𝔾love Gesture Control Device

By

Samarth Makhija

Ahmed Farooqi

Yashovardhan Arora

Final Report for ECE 445, Senior Design, Spring 2020

TA: Shuai Tang

May 2020

Contents

1 Project Motivation

1.1 Problem Statement

Until the early 2010s, voice-controlled technologies were mostly seen in sci-fi movies as a promise of the future. However, since the release of the Amazon Echo devices, voice-controlled systems have become quite ubiquitous nowadays [1]. The convenience it brings is not just limited to intangible tasks like asking for the weather or playing music and has expanded to more tangible actions like controlling lights, home security, etc. It has brought about a revelation in society by introducing us to the concept of a smart home.

However, voice-controlled systems have some major drawbacks. They can neither function in a silent environment nor a very noisy one. It is also not accessible for people with speech/hearing disabilities. In a survey conducted by our team, we found out that 64.29% of users would like to use gesture-based commands over voice-based commands for their smart home even though 78.57% of the responders already owned some kind of voice-controlled system. This proves that there exists a strong market for gesture-controlled smart home systems.

Gesture-based interaction has been around for several years with Microsoft Kinect being one of the first devices to use it in 2010 [2]. The Kinect worked with a 640 x 480 resolution RGB camera and an IR depth sensor. [2]. BMW is also bringing gesture-based technology into their products with BMW Natural Interaction system [3]. This allows users to control in-car functions like the entertainment system and temperature using gestures while also allowing them the freedom to interact with certain features outside the car. These functions include pointing to buildings, streets, and highways to get more information.

Keeping some of these current use cases for gestures in mind, we believe that our project can be the next 'big step' in making smart home systems more convenient and intuitive. It will enable the user to interact with his/her home using gestures detected by wearable technology on the user's hand.

1.2 Solution

Our solution involves creating a gesture-controlled device for smart home - a glove containing IMU sensors and flex resistors. The Glove has the capability of syncing with an Alexa device such as an Echo dot via bluetooth and controlling any of the smart home devices that Alexa has linked up with. The IMU sensors help detect the motion of the arm and hand while the flex resistors help capture the motion of the fingers. By combining them together, we are able to classify complex gestures from the user with fairly high accuracy (greater than 85%)

The gesture is created via an Alexa skill [Section 2.2] which registers a gesture corresponding to a command [Section 2.3]. When this gesture is performed, the Alexa device would receive a bluetooth signal from the glove and perform the function. The glove allows us to add flex sensors [Section 2.4 and 2.5] and get a greater range of gestures than before by allowing finger movements along with wrist movements.

This results in some advantages over our first project which are highlighted in Section 1.2.1.

1.2.1 Comparision between our Old Project vs Current Project

We have reworked our design from a band containing an IMU unit [Section 4] to a glove containing flex resistors combined with an array of IMUs at different points. The additional IMUs increase the degrees of movement that can be detected and thus increase the variety of gestures that can be processed. This expanded hardware design will be a more powerful approach compared to our previous design by allowing greater control of more devices without the need for the plug.

The removal of an IR sensor removes the functionality of being able to point at a device and perform a gesture to control it; however, it also allows us to have faster processing of gesture and no longer requires us to be in the device's line of sight.

The software used has also changed significantly with our new design being compatible with Alexa which allows us to control a wide range of smart home devices without requiring a special smart plug but also restricts to only being able to control Alexa compatible devices. It is possible this dependancy is removed in the future [Section 3.4]. A complete summary of the advantages and disadvantages can be found below:

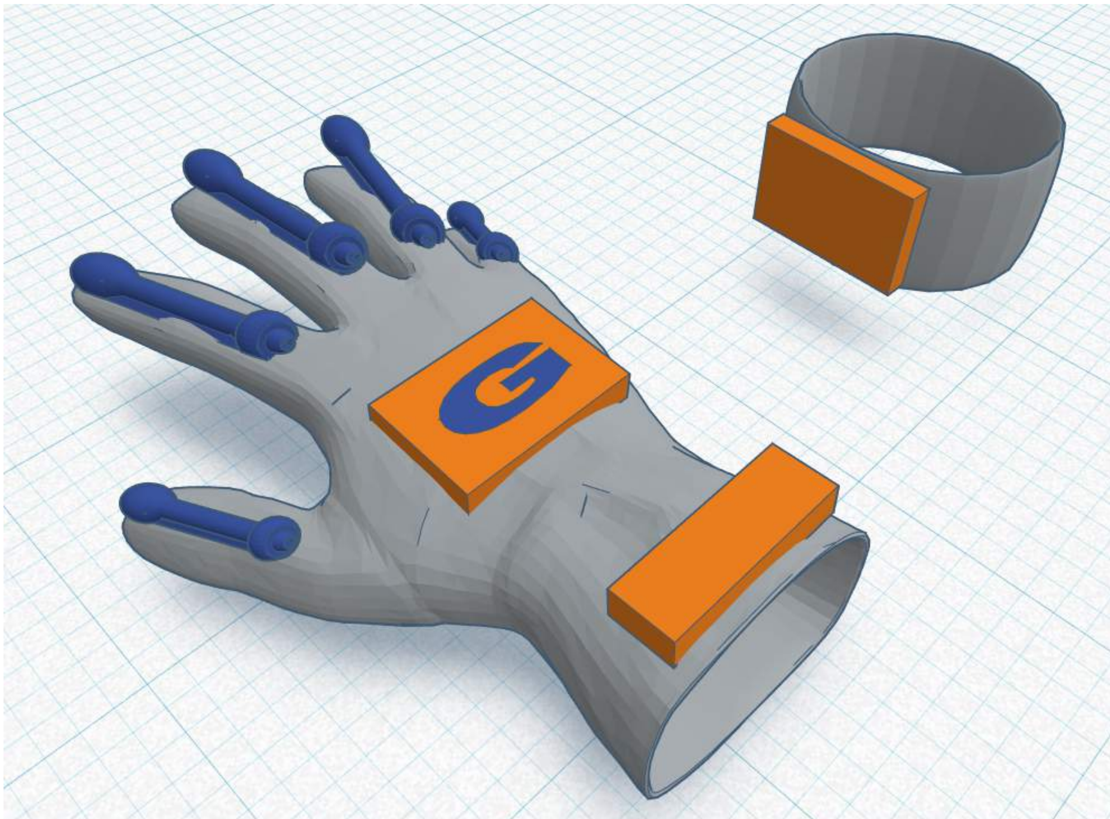
Previous Design	New Design
Cannot detect as many gestures and can only detect simple gestures such as up, down, left, right.	Is able to detect more and increasingly complicated gestures such as rock, paper, scissors, snapping of the fingers or a karate chop.
Cannot link with existing smart home devices and Amazon Echo.	Can perform more than just on/off of a device, Alexa can connect gesture with specialized command (change lights to blue, make a cappuccino)
Has a significant lag due to the IR sensor-receiver handshake	No wait for IR handshake allows instantaneous gestures
Needs to be in line of sight for the device to be controlled	Lack of IR sensor eliminates the need for visual handshake allowing gestures from any location.
Can point at a device and perform a simple gesture for an upwards motion to turn it on.	Needs to perform a specific gesture to control an action on a certain device.
Follows the same general gesture for all devices.	Needs to save unique gestures for each action on each device
Needs the user to only remember the four basic gestures	Needs the user to remember all the saved gestures

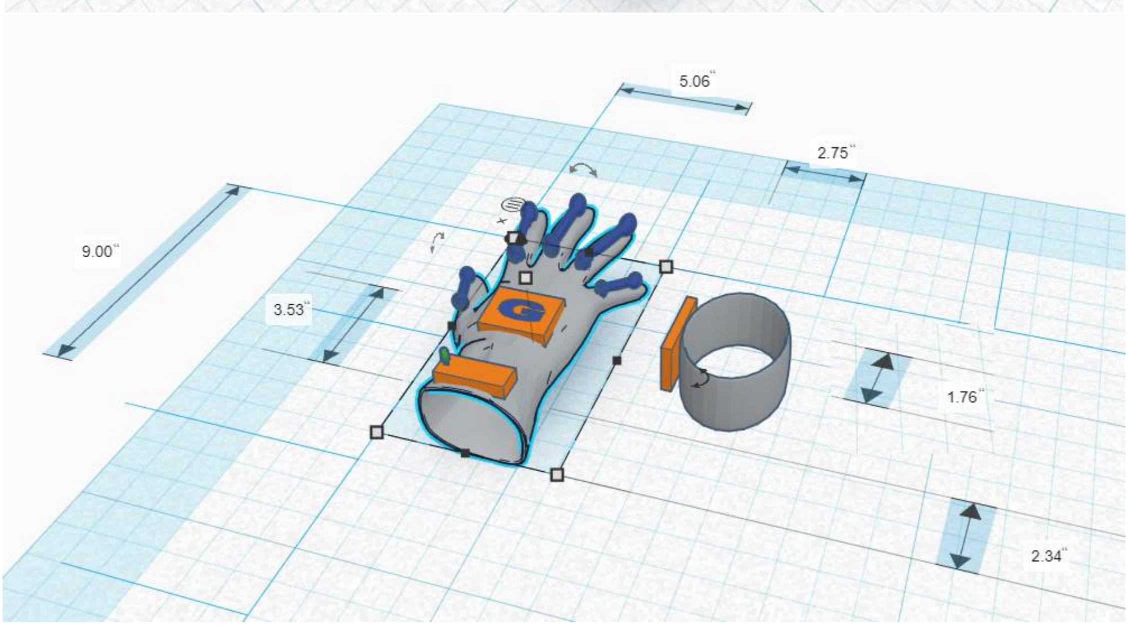
1.3 High-Level Requirements

- Compute and recognize gestures by the band and glove combination. Percentage of gestures detected needs to be over 85%.
- Save at least 4 gestures on the the Alexa app for later use.
- Interact with at least 4 smart-home devices. The 'interaction' in this context could mean switching the device on or off, changing a channel or playing a song depending on the type of device. Smart home devices could include television, speakers, smart bulb, smart plug or thermostat.

1.4 Sketch

1.4.1 Sketch for prototype

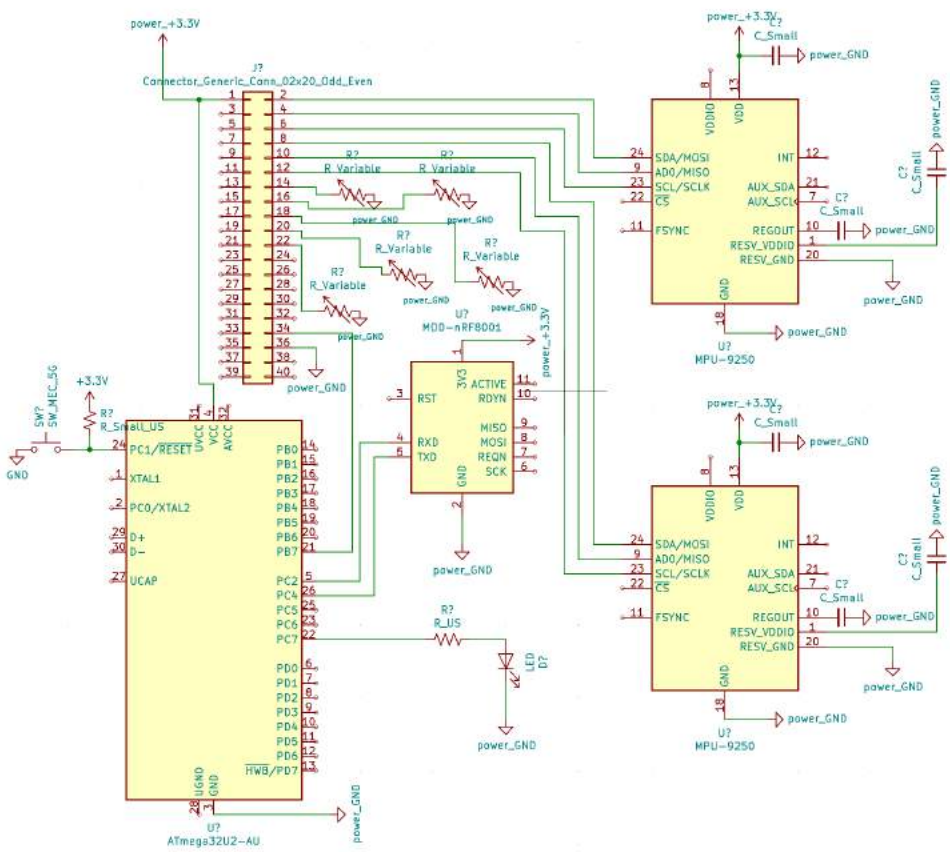




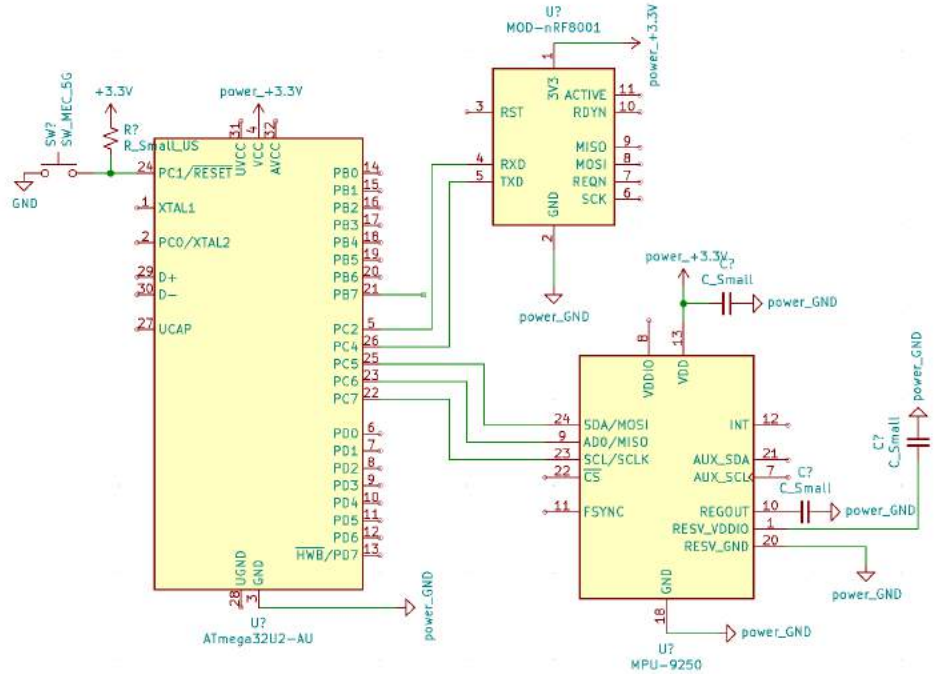
Approximate Weight of System: 1150 grams

1.4.2 Sketch for usage

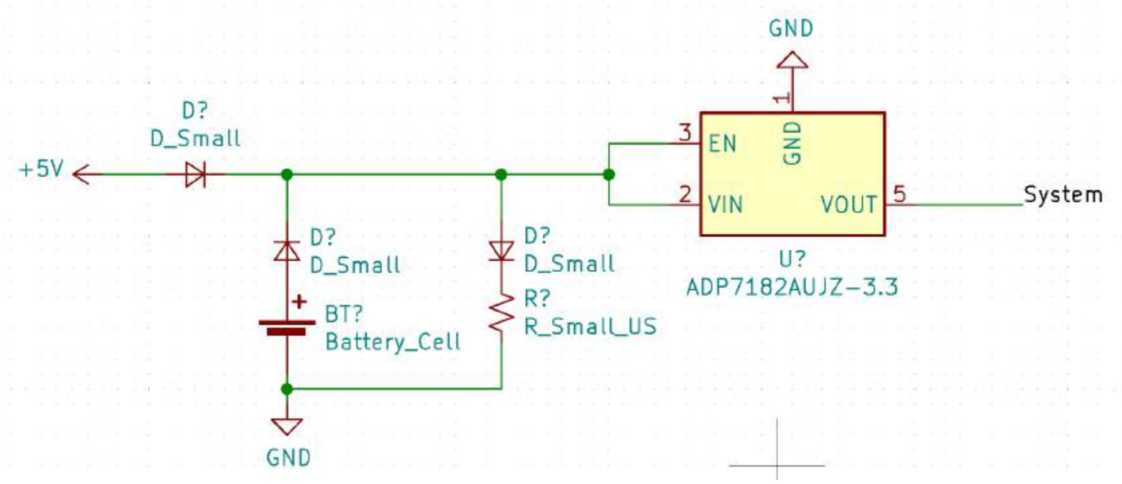
1.5 Schematics



Glove

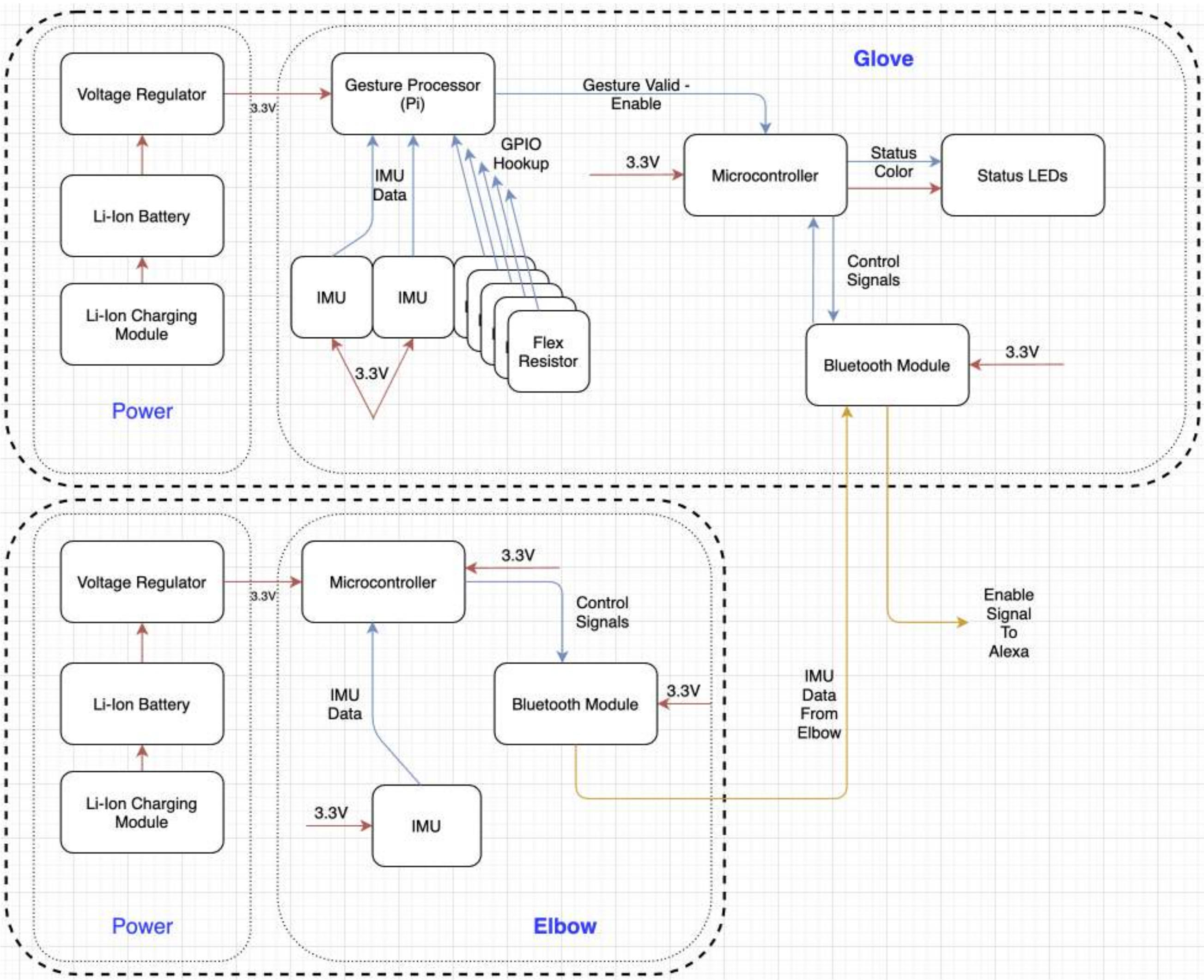


Elbow Band



Charging and 5V to 3.3V Regulator

1.6 Block Diagram



2 Implementation

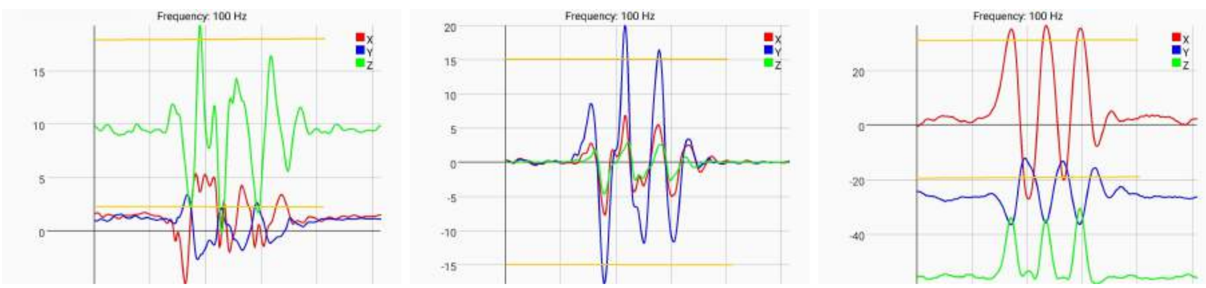
2.1 Shake to Engage Algorithm

Like any voice-controlled device, our gesture-controlled device will also need an engaging signal. This is essential to prevent the user from accidentally sending gesture commands to Alexa when not required. It will be done through a rapid jerk of the wrist that arms the IMU and flex sensors to start detecting gestures. Once the user jerks their wrist, the 'Shake to Engage' algorithm will check if the jerk is over a threshold value of the accelerometer and if so, the device will go into the 'Gesture Detection' State. The user will be informed of the same by the LED turning green. The threshold value mentioned can be easily toggled by changing the sensitivity of the 'Shake to Engage' algorithm in the Alexa app. The device can also be disengaged from the 'Gesture Detection' state by performing the same jerk. The pseudocode of the algorithm is as follows:

Algorithm 1 Shake to Engage

Require: top level module and threshold value
\\Detect whether the jerk produced by the wrist is stronger than the threshold value
Input: IMU Accelerometer data
if accelerometer is greater than threshold value **then**
State is Armed
LED turns Green
else
State is Disarmed
LED turns off
end if

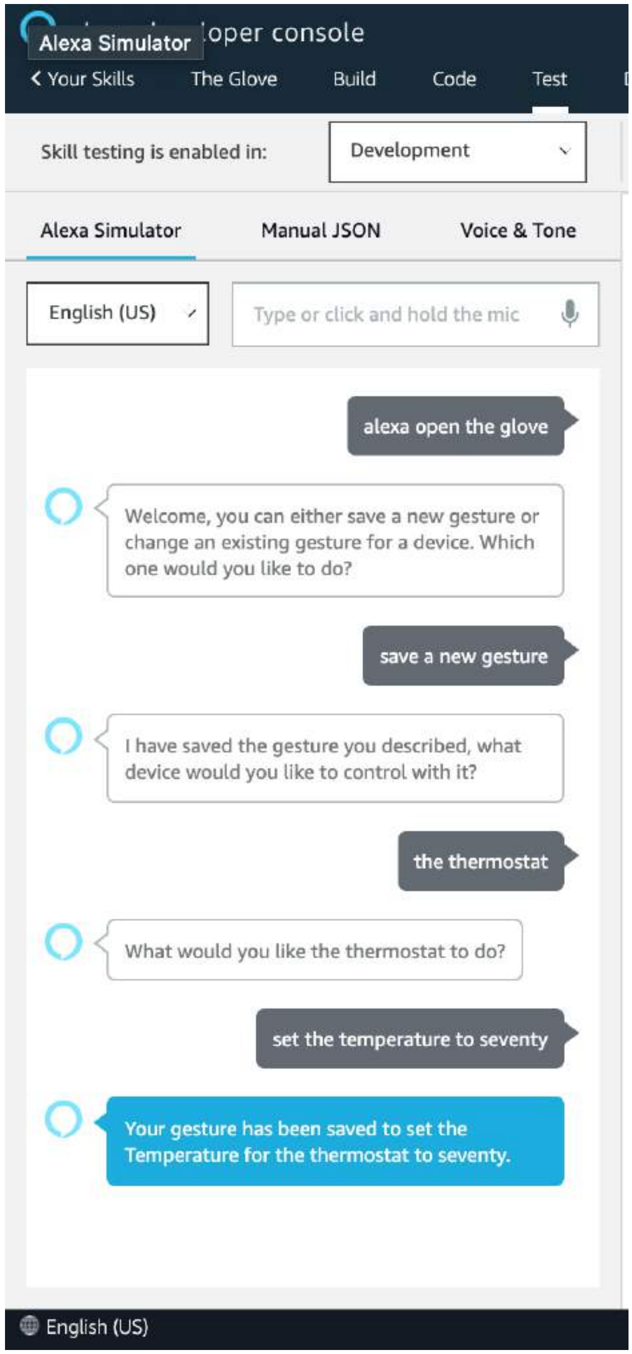
Graphically, we can see the shake invocation and how it effects each axis of the IMU differently depending on how the user shakes the glove. The yellow lines can indicate a possible threshold that the user can set (or a delta value between the highest and lowest values outputted by the IMU) in order to invoke the device. Each axis will need its own treshold to account for how the device is shaked. Uncertainties with this approach is outlined further in section 3.2.1.



Shake: Accelerometer, Gyroscope, Magnetometer Graphs

2.2 The Glove Alexa Skill

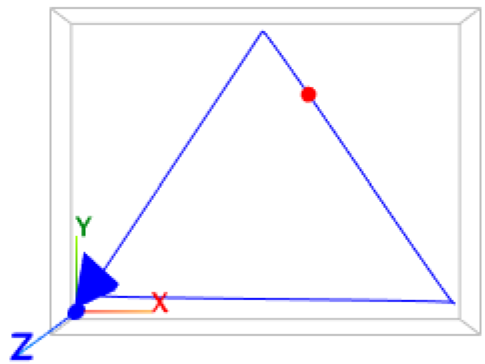
The Alexa Skill is responsible for saving the gesture that are to be used by the user of The Glove later. We created a skill and tested it using the Alexa simulator. The Alexa Simulator allows us to test an Alexa Skill by showing how Alexa would interact with the user given an input. We built an interaction model that allows users to save gestures for a certain device and request Alexa on what needs to be done with a device upon receiving that gesture. An interaction with the user on the Alexa simulator can be seen below:



As depicted, the user is able to interact with Alexa, save gestures for an Alexa compatible device performing a specified action. These gestures can later be detected using a combination of the algorithms described in section 2.3 and 2.4.

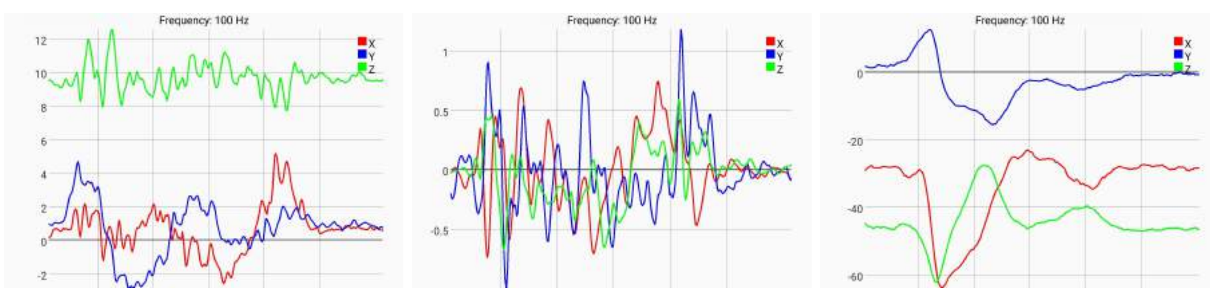
2.3 IMU Gesture Recognition

For the gesture recognition component of our device, we were unable to create the same setup that we would have on a physical glove. To simulate that setup, we have used an IMU logging app to pull the accelerometer, gyroscope and magnetometer data from the built-in IMU on a Samsung SM-N975W (accelerometer and gyroscope: LSM6DSO, magnetometer: AK09918) in CSV format and plug that into visualization tools and our base algorithm. The gesture that is to be detected is a triangle gesture, visualized below.



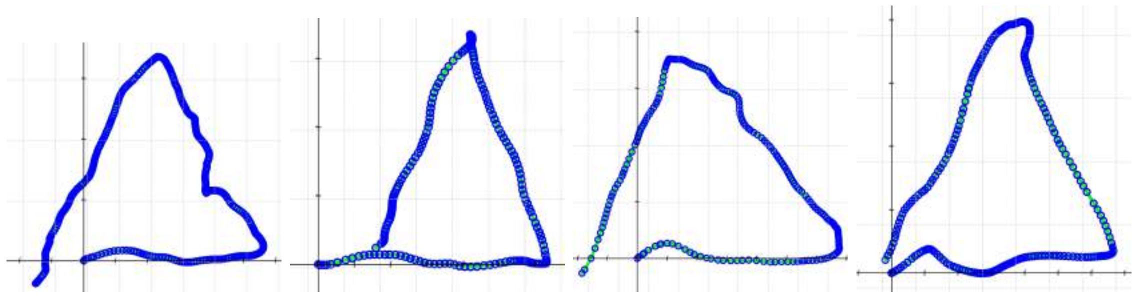
Base Triangle Gesture

This is a two-dimensional gesture and relies on the gesture performed on the android device to also be matching in orientation. In other words, the gesture must be performed along the X and Y dimensions of the IMU (where the screen and back of the phone are the Z axes) by keeping the phone face up and moving it in a two-dimensional plane. A triangle gesture was made while the IMU data was streamed into a csv file. The data was then plotted into the follwoing graphs:



Triangle: Accelerometer, Gyroscope, Magnetometer Graphs

Taking a few measurements of the triangle gesture and using our algorithm as well as some cleanup on the data, we were able to export a few usable examples of the triangle shape. Using a shape recognition algorithm to check for confidence of the created shape via a gesture and the baseline triangle shape, we got a usable confidence for about nine of the ten gestures invoked.



Sample Triangle Positions Cleaned

The IMU data collector sampled at a rate of approximately 4 kHz, leading to a dispersion of values around the path of the gesture. The collector used does not have the option to change the sample rate, which is something that would be done on the physical IMUs that we would use on the glove, so the higher than necessary sample rate needed to be slightly cleaned. The shape recognition algorithm we used, OpenCV, was able to detect simple shapes that were outputted via the IMU data. Section 3.2.2 discusses the uncertainties and additional testing needed with this approach. The psuedocode is provided here:

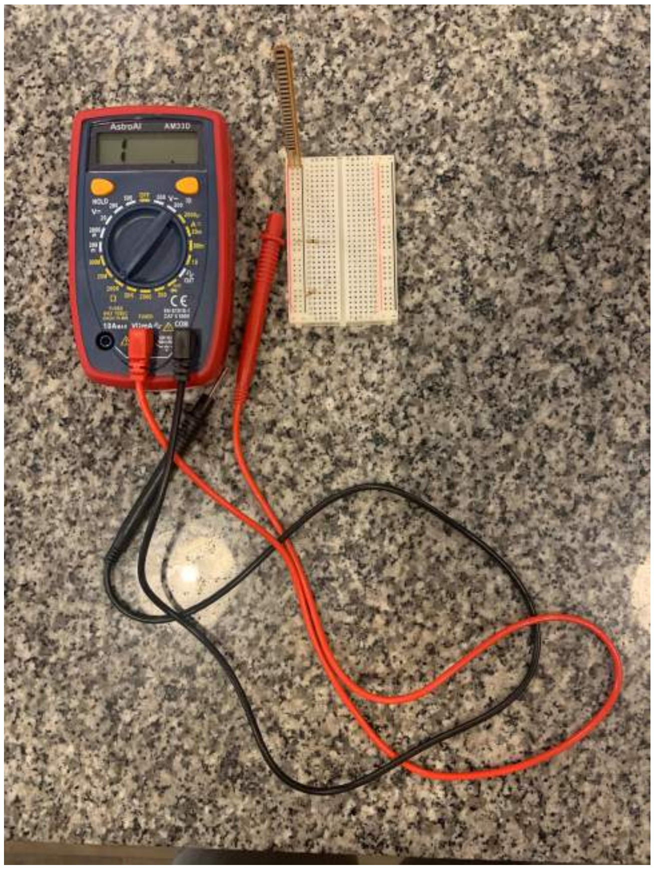
2.3.1 Pseudocode for shape detection

Algorithm 2 Detect Shapes Created From IMU Data

```
Require: The IMU Data drawn on a graph via Tableau from the dead-reckoning and binary colored
  Input: Binary image of graph
  contours = cv.findContours()
  //find all the continuous edges in the image
  for each contour in contours do
    approx_sides = opencv.approxPolyDP(contour),
    //approximate the number of sides for the given contour
    if number of approx_sides = 3 then
      Shape is a triangle
    else if number of approx_sides = 4 then
      Shape is a rectangle
    else if number of approx_sides = 5 then
      Shape is pentagon
      ...
    end if
  end for
```

2.4 Flex Sensor Experiment

An experiment was done to determine the relationship between the bend of a flex sensor vs the resistance. The experimental set up is shown below:



Experimental setup for the Flex Sensor Experiment

The following points describe the experiment performed:

1. Connect a multi-meter across the resistor.
2. Measure the unbent flex resistor's resistance.
3. Bend the resistor by 15 degrees more and measure the new resistance in a table.
4. Perform step 3 a total of 12 times to reach a 180 degree bent in the flex resistor.
5. Plot the collected data in a scatter plot using a Python script.
6. Plot the best fit line across the data (calculated through regression) and record the equation of the line
7. You should now have a relation that correlates the bending angle in degrees to the resistance of the flex resistor in Ohms.

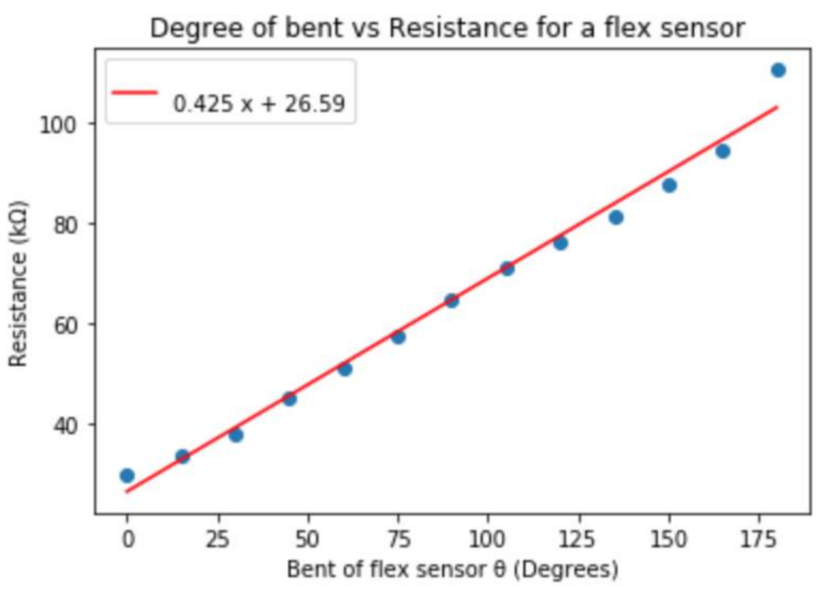
Note: 0 Ohm Resistors were used instead of wires in the experimental setup image since the ECE building is closed due to the pandemic

Hypothesis: We predict that as the bending angle increases, the resistance of the flex resistor increases with an increasing rate.

Degrees	Measurement 1	Measurement 2	Measurement 3	Measurement 4	Measurement 5	Average
0	29.1	32.8	29.4	29	28.9	**29.84**
15	34.5	33.6	32.8	33	35	**33.78**
30	38	36.2	38.8	40.7	37.5	**38.24**
45	44	44.3	47.7	44.6	46.2	**45.36**
60	51.8	54.1	49	47	53.3	**51.04**
75	59	57.6	55.2	58.4	58.5	**57.74**
90	63.6	66.4	64.5	65.4	64	**64.78**
105	70.9	68.8	72.6	73.5	69.5	**71.06**
120	73.6	76.9	77.9	76.3	77	**76.34**
135	79.8	83	81.2	84.6	78.5	**81.42**
150	90.8	91.1	86	86.9	84.6	**87.88**
165	95.6	93.9	92.8	94.8	96.7	**94.76**
180	102.4	108.9	116.5	116.1	109.2	**110.62**

Data collected for Flex Sensor Experiment. All measurements are in kOhms.

The data was then plotted using the matplotlib.pyplot library and a best fit line as plotted using the numpy library. The results can be seen below



Using our best fit line our the equation that classifies the relationship between degrees and bent is as found to be:

$$k\Omega = 0.425 * \theta + 26.59 \tag{1}$$

This equation will be used in our Flex Sensor Gesture Detection Algorithm and referred to as f.

2.5 Flex Sensor Gesture Detection Algorithm

Along with the IMU gesture detection which captures the motion of the arm and wrist, the flex sensors allow us to measure the motion of the fingers allowing us to detect gestures with better accuracy while also increasing the pool of gestures we can successfully detect.

2.5.1 Pseudocode for algorithm to save gestures

Algorithm 3 Save gesture performed using fingers

```
Require: The Glove is on and the Alexa Skill is in the Saving Gestures State
Ensure: Gestures are over 0.5 seconds long.
  Input: Gesture_Name
  Create Empty array A
  while Gesture is active do
    Wait 0.01 seconds
    Record resistances of the four resistances r1,r2,r3,r4.
    Append f(r1),f(r2),f(r3),f(r4) to A, where f is the function to convert resistances to angle of bend.
  end while
Save A in the band's memory and label it as Gesture_Name
```

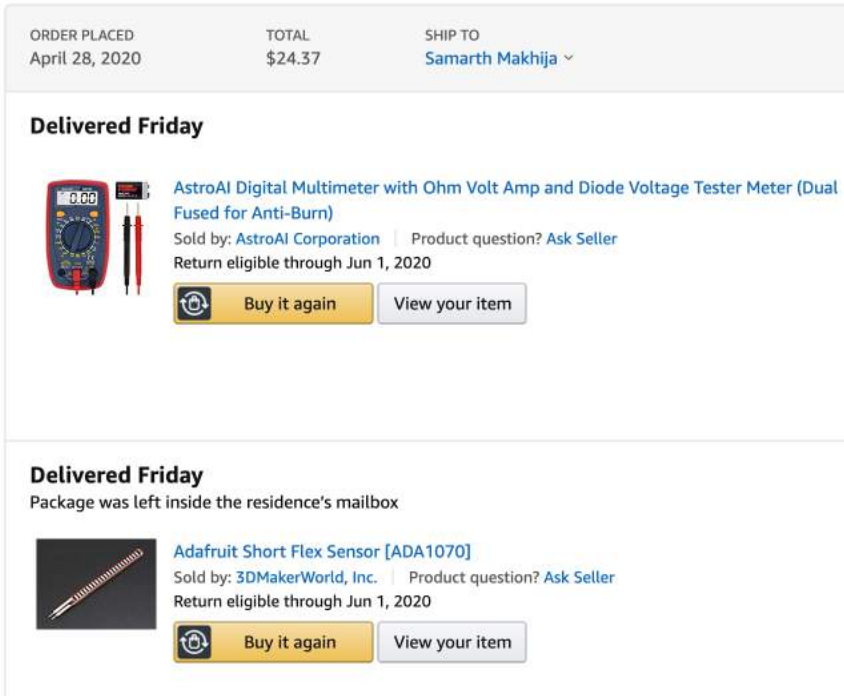
2.5.2 Pseudocode for classification algorithm

Algorithm 4 Classify Bending of fingers into gesture

```
Require: The Glove is on and in the detecting gestures state
  Input: Resistances from the flex resistors: r1,r2,r3,r4, Array of Gestures
  Create Empty array A
  while Gesture is active do
    Wait 0.01 seconds
    Record resistances of the four resistances r1,r2,r3,r4.
    Append f(r1),f(r2),f(r3),f(r4) to A, where f is the function to convert resistances to angle of bend.
  end while
  ClosestMatch = NULL
  MatchPercentage = 0
  for SavedArray in Array of Gestures do
    LocalMatchPercentage = 1
    for index in Array do
      if A[index] is between SavedArray[index]*0.95 andSavedArray[index]*1.05 then
        MatchPercentage = MatchPercentage - 1/len(SavedArray)
      end if
    end for
    if LocalMatchPercentage >MatchPercentage then
      MatchPercentage = LocalMatchPercentage
      ClosestMatch = SavedArray
    end if
  end for
  if MatchPercentage ¡ 0.6 then
    OUTPUT "No valid gesture detected"
  else
    OUTPUT ClosestMatch
  end if
```

2.6 Bill of Materials

The bill of materials for electronic components that were ordered to collect real data on the flex sensor and construct our function have been inluded in the figure below.



3 Conclusions

3.1 Implementation Summary

We were able to corroborate most of our theories pertaining to building a gesture controlled smart home device during our implementation phase. Our previous design had various restrictions with the complexity of gestures it could perform and the number of devices it could control. In Chapter 2, we were able to add improvements and make our device more inclusive, intuitive and faster.

This required completely revamping our design from a bracelet to a glove and band combination. The designing and remodelling was done by Ahmed. He also redesigned the schematics and updated the power requirements.

We developed our Shake to Engage algorithm to prevent the user from accidentally sending commands to Alexa. It also needed identifying and fine tuning the threshold value to for the accelerometer. Yash completed this task.

Samarth worked on the Alexa skill and conducted experiments with the IMU and flex sensors to understand their functioning. He inculcated those findings in developing a gesture detection algorithm. He also worked on the fault tolerance of the flex sensors as it was a new critical component in Chapter 2.

Overall, we enjoyed working on this project and hope to make it into a working prototype in the near future.

3.2 Unknowns, Uncertainties, Testing Needed

3.2.1 Shake Gesture Threshold

The shaking of the wrist to invoke the gesture may be considered differently user to user. Someone with a weaker stature may be unable to perform the same invocation and yield the same deltas in IMU data that someone who can shake their hand to the limit of the IMU. Additionally, the shake can occur in the X, Y, or Z axis depending on what direction the user shakes their hand. This will require all three axes to have a threshold set or using the RMS value of the accelerometer, which needs testing. Making the threshold changeable via Alexa will reduce the possibility of false positives that may show up by relying on the user to feel what is an appropriate value for the invocation.

3.2.2 OpenCV

The OpenCV shape detection algorithm required that the image be a binary image, or that the background should be a different color than the shape. Because we have created a shape based on IMU data and are just plot points on a graph, the graph needs to be modified to match the requirements. This also included cleaning up some of the outliers on the plot that were caused due to the high sample rate of the IMU collection application. We are not going to use this approach inside our gesture recognition algorithm, but it is important to acknowledge this as we do not have the proper parts to be able to test this functionality on the glove. OpenCV also works best on shapes with straight edges, so the jagged nature of a double integrated function is something that may lead to skewed results.

3.2.3 Availability of Parts

Due to the COVID-19 pandemic, there has been a halt on many supply chains as well as shipping, including critical hardware production facilities in china. This has prevented our project from receiving the PCBs we need in order to create the prototypes. Additionally, due to the increased need for concern when it comes to handling packages, many places, such as Amazon, have increased their shipping times for items considered non-essential during this pandemic. To stay safe and in good health, packages shouldn't be accepted at this time as there are many points of contact for an item in the shipping chain. Due to the lack of parts due to either lack of restock or other unavailability's, we have been unable to create a prototype that matches the specifications we have created.

3.2.4 Lack of Proper Tests Due to Lack of Parts

Due to the lack of parts available, some critical testing cannot be done. The microcontroller is one of the most important pieces of the circuit as it will be the communication hub for the rest of circuit. As we are unable to order a PCB or the controller, testing the communications between all the submodules becomes impossible. Individual testing of the IMU could be done, or at least simulated with an Android device. Individual testing of the flex sensors was also possible since we managed to get those, but the specific testing of the bluetooth module or the voltage regulator was not possible. We understand that most of our algorithms are theoretical and have been proven by small isolated experiments. Our next stage of testing would have been building the device and testing its components in its natural environment (On our hands). We strongly believe that if given a chance, we could have made the user experience as intuitive as possible and revolutionized the way a he/she interacts with his/her home devices and appliances.

3.3 Ethics and Safety

Our group has done thorough research of the IEEE Code of Ethics[4]. The following tables shows how we recognise all aspects of the code and abide by it.

IEEE Code of Ethics Point	How we address it
1	We enclose our batteries in heat proof containments and keep them away from humans without prior testing of their safety.
2	We have done research of existing gesture controlled devices, such as the ultraleap or fibaro, and our device benefits from being non-specialized in terms of the types of products that can be controlled.
3	We have provided appropriate proof for our claims and referenced claims we have made using external sources.
4	Our team will never accept bribes for any reason
5	Our device is the first gesture recognition wearable technology specifically designed to work for a smart home. We want to use it to aid society in how it interacts with technology.
6	Our experience in gesture recognition has assisted us in creating a more advanced gesture recognition algorithm for our device. We have more ideas on making our algorithm modular, and have documented the limitation that our implementation has yielded.
7	Our group has taken feedback received from the instructors of ECE 445 as well as from our grades and improved our physical and software design of our device.
8	Our product has no prejudices of any kind.
9	We aim to have proper safety mechanisms in our device to prevent injury by the user, as well as prevent damage to their surroundings willingly. Similarly we do not aim to degrade or slander someone's reputation with the device.
10	We have given peer reviews to colleagues to help them in their professional development and will make sure our colleagues also abide by the code of ethics.

3.3.1 Risk Mitigation and Safety

Making a project in a safe manner is not only important when it comes to creating the project, but any injury that could occur before or after completion of the project is easily avoided by following safety procedures. Our work in the lab will require working with power supplies and batteries. Reviewing the Safe Practice for Lead Acid and Lithium Batteries, it is recommended to avoid battery use if a power supply can replace it [7]. We have also taken note of the different ways lithium batteries can be damaged and what procedures need to be taken when handling them. While powering our circuit in the lab, it is additionally crucial that at least two people work together. When working alone, any sort of issue, whether it be medical or not, is better handled with more than one person in the case that one person may become incapable of calling for assistance.

The University's Division of Research Safety's Laboratory Safety Training is a critical training course that teaches about proper lab procedures and how to handle different materials, hazardous or not. We have completed our training and understood what is expected when working in the lab. If any problems occur when powering our circuits, we will proceed as follows: Power down the affected station, fix the issues that may exist in the setup, double check our design specifications and if proper, turn the power on. For issues that may be more serious, say if a lithium battery were to be damaged and catch fire, the immediate step is to call 911, evacuate any other people in the lab, use the electrical-fire rated fire extinguishers and contact the appropriate people to inform them of this.

We do not plan on running electrical currents through human subjects, but we will ensure that any devices that a user may attach to their wrist, hand or elbow are properly insulated and have no means to injure.

3.4 Project Improvements

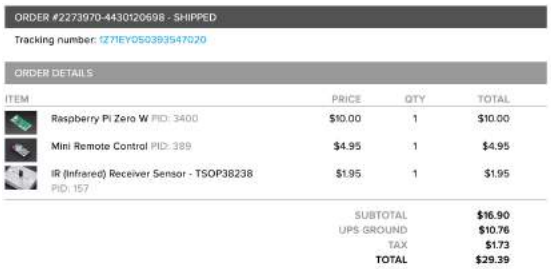
Some design changes that could be made to improve would be:

1. **Going finger-less:** A sleeker minimalistic design that only uses the base for each finger to allow the user to use their hands for day to day work.
2. **Optional elbow bands:** Making the elbow band optional by making a modular IMU algorithm that can adapt to using only the sensors in the glove if necessary.
3. **Integrating with Google Soli:** Google Soli is able to detect fine gestures with a very small footprint hardware wise. It has the potential to replace IMU and flex sensors altogether which would lead to our design being more robust, lighter and smaller.

4 Progress made on First Project

4.1 Parts list ordered

Below is a screenshot of some of the parts we ordered for our first project. Parts such as wires, resistors etc were already available to us.



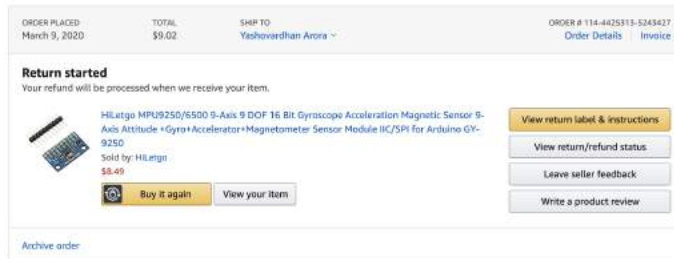
ORDER #2273970-4430120698 - SHIPPED

Tracking number: 1Z71EY050393547020

ORDER DETAILS

ITEM	PRICE	QTY	TOTAL
Raspberry Pi Zero W PID: 3400	$10.00	1	$10.00
Mini Remote Control PID: 389	$4.95	1	$4.95
IR (Infrared) Receiver Sensor - TSOP38238 PID: 157	$1.95	1	$1.95

SUBTOTAL	$16.90
UPS GROUND	$10.76
TAX	$1.73
TOTAL	$29.39



ORDER PLACED March 9, 2020 | TOTAL $9.02 | SHIP TO Yashovardhan Arora | ORDER # 114-4425313-5243427 | Order Details | Invoice

Return started

Your refund will be processed when we receive your item.

HiLetgo MPU9250/6500 9-Axis 9 DOF 16 Bit Gyroscope Acceleration Magnetic Sensor 9-Axis Attitude +Gyro+Accelerator+Magnetometer Sensor Module IIC/SPI for Arduino GY-9250

Sold by: HiLetgo

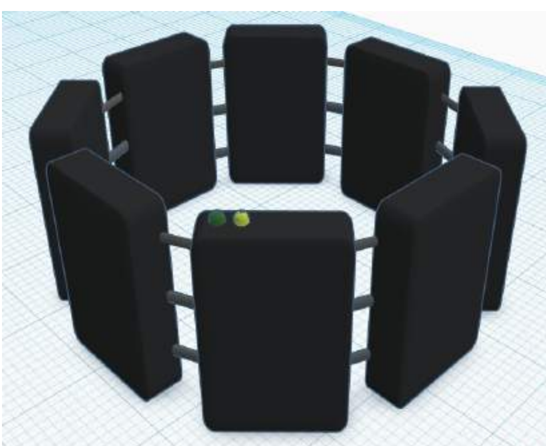
$8.49

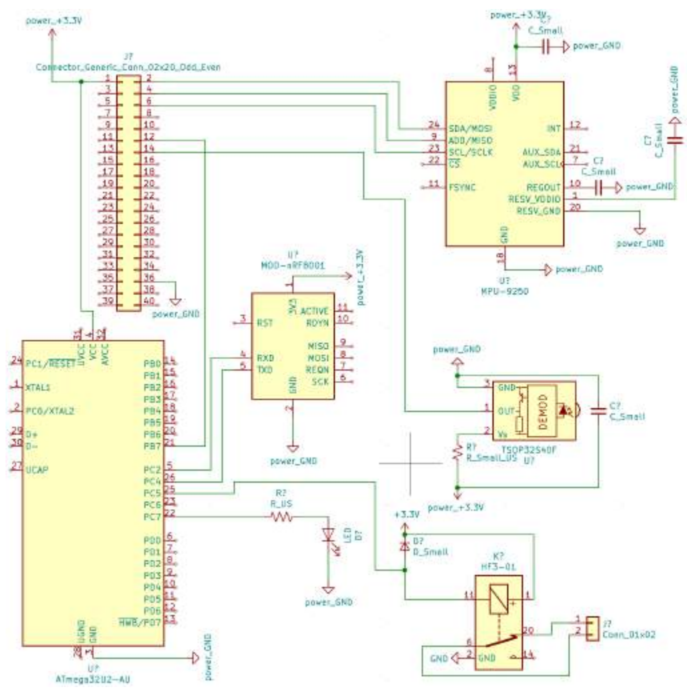
Buy it again | View your item

View return label & instructions | View return/refund status | Leave seller feedback | Write a product review

Archive order

4.2 Design Renders and Schematics





Our first project, called the wand, was CADed so that it could be 3D printed once the parts were assembled together. The schematics for the electronic components of the Wand were also constructed and can also be found here. Each submodule would fit into one of the boxes in the model and wires would run between the boxes for communication.

5 References

1. Tofel, K. (2020). Shhh! We're heading towards a quieter, gesture-based smart home - Stacey on IoT — Internet of Things news and analysis. [online] Stacey on IoT — Internet of Things news and analysis. Available at: https://staceyoniot.com/shhh-were-heading-towards-a-quieter-gesture-based-smart-home/ [Accessed 13 Feb. 2020].

2. Reisinger, Don. "Microsoft Has Finally Killed the Kinect Xbox Sensor." Fortune, Fortune, 25 Oct. 2017, fortune.com/2017/10/25/microsoft-kinect-xbox-sensor/.

3. "BMW Natural Interaction Introduced at the Mobile World Congress 2019." BMW BLOG, 25 Feb. 2019, www.bmwblog.com/2019/02/25/bmw-natural-interaction-introduced-at-the-mobile-world-congress-2019/.

4. "IEEE Code Of Ethics." Ieee.org. N. p., 2020. Web. 14 Feb. 2020.

5. "Samsung Finally Explains What Went Wrong With Those Note 7 Batteries." NBC News. N. p., 2017. Web. 14 Feb. 2020.

6. "Architect Desk Lamp Gesture Control" Amazon, Mon. 26 Sep. 2016, https://www.amazon.com/dp/B01J4LBAW2.

7. "Safe Practice for Lead Acid and Lithium Batteries" ECE 445 Spring 2016 Course Staff, Wed. 13 Apr. 2016, https://courses.engr.illinois.edu/ece445/documents/GeneralBatterySafety.pdf.

8. choudhury, romit. "ECE/CS 498 - Mobile Sensing And Applications." Courses.engr.illinois.edu. N. p., 2020. Web. 25 Feb. 2020.

9. Mummadi, Chaithanya Philips, Frederic Verma, Keshav Kasireddy, Shivaji Scholl, Philipp Kempfle, Jochen Van Laerhoven, Kristof. (2018). Real-Time and Embedded Detection of Hand Gestures with an IMU-Based Glove. Informatics. 5. 28. 10.3390/informatics5020028.

10. Al-rahayfeh, Amer Faezipour, Miad. (2014). Application of Head Flexion Detection for Enhancing Eye Gaze Direction Classification. 2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC 2014. 2014. 10.1109/EMBC.2014.6943753.