ROBT 206 Microcontrollers with Lab

# ROBOHAND - Project Report

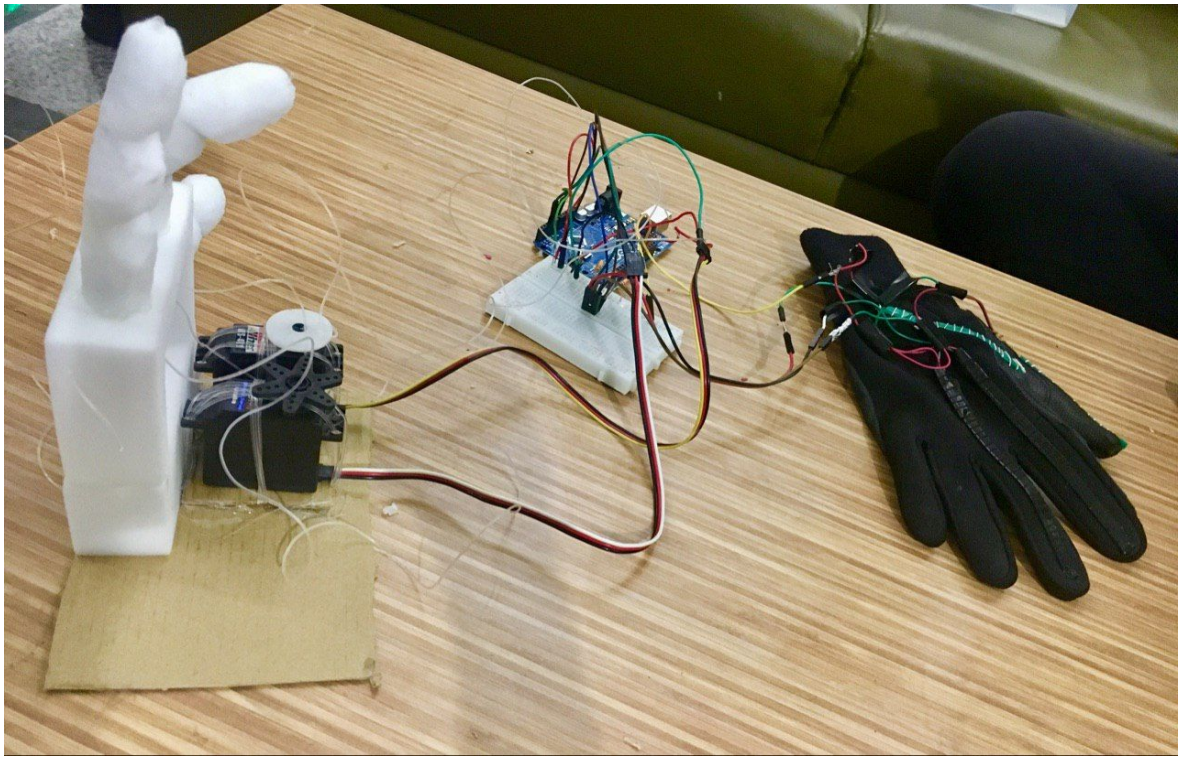


**Figure 1.** RoboHand simulation of a human hand. Source: our own photographs, 2018.

## 1. Introduction

This project was aimed at building the simulation of a basic motion of a human hand - namely, contraction of the fingers and the movement of a thumb in a plane around the wrist. Due to the limited hardware supplied, the initial plan of simulating a hand with all five fingers was realized with three fingers only - a thumb, a pointing finger and a middle finger. During the implementation stage, the problem with the overly noisy servo motors was discovered. The reason behind such behavior was the high resonance of servo motors resulting from the overly excited natural frequencies of the analog input from flex sensors' bending (Collins 2018). The Median Filtering was used to reduce these resonances (Zhu and Huang. 2012). The RoboHand was built from styrofoam with fishline used to reproduce a

muscular system. Arduino board and Arduino ToolKit were used to implement the concepts described above. Flex sensors were used to detect the changes in fingers' position during a hand motion. We also integrated Blender software in our project, that was used to visualize motion of fingers using simple 3D shapes. Thus, the importance of the project lies in the ability of its developers to integrate the different technologies and real human experience (i.e. human motion) in one project with a proper care given to the minimization of deviations from the initial visions of a project. This is in agreement with a broader field of human motion simulation goals that places a high value to the combination of artificial and natural with focus on avoidance of deviations. Moreover, our project is significant because it shows that socially impactful concepts of different complexity can be developed even with the limited finances and equipment. As RoboHand was able to simulate the motion as expected, the project's goals were met.

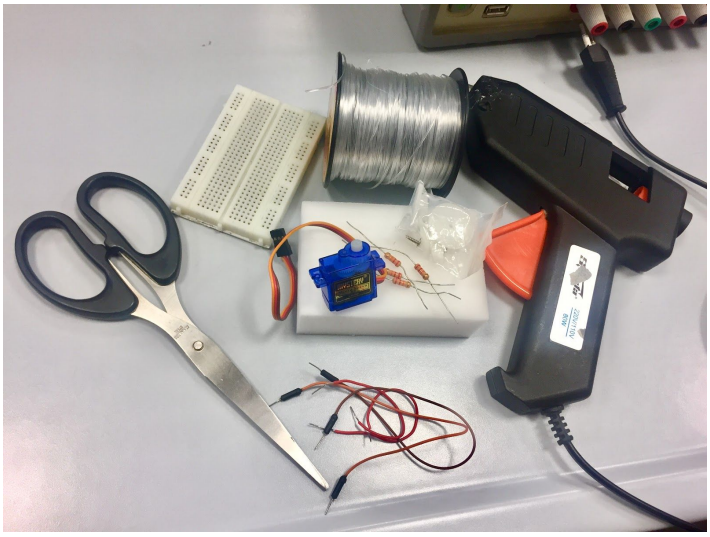
# 2. Materials and Methods

To bring a coherent and comprehensible image of the RoboHand implementation we follow the recommendations proposed by Mano and Kime in "Logic and Computer Design Fundamentals" (2014). Additionally, we include mechanical details of building a RoboHand. Thus, we structure the design process in following steps:

### - Specifications

RoboHand is the simplest version of a Robotic Hand - a mechanical programmable arm that functions similar to a human arm. The functionality of RoboHand are: 1) the fastening of the middle and pointing finder, 2) the rotation of a thumb around a wrist in a plane. The main components of a RoboHand implementation are:

- *Foam;*
- *Fishline;*
- *Wires;*
- *Flex sensors;*
- *Servo motors;*
- *22k ohms resistors.*
- *Breadboard.*



**(a)** **(b)**

**Figure 2.** The core materials a RoboHand was built from: (a) the main components of RoboHand (see the bulleted-point list above); (b) the glove with flex sensors, provided by a Lab Instructor. Source: our own photographs, 2018.

RoboHand itself was made of a foam with muscular motion emulated with help of flex sensors and 1-mm thick fishline. As flex sensors measure the amount of deflection or bending, they were built into the glove to imitate a hand motion. That is, a person wears a glove, moves her fingers, and flex sensors read in this movement to pass it further to servo motors through Arduino board. Servo motors, in their turn, control the movement of a mechanical hand through fishline - they move in response to the input and, consequently, move the fishline. As the signal passes through Arduino board, Arduino ToolKit with Blender's modeling tools are used to map the analog inputs and implement logic required to simulate a real hand motion.

- **Formulation**

Boolean Equations and State Tables were found to be not feasible to derive in our project due to the wide range of analogous input from flex sensor. That is, to derive boolean

equations for this implementation would require to have a separate boolean variable for each 1-step change in inputs, resulting in truth tables of exponentially growing sizes.

Instead, a common in Arduino ToolKit technique for reading analogous input was used. This technique is to map the analogous inputs to the ranges that are acceptable as input values for our devices (in our project - servo motors). As the combinational logic itself was not that challenging, the burden of a working implementation was put on the correctly functioning software.

We have used the PWM (Pulse Width Modulation) to map the analogous readings from flex sensors to the digital inputs of the servo motors. The principle is as follows: in order to simulate the continuous motion of servo motors the signal from the arduino (either HIGH or LOW) is passed to the digital pins for a specific time intervals - we took a time interval equal to 100 ms. During the pulse the servo motor rotates either towards 180 degrees or toward 0 degrees.

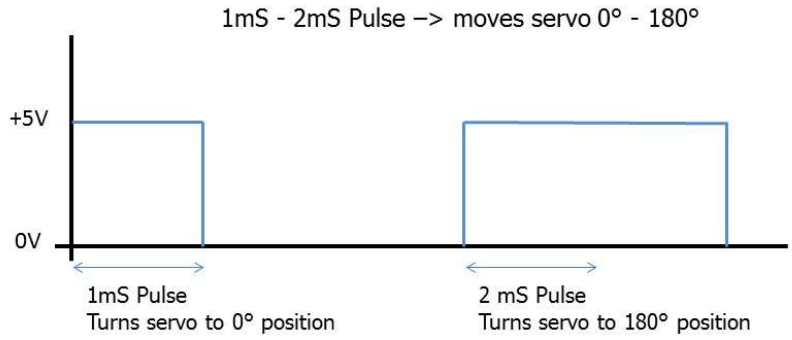



**Figure 3.** PWM for Servo Motors. Source: "How To Control A Servo Using Pulse Width Modulation (PWM) - Smartmicrocontroller.Com" 2018.

- **Circuit Design**

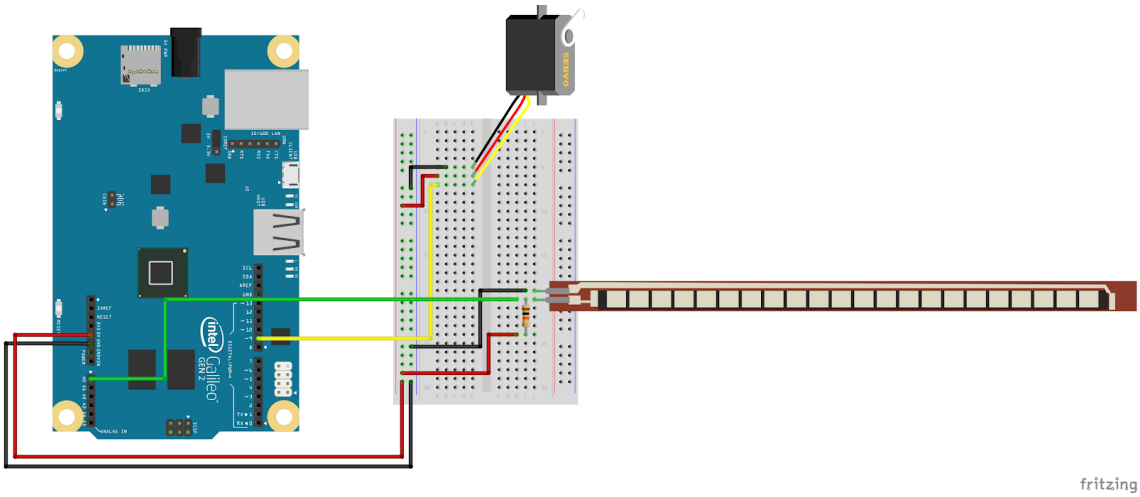



**Figure 4.** Flex sensor as input to the servo motor. Source: ("Galileo Experiment Guide - Learn.Sparkfun.Com" 2018).s

The circuit reflected the basic combinational logic principles. The readings from the flex sensors were passed as analogous inputs to the Arduino Board. The input wires of the servo motors were connected to digital pins of the Arduino Board with proper care given to the grounding and voltage supply. These pins were the actual outputs to which the mapping results were passed in Arduino ToolKit code. This logic can be observed in Figure 2. In the actual circuit design, two more flex sensors were added to read in the motion of 3 fingers, overall.

# 3. Results

We have repeatedly simulated the motion of the hand in the glove and have gained the required results on our prototype - that is, RoboHand managed to imitate a real hand motion. The link for the video with a simulation and process details: https://www.youtube.com/watch?v=4xDLHAeh5DE&feature=youtu.be. Besides, we have arrived at the following results.

Our initial plan required 5 flex sensors to imitate the motion of fingers of the hand. However, after proposals' review we cut off our implementation to 3 fingers with 3 flex

sensors that are now connected to the glove. Initially, the plan was to build the flex sensors on our own with usage of aluminium tape and wires. However, , the glove with motion-reading flex sensors was provided by a lab instructor - Zhanat Kapassov.

Secondly, we encountered the problem of noise in data readings from flex sensors. In order to solve this problem we used filter methods. We have considered two options of dealing with an issue: 1) integrating additional hardware such as capacitors to regulate the range of values the flex sensors can read, and 2) integrating the data readings filtering through a software in Arduino IDE. We have decided to work with a software because it provided a better control of a process as the control mechanisms of the whole RoboHand were defined and maintained through Arduino IDE. After working for a while with building our own filters, we encountered limited success as we got rid of part of the noise. However, during the verification process, when compared to the tutorials and instructions available online (on Arduino forums and YouTube channels), we have understood that our filtering has not been working as reliably as required, Thus, we decided to include a special pre-built Arduino filtering library developped by an experienced Arduino engineer (https://github.com/jeroendoggen/Arduino-signal-filtering-library/blob/master/README.md). Specifically, out of all methods available in a library, we used Median Filtering, which is a nonlinear digital filtering technique to remove extra noise from the signal (Zhu and Huang 2012). Such choice is based on empirical results. We have tested variety of filters and the most suitable is found to be the Median Filtering with more methods in its implementation that suit the project's needs (such as allowing for a better flow of a signal to simulate the motion in real time).

Finally, the flaw in the flex sensor of a middle finger and the inaccuracy of one of the 3 servo motors were discovered. We suspect the flex sensor functions improperly because of the wearing out. The servo motors' abnormal behavior is suspected to flow from the initial manufacturer flaws. Due to the limited lab equipment, we had to abandon the efficiency of a middle finger to let the other two work properly and meet the project aims.

So, we combined all improperly functioning hardware together (i.e. connected the flawed flex sensor to a flawed servo motor) to focus on managing the functionality of a properly

working equipment to develop an efficient implementation of a pointing and thumb fingers' motion.

```
#include <newFilter.h>
#include <iirFilter.h>
#include <firFilter.h>
#include <medianFilter.h>
#include <Servo.h>

medianFilter Filter1, Filter2, Filter3;

// Constants:
const int flexPin1 = A0; //pin A0 to read analog inputasd
const int flexPin2 = A1; //pin A1 to read analog inputasd
const int flexPin3 = A2; //pin A2 to read analog inputasd
Servo myservo1, myservo2,myservo3;  // create servo objects to control a servo
// Variables:
int value1 = 0, value2 = 0, value3 = 0; //save analog value

void setup(){
  Serial.begin(9600);//Begin serial communication
  // Start filtering instances
  Filter1.begin();
  Filter2.begin();
  Filter3.begin();
  myservo1.attach(7);
  myservo2.attach(8);
  myservo3.attach(9);
}
void loop(){
  /* Read and save analog values from flex sensors */
  value1 = analogRead(flexPin1);
  value2 = analogRead(flexPin2);
  value3 = analogRead(flexPin3);
  /* Process values through filters */
  value1 = Filter1.run(value1);
  value2 = Filter2.run(value2);
  value3 = Filter3.run(value3);
  /* Map sensor values so that they could move servos */
  int val1 = map(value1, 670, 800, 0, 180);
  val1 = constrain(val1, 0, 180);
  int val2 =  map(value2, 500, 1023, 180, 0);
  val2 = constrain(val2, 0, 180);
  int val3 =  map(value3, 880, 920, 0, 180);
  val3 = constrain(val3, 0, 180);
  /*
    Print data to the serial in the proper format,
    so that we could read data to Blender
  */
  Serial.print(val1);
  Serial.print(" ");
  Serial.print(val2);
  Serial.print(" ");
  Serial.print(val3);
  Serial.print("\n");
  /* Tell servos to move */
  myservo3.write(val3);
  myservo2.write(val2);
  myservo1.write(val1);
  delay(100); //Small delay
}
```

**Figure 5.** The Median Filtering incorporated into the main void loop() function that continuously runs the simulation. Source: Arduino code, viewed through Sublime Text, screenshot, 2018.

**Blender Simulation**

Blender 3D creation suite  was used for hand motion simulation to visualise the motion of our hand. It was implemented with Python script  that reads data from Arduino Serial. One of the fingers was represented with a cube and another with a cone because our main goal was to show functionality not the design. The motion was similar to what we expected to get and it is shown in the video. The static line of XY-axis on the plane is for illustration.

# 4. Discussion and Conclusions

During the project development we experienced the challenges of implementation of a higher-level abstract models. The initial approaches were found to be either naive or limited by the natural circumstances independent of our work (such as lab equipment limitations, flaws in the hardware available, etc.) Thus, we were able to learn that the project development must also hugely incorporate the accurate approach towards preparing all the essentials for project development. For example, the preliminary understanding of filtering of natural noises would have facilitated the whole process. In the end, we were able to reflect back on our work and be better prepared for future projects of this kind and the whole idea of project management.

## Student contributions

Adambek Aishabibi

- Report, presentation preparation
- The mechanical construction of a hand
- Investigation of available noise filtering
- Project materials preparation

Khuzyakhmetova Assiya

- Report, presentation preparation
- The mechanical construction of a hand
- Investigation of available noise filtering
- Project materials preparation

Sabyrov Arman

- Electrical circuit construction
- Demo video editing

Suleimenov Aidarbek

- All software development
- Electrical circuit design
- Blender simulation

Tuleuov Adilet

- The mechanical construction of a hand
- Project materials preparation
- Electrical circuit design

## 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/. Accessed: April 6, 2018.

"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. Accessed: April 9, 2018.

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. Accessed: April 6, 2018.