

2022

Summer Internship Work Report

Reporter: Ruiqi Mao

9/28/2022

CONTENTS

1

Motion
Sensor Glove

2

Augmented
Cane

3

Prosthetic
Hand

4

Participate in Lab
Experiments

5

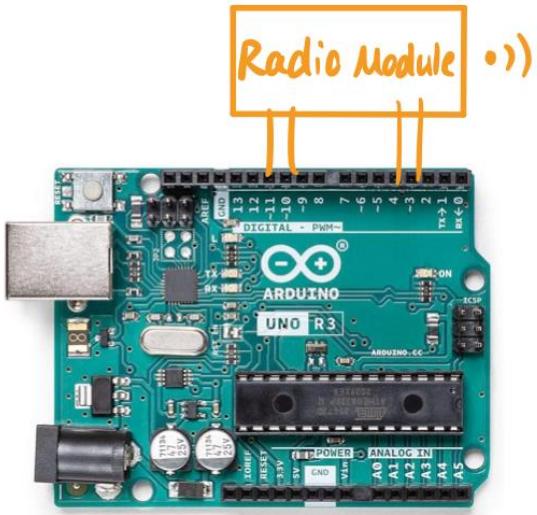
Acknowledgement



Motion Sensor Glove

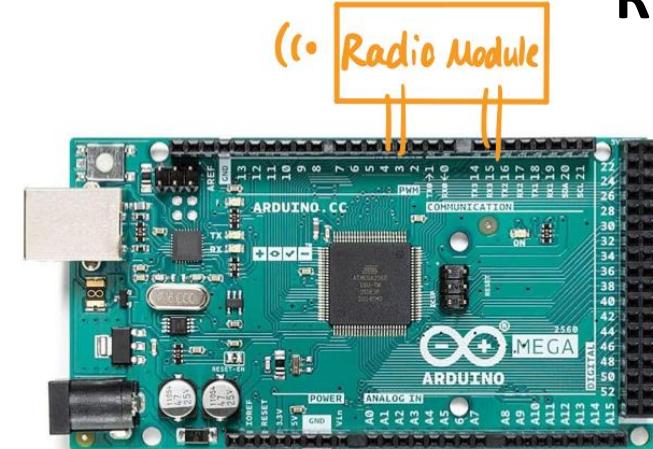
Project Overview

Receiver
(Arduino UNO)

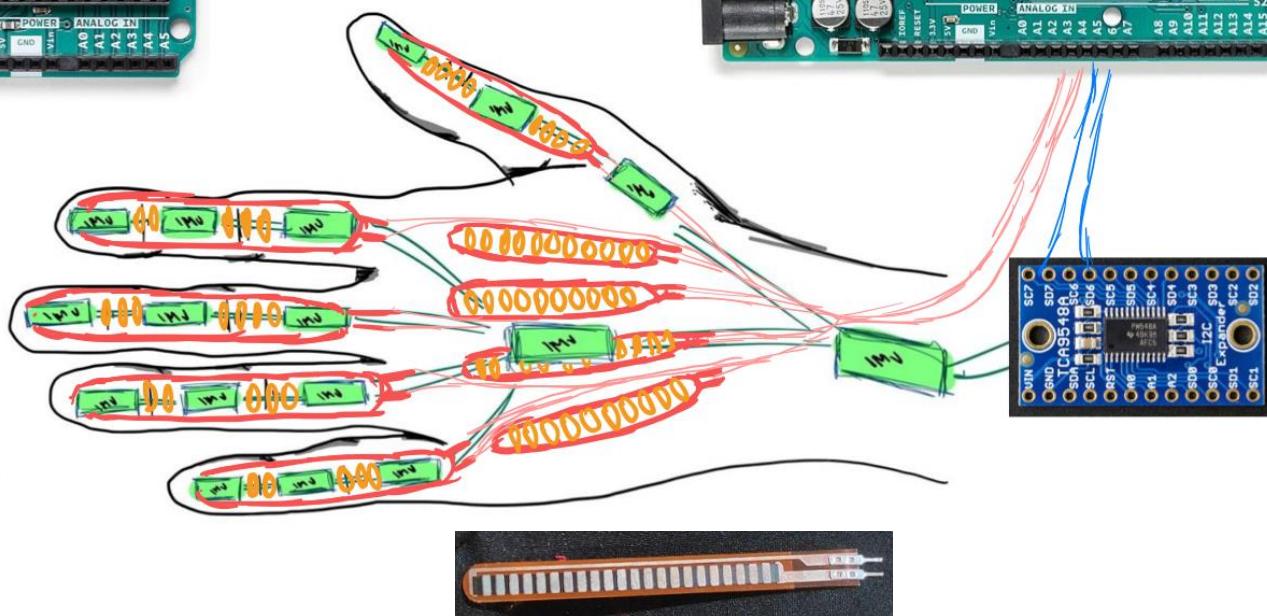


IMU
(LSM9DS1, which
has 9 degrees of
freedom)

Radio Module



Transmitter
(Arduino mega2560)



Multiplex
(TCA9548A)

Flex Sensor

Workflow of the Glove

Workflow

1 Read Data

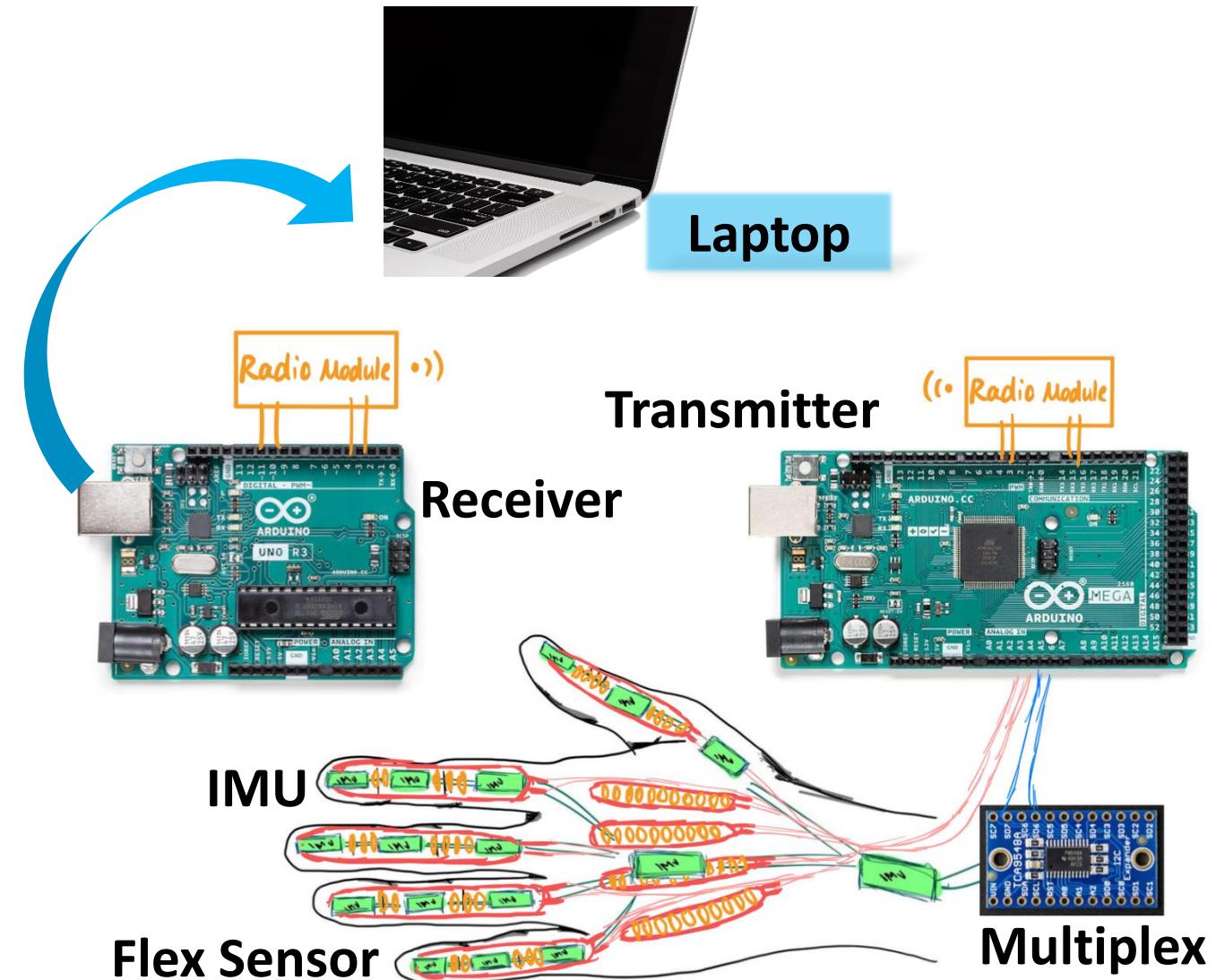
The data from the 16 IMUs and 9 Angle Sensors will be read by the Arduino mega2560 for 7 times per second.

2 Data Processing

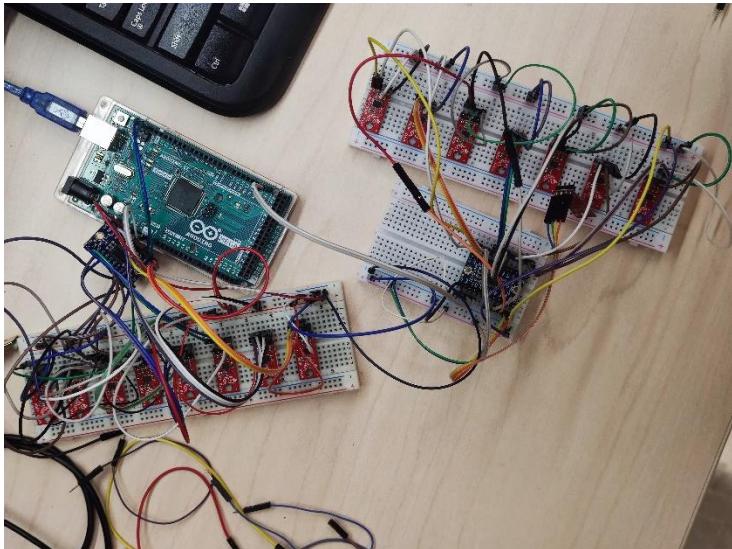
The data read from the sensors will be processed by the Arduino mega2560. Then we can get the acceleration, angular rate, magnetic field, direction data(from IMU) and angle data(from flex sensors)

3 Transmission

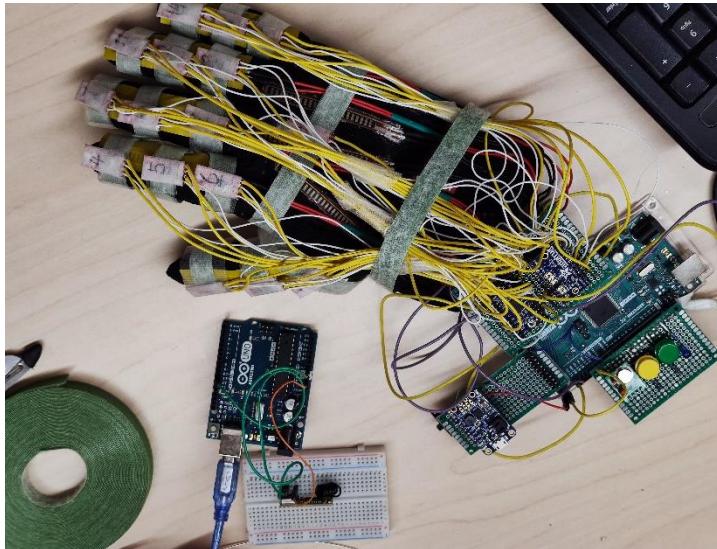
After the Arduino mega2560 get the data, it will work as a transmitter and send the data to the receiver(Arduino Uno), which is connected to the laptop. Then we can see the data from the laptop and do further analysis.



My Contribution to this Project



IMU module test



Overall assembly
and debugging

```
SampleUserPolling_ReadWrite.cs  Receiver_Code_V1.2.ino  Receiver_Code_V2.0.ino
100 }
101 }
102 }
103 }
104 }
105 }
106 }
107 }
108 }
109 }
110 }
111 }
112 }
113 }
114 }

float val8;
uint8_t datalen8 = sizeof(val8);
if (driver.recv((uint8_t*)&val8, &datalen8))
{
    //Serial.print("The angle from the seventh sensor is: ");
    Serial.print(val8);
    Serial.print("\n");
    //delay(100);
    count++;
}

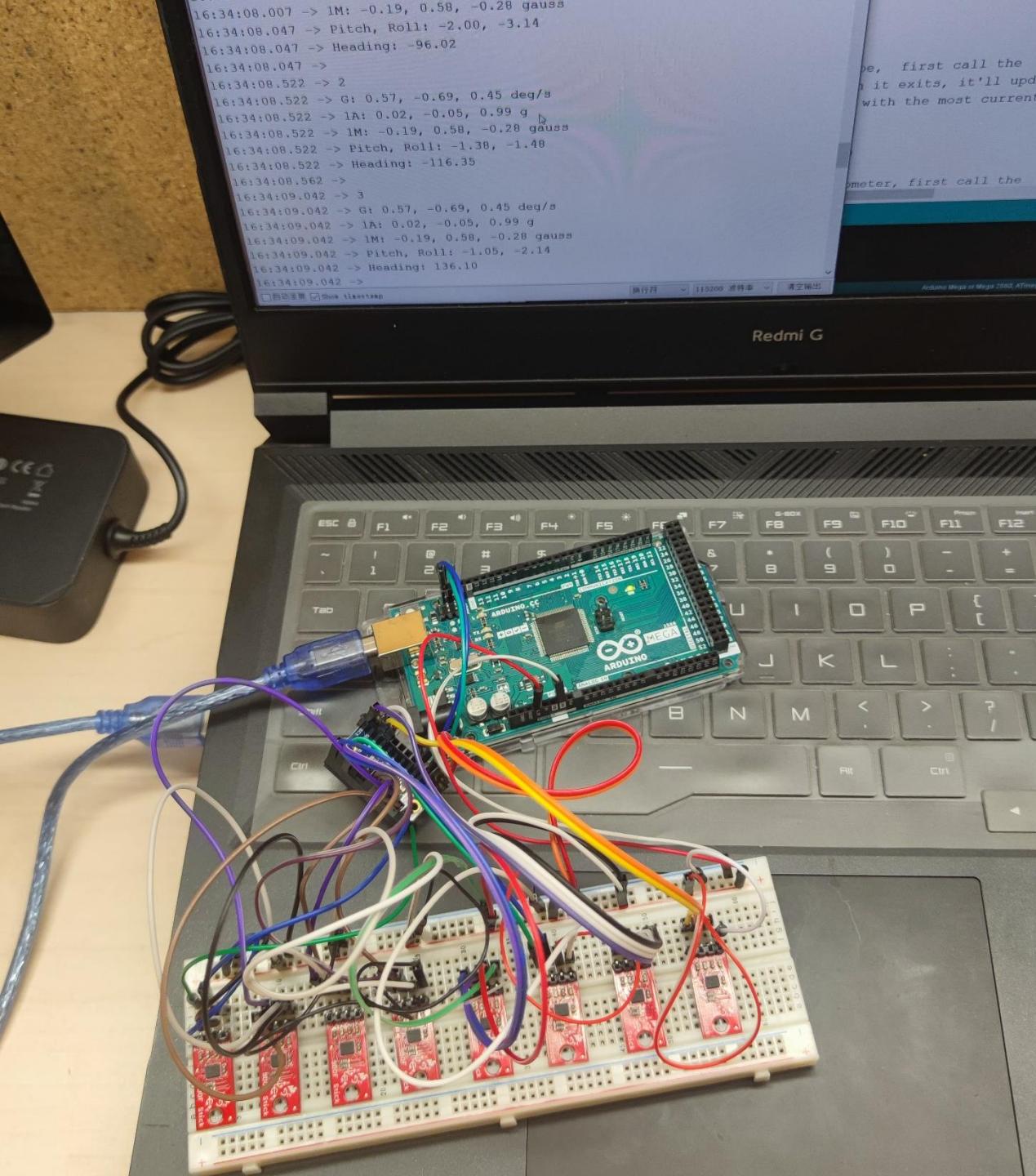
float val9;
uint8_t datalen9 = sizeof(val9);
if (driver.recv((uint8_t*)&val9, &datalen9))
{
    //Serial.print("The angle from the ninth sensor is: ");
    Serial.print(val9);
    Serial.print("\n");
    //delay(100);
    count++;
}

if(count==9){
    Serial.println();
    count=0;
}

Serial.print("\n");
```

The image shows a code editor with three tabs: SampleUserPolling_ReadWrite.cs, Receiver_Code_V1.2.ino, and Receiver_Code_V2.0.ino. The Receiver_Code_V1.2.ino tab is active, displaying C++ code for an Arduino sketch. The code includes sections for receiving data from sensors (using the driver.recv() function) and printing the received values to the serial monitor. The Receiver_Code_V2.0.ino tab shows a similar structure but with some differences in the logic, particularly in the handling of the received data. A red arrow points from line 43 of the V1.2.ino code to line 43 of the V2.0.ino code, highlighting a specific line of code that appears to be a comment or a placeholder.

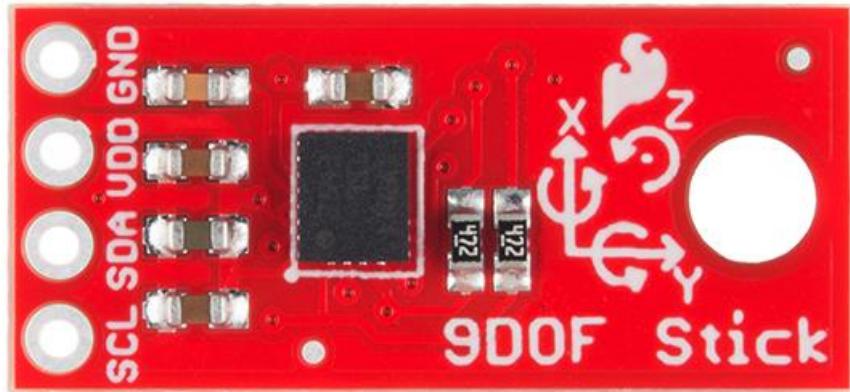
Code improvement



1.1 IMU Module Test

1.1 IMU Module Test

Hardware Overview



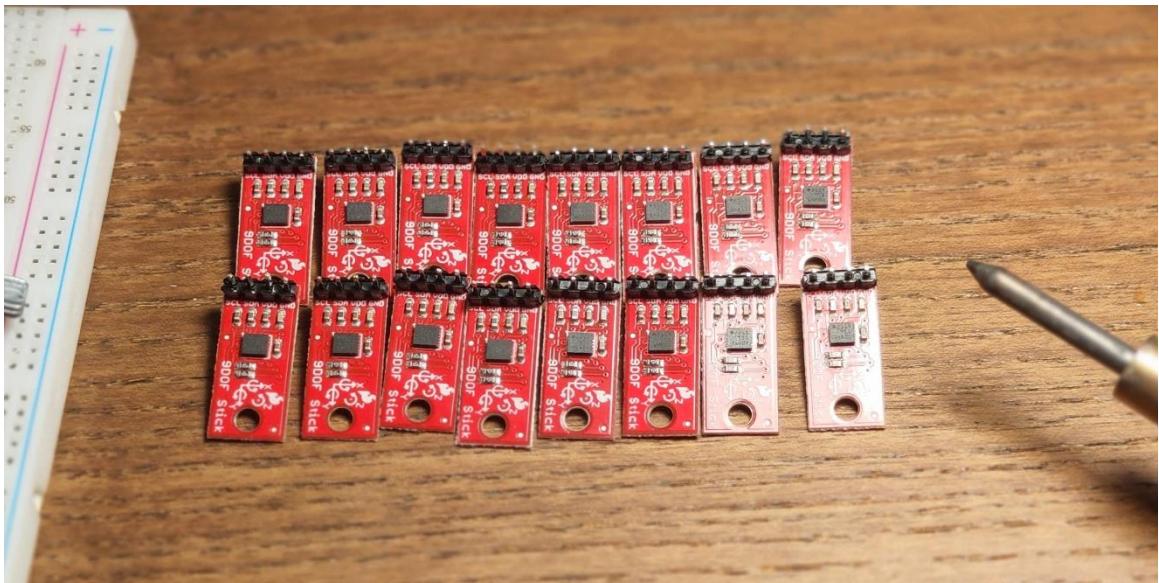
9DoF Sensor Stick (LSM9DS1)

LSM9DS1 Details:

- 3 acceleration channels, 3 angular rate channels, 3 magnetic field channels
- $\pm 2/\pm 4/\pm 8/\pm 16$ g linear acceleration full scale
- $\pm 4/\pm 8/\pm 12/\pm 16$ gauss magnetic full scale
- $\pm 245/\pm 500/\pm 2000$ dps angular rate full scale
- I2C serial interface
- Operating Voltage: 3.3V

1.1 IMU Module Test

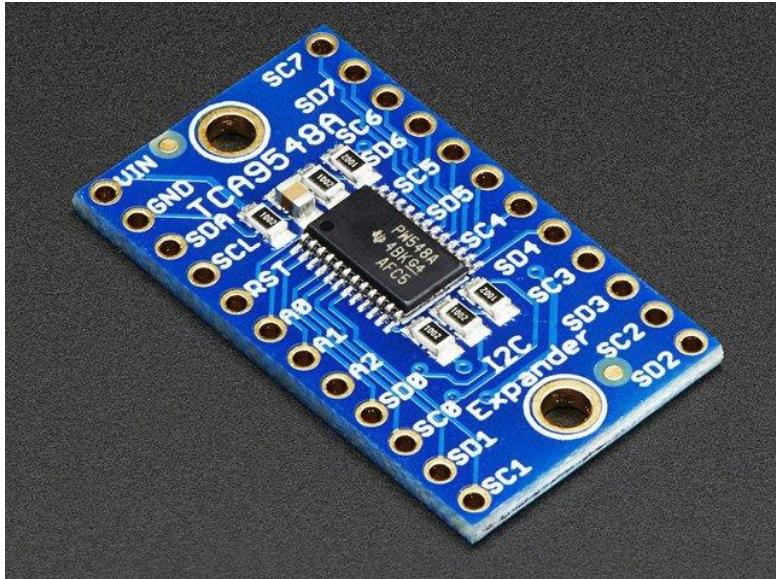
Problem: How to get the data from the 16 Sensors?



We need to use 16 sensors and each IMU need one SCL and one SDA for I2C communication with the controller. However, there are only two pins for each SCL and SDA on Arduino mega2560.

1.1 IMU Module Test

Solution: TCA9548A multiplex



The TCA9548A multiplexer is interesting in that it has an I2C address (0x70 by default) - and we basically send it a command to tell it which I2C multiplexed output we want to talk to, then we can address the board we want to address.

```
#define TCAADDR 0x70
#define TCAADDR2 0x71 //enable the A0 Pin on the second TCA
#define num 16 //the number of the IMUs
void tcaselect (uint8_t i) {

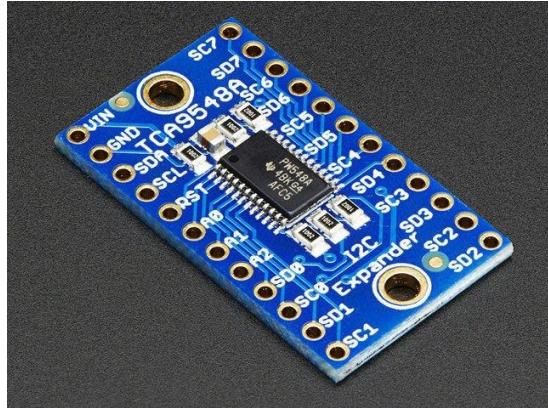
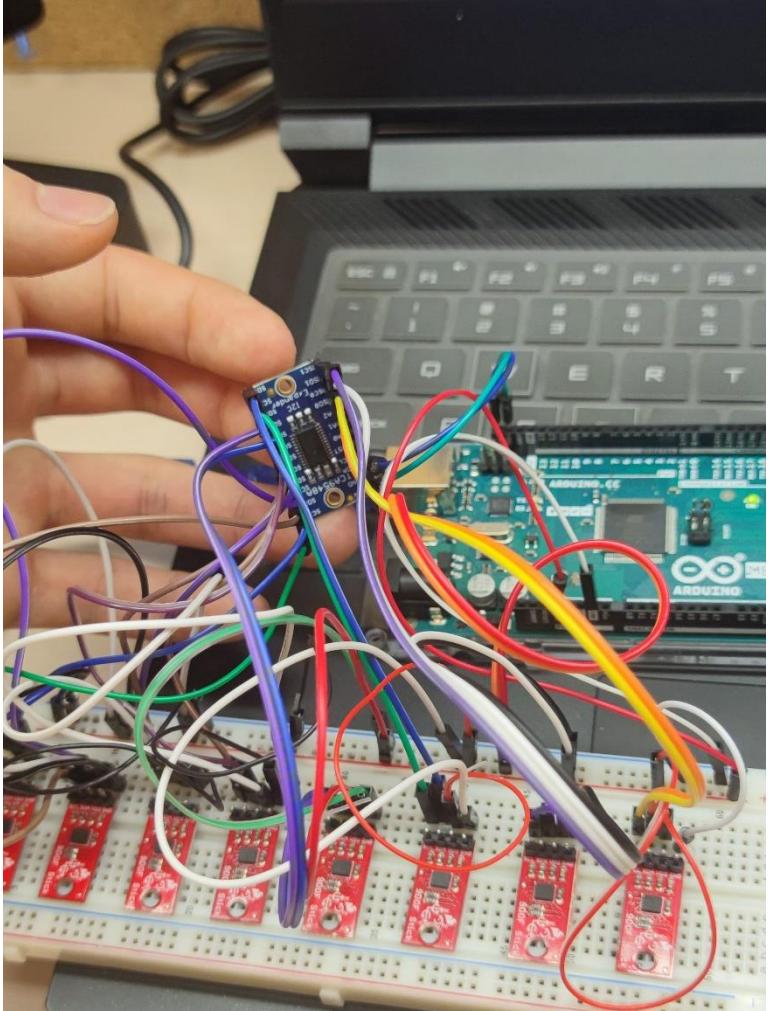
    if (i > num-1) return ;
    if (i<8){
        Wire.beginTransmission(TCAADDR);
        Wire.write(1 << i);
        Wire.endTransmission();
        Serial.println(i);
    }

    if (7<i<16){
        Wire.beginTransmission(TCAADDR2);
        Wire.write(1 << i);
        Wire.endTransmission();
        Serial.println(i);
    }
}
```

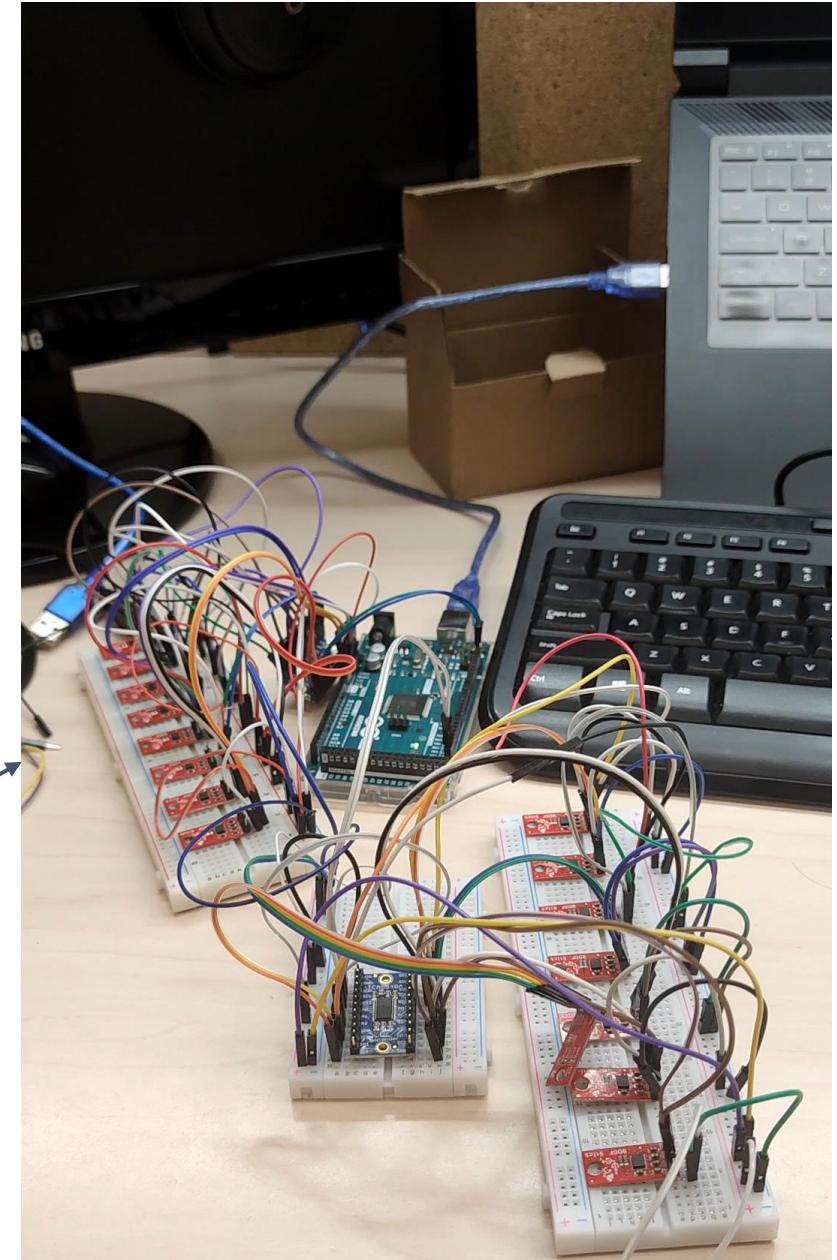
Using this code, we can call from `tcaselect(0)` to `tcaselect(15)` to set up the multiplexer. Then we just need to connect the IMU to the multiplex and use the `tcaselect()` to communicate with whichever IMU. After that, we can get all the data from the 16 sensors with only four SCL and SCA Pins.

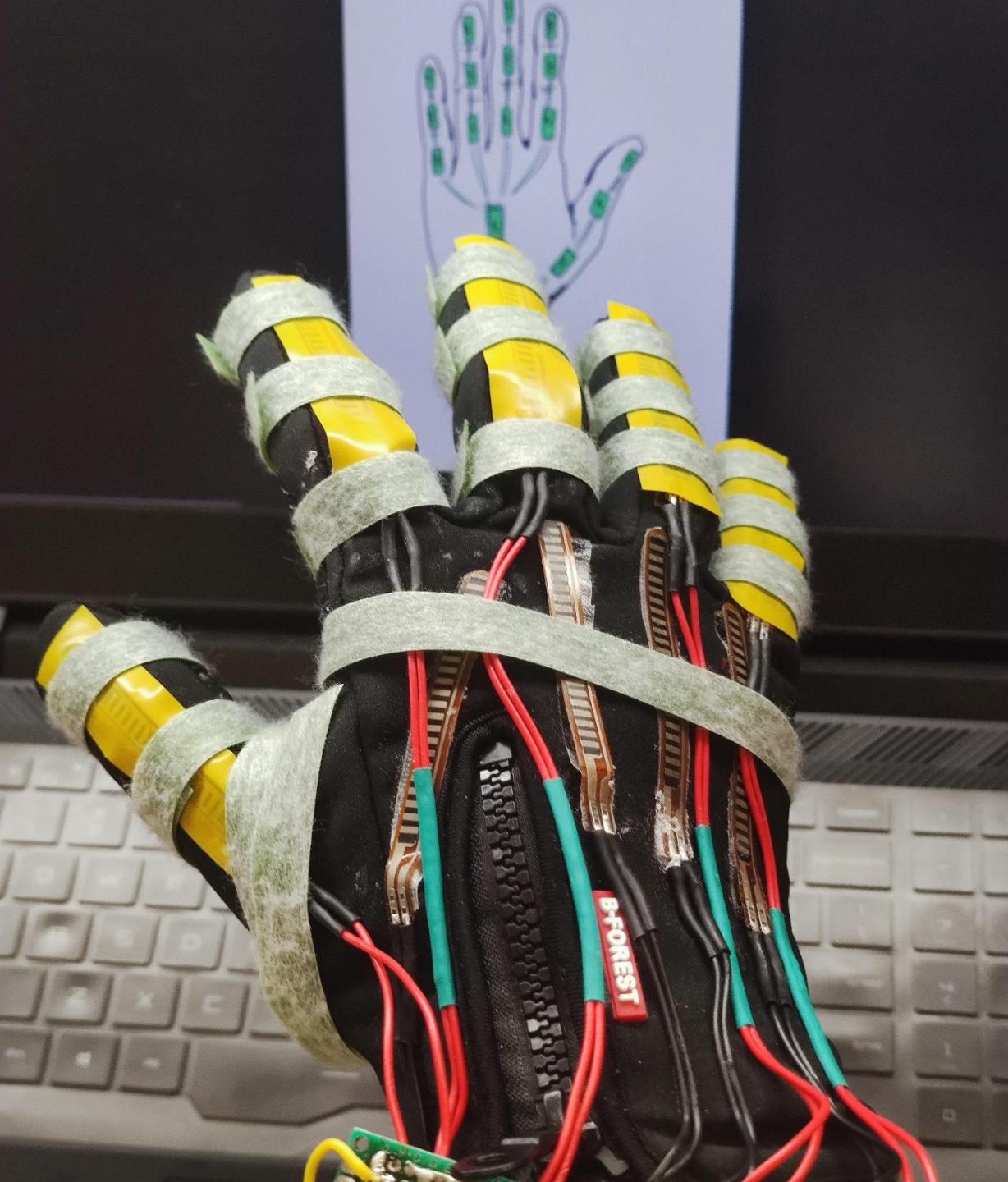
1.1 IMU Module Test

Test Result



We can get all the data from
the 16 sensors with only four
SCL and SCA Pins now!

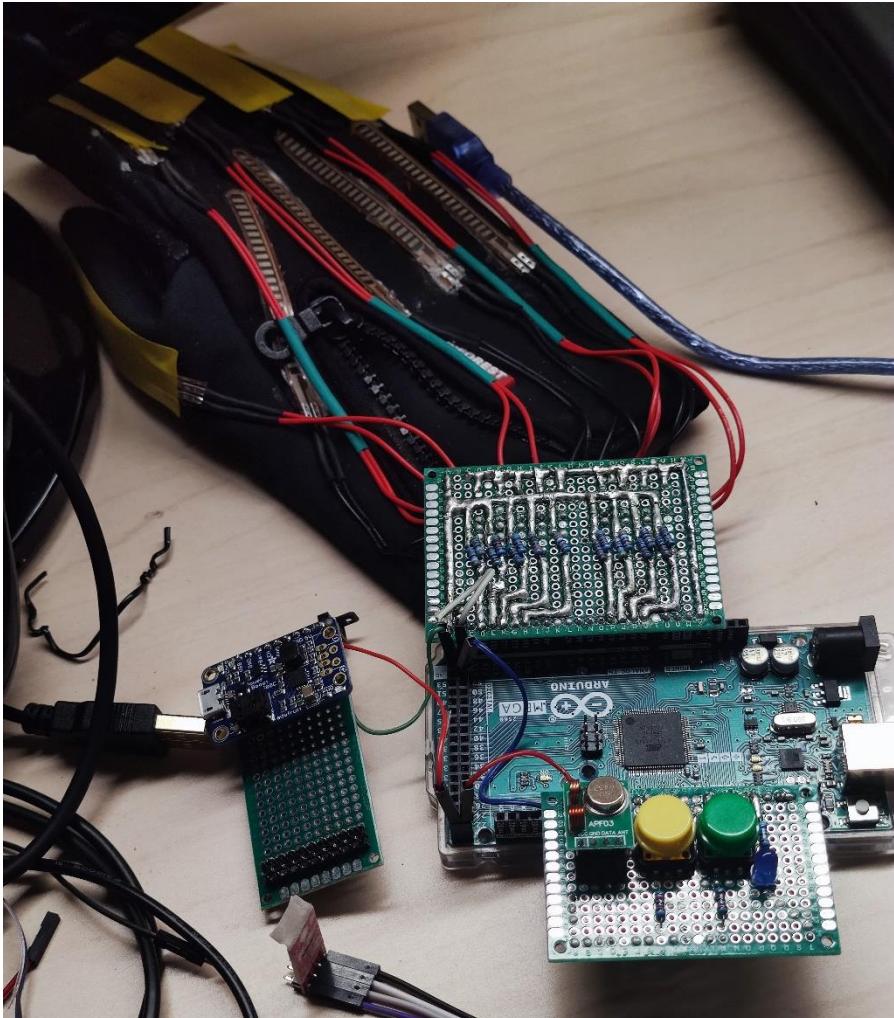




1.2 Overall Assembly and Debugging

1.2 Overall Assembly and Debugging

Problem: How should I add the IMU module to the glove?



When I got the glove, it looked like in the left picture. The glove already had several parts which had some impact on the movement of user's hand. That means I should be careful when I add the IMU module to the glove so that I can let the restriction on movement reduce to minimum as possible.

1.2 Overall Assembly and Debugging

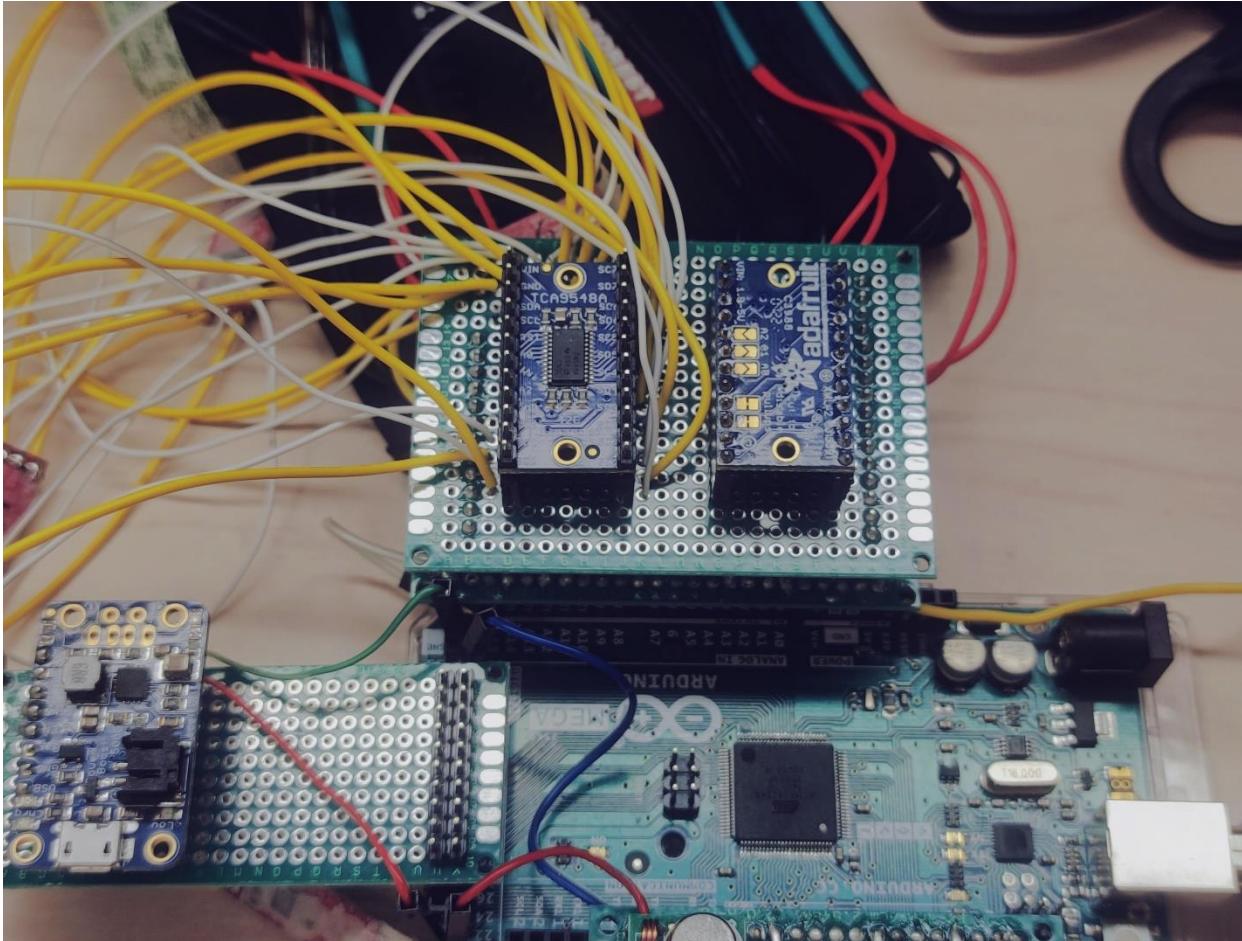
Solution1: Put IMU on Velcro

We decided to put Velcro around the finger first. Then I attached the IMU to the Velcro. In this way, it will be just easier for the user to adjust the position and tightness.



1.2 Overall Assembly and Debugging

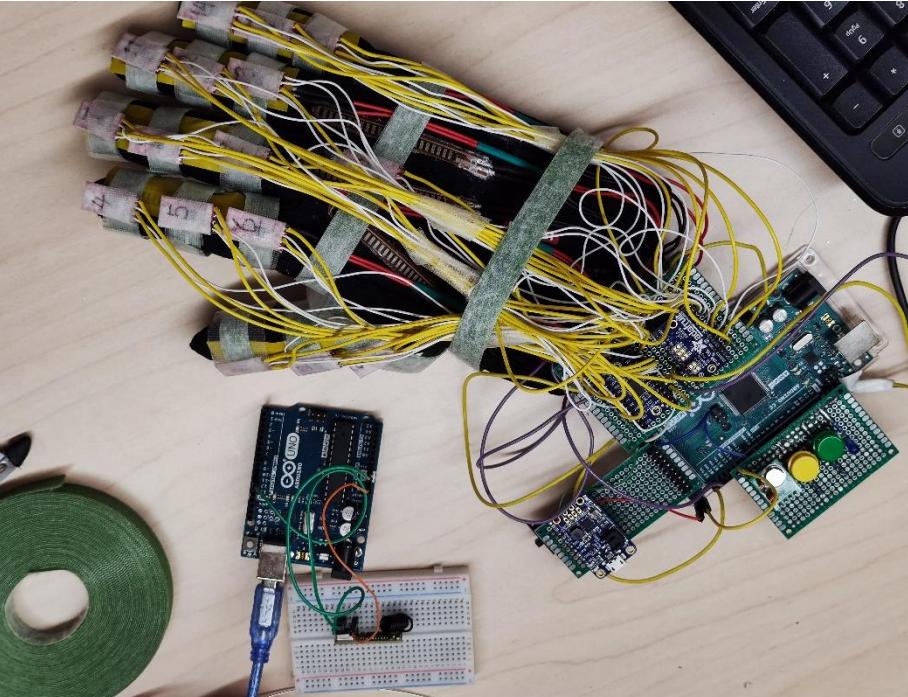
Solution2: Mount the multiplex board on the existing circuit board



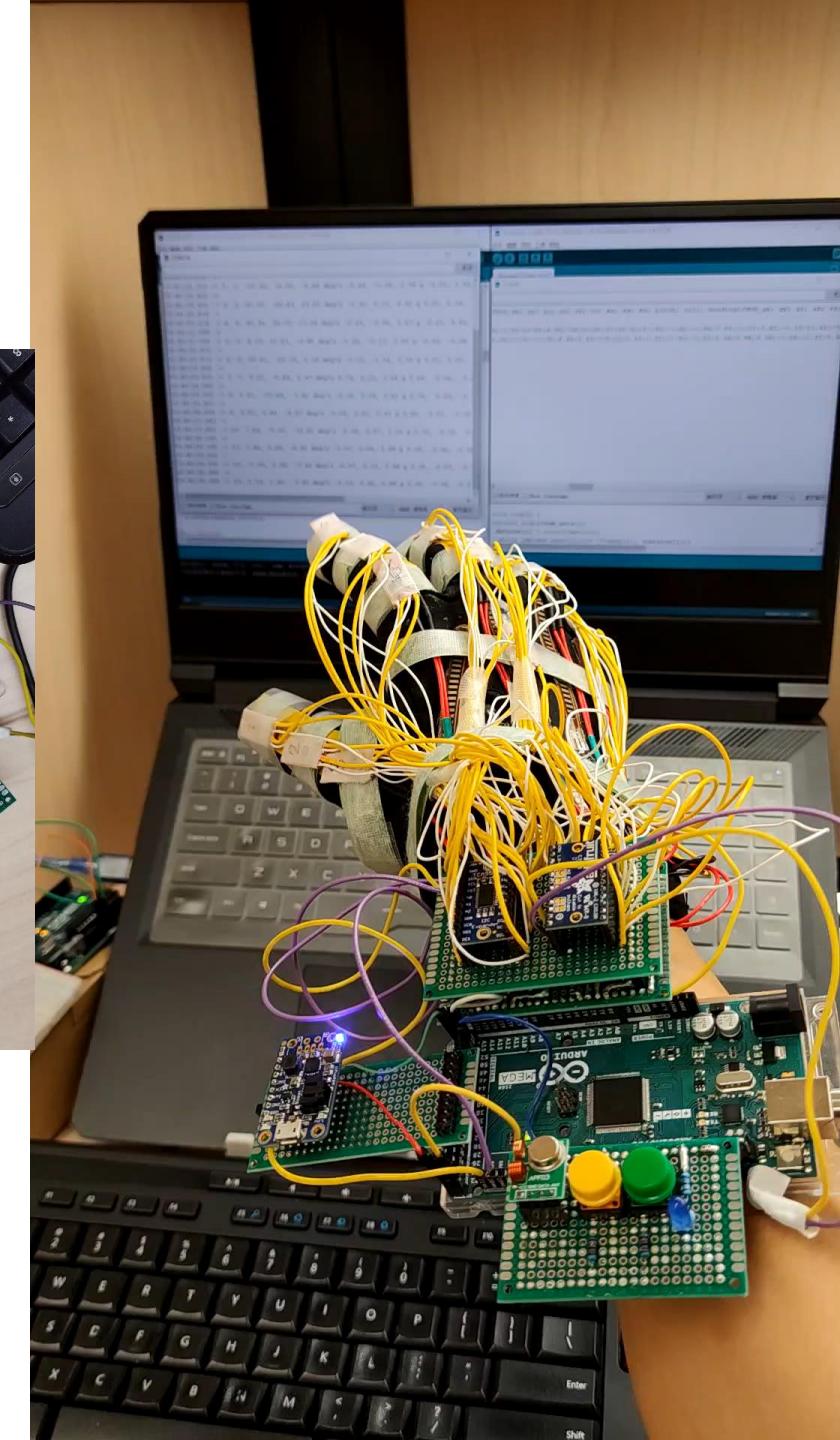
To avoid adding more restriction on movement, I decided to put the multiplex board on the existing board. I soldered some pins on the two boards so they can put together, easy for disassembly as well.

1.2 Overall Assembly and Debugging

Test Result



Although the overall structure is still relatively cumbersome, the basic movement restrictions of the user's hands are minimized



✓ Receiver Code V1.2

Receiver Code V1.2.ino

✓ Receiver_Code_V2.0

Receiver Code V2.0.ino

Transmitter Code V1.2

Transmitter Code V1.2.ino

Transmitter Code V2.0

Transmitter Code V2.0.ino

> SparkFun_LSM9DS1_Arduino_Library-master

✓ testonIMUs

teston|MUs.ino

✓ testonIMUsfor16

€ testonIMUsfor16.ino

✓ testonIMUsfor16v1.50

teston|MUsfor16y2.50.jno

✓ testonIMUsfor16v2.0

teston|MUsfor16v2.0.jno

1.3 Code Improvement

1.3 Code Improvement

Improvement1: Code shortening

```
17 void loop() {  
18     float val1;  
19     uint8_t datalen1 = sizeof(val1);  
20     if (driver.recv((uint8_t*)&val1, &datalen1))  
21     {  
22         //Serial.print("The angle from the first sensor is: ");  
23         Serial.print(val1);  
24         Serial.print(";" );  
25         //delay(100);  
26         count++;  
27     }  
28  
29     float val2;  
30     uint8_t datalen2 = sizeof(val2);  
31     if (driver.recv((uint8_t*)&val2, &dalen2))  
32     {  
33         //Serial.print("The angle from the second sensor is: ");  
34         Serial.print(val2);  
35         Serial.print(";" );  
36         //delay(100);  
37         count++;  
38     }  
39  
40     float val3;  
41     uint8_t datalen3 = sizeof(val3);  
42     if (driver.recv((uint8_t*)&val3, &dalen3))  
43     {  
44  
45  
46
```

Before

```
26 void loop() {  
27     for(int i=0;i<num_data;){  
28         datalen[i] = sizeof(val[i]);  
29         while (driver.recv((uint8_t*)&val[i], &dalen[i]))  
30         {  
31             Serial.print(val[i]);  
32             Serial.print(";" );  
33             //delay(100);  
34             i++;  
35         }  
36     }  
37     Serial.print("\n");  
38 }  
  
76 void loop() {  
77     int t=0;  
78     for(;t<num;t++)  
79     {  
80         tcaselect(t);  
81         if ( imu[t].gyroAvailable() )  
82             imu[t].readGyro();  
83         if ( imu[t].accelAvailable() )  
84             imu[t].readAccel();  
85         if ( imu[t].magAvailable() )  
86             imu[t].readMag();  
87  
88         if ((lastPrint + PRINT_SPEED) < millis())  
89         {  
90             printGyro1(t); // Print "G: gx, gy, gz"  
91             printAccel1(t); // Print "A: ax, ay, az"  
92             printMag1(t); // Print "M: mx, my, mz"  
93             printAttitude(imu[t].ax, imu[t].ay, imu[t].az,  
94                             -imu[t].my, -imu[t].mx, imu[t].mz);  
95             Serial.println();  
96             lastPrint = millis(); // Update lastPrint time  
97         }  
98         delay(5);  
99     }  
100 }
```

After

By using loops and arrays, I shortened the code significantly.

1.3 Code Improvement

Result

```
SampleUserPolling_ReadWrite.cs  Receiver_Code_V1.2.ino  ...
UAProject > glove_code > Sarthak code > Receiver_Code_V1.2 > Receiver_Code_V1.2.ino > loop()
100 }
101
102 float val8;
103 uint8_t datalen8 = sizeof(val8);
104 if (driver.recv((uint8_t*)&val8, &datalen8))
105 {
106     //Serial.print("The angle from the seventh sensor is: ");
107     Serial.print(val8);
108     Serial.print(",");
109     //delay(100);
110     count++;
111 }
112
113
114
115 float val9;
116 uint8_t datalen9 = sizeof(val9);
117 if (driver.recv((uint8_t*)&val9, &datalen9))
118 {
119     //Serial.print("The angle from the ninth sensor is: ");
120     Serial.print(val9);
121     Serial.print(",");
122     //delay(100);
123     count++;
124 }
125
126 if(count==9){
127     Serial.println();
128     count=0;
129 }
130
131 }

UAProject > glove_code > Sarthak code > Receiver_Code_V2.0 > Receiver_Code_V2.0.ino
14     driver.init();
15
16     for(int i=1;i<=16;i++){
17         Serial.print("IMU");
18         Serial.print(i);
19         Serial.print("_gx; gy; gz; ax; ay; az; mx; ");
20     }
21     Serial.print("Little Finger;Little Finger M");
22     Serial.println("\n");
23     // Serial.println("begin");
24 }
25
26 void loop() {
27     for(int i=0;i<num_data;){
28         datalen[i] = sizeof(val[i]);
29         while (driver.recv((uint8_t*)&val[i], &datalen[i]) != -1)
30         {
31             //    if(val[i]==0){
32             //        Serial.print("\n");
33             //        i=0;
34             //    }
35             //Serial.print("The angle from the first");
36             Serial.print(val[i]);
37             Serial.print(",");
38             //delay(100);
39             i++;
40         }
41     }
42     Serial.print("\n");
43 }
```

The length of the overall receiver code(after I added the IMU code) is reduced to 43 lines than before (131 lines, this is the code length for the flex sensors).

1.3 Code Improvement

Improvement2: Printing Speed

```
for(;t<num;t++){  
    tcaselect(t);  
    if ( imu[t].gyroAvailable() )  
        imu[t].readGyro();  
    if ( imu[t].accelAvailable() )  
        imu[t].readAccel();  
    if ( imu[t].magAvailable() )  
        imu[t].readMag();  
  
    if ((lastPrint + PRINT_SPEED) < millis())  
    {  
        printGyro1(t); // Print "G: gx, gy, gz"  
        printAccel1(t); // Print "A: ax, ay, az"  
        printMag1(t); // Print "M: mx, my, mz"  
        printAttitude(imu[t].ax, imu[t].ay, imu[t].az,  
                      -imu[t].my, -imu[t].mx, imu[t].mz);  
        Serial.println();  
        lastPrint = millis(); // Update lastPrint time  
    }  
    delay(5);  
}
```

```
imu_mx[select]=imu[select].calcMag(imu[select].mx);  
imu_my[select]=imu[select].calcMag(imu[select].my);  
imu_mz[select]=imu[select].calcMag(imu[select].mz);  
  
Serial.print(imu_mx[select], 2);  
Serial.print(", ");  
Serial.print(imu_my[select], 2);  
Serial.print(", ");  
Serial.print(imu_mz[select], 2);  
Serial.print(" gauss ");  
  
driver.send((uint8_t *)&imu_mx[select], sizeof(imu_mx[select]));  
driver.waitPacketSent();  
driver.send((uint8_t *)&imu_my[select], sizeof(imu_my[select]));  
driver.waitPacketSent();  
driver.send((uint8_t *)&imu_mz[select], sizeof(imu_mz[select]));  
driver.waitPacketSent();  
  
#elif defined PRINT_RAW  
    Serial.print(imu[select].mx);  
    Serial.print(", ");  
    Serial.print(imu[select].my);  
    Serial.print(", ");  
    Serial.print(imu[select].mz);  
#endif  
}
```

The `delay()` and `driver.waitPacketSent()` functions used in the code will cause blocking and decrease data printing speed.

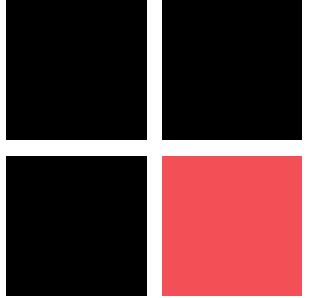
1.3 Code Improvement

Result

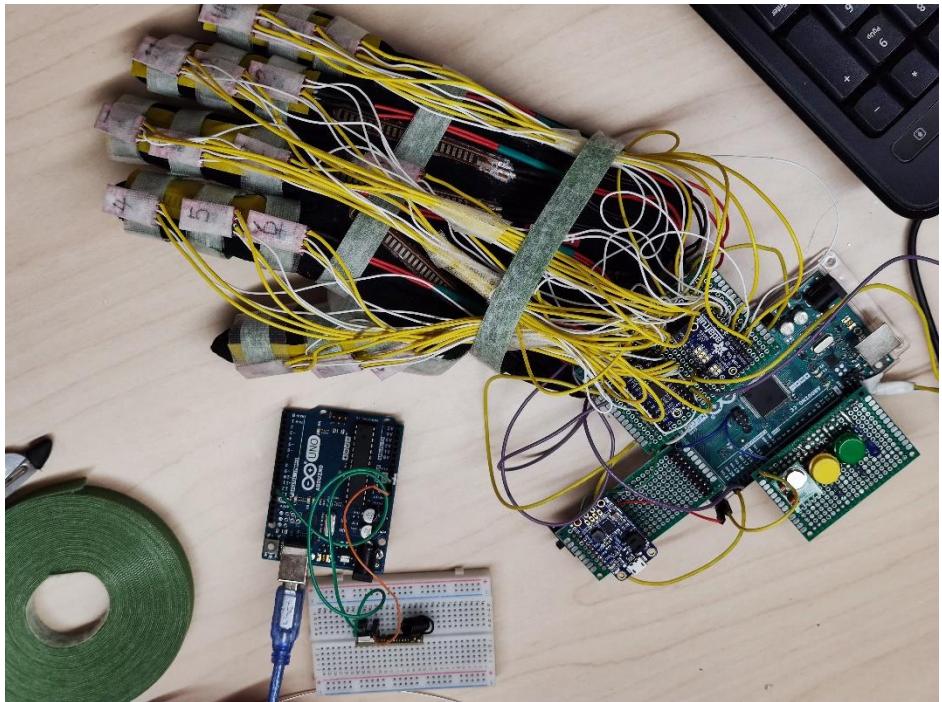
```
16:00:59.995 -> The angle from the first sensor is: 177.81
16:00:59.995 -> The angle from the second sensor is: 156.87
16:00:59.995 -> The angle from the third sensor is: 1667.57
16:00:59.995 -> The angle from the fourth sensor is: 1076.82
16:01:00.041 -> The angle from the fifth sensor is: -259.13
16:01:00.041 -> The angle from the sixth sensor is: 1105.86
16:01:00.041 -> The angle from the seventh sensor is: 592.42
16:01:00.041 -> The angle from the eighth sensor is: -1114.43
16:01:00.041 -> The angle from the ninth sensor is: -682830.00
16:01:00.041 -> 0: 0: 0.70, 1.40, 3.37 deg/s 0.25, -0.50, 0.66 g 0.05, 0.04, -0.55 gauss -16.87, -37.27, -137.32
16:01:00.041 ->
16:01:00.041 -> 1: 1: 0.11, 0.03, -3.77 deg/s 0.06, -0.47, 0.00 g 0.59, 0.03, -0.51 gauss -7.19, -89.99, -168.61
16:01:00.088 ->
16:01:00.088 -> 2: 2: 0.28, 0.14, 0.24 deg/s 0.25, -0.47, 0.89 g 0.05, 0.36, -0.57 gauss -14.16, -27.65, -90.03
16:01:00.088 ->
16:01:00.088 -> 3: 3: 1.16, 2.25, 0.08 deg/s 0.00, -0.25, 0.96 g 0.05, 0.15, -0.53 gauss -0.05, -14.40, -98.28
16:01:00.088 ->
16:01:00.088 -> 4: 4: 1.12, 0.00, 0.00 deg/s 0.06, -0.43, 0.94 g 0.01, 0.31, -0.36 gauss -3.46, -24.34, -83.10
16:01:00.136 ->
16:01:00.136 -> 5: 5: 0.00, 1.12, 0.28 deg/s 0.02, 0.01, 0.01 g 0.04, 0.01, -0.57 gauss -49.02, 66.09, -160.37
16:01:00.136 ->
16:01:00.136 -> 6: 6: 0.28, 3.79, -0.98 deg/s 0.25, -0.46, 0.78 g 0.02, 0.02, -1.15 gauss -15.39, -30.32, -123.85
16:01:00.136 ->
16:01:00.136 -> 7: 7: 5.22, 0.07, 0.03 deg/s 0.00, -0.25, 0.32 g 0.06, 0.00, -0.57 gauss -0.09, -38.19, -170.61
16:01:00.183 ->
16:01:00.183 -> 8: 0.05, 0.48, -6.44 deg/s 0.12, -0.24, 0.94 g 0.35, 0.03, -0.40 gauss -7.19, -14.15, -167.09
16:01:00.183 ->
16:01:00.183 -> 9: 0.18, 0.98, -6.32 deg/s 0.13, -0.24, 0.94 g 0.35, 0.03, -0.40 gauss -7.41, -14.13, -167.00
16:01:00.183 ->
16:01:00.183 -> 10: 0.28, 0.50, -7.44 deg/s 0.13, -0.24, 0.94 g 0.35, 0.03, -0.41 gauss -7.50, -14.05, -167.15
16:01:00.230 ->
16:01:00.230 -> 11: 0.22, 0.52, -6.62 deg/s 0.12, -0.24, 0.94 g 0.35, 0.02, -0.40 gauss -7.31, -14.06, -167.60
16:01:00.230 ->
16:01:00.230 -> 12: 0.16, 0.60, -7.07 deg/s 0.13, -0.24, 0.94 g 0.36, 0.03, -0.40 gauss -7.40, -14.03, -167.17
16:01:00.230 ->
16:01:00.277 -> 13: 0.13, 0.66, -6.78 deg/s 0.13, -0.23, 0.94 g 0.35, 0.03, -0.40 gauss -7.45, -14.06, -167.28
```

The data printing frequency is improved from 0.1Hz to 7.5Hz (to be improved)

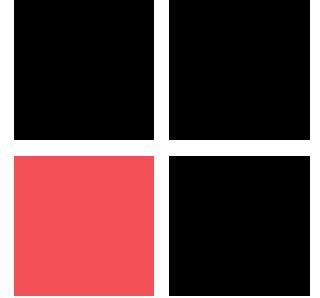
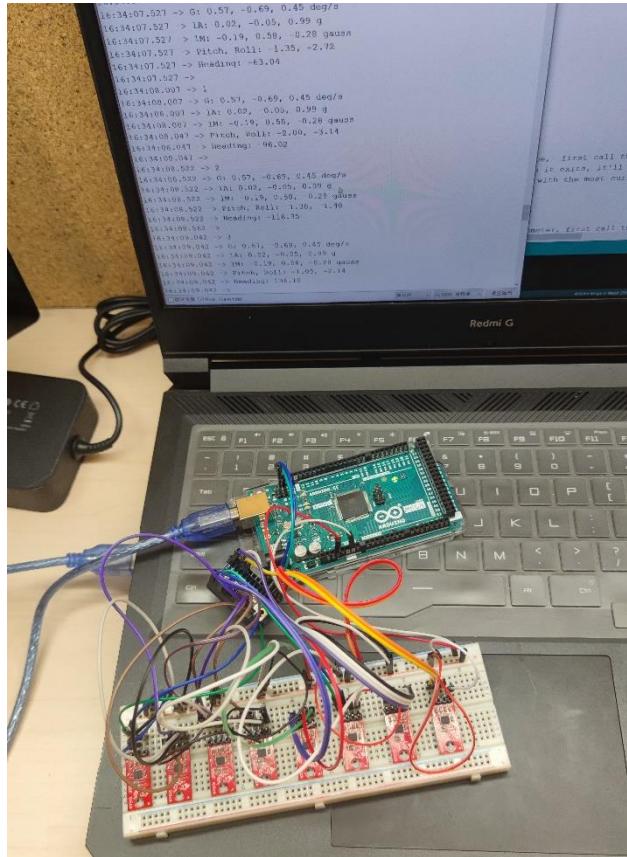
1.4 Further Improvement of the Glove



Structure Simplification



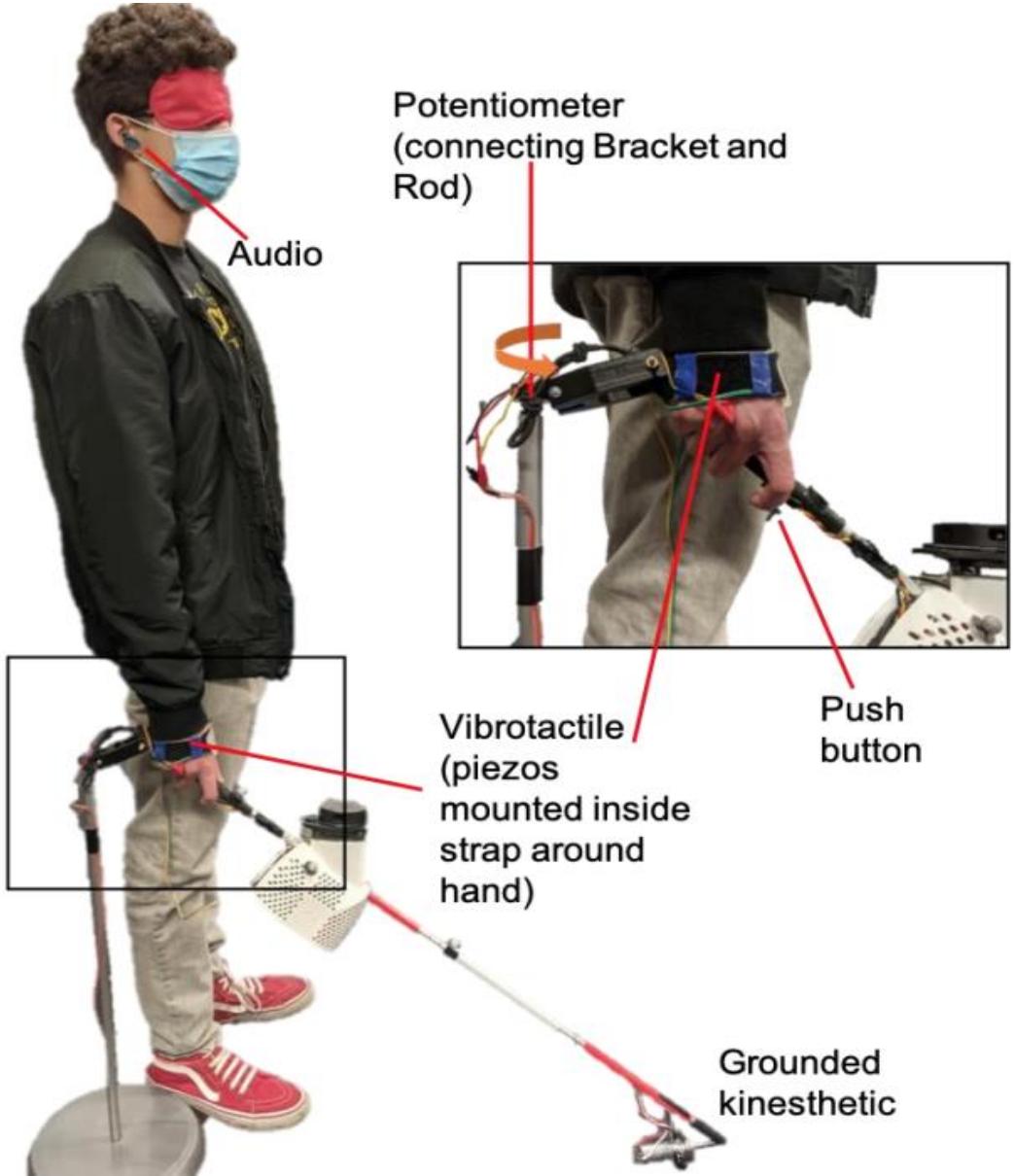
Improve Data Printing Speed





Augmented Cane

Augmented Cane

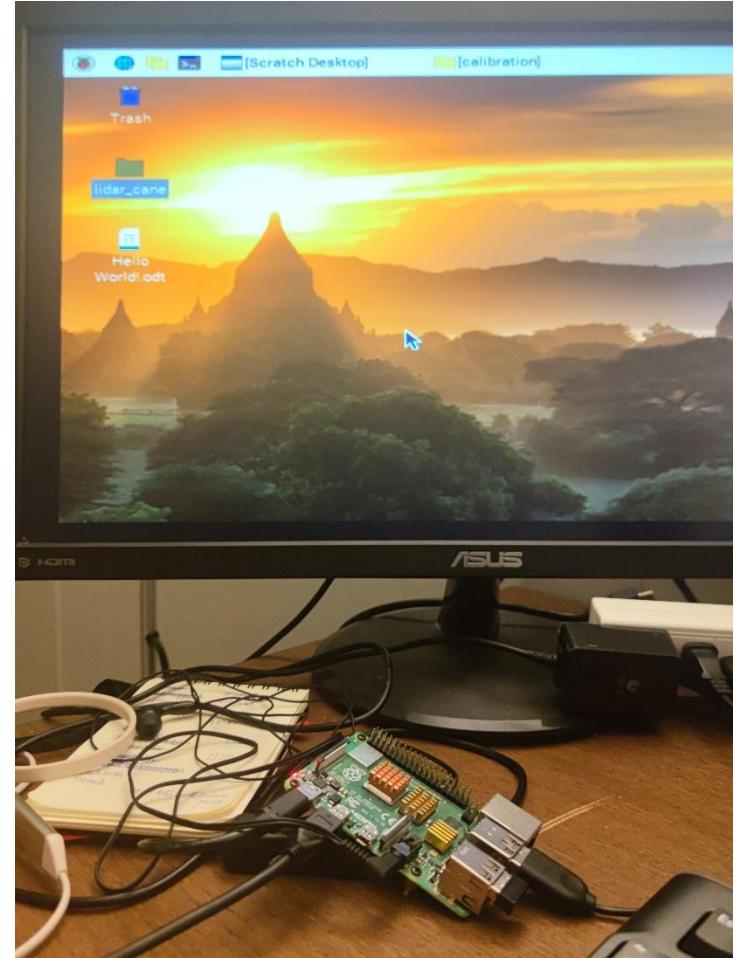


Augmented Cane is a white cane with a comprehensive set of sensors and an intuitive feedback method to steer the user, which addresses navigation challenges and improves mobility for people with impaired vision. It is a cane developed by the students in Stanford University.

What I have done for this Project

	A	B	C	D	E	F	G	H
1	Number	Part Name	Vendor	Quantity	Cost	Shipping	Total Cost	Link
2	1	RPLidar A1	Amazon	1	\$139.99	\$0.00	\$139.99	Slamtec RPLIDAR A1
3	2	Motor(12V 200rpm)	Amazon	1	\$17.03	\$0.00	\$17.03	DC12V 200RPM Gear
4	3	GPS Breakout	Elmwoods	1	\$52.99	\$0.00	\$52.99	https://elmwoodelectro
5	4	NXP IMU	Elmwoods	1	\$19.99	\$0.00	\$19.99	https://elmwoodelectro
6	5	RPi Camera	Elmwoods	1	\$39.99	\$0.00	\$39.99	Raspberry Pi NoIR Ca
7	6	RPi Camera Cable	Elmwoods	1	\$4.99	\$0.00	\$4.99	https://elmwoodelectro
8	7	Pi Cobbler Breakout	Elmwoods	1	\$11.99	\$0.00	\$11.99	https://elmwoodelectro
9	8	Protohat	Elmwoods	1	\$7.99	\$0.00	\$7.99	https://elmwoodelectro
10	9	GPS Antenna	Elmwoods	1	\$19.99	\$0.00	\$19.99	https://elmwoodelectro
11	10	LiPo Battery	Amazon	1	\$39.99	\$0.00	\$39.99	https://www.amazon.ca
12	11	LiPo Charger	Amazon	1	\$22.99	\$0.00	\$22.99	melasta 9.6V 7.2V 8.4
13	12	Motor Driver	Elmwoods	1	\$10.99	\$0.00	\$10.99	https://elmwoodelectro
14	13	White Cane	Amazon	1	\$21.99	\$0.00	\$21.99	https://www.amazon.ca
15	14	Ear Buds	Amazon	1	\$49.89	\$0.00	\$49.89	https://www.amazon.ca
16	15	Aluminum mounting hub	Amazon	1	\$12.49	\$0.00	\$12.49	https://www.amazon.ca
17	17	Omniwheel	Amazon	1	\$75.14	\$0.00	\$75.14	https://www.amazon.ca
18	18	4mmx100mm shaft (18pcs for sale)	Amazon	2	\$1.00	\$0.00	\$2.00	https://www.amazon.ca
19					TOTAL:		\$550.43	
20								

Made the bill of materials for the project



Learned to use Raspberry Pi



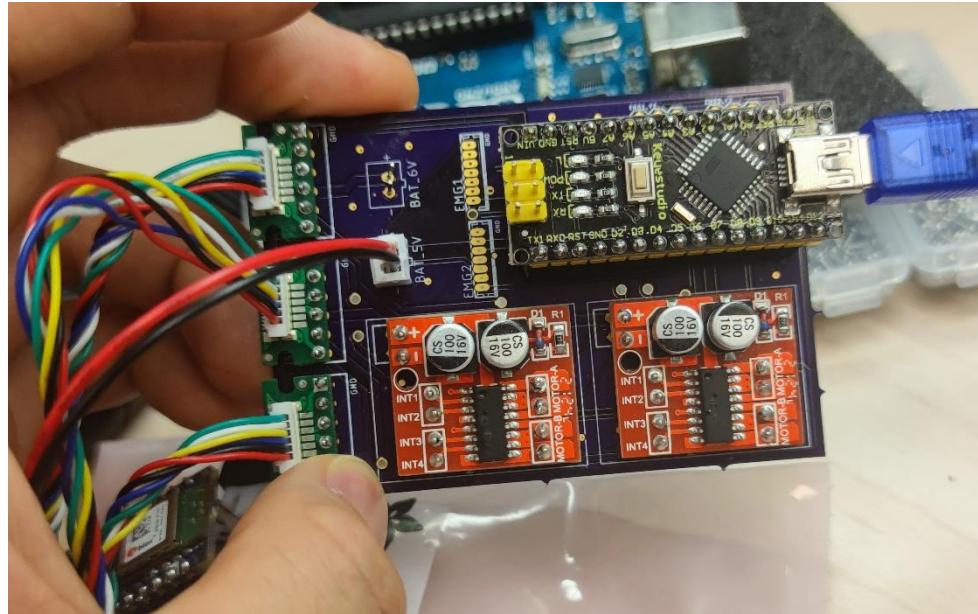
Prosthetic Hand

Prosthetic Hand

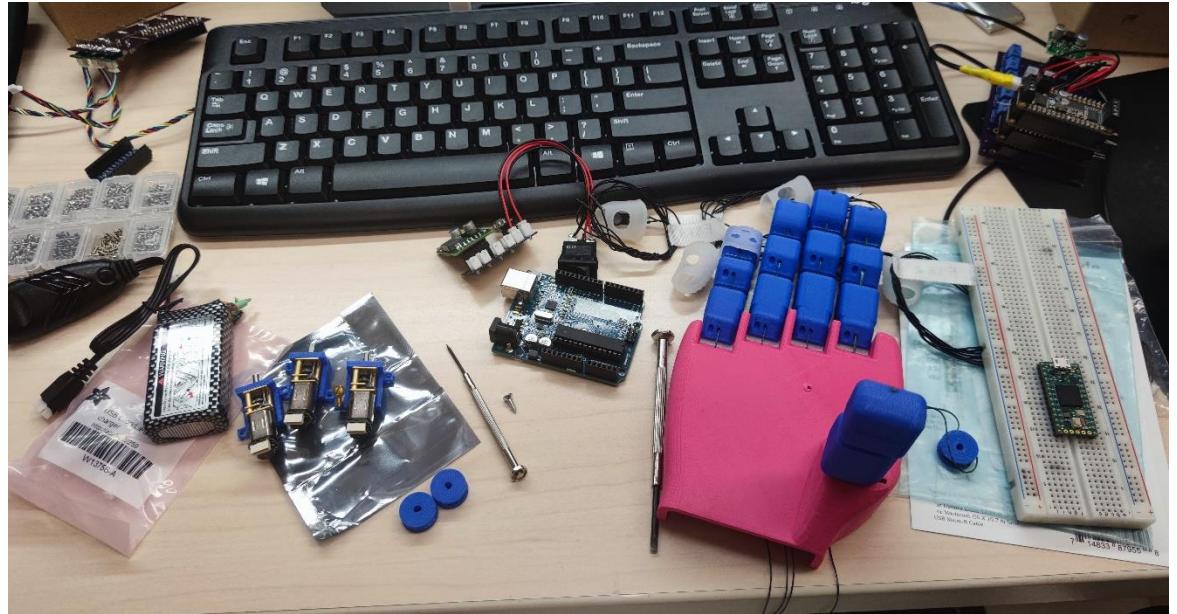


The project proposal was to develop a prosthetic hand that would be accessible with an affordable implantation cost. Functional with an EMG-controlled system, and scalable to customize EMG processing to each user.

What I have done for this Project

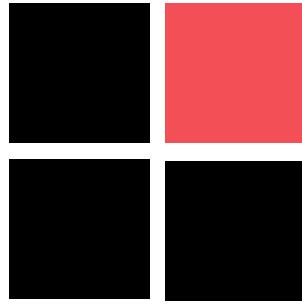


Test on the DC motor driver

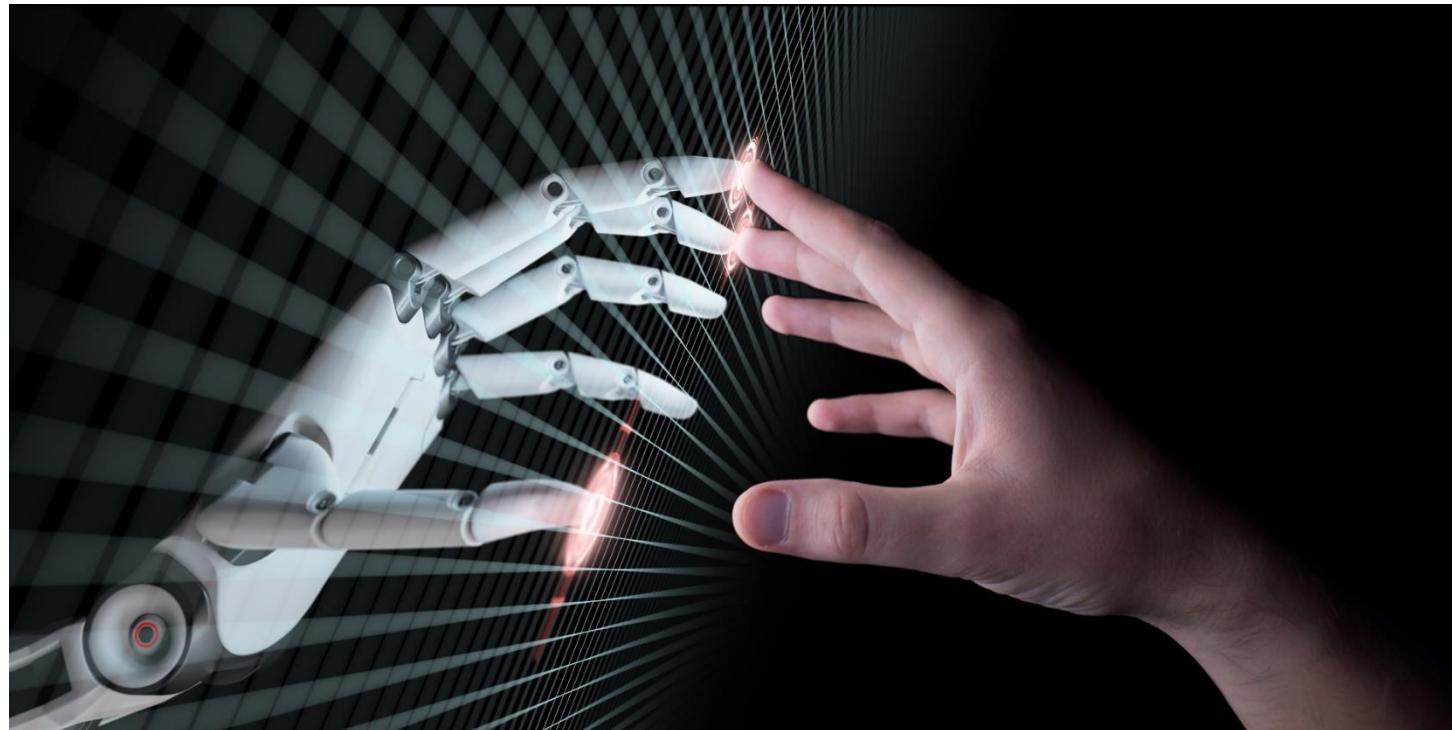


Some assemblage of the
prosthetic hand

Further Development of the Prosthetic Hand



Digital Twin/Prosthetic Mimic

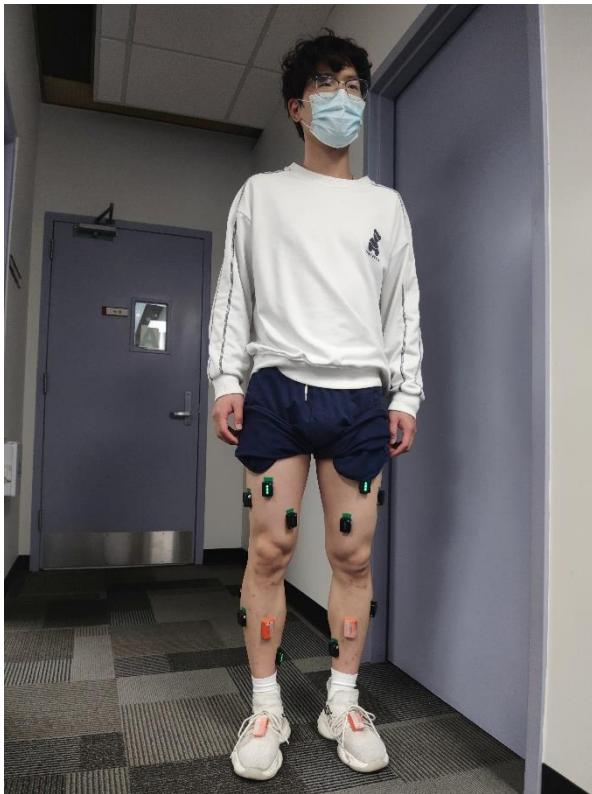


After finishing the Assembly and debugging of the hand, what we can do with the Prosthetic Hand is creating its digital twin on the computer. Connecting Arduino with Unity, we can assign data to the model in unity and drive the model. Then we can get the digital twin of the hand in the computer.



**Participate in Lab
Experiments**

Volunteering for lab experiments



Walking motion capture test for Xianwen and Richa, 26th August and 22nd September



EEG Experiment for Richa and Sarthak,
11th August



Motion Capture test for Maryam, 19th August



Acknowledgement

Acknowledgement



Dr. Rouhani

Thank you for giving me this chance to have a summer internship in University of Alberta.

Karla and Richa

Thanks Karla for keeping track of my progress and pushing me forward. Thanks Richa for your help.

Sarthak and Hugo

Thanks Sarthak and Hugo for your help during the project. Both you really encouraged me and let me feel at home. So nice to meet you this summer:)

All people in the lab

Xianwen, Maryam, Steven, Ramin and all the people in the lab. Thank you all for giving me an unforgettable summer. So nice of you all!

THANK YOU