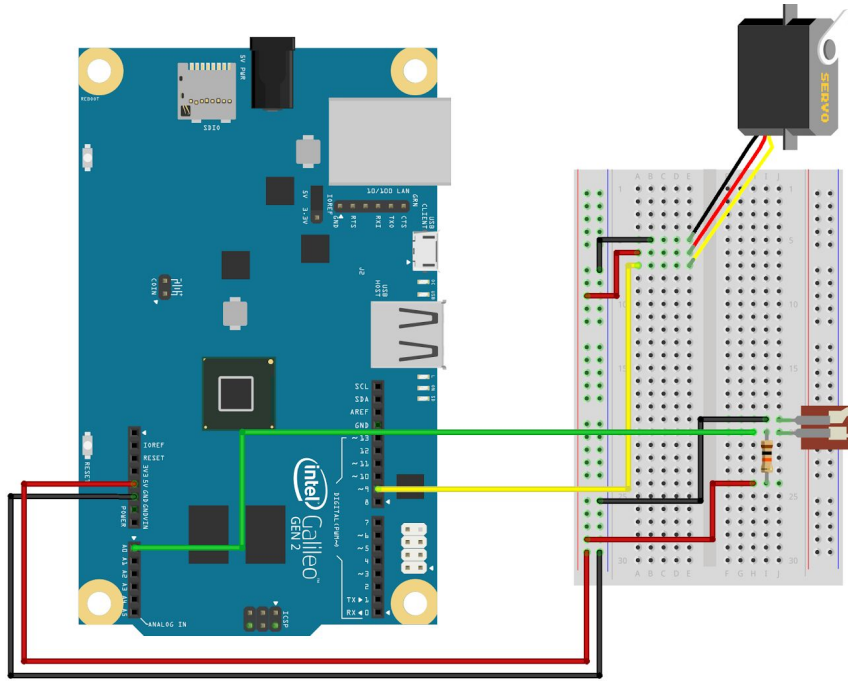
- Basic *combinational logic* principles;
- Readings from the flex sensors = analogous inputs;
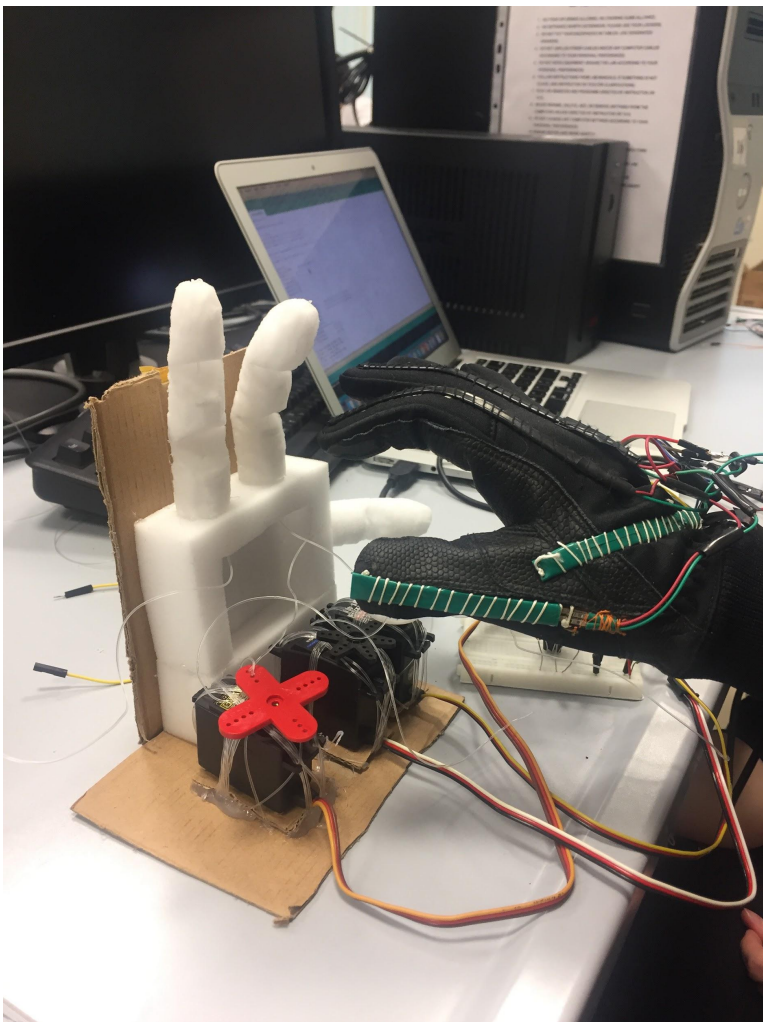- Servo motors connected to the digital pins of the Arduino Board;

In the actual circuit design, two more flex sensors were added to read in the motion of **3 fingers**.

fritzing

Blender

Simulation

Free and Open 3D Creation

Software

Verification

And watch our demo!

https://youtu.be/pKi1RAyJarI

References

Collins, Danielle. 2018. "Notch Filters And Low-Pass Filters For Servo System Resonance". *Motioncontroltips.Com*.
 https://www.motioncontroltips.com/notch-filters-low-pass-filters-reduce-resonance-servo-systems/.

 "How To Control A Servo Using Pulse Width Modulation (PWM) - Smartmicrocontroller.Com". 2018.
 Smartmicrocontroller.Com.
 http://smartmicrocontroller.com/how-to-control-a-servo-using-pulse-width-modulation-pwm/.

"Galileo Experiment Guide - Learn.Sparkfun.Com". 2018. *Learn.Sparkfun.Com*.
 https://learn.sparkfun.com/tutorials/galileo-experiment-guide/sik-galileo---part-9-using-a-flex-sensor.

Zhu, Youlian, and Cheng Huang. 2012. "An Improved Median Filtering Algorithm For Image Noise Reduction". *Physics Procedia* 25: 609-616. doi:10.1016/j.phpro.2012.03.133.