ECE 445 Final Report

Glove For Programmable Prosthetic Hand

Team Number: 55

Ryan Metzger (rtm4)

Quang Nguyen (quangpn2)

Sohil Pokharna (sohilp2)

Mentor TA: Barbhai, Sainath

Date:5/3/2023

Table of Contents

1 Introduction

1.1 Problem: Modern robotic prosthetics may achieve fine motor control through predefined hand motions encoded into the prosthetic. Modern prosthetics may have the ability to save preset positions but don't have the ability to adjust and tweak positions on the fly. We implemented a hardware/software solution that is able to measure the positions of a functional organic hand and translate this motion into a prosthetic hand in order for this prosthetic to mimic this motor control on the move. With features such as mirroring mode, we are able to have 2 hands, 1 organic and a simulated prosthetic. This glove is able to introduce a level of dynamic programmability through multiple preset positions through organic hand gestures. By adding sensors to individual fingers, we can combine combinations of gestures in order to control the prosthetic beyond mirror mode and be able to change the preset positions using these gesture controls.

1.2 Solution: We propose to create a glove with flex sensors and hall effects that can measure the motion of the fingers and detect gestures that the user gives. From this the user can then control a robotic hand with their organic hand making it easier to adjust position as well as record motions for the robotic hand to execute.

1.3 Functions: The product records an organic hand and either saves positions from the organic hand or mimics the movement of the organic hand.

1.4 Benefits: It provides the user with the ability to make quick and easy changes to their robotic hand while on the go. It also allows the user to mimic the user's organic hand's movement.

1.5 Features: The parts of this project that make it marketable is the quick and easy reprogrammability on the fly and the ability to mimic the user's hand movement.
1.6 Visual Aid:

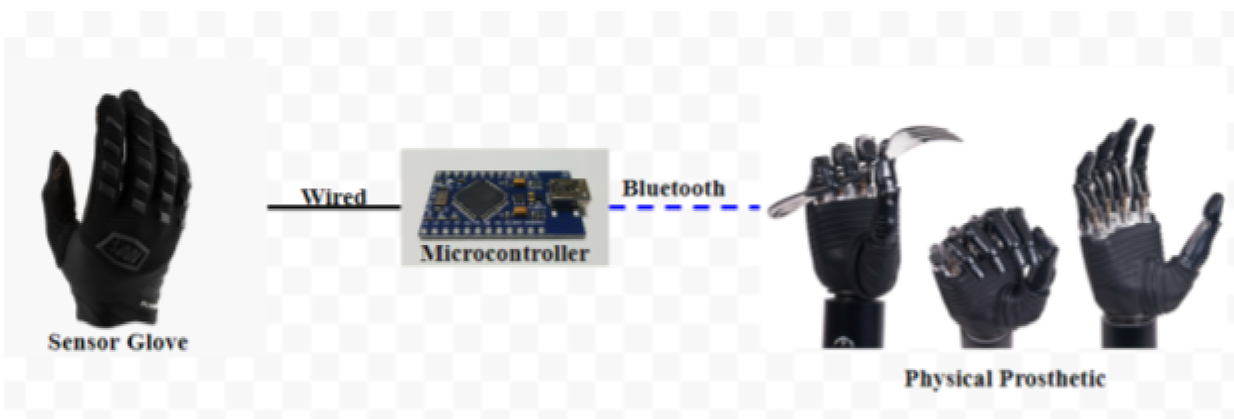


Figure 1.6.1: Visual Aid for Glove for Programmable Prosthetic Hand

1.7 High-level requirements list:

1. Have the ability to detect each of the 5 fingers' movement and also move the digital/physical robotic hand appropriately
2. Must detect 4 gestures at minimum
3. Must have the ability to move to 3 preset positions and mirroring mode

1.8 Block Diagram:

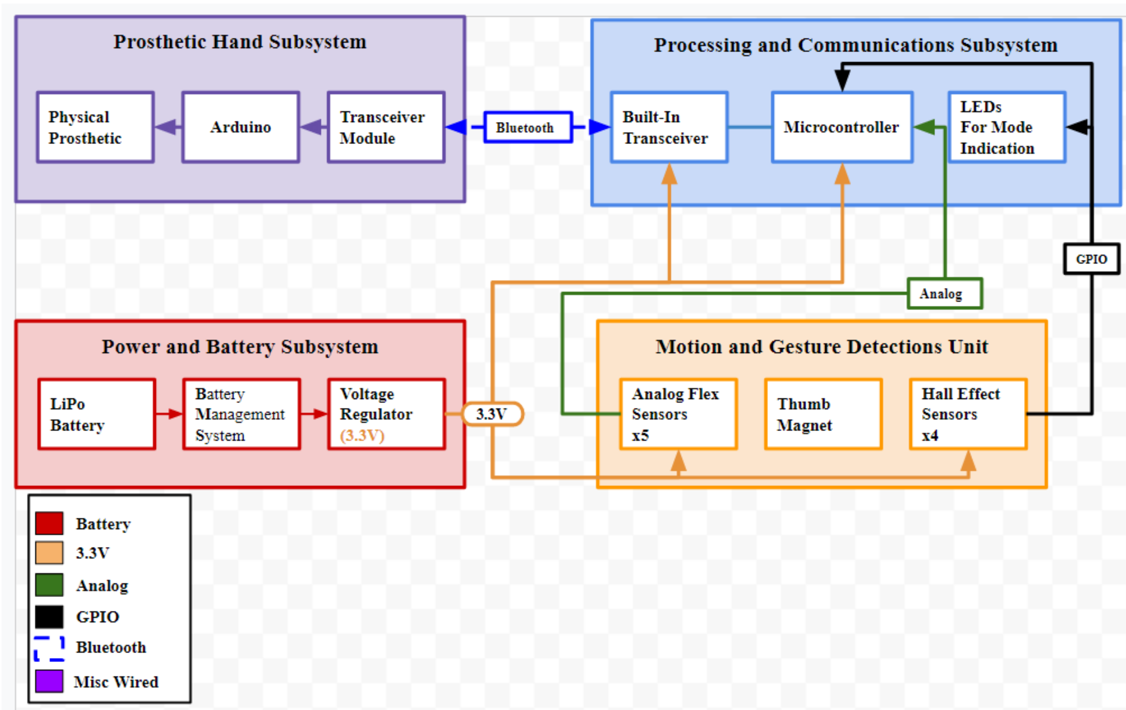


Figure 1.8.1 Block Diagram Showing Subsystems

As seen in Figure 1.8.1 we have four main subsystems, the Prosthetic Hand subsystem, the Processing and Communications Subsystem, the Power and Battery Subsystem, and the Motion and Gesture Detections Unit. The prosthetic hand subsystem is the subsystem that encompasses the robotic hand for the demo. It is made up of servos and an Aduino to control it. The prosthetic hand subsystem gets the information of the flex sensors from the processing and communication subsystem which then translates that in movements in the fingers and hand using the servos. The information from the motion and gesture detections unit is sent into the processing subsystem which is connected to the prosthetic hand through bluetooth.. The motion and gesture detection unit is made up of 4 hall-effect sensors and 5 flex sensors to get the orientation of the hand and the mode the glove is in. This system detects changes in the fingers which is read as change in the resistance values, it also takes in user inputs through the use of hall effect sensors which function to change modes and the behavior seen in the robotic hand. The power and battery subsystem regulates the voltage from a LiPo battery to a voltage usable to the microcontroller and sensors suite. Keep in mind that the sensors/microcontroller are isolated systems from the prosthetic hand and run off of different power supplies, as such the power subsystem does not affect the prosthetic hand. This system also features a back current protection in case the user connects the battery backwards. The processing and communication subsystem is used to interpret values from the motion and gesture detections unit. It uses the microcontroller on board ADC to read the change in voltage from the flex sensors in order to determine flex, these values are then sent and mapped onto the hand which is seen as motion. A similar process is used where we set the hall effect sensors as digital inputs and determine which finger was tapped and how many times. This is then interpreted by the microcontroller and various functions are achieved in software. The hand itself does not know about modes as it only receives the values sent by the microcontroller, as such all of the modes are programmed into the microcontroller and data sent is changed based on these modes.

1.9 Physical Design

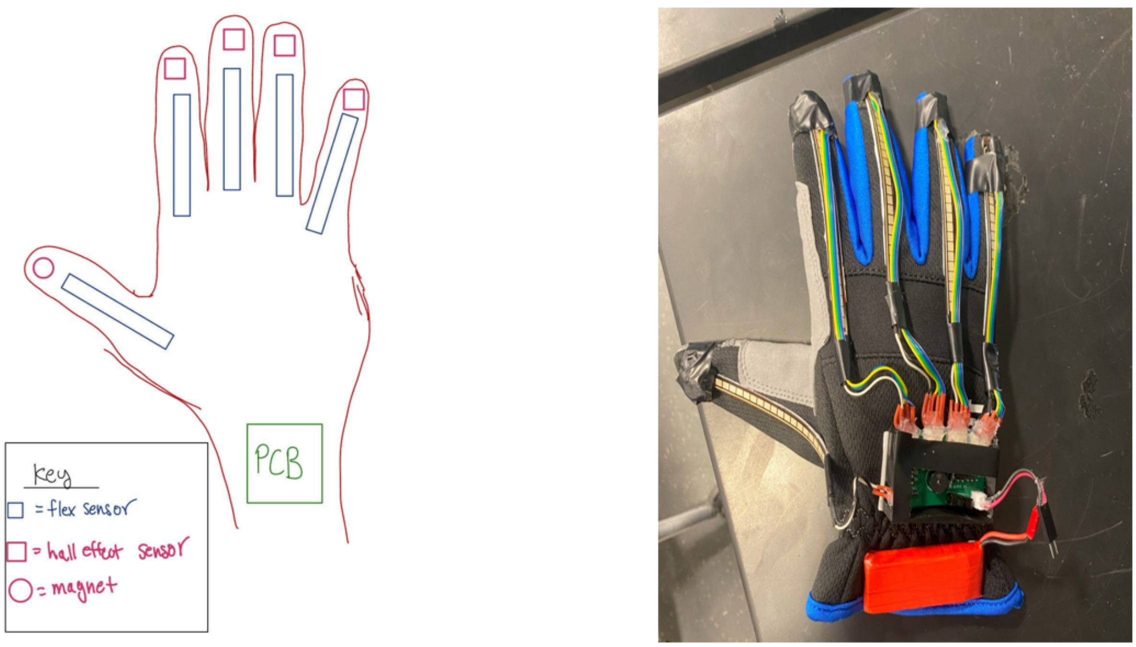


Figure 1.9 Physical Design

In the physical design you can see that the hall effect sensors are located in the tips of the four fingers and the magnet resides on the thumb. The flex sensors have to cover the middle and top knuckle at minimum. The pcb and battery pack will either reside on the wrist or the back of the hand.

1.10 Subsystem 1: Processing/Communication Unit Microcontroller -

Requirements
1) Make sure the processor can distinguish between all the different hall effects and set

the LEDs to the correct modes
2) Check to see if Bluetooth can transmit and receive information *introduce bitrate standard
3) Flex sensors are successfully read

1.11 Subsystem 2: Motion/Gesture Detections Unit.

Requirements
1) The hall effects getting triggered should set the modes/try and move the hand to preset positions
2) The flex sensors bending causes the finger to move to correct location with ±5%

1.12 Subsystem 3: Power/Battery System.

Requirement
1) Able to power both the sensors and microcontroller for 8 hours in continuous transmission mode

1.13 Subsystem 4 (Prosthetic Hand):

Requirements
1) Replicate hand position and commands given by sensor glove

2 Design:

2.1 Subsystem 1 Processing/Communication Unit Microcontroller:

The Microcontroller is responsible for processing Motion/Gesture Unit inputs, processing the data and sending the results via Bluetooth to the Hand. This unit features small LEDS that give the user feedback on what mode they are in/the preset options they are selecting.

Microcontroller must contain on-chip support for bluetooth LE and be able to run off 3.3V. The microcontroller must have on-chip Flash in order to support saving user configurations or an off chip-flash interfaces via SPI to achieve the same function. The micro has a power saving/sleep mode that will increase battery life of the system. The microcontroller must have at least 5 analog in pins and 4 digital in pins at the minimum and be able to run off the voltage regulator ripple specs. The User Feedback LEDs must be able to operate off of 3.3V.

2.2 Subsystem 2 Motion/Gesture Detections Unit:

The Motion/Gesture Unit is used to collect data about the user's finger positions as well as to control the unlocking/programmability it sends data to the Processing Unit.

The Detection Unit consists of 5 Analog Flex Sensors and 4 Hall Effect Sensors that interfaces with the microcontroller for sensor feedback and draws power at 3.3V from the Power/Battery System. The noise is small enough and magnetics tuned so that each hall effect sensor input is distinguishable from one another and can be repeatedly triggered. The flex sensors will interface with a resistor of the same nominal Resistance to form a voltage divider to read flex/finger positions may need to consider input impedance of ADC pins as some flex sensors have high impedance .

We started off by picking a hall effect sensor to use for our design. Ideally we wanted a hall effect that had a low sensitivity so that each finger could be triggered individually, without accidentally triggering hall effect sensors on nearby fingers. We did some research on low power hall effect solutions and found the DRV5032 line had current draw in the microampere range. To pick the sensitivity we went with the specific SMD model that had the medium sensitivity in this device line.

$$\vec{B} = \frac{B_r}{2}\left(\frac{D+T}{\sqrt{\left((0.5C)^2 + (D+T)^2\right)}} - \frac{D}{\sqrt{(0.5C)^2 + D^2}}\right)$$

Figure 2.2.1 Field Strength for Cylindrical Magnets

In order to pick a magnet that will trigger the hall effect sensors at the right distance, we used the equation provided in figure 2.2.1 for cylinder magnets that gives the magnetic strength at a given distance based on a magnet's dimensions and field strength.

To build the flex sensor unit we simply added resistors in series with each flex sensor with a resistance value of 20K shown in figure 2.2.2, the center point of the flex sensor's resistive range. We decided to go with flex sensors as they are low profile, consume minimal current , and are fairly linear devices. We also get a decent voltage range for the flex sensors with a output range of around (3.3V) * (0.6-0.33)V -> 27% of ADC range. Our ADC was set up for 10 bit resolution leading to 1024 possible values, 27% of this range is about 276 values. Since this range maps 0 to 180 degrees of flex in the sensors, our LSB equals 0.65 degrees which means that our **finger angle detection minimum resolution is 0.65 degrees.** We decided this was a granular enough resolution for our glove sensing.

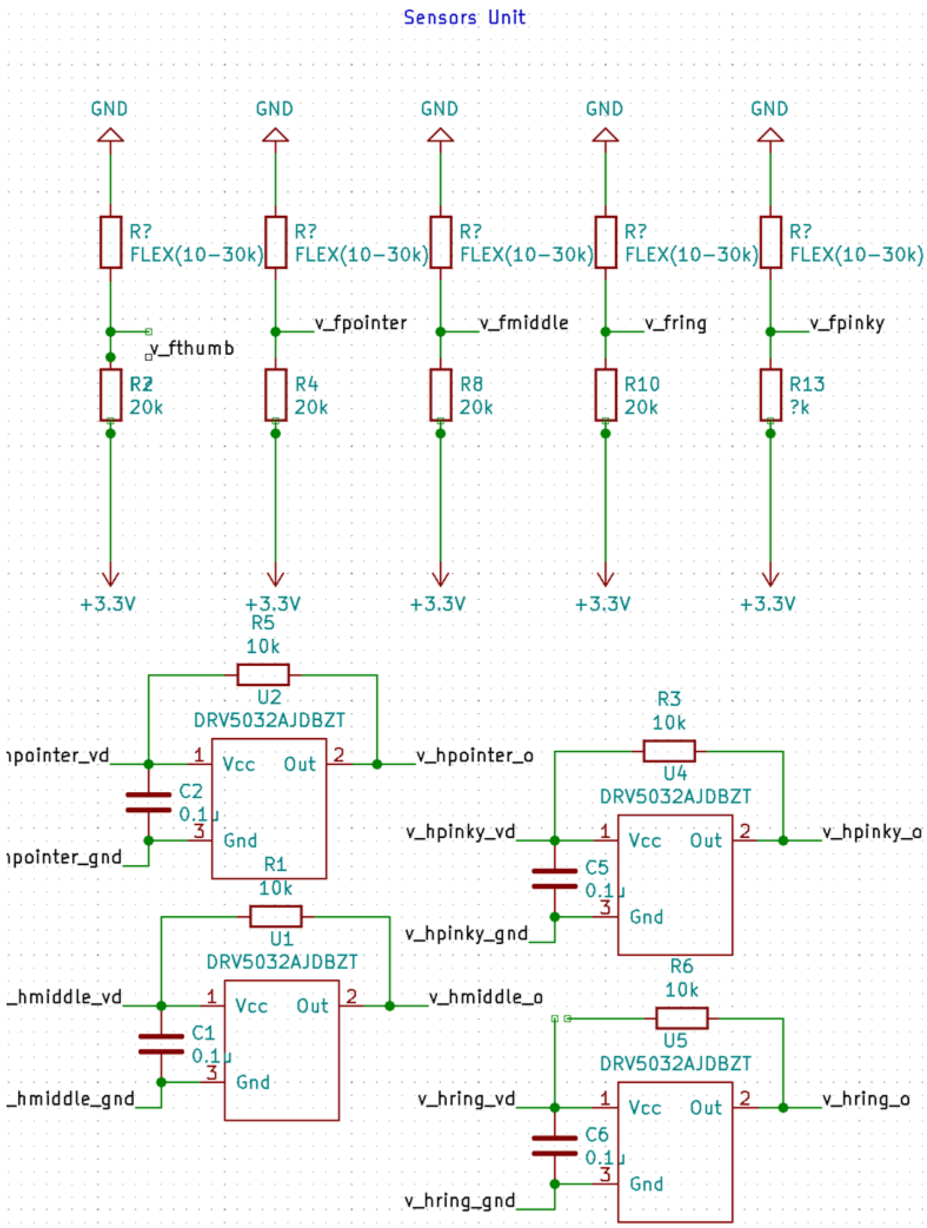


Figure 2.2.2 Schematic for Flex Sensors and Hall Effect Sensors

2.3 Subsystem 3 Power/Battery System:

The Power/Batter System takes a LiPo rechargeable battery and regulates the voltage/current to 3.3V so that our system functions. This is responsible for powering the Motion/Gesture Unit and the Processing Unit. There were two main parts to designing the power unit. We picked a Schottky Diode for reverse polarity protection on the battery connection, as well as an LDO to regulate the battery voltage to 3.3V to power our sensors/micro. For the regulator to function properly, the voltage from the battery has to overcome the forward drop of the diode as well as the dropout voltage of the LDO. We picked the diode and LDO to achieve the smallest voltage drops possible while still being able to handle the current of our loads with a peak of around 35mA. Also, smaller voltage drop across the diode/LDO means a better power efficiency[6.7], but at the cost of a higher reverse polarity leakage current. The circuit described is shown in the figure below Figure 2.2.3.

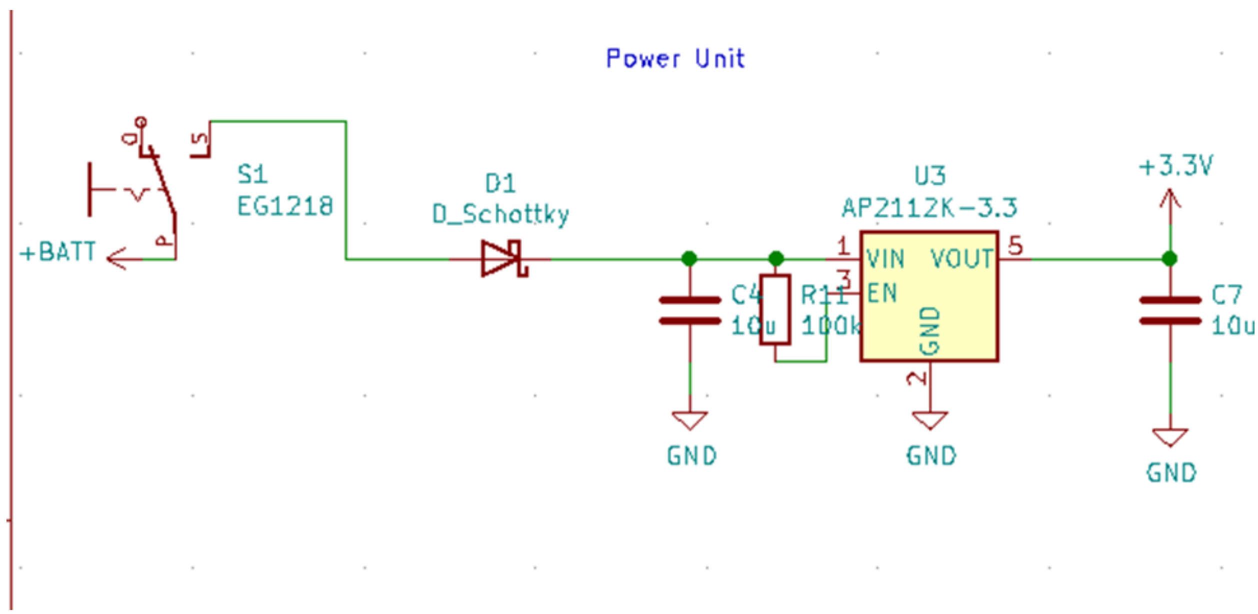


Figure 2.2.3: Power Unit

$$P_D = \left[\left(V_{IN} - V_{OUT} \right) \times I_{OUT} \right] + \left(V_{IN} \times I_{ground} \right)$$

Figure 2.2.3.1: Power Dissipated in an LDO

2.4 Subsystem 4 Prosthetic Hand:

This robotic hand was used to demonstrate the effectiveness of our project and interfaced with the Communication/Processing Unit via Bluetooth. 5 digits were modeled and the angle of the finger is adjustable. This hand responds to the 4 gesture commands such as prerecorded positions and mode switching. We modeled our hand using an open source robotic hand from InMoove.[6.2] Since our project is focused on modeling the organic hand and mimicking the movement into the robotic hand, we opted to use an open source hand as a way to devote more time into the sensing of the hand rather than the design of the robotic prosthetic.

This robotic hand uses the MG996 Servo Motor as the robotic hand is designed to integrate them into the model, this servo is able to provide a maximum of 1.49 N*m of torque at 6 V. These servos will transmit movement through multi braided fishing line wrapped around a 3D printed pulley since the breaking point of the fishing line will be beyond what the servos can provide, and the multi braided line provides repeated abrasion resistance compared to single filament line. For the 3D print material of the hand we are using PLA or Polylactic Acid filament. Compared to similar materials such as PETG or ABS, PLA is much easier to work/print with, has a lower cost per kilogram, and overall has a higher rigidity. Though PETG and ABS have a higher temperature resistance before deformation, the temperature range we plan to use the robotic hand in will be well within the acceptable range for PLA since PLA begins to deform at 140 Degrees Fahrenheit. For the fasteners and hardware we plan to receive assistance from the ECE Machine Shop such as springs, screws, nuts, etc.

To control the hand we are using an Arduino Nano BLE which is used to control all the servos in the hand. This Arduino then communicates with the sensor glove's microcontroller via bluetooth. The Arduino and Servos are powered using 2x 3.7V, 18650 Li-ion Batteries placed in series in order to provide a 7.4 V for the Arduino and Servos.

3 Verification:

The verifications that are not shown below are discussed in the appendix, section 7.1, and where conducted throughout the design process. These verifications did not require complex measurements

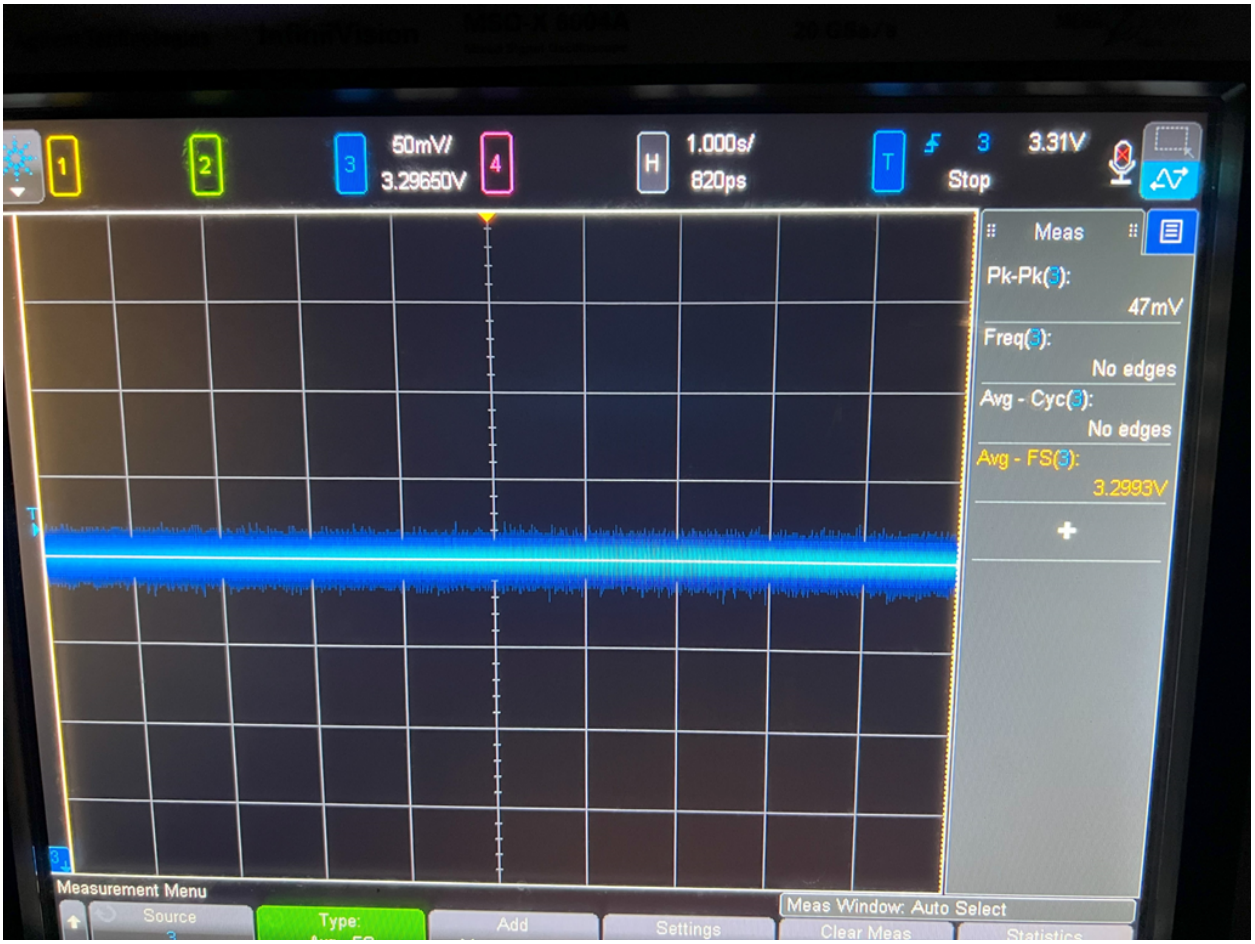


Figure 3.1: Battery Life Calculations

To verify our power subsystem we connected a probe to the output of our LDO to measure DC voltage as well as ripple at this node. In the design stage we aimed for 3.3V +- 0.1 for the DC accuracy and minimal supply ripple. The main reason we wanted minimal ripple is that our flex sensor unit is essentially a resistive voltage divider between the LDO output and ground, so any ripple across the output of the LDO will appear at the measuring node of the flex sensor.We far exceeded these specs as our measured DC output was 3.299V and our ripple was <1% show in Fig 3.1.

Power Calculations

	Battery Voltage(V)	Battery Current(mA)	Power(mW)
Idle	4.168	19.12	79.69
Mirroring	4.168	35.2	146.71

Passive(mW)	80
BLE(mW)	67

Solving for time we can power glove for approximately 38 hours.

LDO loses regulation closer to **30 Hours**.

Energy Equations

$$Energy_{Battery} = Energy_{passive} + Energy_{BLE}$$

$$Energy_{passive} = Power_{passive} * time$$

$$Energy_{BLE} = Power_{BLE} * (T_{packet} * \frac{time}{T_{rate}})$$

T_{rate} = 60ms time between transmissions

T_{packet} = 200us time transmitting each packet

$E_{battery}$= 11,000J

Figure 3.2. Battery Life Calculations

One of our requirements was to support at least 8 hours of continuous glove use. To estimate the battery life of our system we measured the power that our battery provided in passive mode(no BLE transmissions), as well as active mode(BLE transmitting). The BLE energy is only consumed when the device is transmitting data packets. We used an oscilloscope to measure the current draw, voltage draw and solved for power in each mode P = V*I. We found that our system draws an additional 15 mA when transmitting data compared to being idle.

Using Power/Energy relationships in Fig 3.2. We were able to solve for the estimated time that our system could run based on our LiPo model of storing 11,000J of usable energy. We found that our system could run for about 30 Hours before needing a charge. The energy equations also show that the majority of energy is consumed while in passive mode, with the additional energy from transmitting being small.

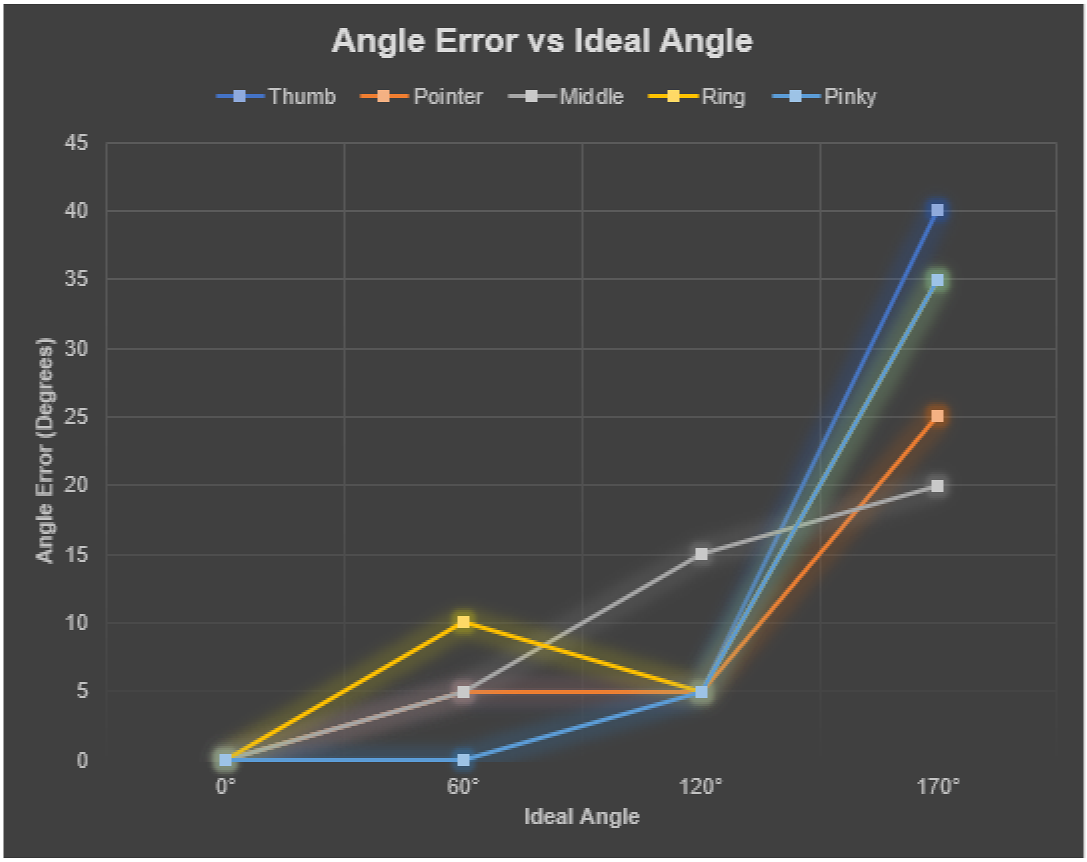


Figure 3.3: Organic Hand Vs Robotic Hand Accuracy

Another criterion was that the angle of the organic hand and the angle of the flex sensors be within 5% of each other. We quickly learned, however, that this level of accuracy was not possible with the robotic hand we were using. The open-source hand's tensioning system, as well as the physical restrictions of the joints, provided challenges that prevented us from obtaining this degree of precision. As demonstrated in the figure 3.3, the angle we selected for the hand and the actual angle of the robotic hand increased as it approached its lower limit. This was primarily due to issues with the robotic hand itself, as well as the strict constraints we had to impose on the servos to prevent the fishing lines attached to them from breaking.

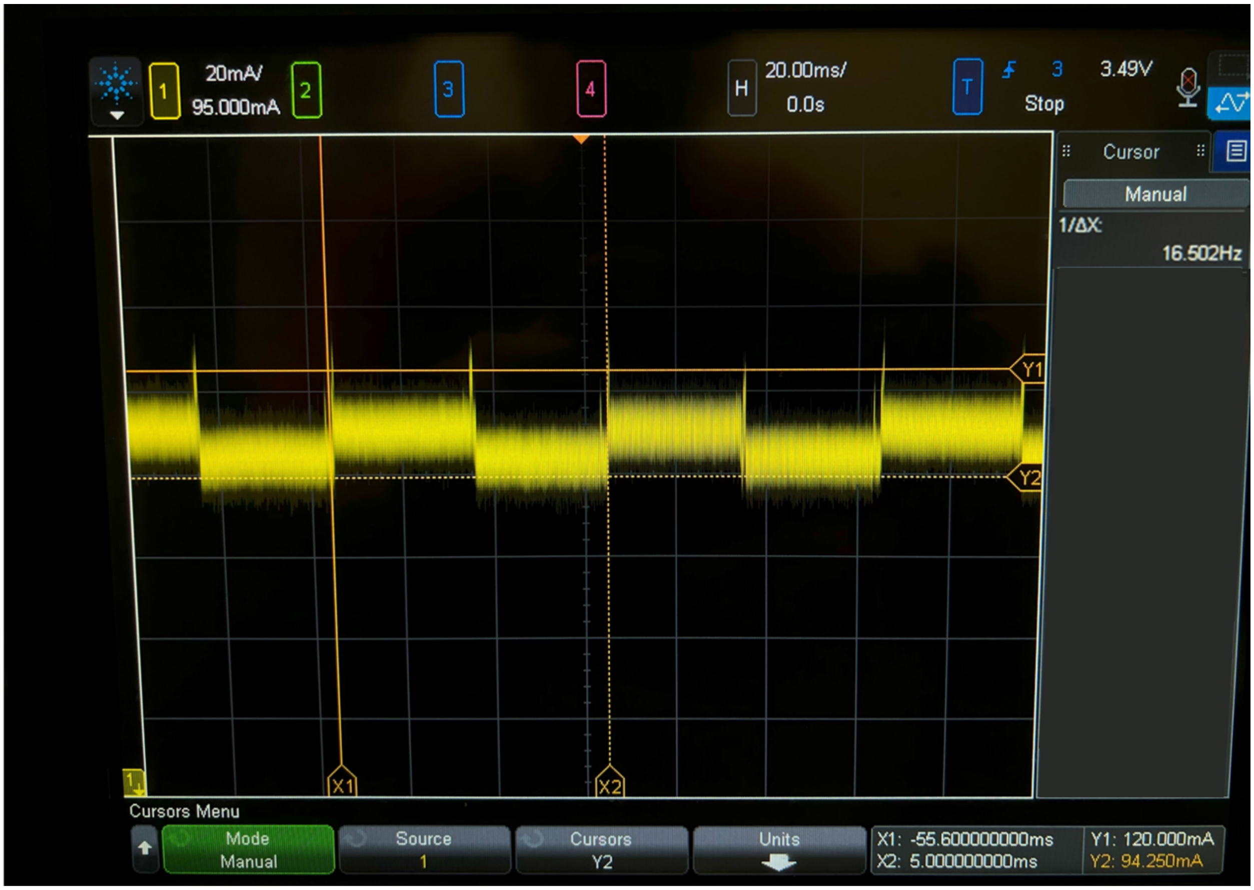


Figure 3.4: Probing Battery while in Mirroring Mode

Verification of Bluetooth Transmission Rate: We ran two tests to establish the rate at which Bluetooth could deliver data. The first test comprised measuring the current of the battery while the glove was in mirroring mode and measuring the resulting waveform, as seen in figure 3.4. Based on this, we calculated that the waveform's period was 16.502Hz.

The second test entailed connecting the Adafruit board to the Adafruit Bluetooth app and then setting a 20-second timer. We proceeded to count the number of data packets sent during this time period and determined the transmission rate by dividing the total number of packets by the 20-second timeframe. This approach likewise produced a 16.5Hz transmission rate.

4 Costs and Schedule:

4.1 Costs

Part Name	Manufacturer	Model	Unit Price($)	# Units	Cost($)
Flex Sensor	Spectra Symbol	FS-L-0055-253-ST	7.95	5	39.75
Glove Pair	Home Depot	Home Depot	15	1	15
Hall Effect	Texas Instruments	DRV5032AJDBZT	1.45	4	5.8
Magnet	K&J Manetics	D62 (K and J Magnetic	0.61	1	0.61
BLE Micro	Adafruit	1528-2835-ND	8.75	1	8.75
3.3V LDO	Diodes Incoporated	AP7387Q-33Y-13	1.55	1	1.55
Schottky Diode	Fairchild	Bat54	0.31	1	0.31
7.4V LiPo	Blomik	850mAh	11.99	1	11.99
25k Resistors	TE Connectivity	CPF-A-0805B25KE	0.48	5	2.4
330 Ohm Resistor	ROHM	SDR03EZPD3300	0.22	3	0.66
0.1uF Cap	Murata	GRT155R71C104KE01I	0.1	8	0.8
1uF Cap	Kemet	C2220X105K2RACAUT(	0.5	2	1
PCB	N/A	N/A	5	1	5
5050 SMD LED	ROHM	CSL1901VW1	3	0.5	1.5
				Glove Cost	$93.62

Part Name	Manufacturer	Model	Unit Price($)	# Units	Cost($)
Power Supply	MEAN WELL	323282	15	1	15
Servos	Towerpro	MG996R	4.831	5	24.155
Nano 33 BLE	Ardunio	Nano 33 BLE	19.99	1	19.99
Printer Filament	Polymaker	PLA	10	1	10
Fishing Line	Berkly Line	BGB20-22	10	1	10
				Robot Cost	$79.15

$/Hour	Hours	Employees
50	80	3
		X2.5
	Labor Cost	$30,000

Total Cost: $30,172.77

4.2 Schedule:

Week	Objectives	Notes
2/20	Design Document complete, worked on PCB layout, started 3d printing hand, work on firmware	All of us worked on the design document, Quang started 3d printing the hand, Ryan worked on the PCB layout, Sohil worked on the firmware
2/27	Get PCB checked, worked on Firmware, assemble 3D printed hand/ more 3D	Ryan checked V1 of the PCB. Sohil had a dev board and started validating some

	printing	of the firmware, Quang assembled the robotic hand/3d printed more parts
3/6	Ordered PCB, continued work on Firmware, tested robotic hand	This week we ordered our PCB and worked more on the firmware. We also tried control the robotic hand by itself (Ryan, Quang, and Sohil)
3/13	Spring break	N/A
3/20	Continued working on firmware	Finished up the firmware for each specific subsystem (Ryan, Quang, and Sohil)
3/27	Assembled PCBs (main PCB for controlling and the PCBs that hold the hall effect sensors), continue working on firmware, print hall effect sensor mounts and other small 3d prints	Ryan took point on assembling all the PCBs with help from Sohil and Quang. Quang was in charge of making sure all the 3D printing is complete, Sohil was in charge of trying to integrate all the different parts of the firmware together
4/3	General Debugging, placed order for V2 of the PCB	This week we mainly worked on debugging all the issues with the firmware and hardware debugging as well (Ryan, Quang, and Sohil)
4/10	General Debugging	Continued debugging issues as they come up (Ryan, Quang, and Sohil)
4/17	Started manufacturing the final PCB revision	Continued debugging (Ryan, Quang, and Sohil)
4/24	Got the demo ready and assembled the hand battery power circuitry	Ran a demo by the TA
5/3	Worked on Final Paper	Work on Final Pape, gave final presentation to Professor/TA

5 Conclusions:

5.1 Conclusions

Our solution resulted in the creation of a sensing glove and a robotic prosthetic which is used to record and demonstrate the movements in the hand. This sensing glove used a suite of sensors in order to record and sense movements in the hand as well as accept input from the user. These inputs led to mode switches including a mirroring mode and position recall/record. By combining these modes we allow the user to be able to have the freedom to change positions when needed as well as dynamically move the prosthetic when the need arises. These positions can be saved in memory which can save time in case there is a loss of power. The glove also lasts for the required time and more based on the requirement and calculations provided. These will give the user the peace of mind that our product will work for as long as needed until charging is needed. This glove also relies on bluetooth as the communication scheme, this is to add compatibility and ease of implementation across many devices. In our work, we found that our robotic hand was our limiting factor in accuracy since the open source design held many tradeoffs for ease of construction and printability. We found that even though they lost a lot of accuracy towards the outer range of the movement, it was able to replicate many simple hand gestures as needed. This allows a user to build one on a budget.

5.2 Ethics and Safety

Our team made sure to comply with both the IEEE ethics and safety guidelines as well as the ACM code of ethics while detailing out our project. The safety of our users is our number one priority and we want to make sure that we are not abusing or using anyone in the creation and distribution of our product. This includes our users who will most likely be disabled people. Some issues we can see with the usage of our project is that it is a wearable so we want to make sure that the usage of our project doesn't cause harm to anyone. Since our sensor glove directly interfaces with a user's limb, there is a risk of injury from exposed circuitry and electric shock. We can ensure our users safety by correctly grounding all components, insulating exposed metal and leads, and using an electrically insulating material on the glove. In doing this we ensure the safety of our users. [6.1]

Furthermore to comply with section 1.5 in IEEE ethics and safety guidelines we have all parts of the project will be looked over by all team members, credit all sources we use/draw inspiration from, and will ensure the high quality of our work and keep each other responsible. Furthermore we will also be open to criticism and suggestions from our TA and our peers and make necessary adjustments to our methods, design, and practices if quality or a breach of ethics is called into question. This also relates with Section 1.3 in the ACM code of ethics where we will make sure to gain the knowledge or seek out help when working on technical aspects that we have never tried before, since a lack of experience working on systems and ideas can lead to dangerous hardware for both us as the creators and the final end users. Since our project involves many disciplines of engineering from mechanical design, communications, and coding, we will try our best to stick to our strengths to ensure the most qualified team member is working on the respective aspect of the project.[6.4]

We were able to meet all of our ethical requirements since our device is in physical contact with the users. We made sure that all of our exposed connectors and wires were correctly insulated. We also constructed a container in order to house our PCB as that has a large number of bare components and connectors and would have led to a major safety concern. We also wrapped our battery in electrical so that it would be insulated and provide additional protection from puncture and contact with our users. The flex sensors contained exposed conductors so we also insulated them with electrical tape. Overall, anything that was exposed and in direct contact with our users were correctly insulated to prevent shocks and danger.

5.3 Future Work and Alternatives:

Moving forwards, the project could be expanded upon and improved with a better budget and another set of design revisions.

A huge area that could be improved upon is the robotic hand. One problem we found was the tensioning system needed serious improvement. The robotic hand could not grip very well and was not accurate. We could try and make a custom robotic hand to implement these improvements rather than ripping the open source models.

We should also use buck converters that can better handle the currents that our servos would sometimes draw, as we were operating the buck converter towards the edge of their rated range.

Another feature that would be nice is LiPo charging circuitry and some sort of battery level indicator for the user to know when to recharge the glove. Additionally, implementing a piezo buzzer for another source of user feedback would be ideal. We attempted to implement this in the last PCB revision, but made an error in the NMOS package routing that was supposed to actuate the buzzer.

Finally, we could also clean up the bugs in the preset recording mode. In this mode we have the ability to record motions and save it as presets on the hand. However, when trying to implement this we had a lot of problems in trying to get a smooth recording as well as getting out of recording state. So moving forward we are hoping to fix this part of the glove and have it work flawlessly and integrate well into our project.

6 References:

6.1 "IEEE code of Ethics," *IEEE*. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html. [Accessed: 09-Feb-2023].

6.2 "Hand and Forarm," *InMoov*. [Online]. Available: https://inmoov.fr//hand-and-forarm/. [Accessed: 09-Feb-2023].

6.3 J. N. Ingram, K. P. Körding, I. S. Howard, and D. M. Wolpert, "The statistics of Natural Hand Movements," *Experimental brain research*, Jun-2008. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2636901/. [Accessed: 09-Feb-2023].

6.4 "The code affirms an obligation of computing professionals to use their skills for the benefit of society.," *Code of Ethics*. [Online]. Available: https://www.acm.org/code-of-ethics. [Accessed: 09-Feb-2023].

6.5 "DRV5032 Ultra-Low-Power digital-switch hall effect sensor datasheet ..." [Online]. Available: https://www.ti.com/lit/ds/symlink/drv5032.pdf?HQS=dis-mous-null-mousermode-dsf-pf-null-wwe&ts=1662340198537&ref_url=https%253A%252F%252Fwww.mouser.com%252F. [Accessed: 24-Feb-2023].

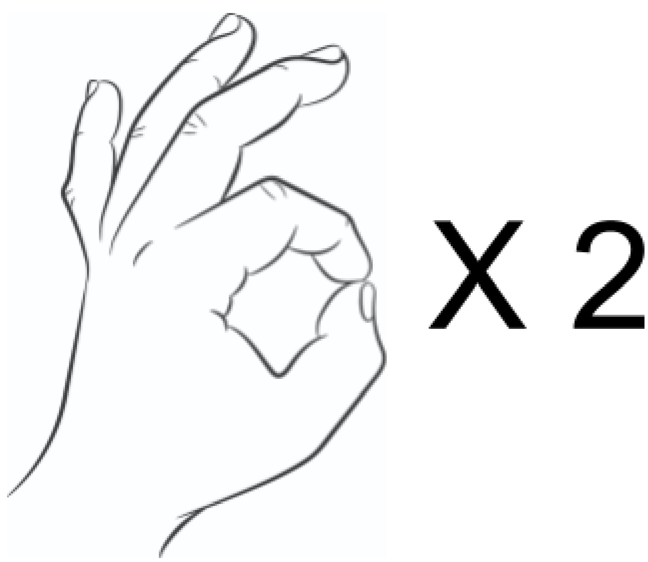
6.6 "D62," *K&J Magnetics: D62*. [Online]. Available: https://www.kjmagnetics.com/proddetail.asp?prod=D62. [Accessed: 23-Feb-2023].

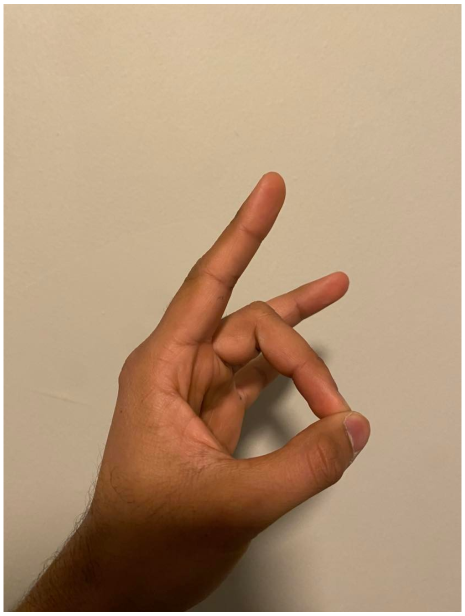
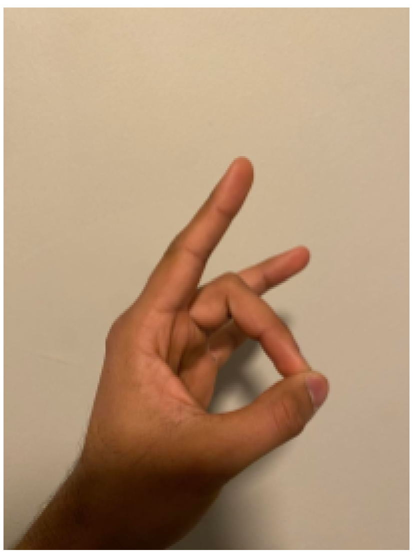
6.7 "Ldo Basics (rev. A) - ti.com." [Online]. Available: https://www.ti.com/lit/eb/slyy151a/slyy151a.pdf?ts=1613818301243&ref_url=https%253A%252F%252Fwww.google.fr%252F. [Accessed: 04-May-2023].

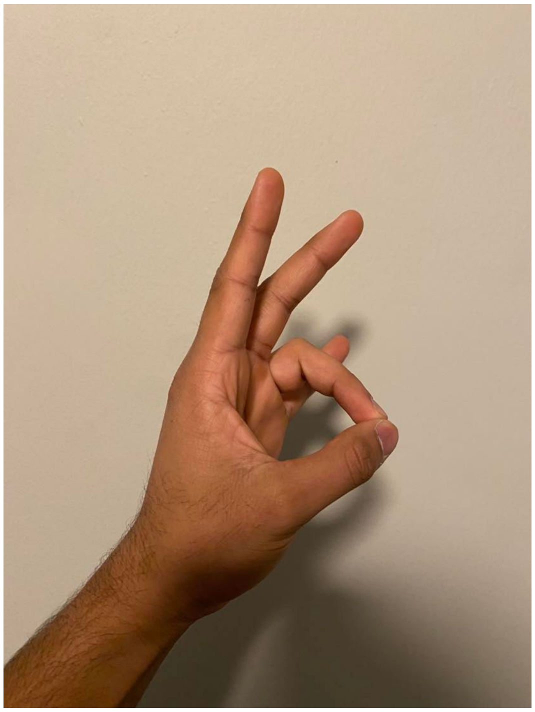
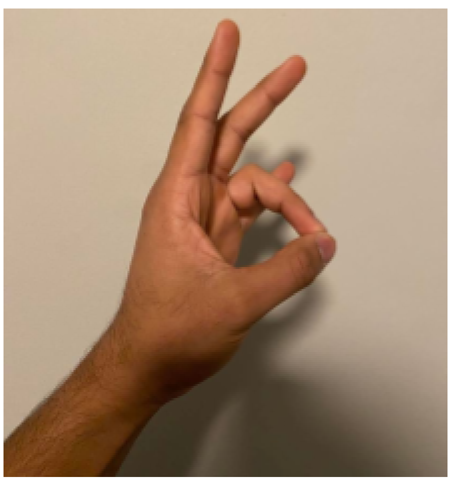
6.8 L. Ada, "Li-Ion & LiPoly batteries," *Adafruit Learning System*. [Online]. Available: https://learn.adafruit.com/li-ion-and-lipoly-batteries/voltages. [Accessed: 03-May-2023].

7 Appendix

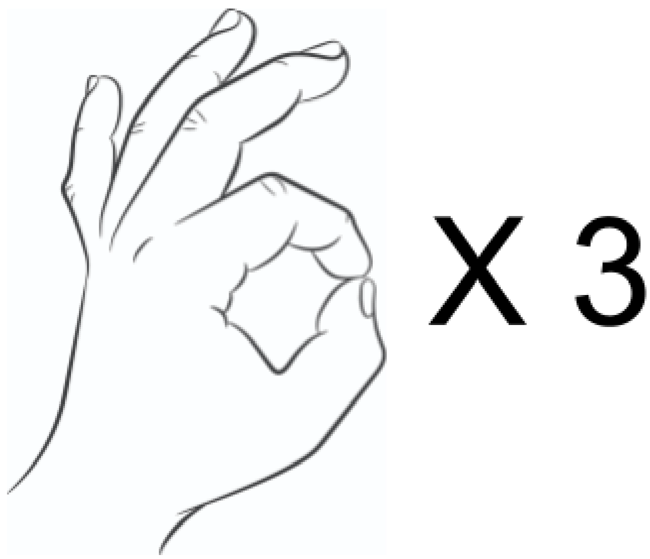
7.1 Requirements and Verification Tables

Gestures	Meaning
	Turns mirroring mode on Mirroring mode is when the robotic hand copies the movement of the hand
	Turns preset mode on Preset mode is when the user is in this mode the gestures labeled preset 1 - 3 will set the hand in the saved configuration the user wants it in. Also while in this mode gestures recorded 1-3 will record the hand position of the user and save it as the preset position. For example, if the user does the gesture “record 2” the glove will record the position of the hand and then save the position of the hand that way if “preset mode 2” is triggered the robotic hand will go to that position.

	Preset 1: this will set the robotic hand to the first of the three preset positions
 X 2	Record 1:This sets the glove in record mode and will wait 5 seconds to take a snapshot of the user's hand to override the saved position for preset mode 1

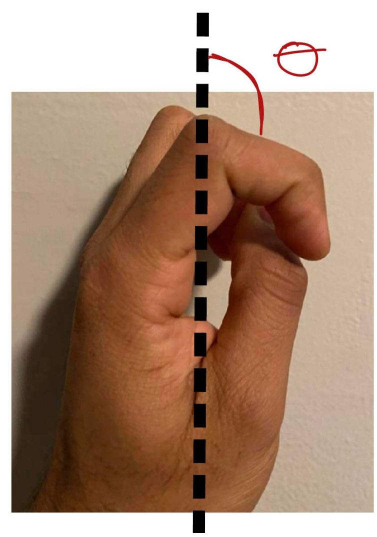
	Preset 2: this will set the robotic hand to the second of the three preset positions
 X 2	Record 2:This sets the glove in record mode and will wait 5 seconds to take a snapshot of the user's hand to override the saved position for preset mode 2

	Preset 3: this will set the robotic hand to the third of the three preset positions
 X 2	Record 3:This sets the glove in record mode and will wait 5 seconds to take a snapshot of the user's hand to override the saved position for preset mode 3

	Turns all modes off, in this mode the hand just waits to be put in either mirroring mode or preset mode before moving the robotic hand

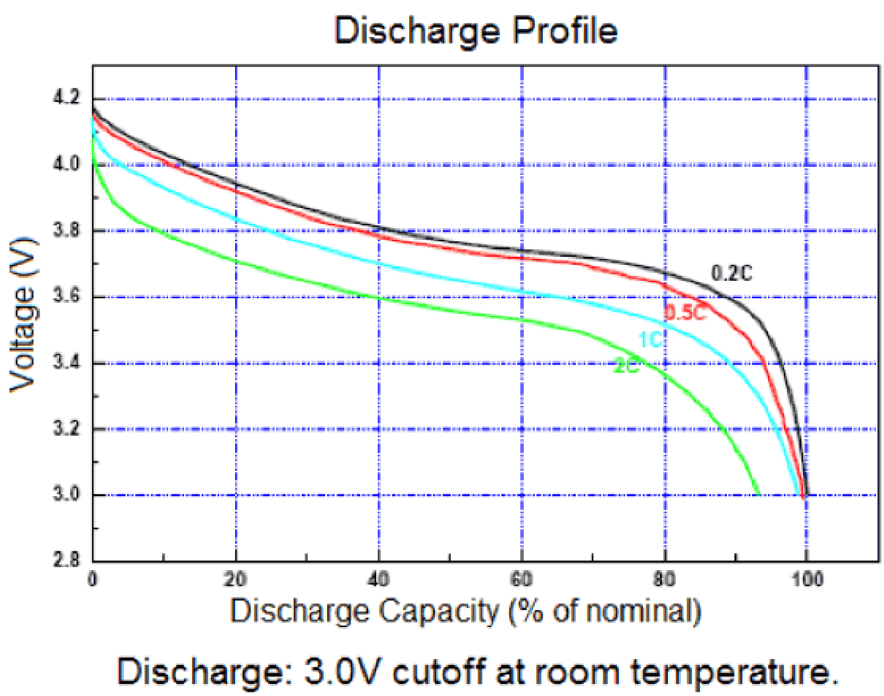
Requirement	**Verification**
1) Make sure the processor can distinguish between all the different hall effects and set the LEDs to the correct modes	a) Check to see if we are successfully saving the different present values for the present positions b) When the system is complete go through all of the gestures and see if the LED signifying what mode is correct as well as the robotic hand going to the correct position as shown in the gestures table below
2) Check to see if Bluetooth can transmit and receive information *introduce bitrate standard	a) First check if we can transmit data between the micro and the Arduino. try and send a bit stream from the micro to the Arduino b) Try and send which hall effects are triggered and send the resistance values of the flex sensors from the micro to the Arduino
3) Flex sensors are successfully read	a) First get the input of the values of all the flex sensors and make sure we get any input at all

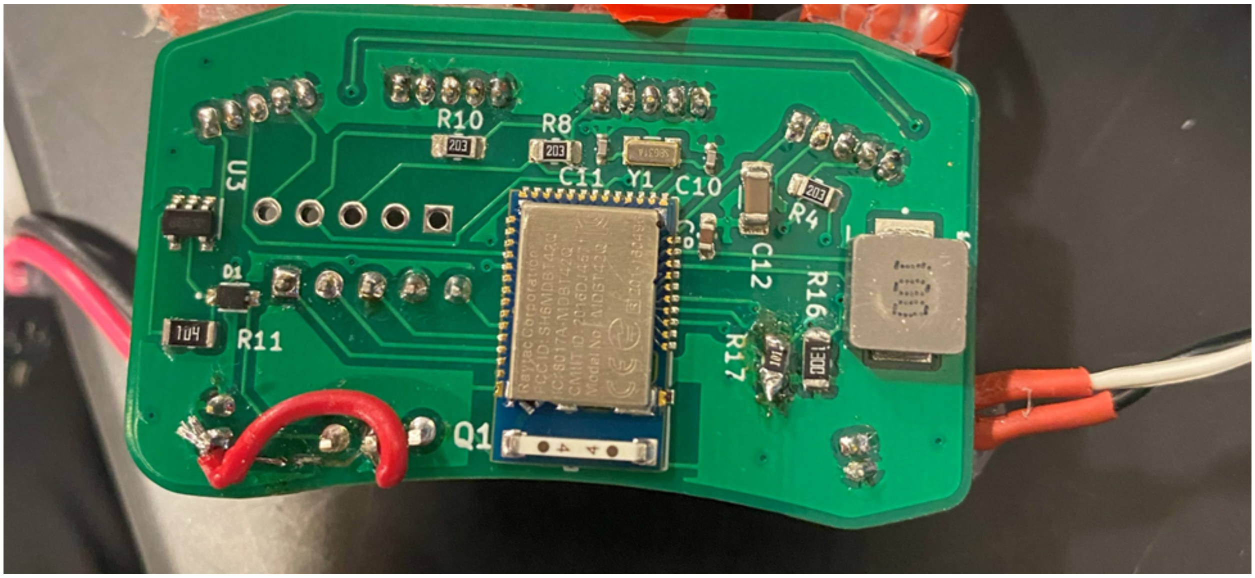
	b) Then we need to at minimum do a two-point calibration for the flex sensors. Have it at its minimum value and measure the resistance and its maximum value and check its resistance c) Finally make sure we can successfully move the hand to the correct position depending on the flex sensor values

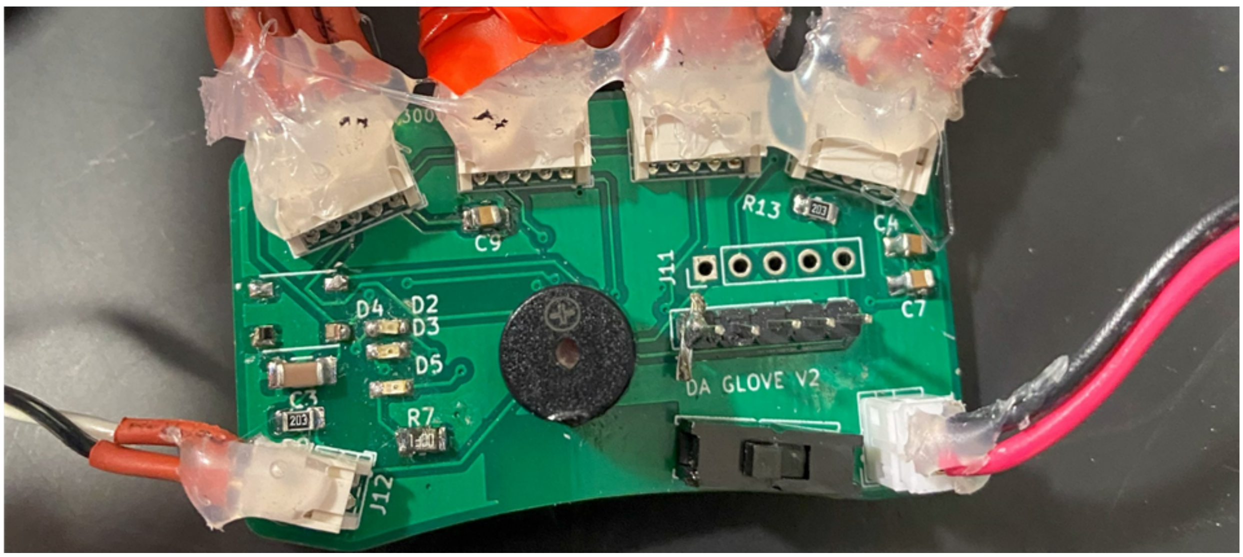
Requirements	**Verification**
1) The hall effects getting triggered should set the modes/try and move the hand to preset positions	1 while designing the system touch the magnet to the hall effect and see in the code if the correct flags are set high
2) The flex sensors bending causes the finger to move to correct location with ±5%	1a) First through the software make sure we can see the change of resistance for the flex sensors and see the full range of motion b) then measure the angle of the finger and compare it to the angle of the robot. Ex of how the angle is measured: 

Requirement	Verification
1) Able to power both the sensors and microcontroller for 8 hours in continuous transmission mode	We will test by placing the glove into mirroring mode which is the most intensive mode since the glove must continuously transmit data through bluetooth. Since our code transmits continuously regardless of movement in mirroring mode, we will set a timer and time how long it takes for the battery to drain in the glove.

7.2 Remaining Elements:




R10
R8
C11
Y1
C10
U3
R4
C12
R16
R17
D1
R11
Q1


R13
C9
J11
C7
D4
D2
D3
D5
C3
R7
DA GLOVE V2
J12