EMG-Controlled Mechanical Hand

Rodolfo de Curtis and Poposky Cherichel

Supervised by Dr. Masood Ejaz

Department of Electrical and Computer Engineering Technology, Valencia College, FL, USA

Abstract

Current non-invasive hand prostheses focus on the flexion and extension of all or most of the fingers simultaneously. This provides amputees with the ability to grasp objects and do simple hand gestures. While these prostheses tend to be prohibitively expensive, in recent years 3D printing has become affordable and available to an increasing sector of the population, making the production of prostheses more affordable and accessible. The EMG-Controlled Mechanical Hand is a fully functional 3D-printed mechanical hand that uses surface electromyography to control the flexion of its fingers individually. This is accomplished by placing electrode pairs on the user's forearm on top of the flexor muscles for each finger. The raw EMG signal from the electrodes is processed by five MyoWare Muscle Sensor EMG board into a signal suitable for the microcontroller unit. The Arduino Uno microcontroller uses an algorithm written by one of the authors to control the flexion and extension of the mechanical hand's fingers using the EMG inputs. This project focuses on the complete flexion and extension of individual fingers, giving the user an intuitive control of the mechanical hand. This paper covers the background research, planning, methodology, testing, and results done for this project. The results of the tests show that it is possible to control the fingers of a mechanical hand individually using electromyography. Additional EMG channels and logic would provide a precise control of flexion and extension as well as adding more features, such as wrist rotation.

Acknowledgements

The authors would like to thank Dr. Masood Ejaz for his guidance, advice, and encouragement throughout this endeavor, and Thomas Dillen for his help, advice, and support regarding 3D modeling and 3D printing

1. Introduction

1.1 Introduction

This project was conceived to address the problem of individual finger control in hand prosthetics. Current prosthetics focus on different types of grips rather than an intuitive control of each finger. While this approach is utilitarian, it has limitations when it comes to more specialized finger uses and user expression.

1.2 Motivation

The cost of a prosthetic arm ranges from $3,000 to $100,000 [1]. Even the most durable prosthetics are made to last 3 to 5 years of constant use [2]. There is currently a trend to use 3D printing to produce affordable prosthetics.

Previous projects using 3D-printing and electromyography use different grip patterns to completely close or open the hand and let the user grab objects. By using an increased number of surface electrodes, it is possible to discern which finger is being actuated, allowing the user to control the prosthesis intuitively and with increased versatility.

1.3 Current Solutions

1.3.1 Body-Powered Prostheses

Body-powered prostheses use harnesses and cables to open and close a hook using the user's upper-body muscles.

Body-powered prostheses are controlled by the user moving his or her shoulders or arms, depending on the system. The hooks can be normally open or normally closed.

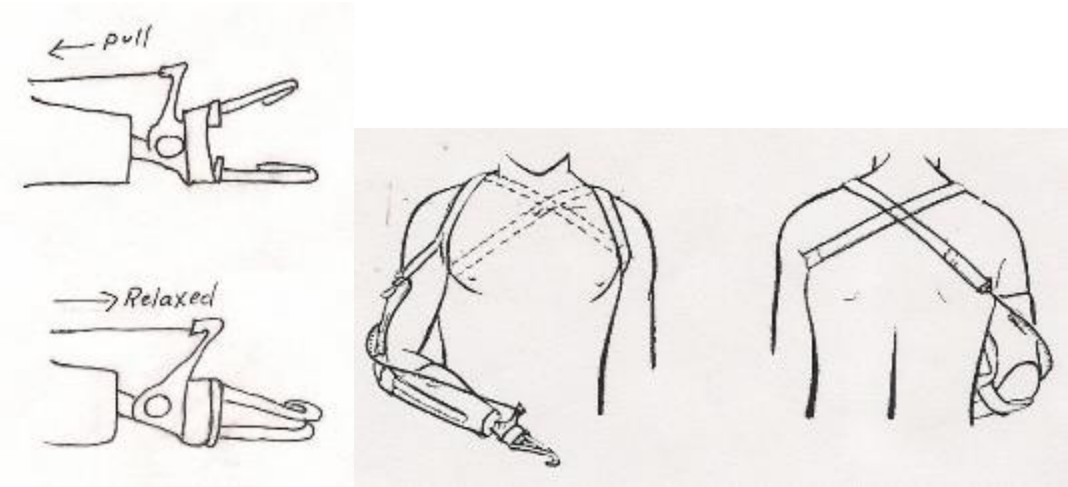


Figure 1.1: Body-Powered-Prosthesis. Hook and Harness [3].

1.3.2 Neural Prostheses

Neural prostheses use brain-computer interfaces to connect the brain of the user to a computer to control a peripheral such as a robotic arm or the user's muscles via stimulating electrodes [4]. An array of electrodes is implanted directly on the motor cortex and the user's neural activity is sent directly to a computer, where the signal is decoded and sent to the actuator.

Figure 1.2: Brain-Computer Interface User Controlling His Own Arm [4].

1.3.3 Myoelectric Prostheses

Myoelectric prostheses use surface electromyography to control prostheses using the user's muscle activity. Electromyography or EMG is a technique similar to electroencephalography (EEG) and electrocardiography (ECG and EKG) that measures and records the electrical activity generated by the motor neurons in muscles. EMG-based prosthetics uses electrodes placed on the skin above one or more muscles to control a prosthesis.

Figure 1.3: 3D-Printed Myoelectric Prostheses from Limbitless Solutions [5].

1.4 Similar projects

In order to set realistic goals for this project, it is important to look at the current state of hand prostheses. We will look specifically to projects done by engineering senior students and the approaches they used.

1.4.1 Intelligent Programmable Prosthetic Arm

In 2014, students from the University of Central Florida designed a prosthesis that uses electromyography on a single muscle group on the user's forearm to automate grasping in different grasping patterns and developed a mobile application for the user to select from the different grasps as well as create their own. They used five servo motors to actuate all fingers.

1.4.2 Multiplex Bionic

In 2015, another group of UCF students designed an EMG prosthesis using a single muscle group, this time on the upper arm, to perform different grasping patterns. They used three servo motors to control the first three digits (thumb, index, and middle fingers).

1.4.3 Electrical System for Exoskeletal Arm

In 2016, a group of UCF students, with the help of Limbitless solutions, designed an exoskeletal electro-pneumatic arm that enabled the user to regain basic control of his arm and hand.

1.4.4 EEnable: An Affordable Myoelectric Powered Prosthetic Hand

In Spring 2019, students from the University of Notre Dame used four EMG channels to recognize the user's intended hand gesture and replicate it on a mechanical hand. The project can recognize four different hand gestures.

1.4.5 Mechanical EMG-Controlled Hand v2.0

In 2015, students from the University of California, Santa Cruz created a prosthetic hand using one EMG channel with the electrodes placed on the upper arm. The project focused on improving the hand's grip from a rigid grip most projects have, to a dexterous grip.

While these projects focus on the simultaneous flexion of all the fingers to perform grasping using few EMG channels, this project uses five EMG channels to identify and perform individual flexion of fingers.

1.5 Engineering Requirements

Table 2.1 shows the project requirements divided into modules, with specifications for the parts in each module, justification, verification, and who will be responsible for each.

Table 1.1: Engineering Requirements

Block Name	Specific Components	Engineering Specifications	Justification & Verification	Responsibilities
Power Module	-Rechargeable battery -Voltage Regulator	● Capacity: at least 2 Ah ● Voltage: 5V DC	Justification: Provides power to the other modules. Verification: MCU and servos work as expected.	Rodolfo: Verify expected results. Poposky: Design and route the power module dependent on the requirement.
Control Module	-Micro Controller Unit	● Programmable memory. ● 10-bit ADC ● 5 to 10 analog inputs ● At least 5 PWM output pins.	Justification: It processes the EMG input using code written by the author to provide output to each finger individually. Verification: Sending multiple input signals and verifying the expected output.	Rodolfo: Program the logic needed to control the output. Develop algorithm to use user EMG input values to control the output. Poposky:

				Verify expected outputs.
Sensor Module	-Electromyogram signal processing circuit. -Wet gel electrodes	● Senses the user’s muscle activity and outputs a rectified and integrated EMG signal to the MCU. ● 5 Channels + ground (11 electrodes)	Justification: The Electrodes pick up the electrical activity of the user’s motor neurons. The circuit processes the EMG signal, so it is more suitable for the MCU. Verification: Measure the rectified and integrated EMG signal that is produced to see if it meets requirements.	Rodolfo: Verify the output of the sensor. Poposky: Route sensor to the power module and MCU unit.
Mechanical Module	-3D printed hand w/ elastics & fishing lines -Five servo motors	● Five Servo Motors to control individual fingers.	Justification: The servo motors are the actuators responsible for moving the fingers of the hand. The hand is the end effector that interacts with its environment, mainly by grasping. Verification: Each finger should close completely, quickly, and with	Rodolfo: Print hand, assemble it, install fishing lines, and install servo motors. Poposky: Design 3D printed forearm.

			some force (to be determined).	

1.5.1 Success Criteria

This project had two criteria to fulfill. First, the hand should be able to grasp an object, and second, the user should be able to control the flexion and extension of each finger individually. Full flexion and extension of the fingers was our goal.

1.6 Block Diagram

Figure 2.1 shows the block diagram of the system with different stages in each block. The next chapter contains a discussion of the implementation of each part.

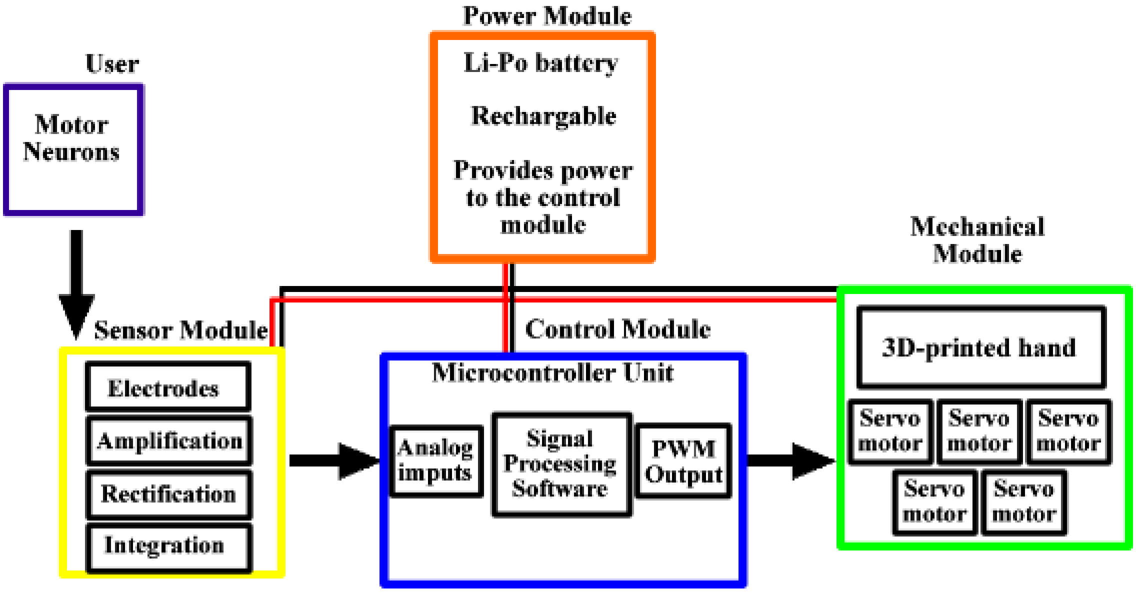


Figure 1.4: Block Diagram

2. Project Implementation

2.1 Introduction

The EMG-controlled mechanical hand consists of 3D-printed pieces assembled into a mechanical hand and a forearm that houses the rest of the components. Each finger is attached to a pair of fishing lines for flexion and extension. The fishing lines are connected to pulleys attached to the servo motors housed in the forearm that flex and extend the fingers. The control of each servo is done by a microcontroller unit using custom software that takes input provided by the sensor unit and identifies what finger to move. The sensor unit consists of several electrodes connected to the user’s forearm and electronics to process the raw signal into a signal that is suitable for the controller unit.

2.2 Implementation of the Project

The following is a discussion of parts and modules that are used in this project, as shown in the block diagram in figure 1.4.

2.2.1 Hardware

Microcontroller

The project requires a controller that can take five analog inputs, has an ADC with a decent resolution, has at least five outputs capable of driving a servo motor, and a programmable memory. The Arduino Uno was selected to be used as the project's controller unit.

Figure 2.1: Arduino Uno [6]

Power Supply

A battery supplies power to the microcontroller unit, the servo motors, and EMG boards. The project uses an Ovonic 7.4V 5000mAh 50C 2S LiPo Battery Pack. A LM2596 DC-DC buck converter is used to step down the battery voltage to 5V.

Servo Motors

Five servo motors pull on the fishing lines to flex their respective fingers. The TowerPro MG946R servo motor has an operating voltage range of 4.8 to 6.6V, an operating speed range of 0.2seconds/60degrees at 4.8V to 0.17seconds/60degrees at 6.0V, and a stall torque range of 10.5kg/cm at 4.8V to 13kg/cm at 6.0V.

Figure 2.2: TowerPro MG946R [7]

Sensor Module

The raw signal that electrodes produce is a high-frequency alternating current that is not suited to be used with a microcontroller's ADC. EMG circuits designed to work with a microcontroller amplify, rectify, and integrate the raw EMG signal to output an EMG linear envelope.

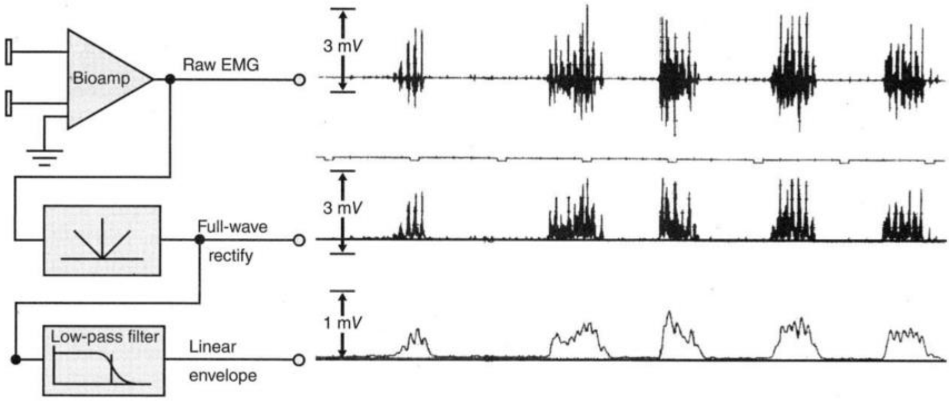


Figure 2.3: EMG Linear Envelope Detection [8].

The MyoWare Muscle Sensor is an EMG sensor module manufactured by Advancer Technologies that measures the electrical activity of a single muscle and outputs the linear envelope of the filtered and rectified signal. This output has a range of 0V to Vs, where Vs is the source voltage connected to the muscle sensor board.

Five MyoWare muscle sensors are used to get the electrical activity of the muscles that control each finger.

Figure 2.4: MyoWare Muscle Sensor [9].

Electrodes

For this project, 24 mm surface EMG gel electrodes were used.

Eleven electrodes, two for each finger and one for reference, are placed on the forearm and sense the electrical activity of the user's muscles. The electrode pairs for each finger are placed on the ventral extrinsic flexor muscles in the forearm and the reference electrode is placed on the dorsal side of the hand, but it can be placed in any other unrelated muscle group or bony part of the arm. The elbow can be used for an alternate placement of the reference electrode.

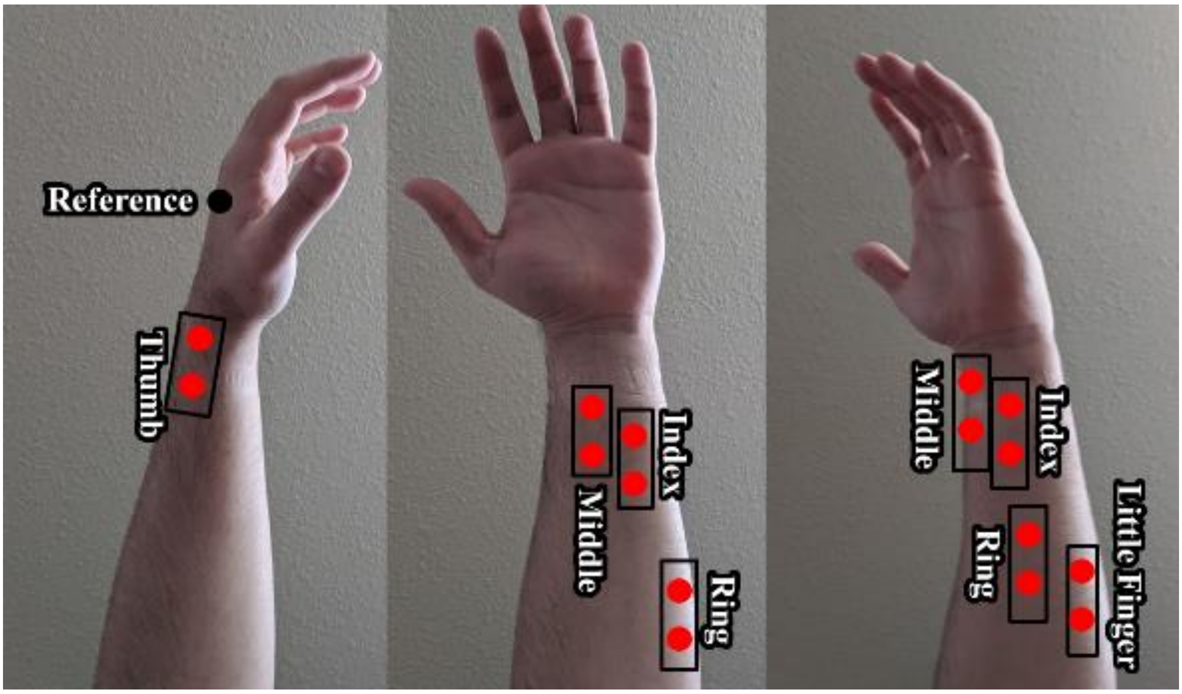


Figure 2.5: Electrode Placement.

2.3.2 Software

The Arduino code reads the analog data coming from the EMG board and uses it to operate the servo motors connected to the PWM outputs.

2.3.3 3D-Printed Arm

The hand and forearm design used are from Gael Langevin's InMoov robot [10]. The fingers of the hand are flexed and extended using fishing lines and pulleys connected to each servo motor. During flexion of a finger, the servo motors pull on one of its fishing lines. During extension, the servos turns in the opposite direction, pulling on the fishing line that extend the finger.

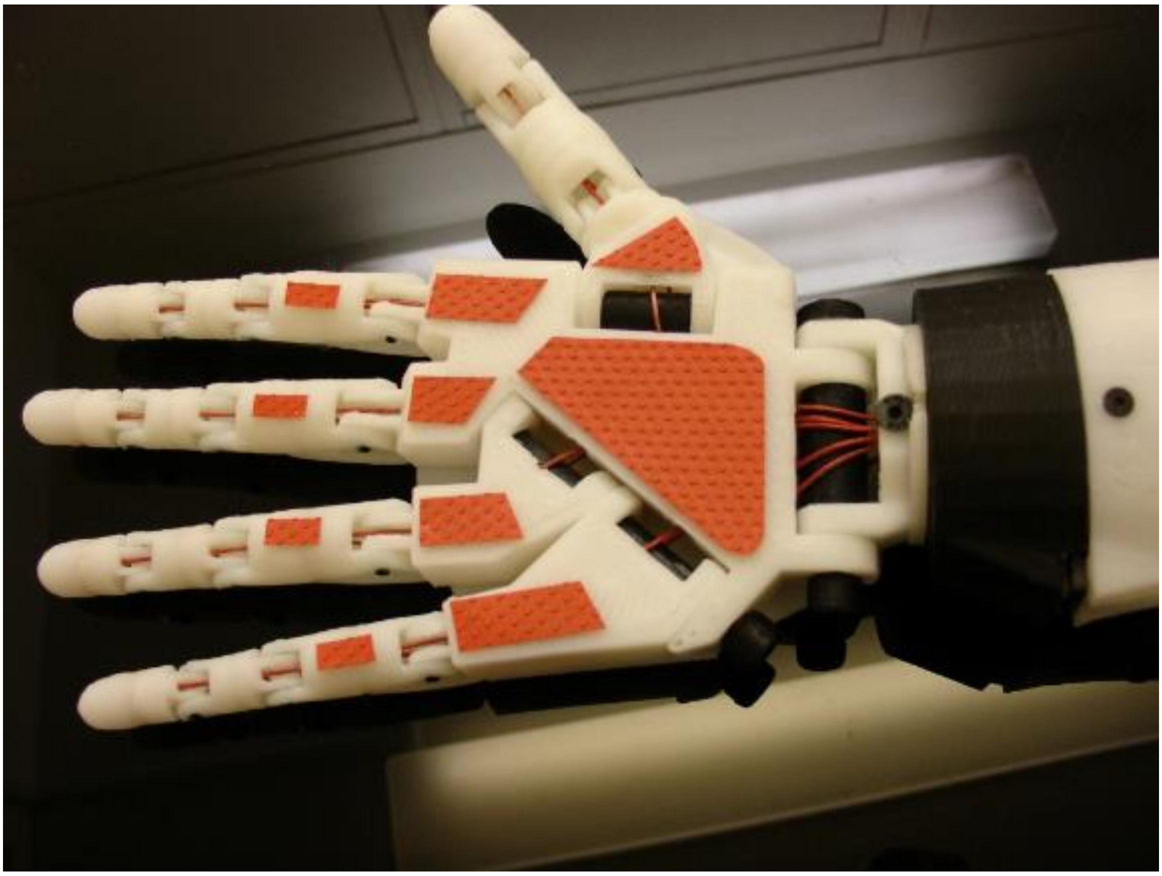


Figure 2.6: The InMoov hand and forearm, designed by Gael Langevin [10].

3. Methodology and Results

3.1 Introduction

The development of the EMG-Controlled Mechanical Hand took a lot of work and diligence from the group. Responsibilities were divided in order to increase efficiency and productivity. Rodolfo printed the mechanical hand with its forearm and assembled it; wrote the code to control it; tested the hand with the sensors; designed the circuit board, built it, and soldered it. Poposky chose the battery and buck converter and designed a new forearm to house all the components and circuitry.

3.2 Design Implementation

3.2.1 Finger Tester

In November of 2019, the InMoov finger tester was printed and assembled, as shown in figure 3.1, to understand how the mechanical finger works, test the MG946r servo motor, and to see if the InMoov hand would suit the requirements of the project. The pieces for the finger tester were printed on a Creality Ender 3 3D printer using black PETG filament of the AmazonBasics brand.

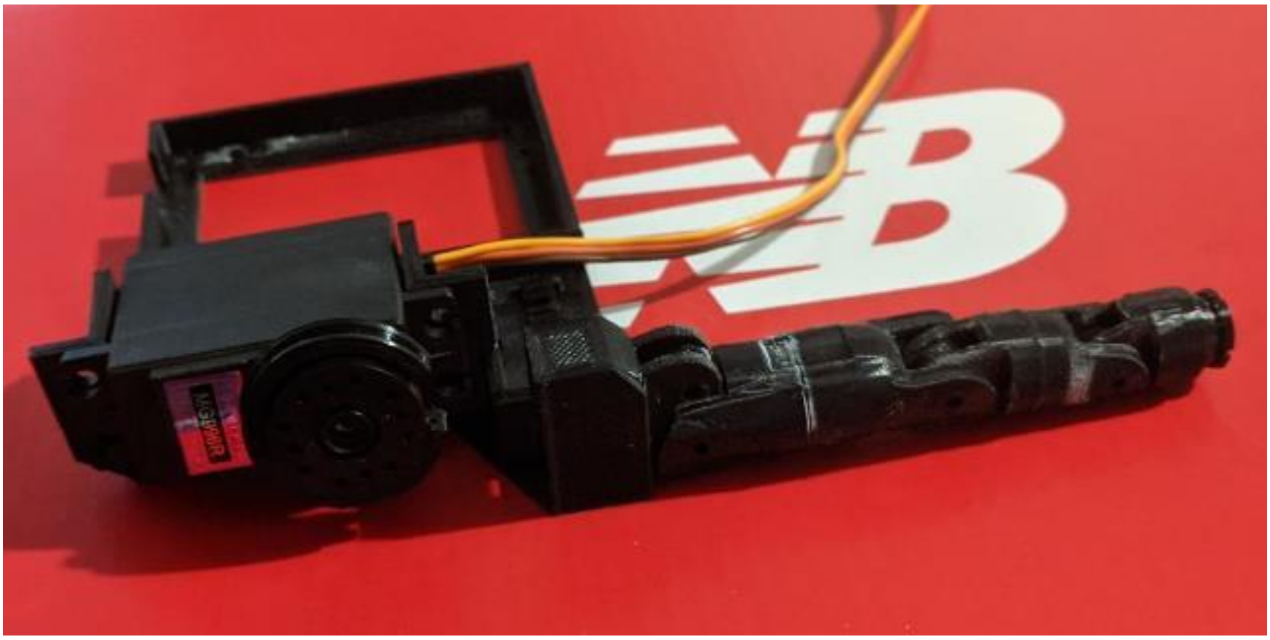


Figure 3.1: InMoov Finger Starter.

The finger consists of three phalanges that are printed as five parts. The first two phalanges consist of two pieces each that were glued using cyanoacrylate glue. The phalanges connect to each other using hinge joints, using 1.75mm filament as a pin for each hinge. An exploded view of the InMoov's index finger can be seen in figure 3.2. Flexion and extension of the finger is achieved using two separate fishing lines that go through separate chambers inside the phalanges and tied together at the distal end of the finger. Braided fishing lines are used for their high strength and small diameter. Flexion is achieved by pulling on the fishing line that goes through the ventral side of the finger. Extension is achieved by pulling on the fishing line that goes through the dorsal side of the finger.

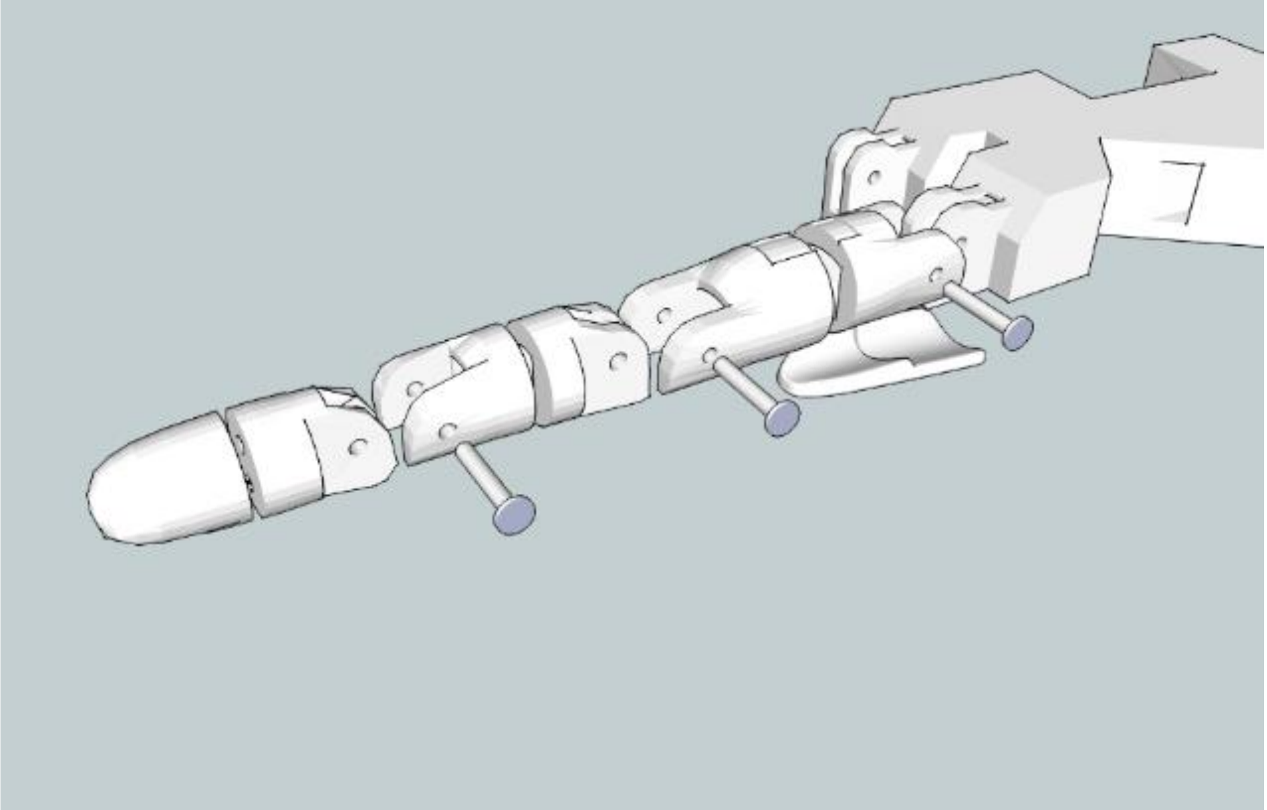


Figure 3.2. Parts of the InMoov index finger [11].

The two fishing lines at the proximal end of the finger are connected to a pulley that is coupled to the shaft of an MG946r servo motor. When the servo motor's shaft is rotated 90° counterclockwise, the finger is flexed. Likewise, when the shaft is rotated 90° clockwise, the finger is extended.

One MyoWare muscle sensor and an Arduino Uno microcontroller board were used to control the MG946r servo motor, as shown in figure 3.3. A simple code was written to test the muscle sensor and control the flexion and extension of the finger.

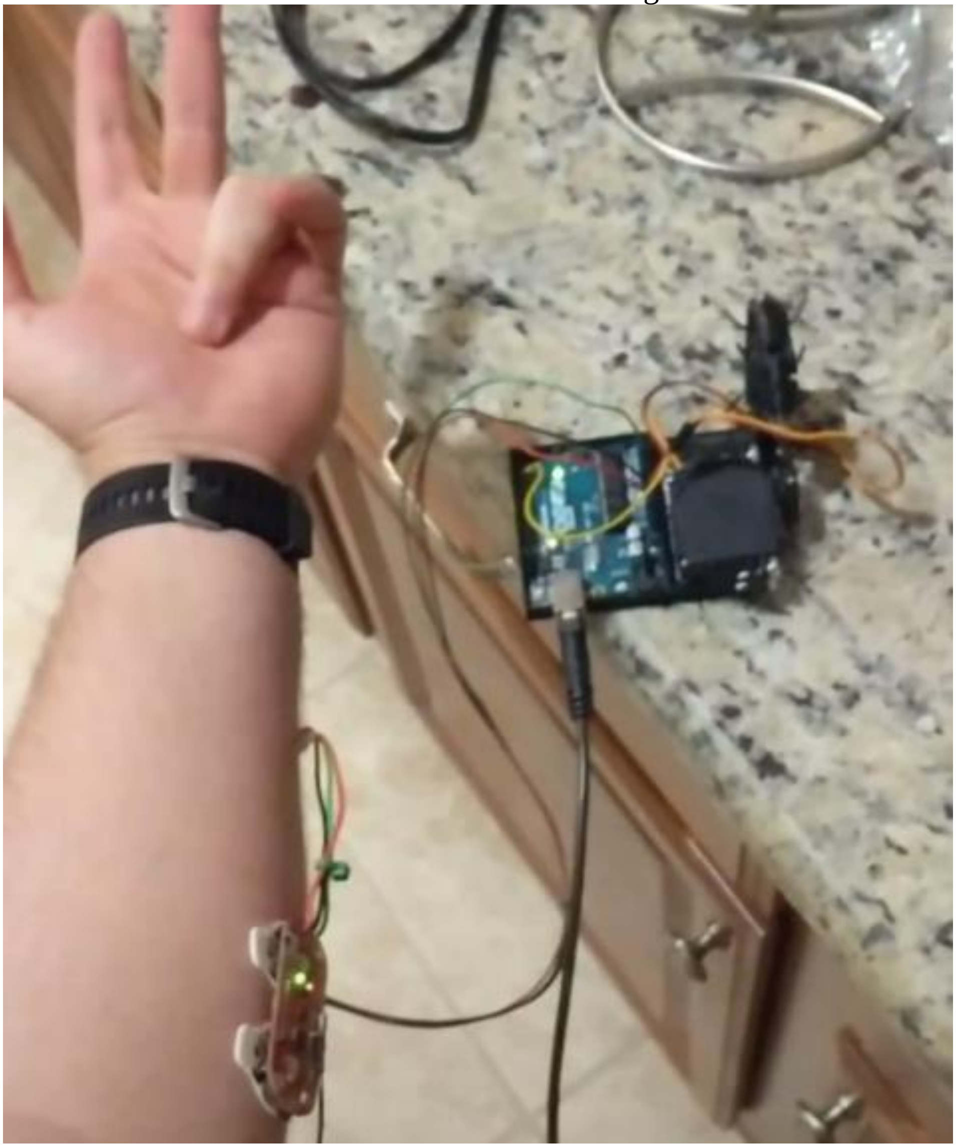


Figure 3.3: Controlling Finger Starter using the MyoWare muscle sensor board.

3.2.2 Early Prototype

In early December of 2019, the InMoov hand and forearm were printed and assembled to test the circuit, motors, sensors, and program.

The pieces of the hand and forearm were printed on a Creality Ender 3 printer using black PETG filament of the AmazonBasics brand. The hand consists of forty-five different pieces, the wrist consists of seven pieces, and the forearm consists of eleven pieces. Figure 3.4 shows the sixty-three printed pieces in front the 3D printer. The housing piece for the servo motors was screwed to the dorsal side of the forearm. The servo motors were screwed to the servo housing piece. The guiding pieces for the fishing lines were mechanically connected to the servo-motor housing and glued to it using cyanoacrylate glue.

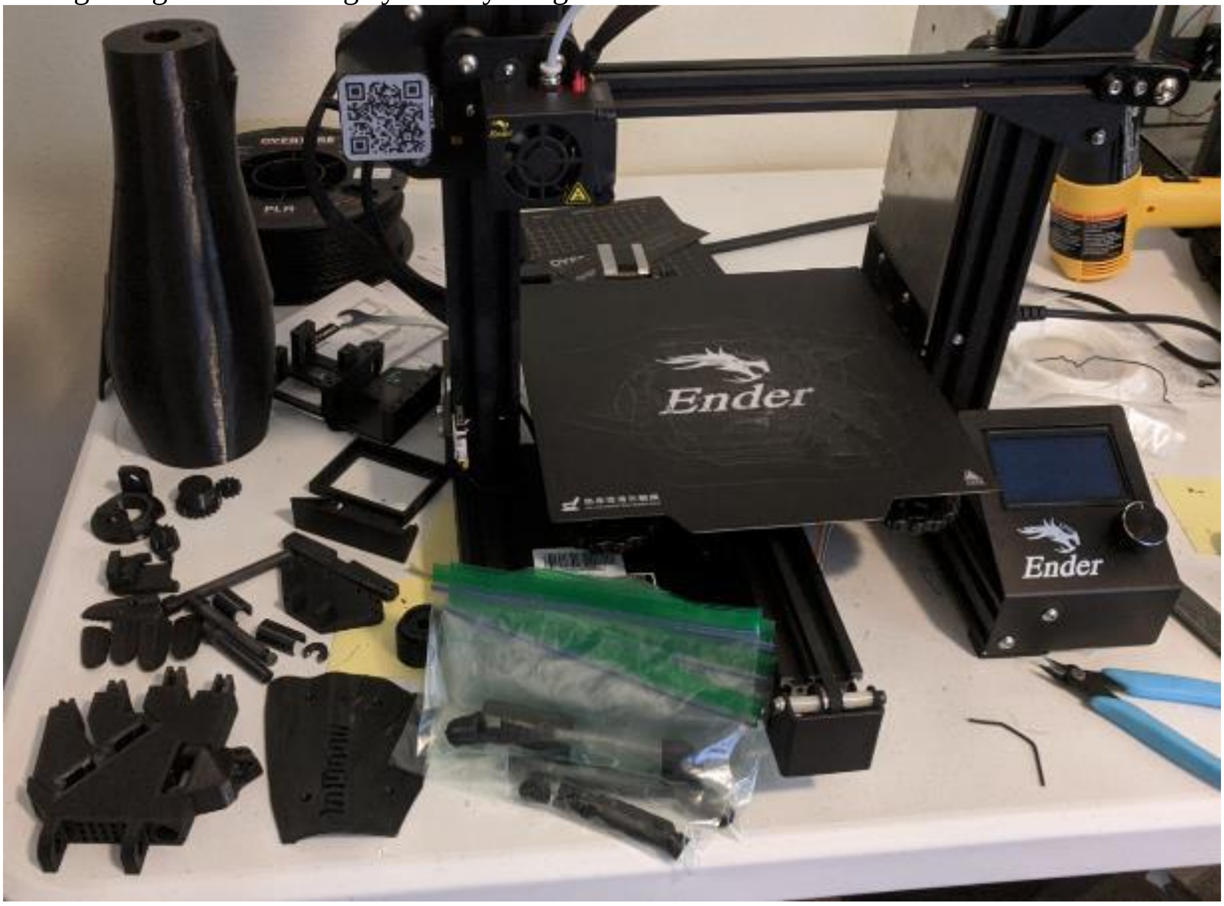


Figure 3.4: Printed pieces of the InMoov hand, wrist, and forearm.

Joint connectors of moving pieces and connector pins needed to be sanded prior to assembly to ensure smooth movement of the mechanical pieces. Ten 70cm-long pieces of braided fishing line were used to connect the fingers of the hand to pulleys in the same way as it was done for the InMoov finger tester. The fishing lines were connected to the fingers of the InMoov hand and each pair was tied at the distal end of the fingers. Before connecting the hand and wrist to the forearm, the fishing lines were passed through a guiding piece inside the forearm ulnar crest, as shown in figure 3.5a. After that the ulnar crest piece and the wrist piece were screwed together. The fishing lines were passed through the guiding pieces on the servo-motor housing and each pair corresponding to a finger were connected to the pulley of a servo motor shaft, as shown in figure 3.5b.

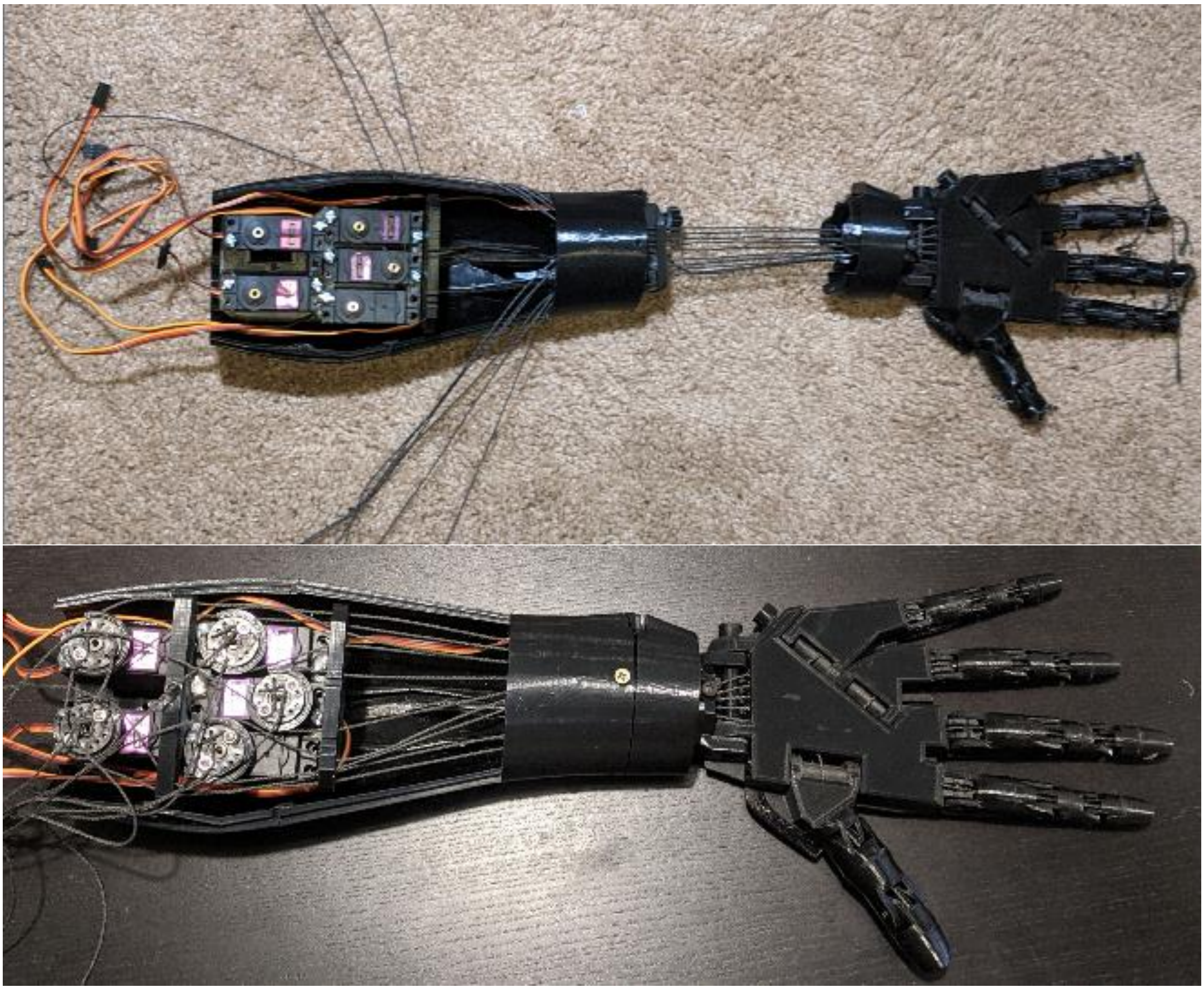


Figure 3.5: (a) Hand and wrist prior to being connected to the forearm (b) Assembled InMoov Hand and Forearm.

3.2.3 Circuit Board.

After verifying the step-down voltage regulator output was 5V, the circuit for the project was built on a breadboard. The circuit schematic is shown in figure 3.6.

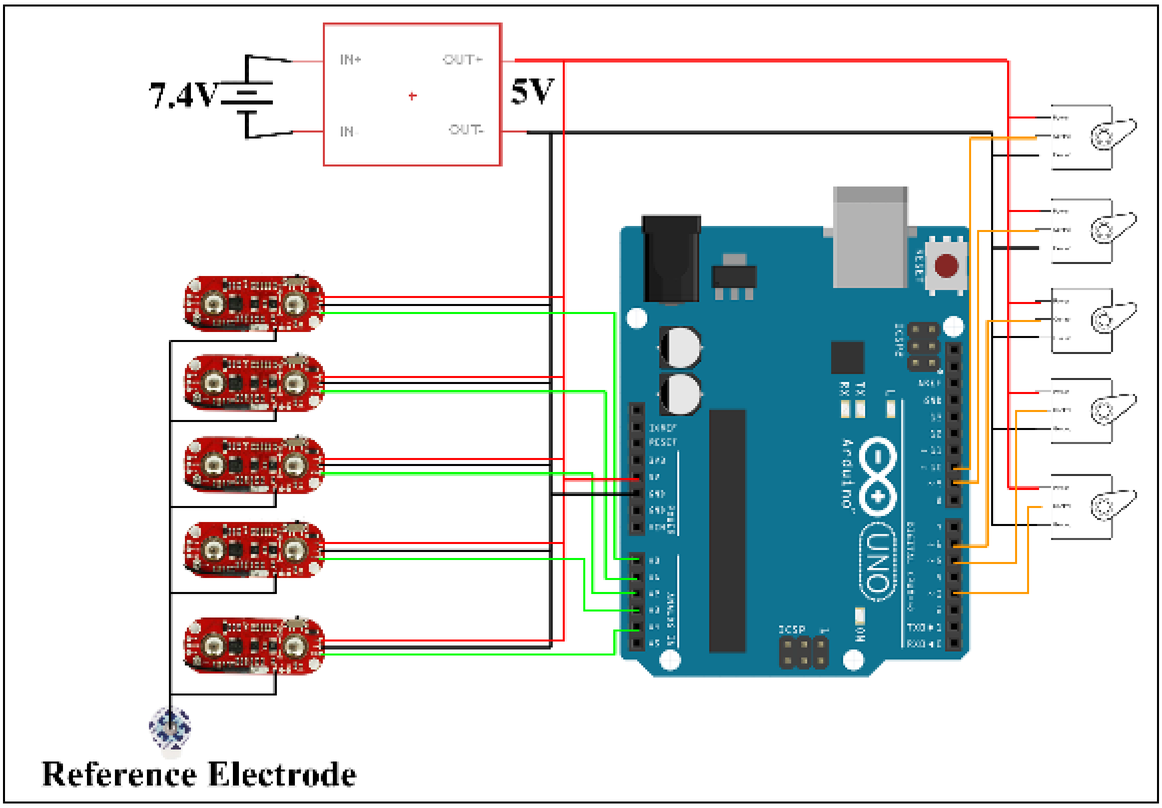


Figure 3.6: Circuit Schematic.

The operating voltage of the Arduino Uno is 5V. The operating voltage range of the MG946r servo motors is from 4.8V to 6.6V. The recommended supply voltage for the MyoWare muscle sensor is 5V. Because of this, all the components receive power from the output of the step-down voltage regulator. The references of the five muscle sensor boards are connected to a single electrode cable. The signal output cables of the muscle sensors are connected to analog input pins of the Arduino board. The control input cables of the servo motors are connected to the PWM signal output pins of the Arduino board.

A circuit board for the project was designed in Autodesk Eagle, as shown in figure 3.7, and printed at Valencia College using a Voltera V-One PCB Printer, as shown in figure 3.8a.

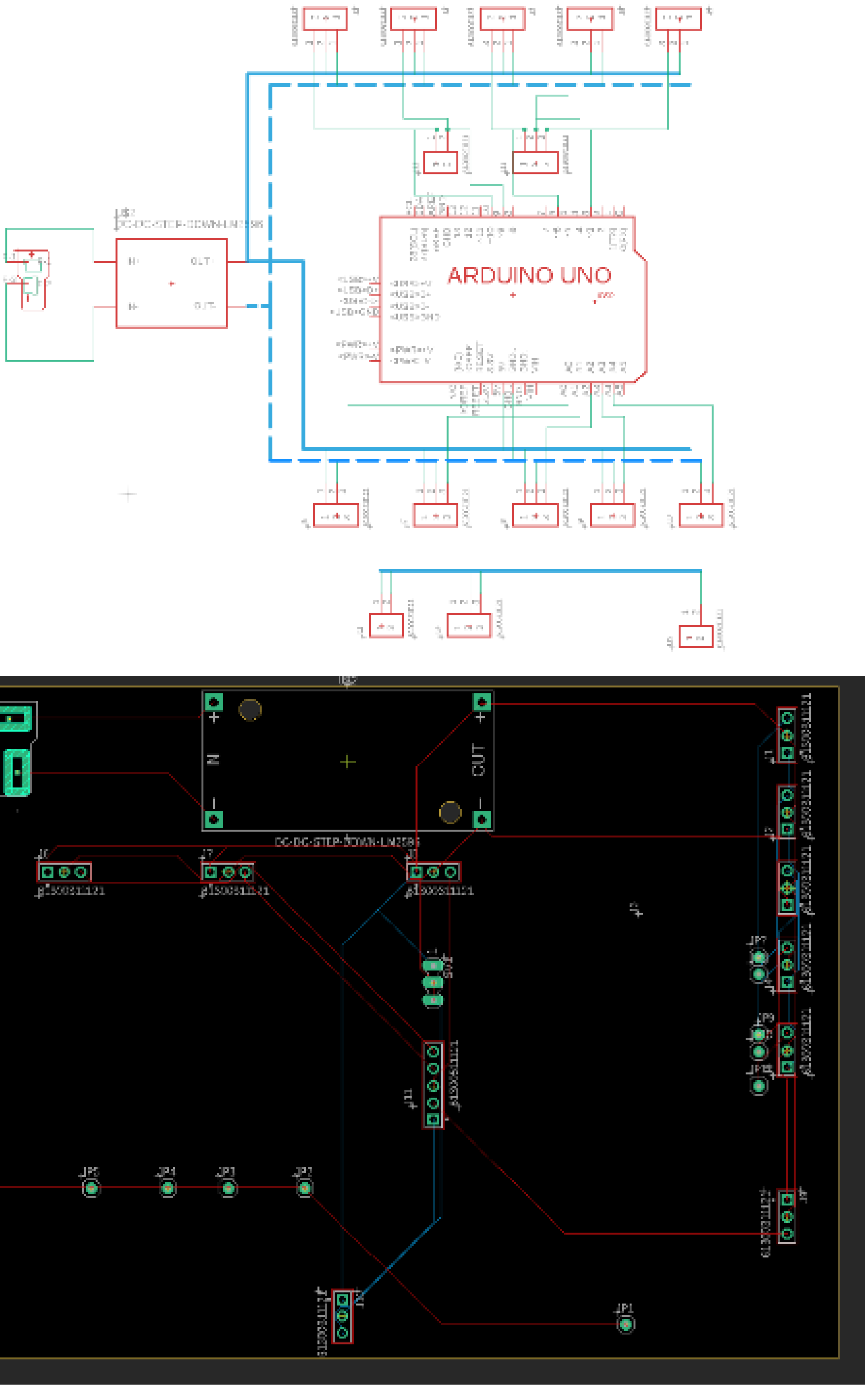


Figure 3.7: (a) Circuit Schematic on Eagle (b) PCB Design on Eagle.

PCBs printed with the Voltera PCB printer require a low-melting point solder. Not knowing this let to one of the silver traces on the board to be burnt away, as shown in figure 3.8b.

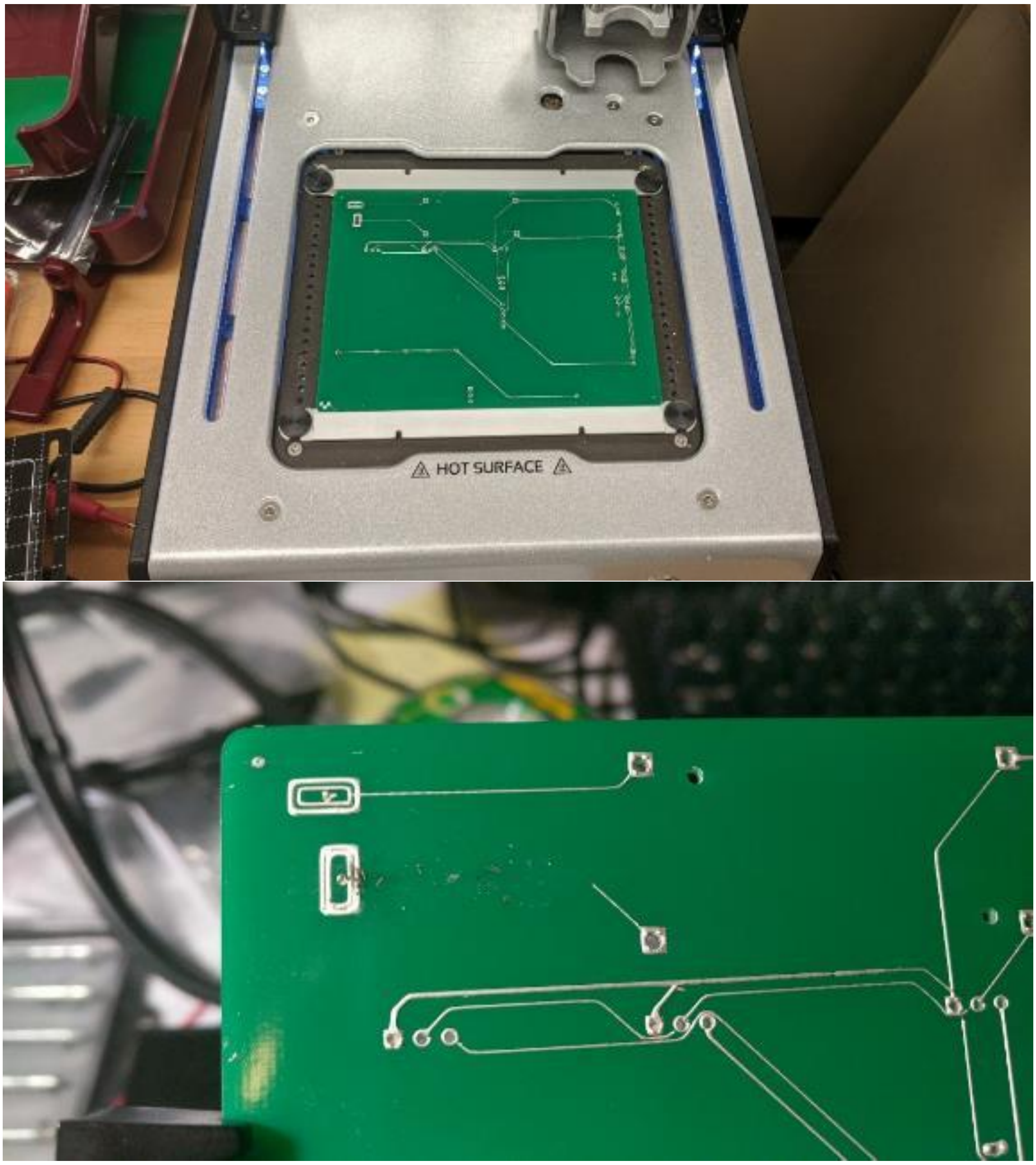


Figure 3.8: (a) Printing of the Circuit Board (b) Burnt Silver Trace.

Due to the COVID-19 pandemic, Valencia College was closed, and the circuit trace could not be fixed. As a result, the circuit was made on a perforated board as shown in figure 3.9.

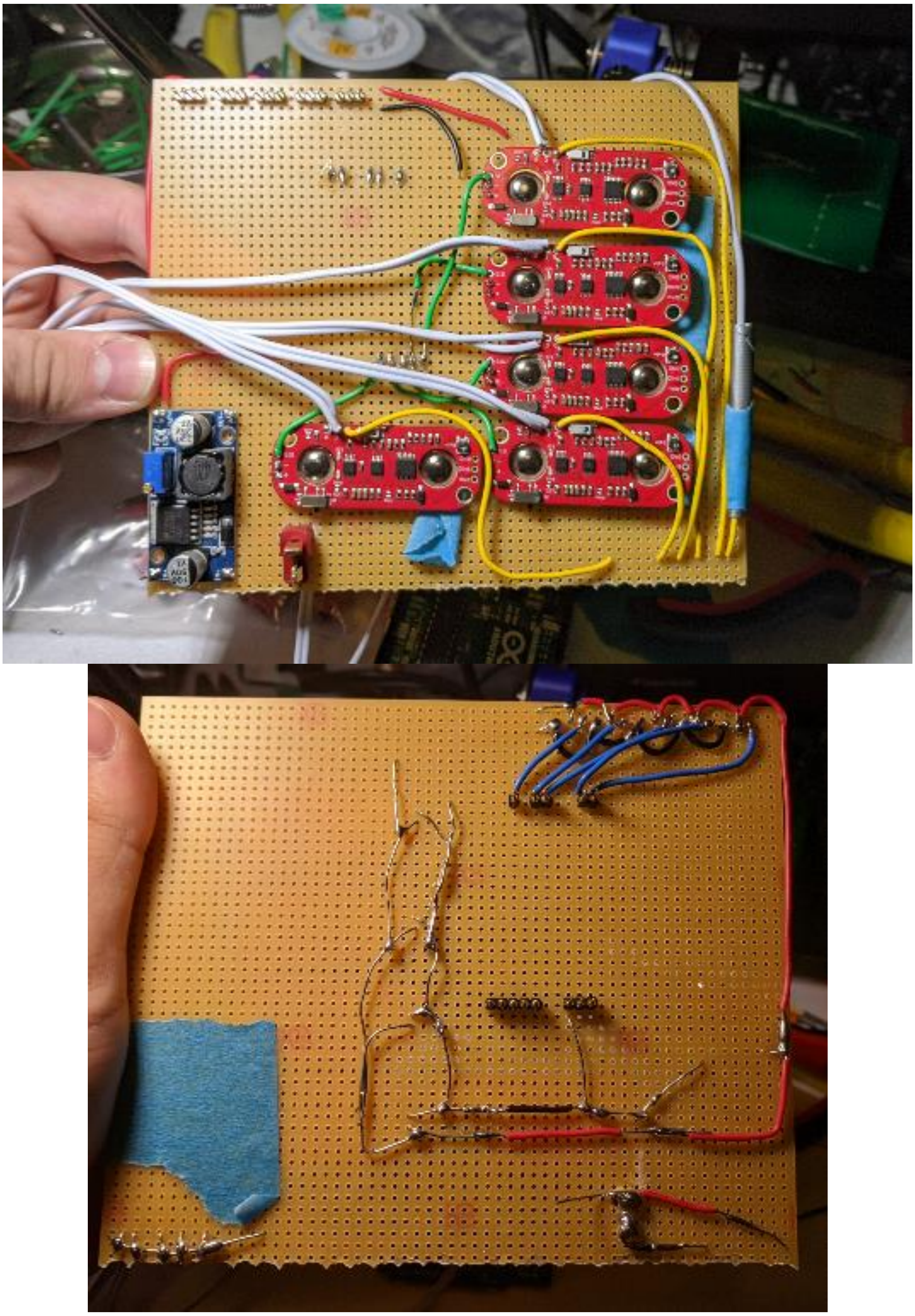


Figure 3.9: (a) Front of the circuit on a perforated board (b) Back of the board.

3.2.4 Forearm Design

The forearm design can be considered a very challenging section in the project. Poposky had no experience in using 3D printers and designing in Inventor or SolidWorks. Poposky spent a lot of days learning the design software Inventor, which is very similar to SolidWorks, with the help of YouTube and other online resources. After investing a lot of time in learning the software, Poposky started practicing using the software to design the forearm for the project.

The purpose of designing a new forearm was to house the servo motors, battery, the Arduino microcontroller, and the circuit board. The parts designed are shown in figure 3.10.

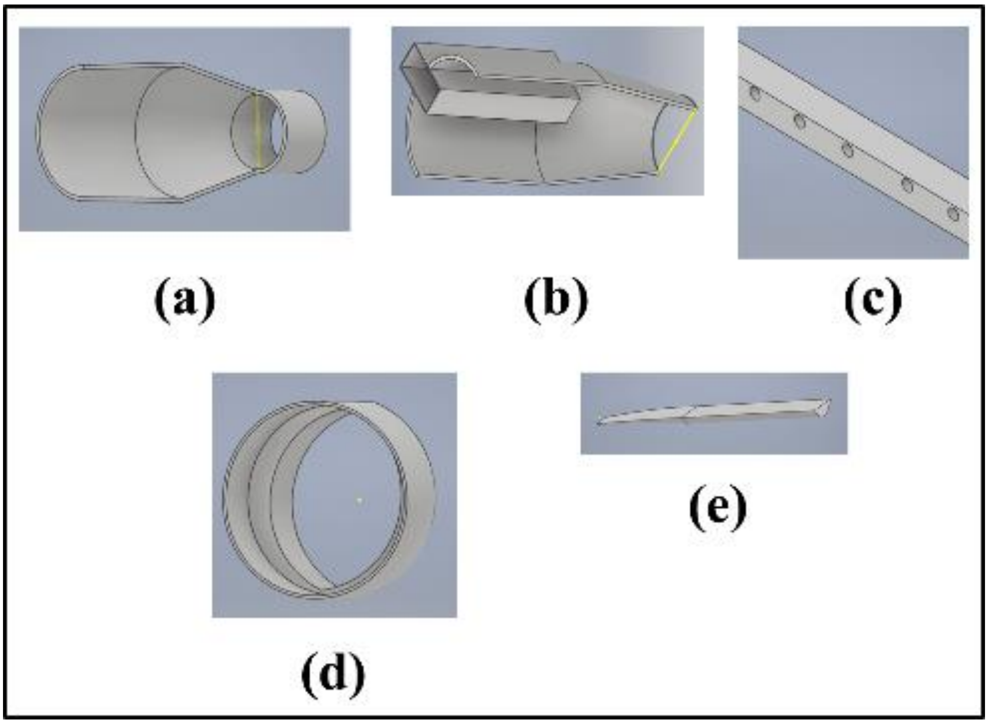


Figure 3.10: (a) Forearm (b) Forearm Cover (c) Fishing Lines Guide (d) Side Support (e) Side Support.

These parts were printed and assembled, as shown in figure 3.11.

Figure 3.11: New forearm pieces.

Another InMoov hand, wrist, and servo bed were printed, sanded, and assembled for this forearm. Screw holes to attach the servo bed and side support needed to be drilled, as they are absent from the newly designed pieces. A problem with this design is that the main support is not made to lie flat at the base of the forearm and is instead intended to rest between the bottom and the bottleneck with glue, as shown in figure 3.12.

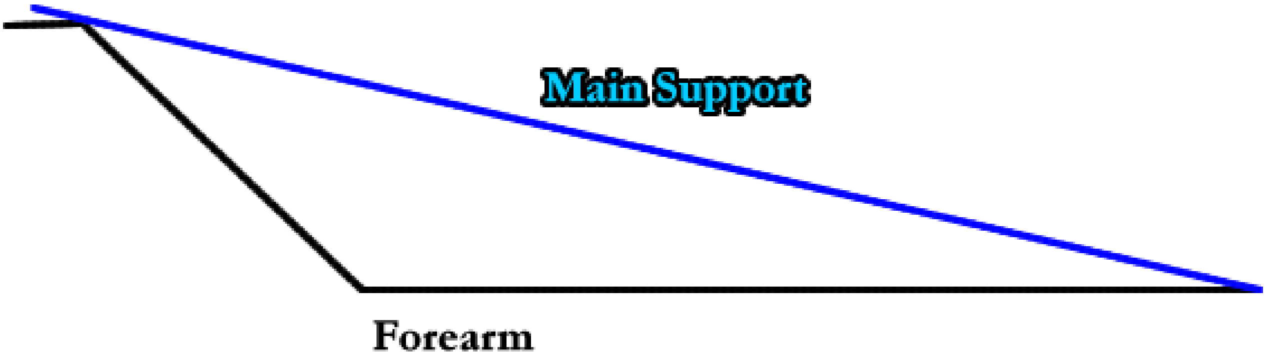


Figure 3.12: Cross section representation of the main support and forearm.

Due to the minimal contact surface area, the main support did not remain attached to the forearm with regular cyanoacrylate glue. The solution was to use generous amounts of cyanoacrylate gel. Figure 3.13 shows the assembled forearm pieces connected to the InMoov wrist.

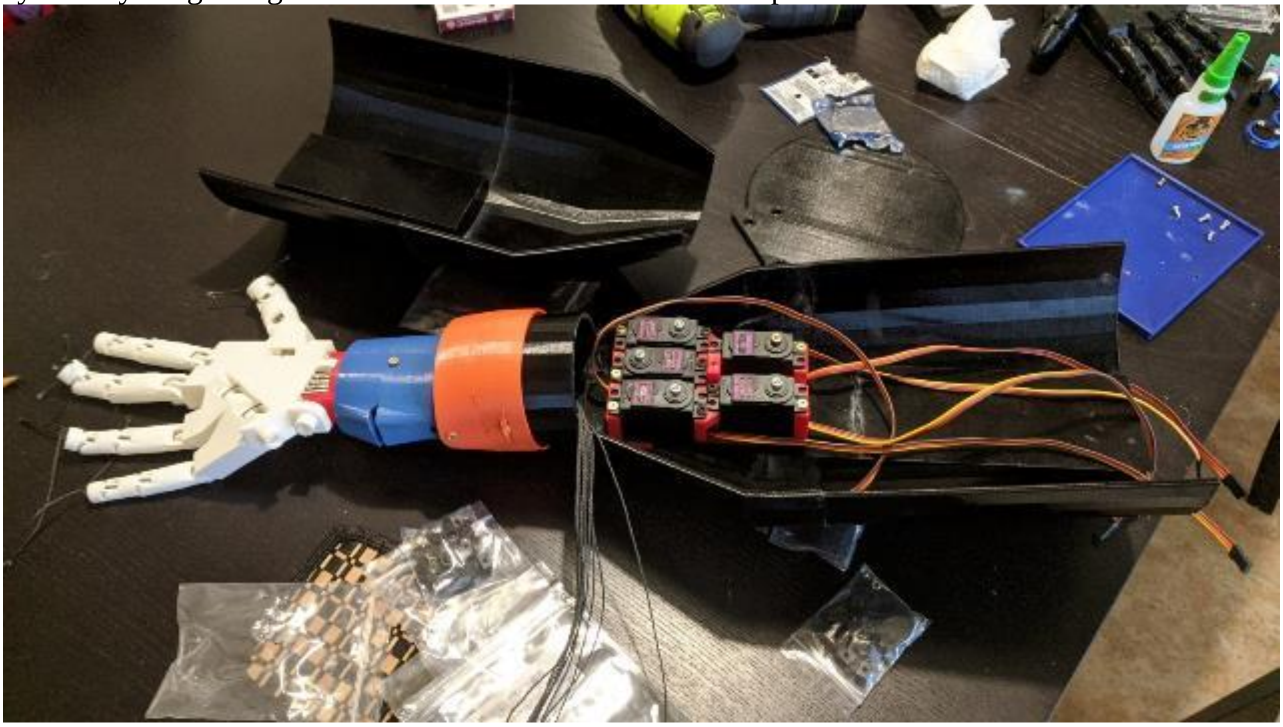


Figure 3.13: New forearm with servo motors and InMoov hand.

The next step taken was to connect the servo motors to the circuit board, as shown in figure 3.14, use a simple Arduino code to center the servomotor shafts, and installing the pulleys.

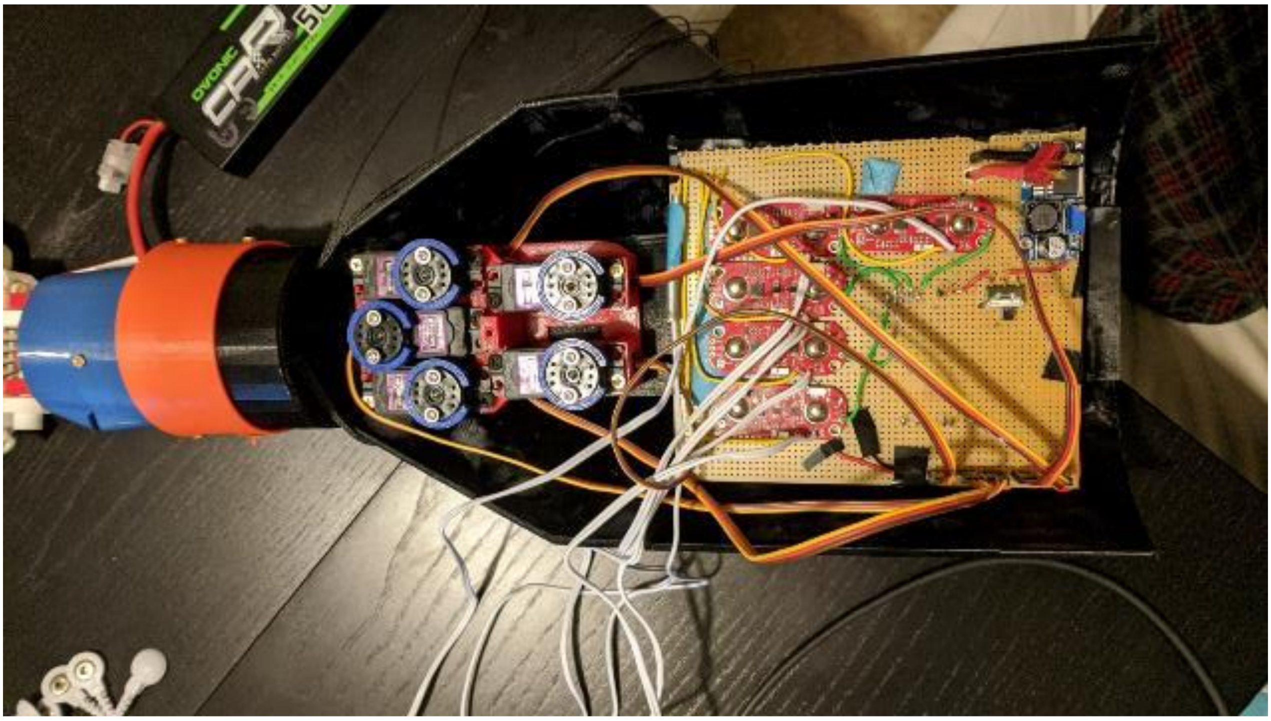


Figure 3.14: Circuit board and servo motor pulleys in new forearm.

The next step was to attach the fishing lines to the pulleys. It was at this step that it was noticed that the guiding pieces for the fishing lines were not printed, as shown in figure 3.15.

Figure 3.15: Missing fishing line guides.

Without these pieces, the fishing lines would get in each other's way and get entangled. Due to the COVID-19 pandemic, these pieces could not be printed at Valencia College. Rodolfo's 3D printer was out of commission with no evident way to repair it. For these reasons, the new forearm design could not be tested. Figure 3.16 shows the final product to the left in contrast to the InMoov hand and forearm to the right.

Figure 3.16: Designed forearm on the left and InMoov forearm to the right.

3.3 Software

The code used in this project was expected to be more complex than what it turned out to be. The code checks the value of each sensor and compares them one by one to the threshold values set for each finger. If the value for a finger is greater than the value of the upper threshold, the finger on the mechanical hand moves to the closed position. If the value for a finger is less than the value of the lower threshold, the corresponding finger moves to the open position. The code can be seen in its entirety on appendix A. A flow diagram of the code is shown in figure 3.17.

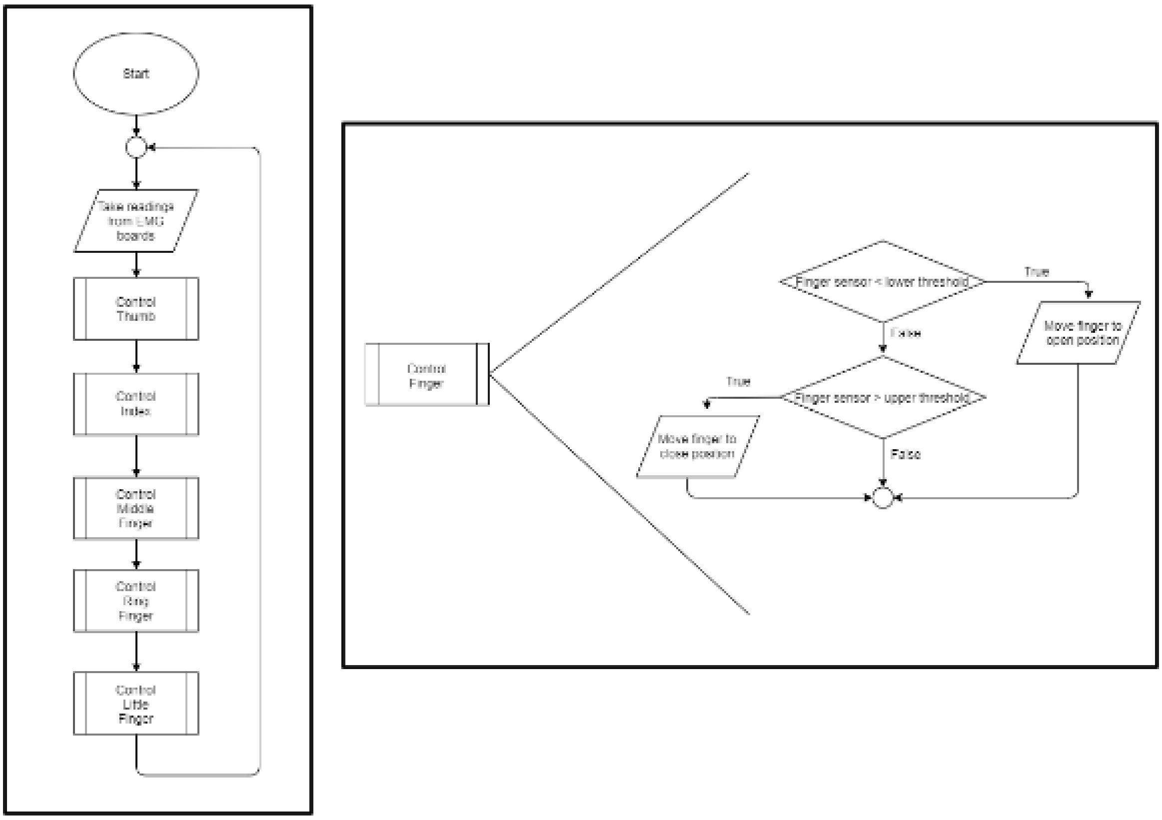


Figure 3.17: Code flow diagram.

3.4 Testing and Troubleshooting

Each test started by placing the electrode pairs one by one on their target muscle on the forearm and flexing the finger corresponding to the muscle while looking at the serial plot of the sensor output to see if the electrode placement was correct. Once this was done with each finger, the motors were connected to the circuit and individual finger movement was tested. Threshold values were tweaked as needed until each finger could be flexed and extended individually.

One major problem was electrode placement. Position and orientation of the electrodes has a great impact on the strength of the muscle sensor signal and a small variation in the placement of the electrode pairs can result in a large difference in sensor output. Flexor muscles for the fingers are close to each other. This results in a sensor for one muscle picking up the signal of a different muscle. Optimal electrode placement is such that each sensor can pick up a relatively strong signal from its intended muscle while minimizing the crosstalk from adjacent muscles. With proper electrode placement, the effect of crosstalk can be eliminated by using threshold values specific to each finger sensor.

Figure 3.18 shows the sensor outputs when a bad electrode placement is used.

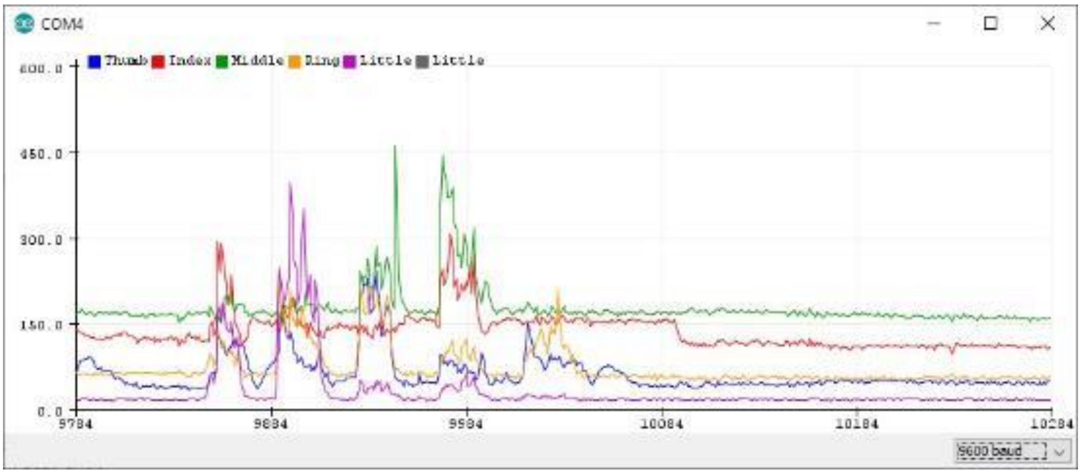


Figure 3.18: Plot of sensors' output during flexion with poor electrode placement. From left to right: little finger, ring finger, middle finger, index, and thumb.

Figure 3.19 shows a proper electrode placement.

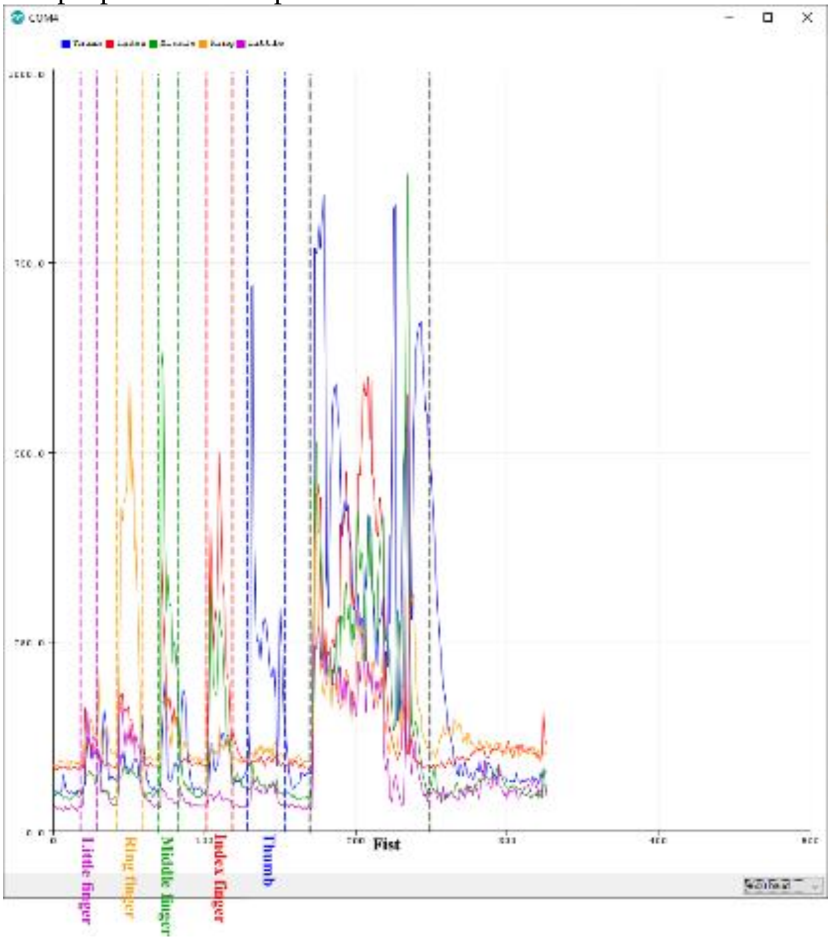


Figure 3.19: Plot of sensors' output during flexion with a better electrode placement.

The placement of the electrode pair for the thumb was the most difficult. For the index, middle, ring, and little fingers, the electrodes are placed on their extrinsic ventral muscle, the flexor digitorum profundus. When testing the muscle sensor on the thumb's extrinsic ventral muscle, the flexor pollicis longus, the output values were too low and required an extreme force to produce any significant reading. The strongest sensor output was found when the electrode pair was placed on the intrinsic flexor muscle that lies between the metacarpals of the thumb and index fingers, the flexor pollicis brevis. Since the flexor pollicis brevis is in the hand, it could not be used for an amputee. A good compromise was placing the electrode pair just below the wrist, close to the opponens pollicis, as shown on figure 2.5, as it produced a useable sensor output. It should be noted than when the flexor pollicis brevis was used and the hand was clenched into a fist and then released, the thumb sensor output was kept high for a few seconds after the finger was relaxed, as shown in figure 3.20. The reason for this is not very clear, but it could be something with the physiology of that particular muscle.

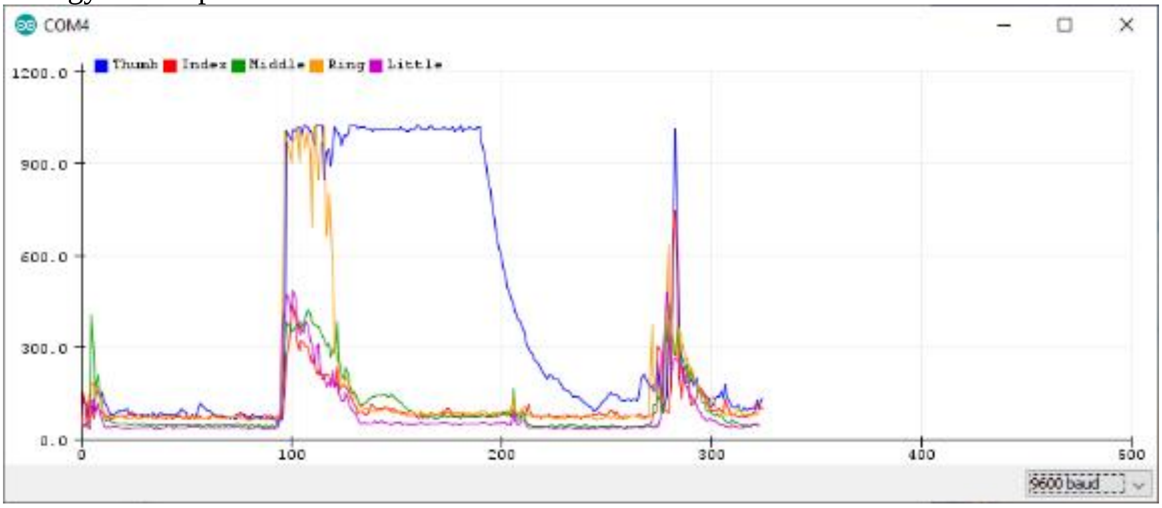


Figure 3.20: Thumb sensor output remaining high after clenching fist.

Another problem encountered early on was that having the sensor boards placed directly on the arm, as shown in figure 3.21, was unwieldy and produced high outputs with when moved. Rotation of the forearm resulted in some of the sensor boards being pulled and pressed against the forearm and producing a high output value.

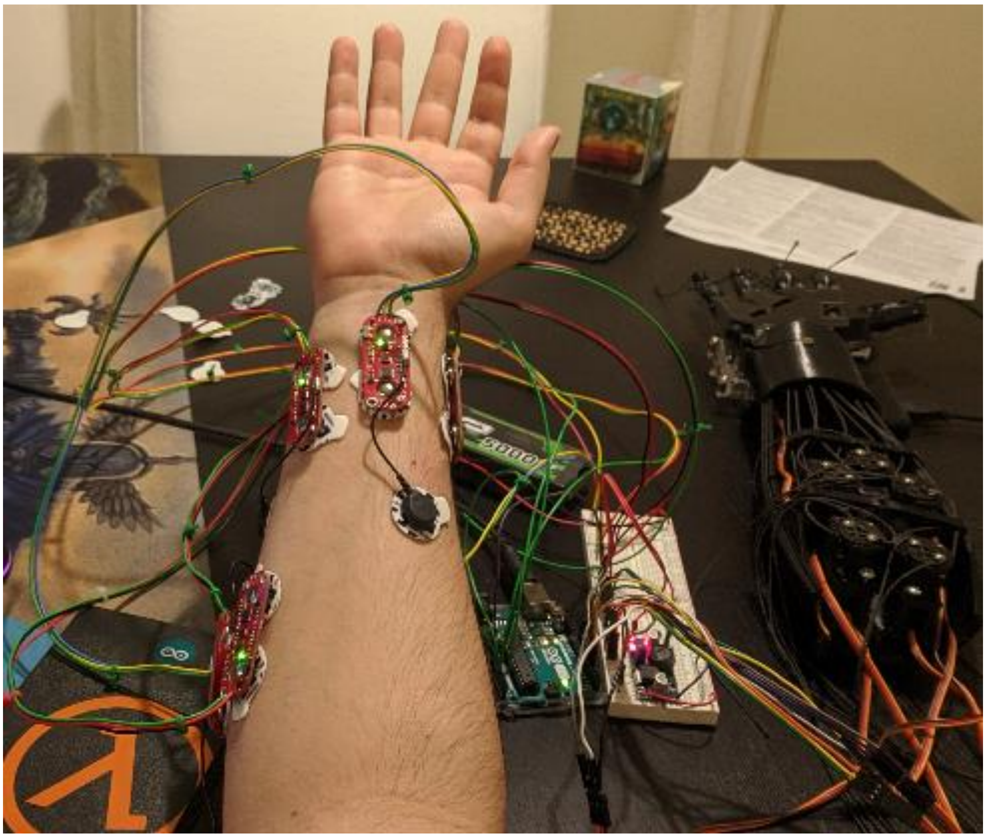


Figure 3.21: Early testing setup.

This was solved by soldering electrode snap cables to the sensor boards, as shown in figure 3.22.

Figure 3.22: Testing with electrode cables.

One interesting problem was that placing an electrode over a vein resulted in the sensor board picking up the user's heartbeat.

3.5 Results

The initial tests with the sensors started in late January. The program worked very well from the beginning, only needing to change threshold values and the position of the electrodes to get the mechanical hand to mimic the individual finger movement of the user.

Once a good electrode placement was found and the threshold values were tweaked by observing the output on the serial port of the Arduino, several trials with different conditions were performed. First, individual control of a single finger was tested, as shown in table 3.1.

Table 3.1: Single finger control

Test	Number of Trials	Number of Passes	Number of Fails
1st digit (thumb)	50	50	0
2nd digit (index)	50	49	1
3rd digit (middle)	50	50	0
4th digit (ring)	50	49	1
5th digit (little finger)	50	50	0

Control of single fingers had a fail rate of 0.4%. The extrinsic muscle flexors for the index and middle fingers are very close to each other and the activation of one results in the partial activation of the other. The same can be said for the ring and little finger. Proper electrode placement and adequate threshold values take care of most these false positives. The two failures in these trials were of this kind.

Following this, control of two fingers simultaneously was tested, as sown in table 3.2.

Table 3.2: Dual finger control

Test	Number of Trials	Number of Passes	Number of Fails
1st & 2nd	30	27	3
1st & 3rd	30	30	0
1st & 4th	30	29	1
1st & 5th	30	30	0
2nd & 3rd	30	28	2
2nd & 4th	30	30	0
2nd & 5th	30	29	1
3rd & 4th	30	30	0
3rd & 5th	30	30	0
4th & 5th	30	29	1

For control of two fingers simultaneously, the fail rate was 0.267%. These failures consisted on the flexion of an additional third finger to the two that were intended to be flexed. These were most likely caused by the test user involuntarily flexing adjacent muscles. This could be fixed with better threshold values. Next, the flexion of three fingers was tested, as shown in table 3.3.

Table 3.3: Control of three fingers

Test	Number of Trials	Number of Passes	Number of Fails
1st, 2nd, & 3rd	30	26	4
1st, 2nd, & 4th	30	27	3
1st, 2nd, & 5th	30	29	1
1st, 3rd, & 4th	30	29	1
1st, 3rd, & 5th	30	25	5
1st, 4th, & 5th	30	27	3
2nd, 3rd, & 4th	30	28	2
2nd, 3rd, & 5th	30	27	3
2nd, 4th, & 5th	30	25	5
3rd, 4th & 5th	30	27	3

The fail rate for three fingers was 1%. These failures are most likely the result of the difficulty of exerting force with three different fingers without also applying some involuntary force with the muscle of an adjacent finger. Following this, the flexion of four simultaneous fingers was tested, as shown in table 3.4.

Table 3.4: Control of four fingers

Test	Number of Trials	Number of Passes	Number of Fails
Sans 1st	40	38	2
Sans 2nd	40	35	5
Sans 3rd	40	37	3
Sans 4th	40	40	0
Sans 5th	40	40	0

For flexion of four fingers, the fail rate was 5%. Flexing four fingers simultaneously require quite a bit of force to be exerted for the sensor signals to pass their respective threshold values at the same time. This makes it difficult to the user not to apply some force to the finger that is intended to remain extended. Then, the closure of the fist was tested, as shown in table 3.5.

Table 3.5: Closure of all fingers

Test	Number of Trials	Number of Passes	Number of Fails
All	50	45	5

For flexion of all the fingers, the fail rate was 10%. However, these failures consisted of the thumb remaining flexed for a few seconds after the user had relaxed his thumb as discussed before figure 3.20.

The failures in these trials are mostly due to the involuntary contraction of a muscle that is not intended to be contracted, suboptimal threshold values, and in the case of the flexion of all the fingers, to suboptimal electrode placement for the thumb.

Finally, grasping and holding of different objects was tested, as shown in table 3.6.

Table 3.6: Grasping and holding different objects

Test	Number of Trials	Number of Passes	Number of Fails
Tennis ball	30	19	11
Coffee mug	30	30	0
Bottle of pills	30	24	6
Bottle of mustard	30	2	28
Brick of ground coffee (283g)	30	10	20
Scissors	30	30	0
Cyanoacrylate bottle	30	30	0

For larger and heavier objects, like a brick of coffee and a bottle of mustard, the hand could not keep a hold of the object as they would slip away. Flat pieces of rubber were cut and glued to the palm of the hand and ventral sides of the first phalanges, as shown in figure 3.23, to improve gripping, but it did not help much for these heavy objects.

Figure 3.23: Rubber paddings for better gripping.

For objects with handles, like scissors or a coffee mug, grabbing and keeping a hold of the object is not a problem.

For cylindrical objects, such a bottle of pills, grasping is not a problem. Holding the object, however, was very difficult. After gluing the rubber pads this was solved, only letting go of the object when vigorous movement was performed.

For spherical objects, like a tennis ball, grasping is the main problem. Successful grasping depended on positioning the object between the first three digits rather than on the center of the palm. Once grabbed, holding the object was not a problem.

4. Non-Technical Issues

4.1 Budget

Tables 4.1 and 4.2 show the proposed budget and the actual budget of the project.

Table 4.1: Proposed Budget

Part	Price	QTY.	Total
Arduino Uno R3	$18.20	1	$18.20
MyoWare Muscle Sensor	$37.99	5	$189.95

MyoWare Electrodes (50-Pack)	$24.99	1	$24.99
Dorisea Extreme braided fish line 200LB (300m)	$23.98	1	$23.98
Tower Pro MG996R 180º Robotic Servo	$15.95	5	$79.75
Flexible Nylon Paracord (100ft)	$10.99	1	$10.99
3D printed hand and forearm (based on the InMoov hand and forearm)	$217.18	1	$217.18
EMG/ECG Snap Electrode Cables (10-pack)	$39.99	1	$39.99
Ovonic 7.4V 5000mAh 50C 2S LiPo Battery Pack	$16.99	1	$16.99
Total			$622.02

Table 4.2: Actual Budget

Part	Price	QTY.	Total
Arduino Uno R3	$18.20	1	$18.20
MyoWare Muscle Sensor	$37.99	5	$189.95
MyoWare Electrodes (50-Pack)	$24.99	3	$74.97
Emma kites 100% Braided Kevlar String 100ft 200lb	$12.99	1	$12.99
Tower Pro MG996R 180º Robotic Servo	$15.95	5	$79.75
Filament to print InMoov hand and forearm (1kg spool)	$22.99	1	$22.99
EMG/ECG Snap Electrode Cables (10-pack)	$39.99	1	$39.99
Ovonic 7.4V 5000mAh 50C 2S LiPo Battery Pack	$16.99	1	$16.99
Tenergy TN267 1-4 Cells Li-Po/Li-Fe Balance Charger	$24.99	1	$24.99
Total			$480.82

The difference between the proposed budget and actual budget is $141.20. This difference is mainly due to having the hand and forearm pieces printed at home and at Valencia College's 3D printing lab instead of using a 3D printing service.

4.2 Timeline

Table 4.3 shows the timeline of the project.

Table 4.3: Timeline

Week	Proposed	Actual (Rodolfo)	Actual (Poposky)
November 11, 2019	Proposal Report Submission.	Proposal Report Submission.	
November 18, 2019	Proposal Presentation.	Proposal Presentation.	
November 25, 2019	Start 3D modeling of hand and forearm. Ordering parts for sensor and power modules. Start working on the microcontroller code.	Printed, assembled and wrote code for finger prototype. Tested prototype.	

December 2, 2019	Printing, assembling and testing finger prototype.	Printed and assembled hand and forearm.	
December 9, 2019	Finish modeling hand and forearm. Printing and assembling. Test flexion and extension of the fingers with the microcontroller.	Waiting to receive power module from Poposky.	
December 16, 2019	Testing sensor and power modules. Continue working on the code.	Waiting to receive power module from Poposky.	Work on Code to test Sensors
December 23, 2019	Christmas	Tested flexion and extension of the hand's fingers with the microcontroller.	Keep working on the code.
December 30, 2019	New year.		Troubleshoot the sensors with the Code
January 6, 2020	Put testing modules together and start testing with the microcontroller and hand. Work on the sensor part of the code.	Waiting to receive the rest of the sensors from Poposky.	Practice Design in Inventor.
January 13, 2020	Troubleshooting and testing.	Waiting to receive the sensors from Poposky.	Forearm Design
January 20, 2020	Troubleshooting and testing.	Put modules together and started testing with the microcontroller and hand.	
January 27, 2020	Troubleshooting and testing.	Troubleshooting and testing. Started working on the report.	Forearm Design
February 3	Final tests.	Troubleshooting and testing.	Forearm Editing and Reprint
February 10	Start working on report.	Final tests.	Start working on Report.
February 17	Continue the report.	Continue the report.	Continue the report.
February 24	Finish first draft of the report and start working on the presentation.	Continue the report.	Continue the report.
March 2	Make corrections to the report and submit it. Finish presentation.	Worked on the PCB.	Edited the designed parts.
March 9	Practice presentation.	Build and tested circuit board.	Printed the edited designed parts.

March 16	Improve website.	Printed additional hand. Worked on website and report	Assembled the designed parts together.
March 23	Make additional corrections to the report if needed and submit it.	Worked on the report.	Adjusted and Reprinted the Forearm, since the designed part was smaller than the PCB dimensions.
March 30	Project Report Submission	Assembled new forearm as much as possible. Tested circuit board again with the old hand. Finished the report.	Report Editing
April 6	Project Presentation	Project Report Submission (April 10)	
April 13		Project Presentation (April 17)	

4.3 Environmental Aspects

The hand is made of relatively inexpensive polymer, making it recyclable. The electronics and other components could be reused if needed. If a piece of the hand or arm breaks, it can be shredded and melted to be reused as filament for 3D printing.

4.4 Health and safety

The system developed is electrically safe and not harmful when used properly. The main safety concern is the Lithium polymer battery, which can be a fire hazard if damaged, mishandled, or not charged properly. It is recommended to charge the battery using a charger designed for Lithium polymer batteries, like the Tenergy TN267 1-4 Cells Li-Po/Li-Fe balance charger used.

4.5 Sustainability

All the electronics components are commercially available and easy to replace. If any part of the hand itself breaks, a replacement can be printed relatively quickly and installed. One major consideration is the gel electrodes. At least eleven electrodes are needed for every day that the EMG-Controlled Mechanical Hand is used, and in the long run they are quite expensive. They alone accounted for 15.6% of the project's budget.

5. Conclusion

5.1 Summary and Conclusions

The EMG-Controlled Mechanical hand proved that control of individual fingers of a mechanical hand could be achieved with electromyography. The main application of this project would be its implementation in low-cost prosthetic hands, providing amputees with an intuitive control of each of their fingers.

Other uses for this project are the control of a robotic hand through telepresence and the control of a virtual hand in virtual reality for amputees.

5.2 Suggestions for Future Work

This project was constrained in time, budget, and group members. With more time, money, and a group with more diverse and specialized knowledge, this project could be improved in the following ways:

First, wrist rotation could be added. The muscles that control flexion, extension, radial deviation, pronation, and supination of the hand are underneath the muscles in the forearm that control flexion and extension of the fingers, and the electrodes we use can pick up their electrical activity. Additional signal processing to differentiate between the surface muscles and deep muscles will be needed.

Second, finger radial deviation and full range of motion for each finger could be added. Besides the extra amount of servo motors and the added complexity to the design of the mechanical hand, this would require an additional amount of signal processing and pattern recognition.

Third, custom components can be made. Using custom-made servo motors that could be installed in the palm of the mechanical hand would free up space in the inside of the forearm. A new PCB could be designed with a custom five-channel EMG circuit. An ATmega328 microcontroller could be integrated into the PCB instead of using an Arduino Uno. A 5V battery could be used instead of the 7.4V one, eliminating the need for a step-down voltage regulator.

Fourth, machine learning could be implemented to set the threshold values instead of entering them manually every time electrodes are placed on the user.

Fifth, conductive fabric electrodes could be used instead of traditional electrodes. Conductive fabric electrodes can be sown on clothing, like as a compression arm sleeve, such that the electrodes are always placed on the intended muscle group. This type of electrodes are reusable and less expensive over time than gel electrodes.

References

[1] G. McGimpsey and T. Bradford, *Limb Prosthetics Services and Devices Critical Unmet Need: Market Analysis*, Bioengineering Institute Center for Neuroprosthetics, Worcester Polytechnic Institution, 2017.

[2] R. Turner, “Prosthetics Costs: The High Price of Prosthetic Limbs,” *Disabled World*, 08-Nov-2018. [Online]. Available at: https://www.disabled-world.com/assistivedevices/prostheses/prosthetics-costs.php. [Accessed: 05-Nov-2019].

[3] Upper Limb Prosthetics, “Upper Limb Prosthetics info,” *Upper Limb Prosthetics*, [Online]. Available at: http://upperlimbprosthetics.info/index.php?p=1_9_Body-Powered [Accessed: 14-Nov-2019].

[4] Brain Gate, “Movement Restoration,” *Brain Gate*, [Online]. Available at: https://www.braingate.org/research-areas/movement-restoration/ [Accessed: Nov-14-2019].

[5] Zenaida Gonzalez, “Spam Email Leads to Big Gift,” *UCF*, 11-July-2018. [Online]. Available at: https://www.ucf.edu/news/spam-email-leads-big-gift/ [Accessed: Nov-14-2019].

[6] Arduino, “Arduino Uno Rev3,” *Arduino*. [Online]. Available at: https://store.arduino.cc/usa/arduino-uno-rev3 [Accessed: 08-Nov-2019].

[7] TowerPro, “MG946R,” *TowerPro*. [Online]. Available at: http://www.towerpro.com.tw/product/mg946r/ [Accessed: 08-Nov-2019].

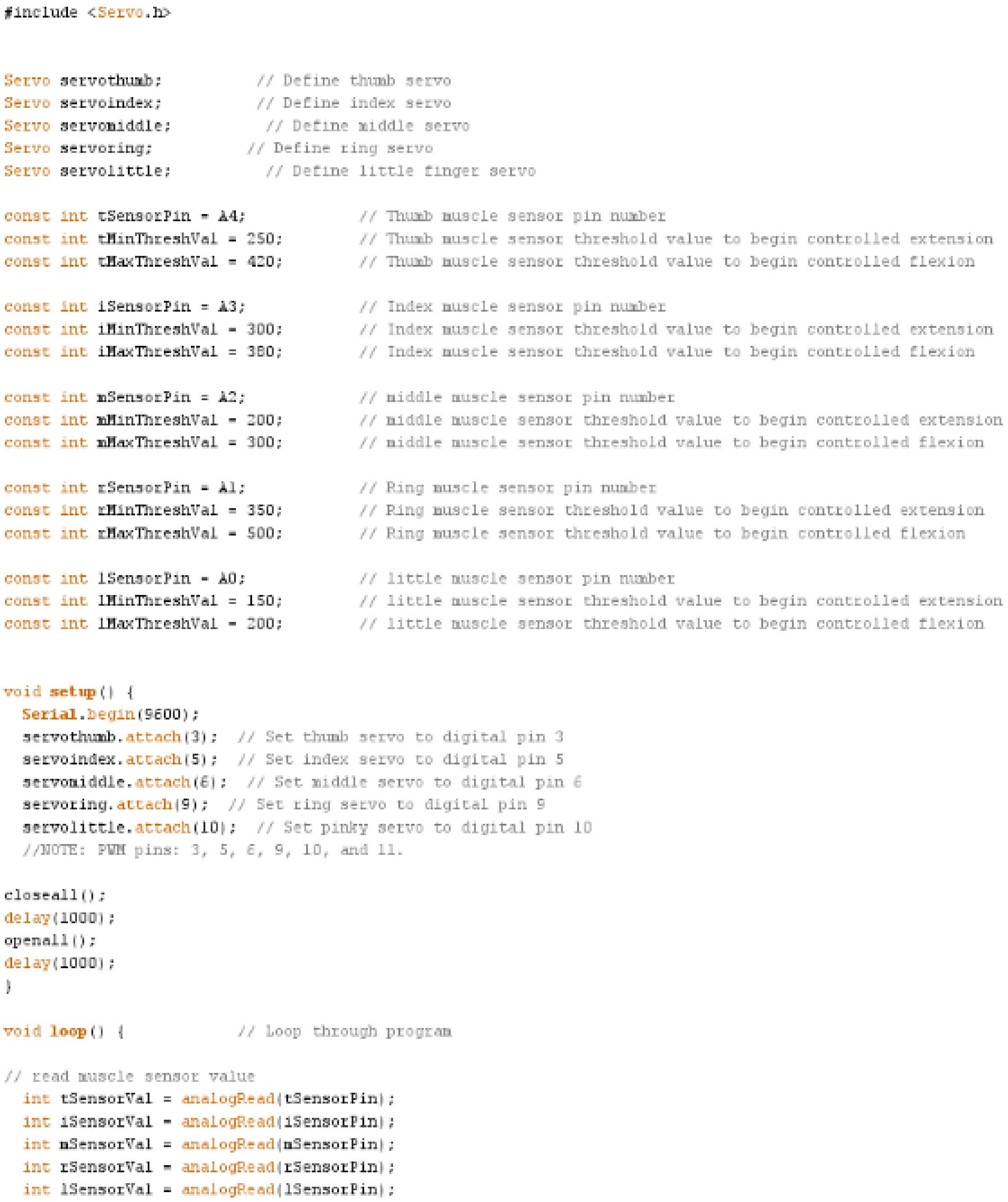
[8] Olstad, Bjørn. *Muscle activation and kinematics in contemporary breaststroke swimming*, Norwegian School of Sport Sciences (NIH), 2015.

[9] Advancer Technologies, “MyoWare Muscle Sensor,” *Advancer Technologies*. [Online]. Available at: http://www.advancertechnologies.com/p/myoware.html [Accessed: 08-Nov-2019].

[10] Gael Langevin, “Hand and Forearm,” *InMoov*. [Online]. Available at: https://inmoov.fr/hand-and-forarm/ [Accessed: 11-Nov-2019].

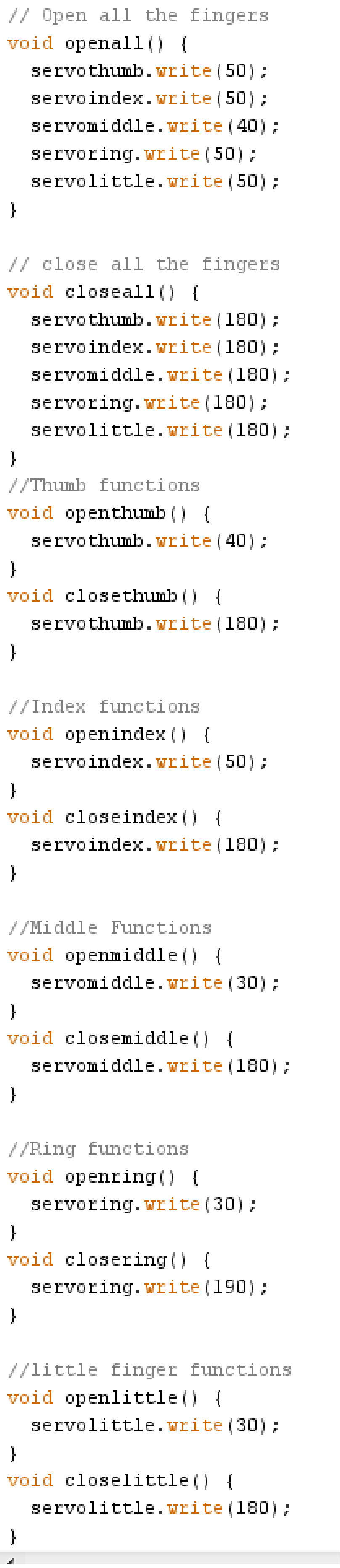
[11] Gael Langevin, “Dilemna with "WristsmallV3" part,” *InMoov*. [Online]. Available at: http://inmoov.fr/dilemna-with-wristsmallv3-part/ [Accessed: 11-Nov-2019].

Appendix A - Code



```
#include <Servo.h>


Servo servothumb;          // Define thumb servo
Servo servoindex;          // Define index servo
Servo servomiddle;          // Define middle servo
Servo servoring;          // Define ring servo
Servo servolittle;          // Define little finger servo

const int tSensorPin = A4;            // Thumb muscle sensor pin number
const int tMinThreshVal = 250;        // Thumb muscle sensor threshold value to begin controlled extension
const int tMaxThreshVal = 420;        // Thumb muscle sensor threshold value to begin controlled flexion

const int iSensorPin = A3;            // Index muscle sensor pin number
const int iMinThreshVal = 300;        // Index muscle sensor threshold value to begin controlled extension
const int iMaxThreshVal = 380;        // Index muscle sensor threshold value to begin controlled flexion

const int mSensorPin = A2;            // middle muscle sensor pin number
const int mMinThreshVal = 200;        // middle muscle sensor threshold value to begin controlled extension
const int mMaxThreshVal = 300;        // middle muscle sensor threshold value to begin controlled flexion

const int rSensorPin = A1;            // Ring muscle sensor pin number
const int rMinThreshVal = 350;        // Ring muscle sensor threshold value to begin controlled extension
const int rMaxThreshVal = 500;        // Ring muscle sensor threshold value to begin controlled flexion

const int lSensorPin = A0;            // little muscle sensor pin number
const int lMinThreshVal = 150;        // little muscle sensor threshold value to begin controlled extension
const int lMaxThreshVal = 200;        // little muscle sensor threshold value to begin controlled flexion


void setup() {
  Serial.begin(9600);
  servothumb.attach(3);  // Set thumb servo to digital pin 3
  servoindex.attach(5);  // Set index servo to digital pin 5
  servomiddle.attach(6);  // Set middle servo to digital pin 6
  servoring.attach(9);  // Set ring servo to digital pin 9
  servolittle.attach(10);  // Set pinky servo to digital pin 10
  //NOTE: PWM pins: 3, 5, 6, 9, 10, and 11.

closeall();
delay(1000);
openall();
delay(1000);
}

void loop() {            // Loop through program

// read muscle sensor value
  int tSensorVal = analogRead(tSensorPin);
  int iSensorVal = analogRead(iSensorPin);
  int mSensorVal = analogRead(mSensorPin);
  int rSensorVal = analogRead(rSensorPin);
  int lSensorVal = analogRead(lSensorPin);
```

```
//Debugging to see values

  Serial.print("Thumb:");
  Serial.print(tSensorVal);
  Serial.print(" ");

  Serial.print("Index:");
  Serial.print(iSensorVal);
  Serial.print("  ");

  Serial.print("Middle:");
  Serial.print(mSensorVal);
  Serial.print("  ");

  Serial.print("Ring:");
  Serial.print(rSensorVal);
  Serial.print("  ");

  Serial.print("Little:");
  Serial.println(lSensorVal);
  Serial.print("  ");


  if(tSensorVal < tMinThreshVal)      //if statement for Thumb
  {
     openthumb();
  }
  else if(tSensorVal > tMaxThreshVal)
  {
    closethumb();
  }


  if(iSensorVal < iMinThreshVal)      //if statement for Index finger
  {
     openindex();
  }
  else if(iSensorVal > iMaxThreshVal)
  {
    closeindex();
  }


  if(mSensorVal < mMinThreshVal)      //if statement for Middle finger
  {
     openmiddle();
  }
  else if(mSensorVal > mMaxThreshVal)
  {
    closemiddle();
  }


  if(rSensorVal < rMinThreshVal)      //if statement for Ring finger
  {
     openring();
  }
  else if(rSensorVal > rMaxThreshVal)
  {
    closering();
  }


   if(lSensorVal < lMaxThreshVal)      //if statement for Little finger
  {
     openlittle();
  }
  else if(lSensorVal > lMinThreshVal)
  {
    closelittle();
  }

  delay(100);
}
```



```
// Open all the fingers
void openall() {
  servothumb.write(50);
  servoindex.write(50);
  servomiddle.write(40);
  servoring.write(50);
  servolittle.write(50);
}

// close all the fingers
void closeall() {
  servothumb.write(180);
  servoindex.write(180);
  servomiddle.write(180);
  servoring.write(180);
  servolittle.write(180);
}
//Thumb functions
void openthumb() {
  servothumb.write(40);
}
void closethumb() {
  servothumb.write(180);
}

//Index functions
void openindex() {
  servoindex.write(50);
}
void closeindex() {
  servoindex.write(180);
}

//Middle Functions
void openmiddle() {
  servomiddle.write(30);
}
void closemiddle() {
  servomiddle.write(180);
}

//Ring functions
void openring() {
  servoring.write(30);
}
void closering() {
  servoring.write(190);
}

//little finger functions
void openlittle() {
  servolittle.write(30);
}
void closelittle() {
  servolittle.write(180);
}
```

Appendix B – InMoov Hand, Wrist, and Forearm

STL files of the InMoov hand, wrist, and forearm. Files are available at: inmoov.fr/inmoov-stl-parts-viewer/

Rodolfo de Curtis

- Major: Electrical & Computer Engineering Technology - Electrical Systems Concentration.
- Research interests include brain-computer interfacing, biomedical applications of electronics engineering, neuroscience, and space exploration.
- GPA: 3.46
- Recipient of the 2019 IEEE Orlando Section Outstanding Undergraduate Student Scholarship Award.
- Member of IEEE.
- Graduated from high school in Caracas, Venezuela in 2013 with honors.

✉ rodolfodecurtis@ieee.org

Poposky Cherichel

- Major: Electrical & Computer Engineering Technology - Electrical Systems Concentration
- Member of Florida Engineering Society and STEM @ SIEMENS
- Graduated from Fort Lauderdale High School in 2014