2022

Summer Internship

Work Report

Reporter: Ruiqi Mao

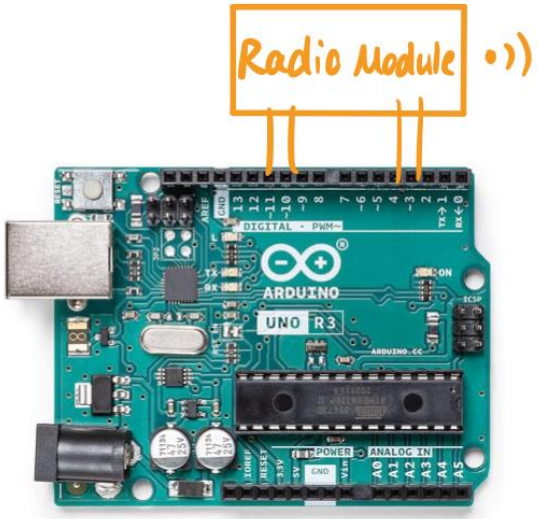
9/28/2022

CONTENTS

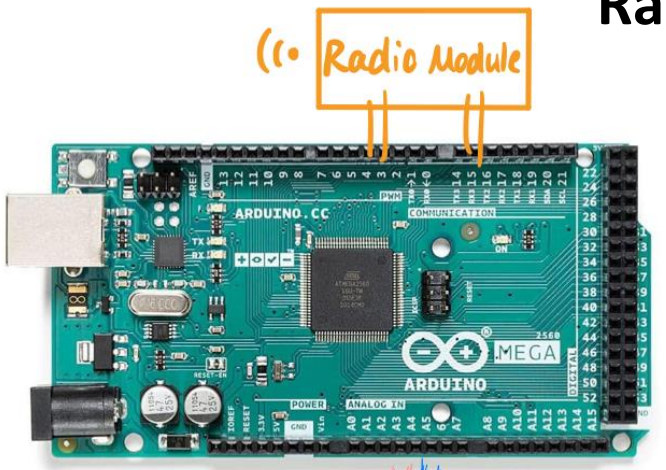
Motion Sensor Glove

Project Overview

Receiver
(Arduino UNO)

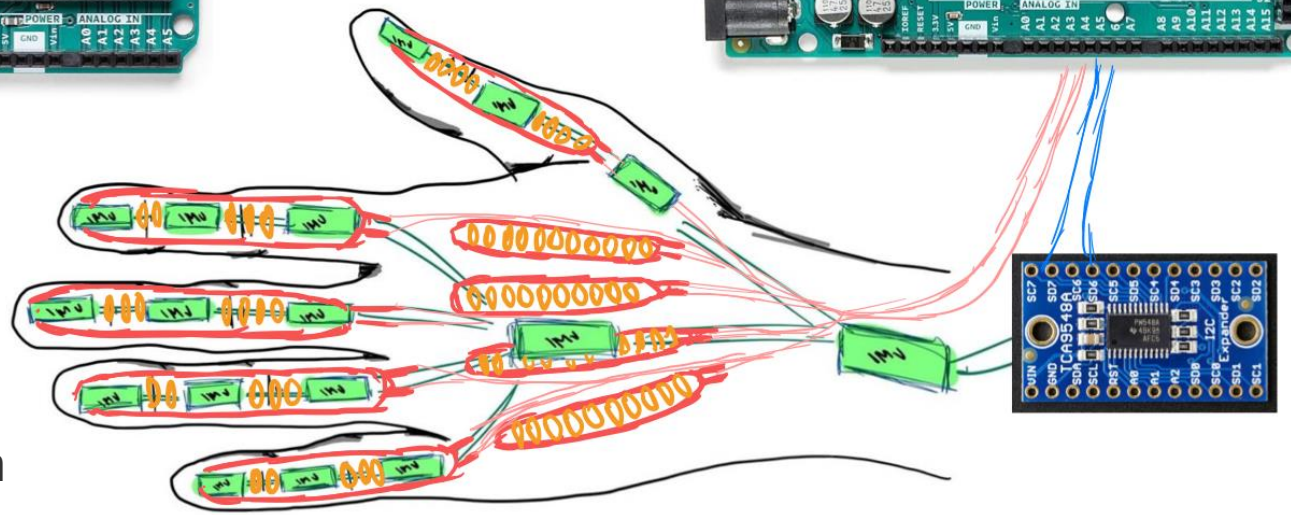


Radio Module



Transmitter
(Arduino mega2560)

IMU
(LSM9DS1, which has 9 degrees of freedom)



Multiplex (TCA9548A)

Flex Sensor

Workflow of the Glove

Workflow

1 Read Data

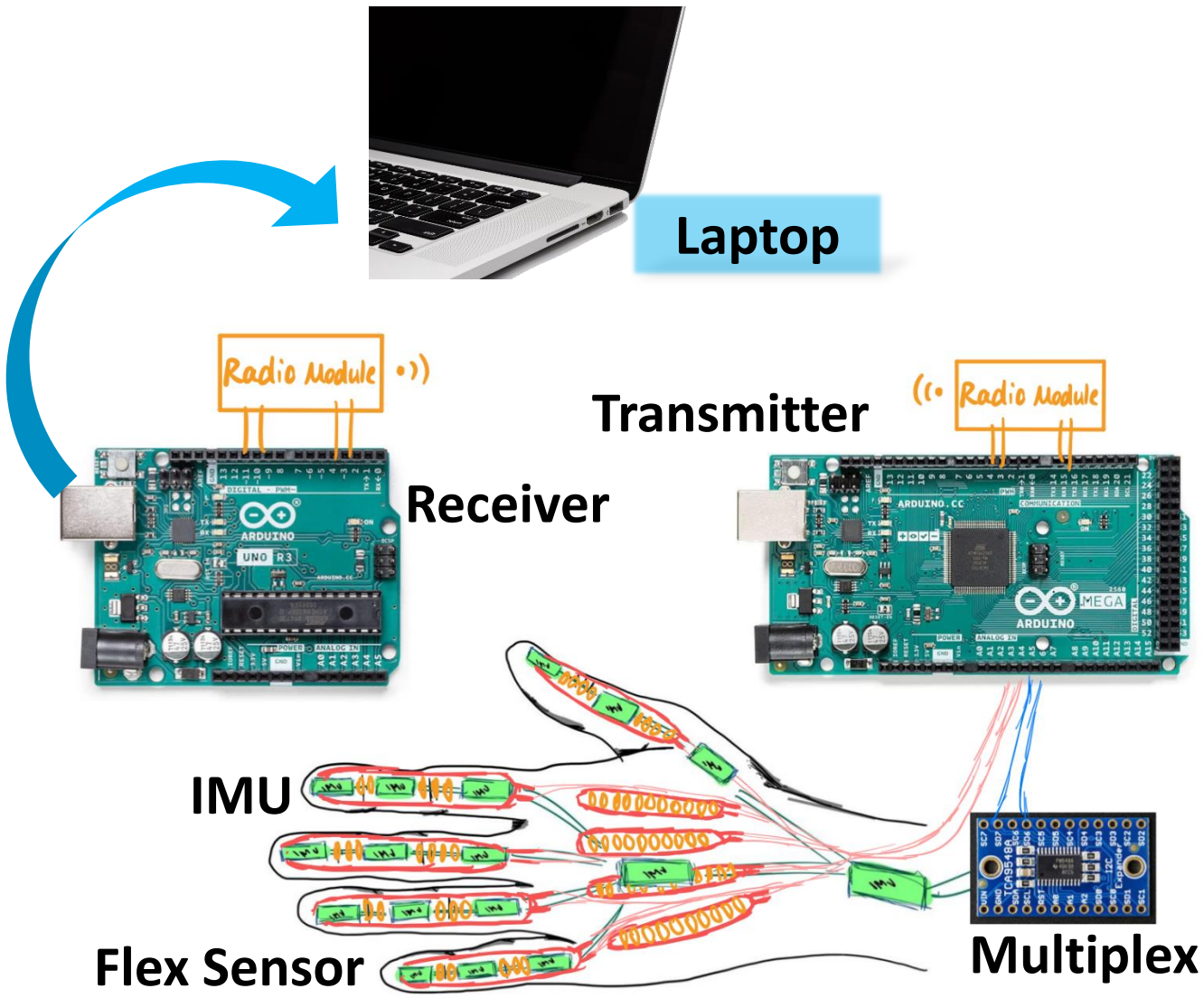
The data from the 16 IMUs and 9 Angle Sensors will be read by the Arduino mega2560 for 7 times per second.

2 Data Processing

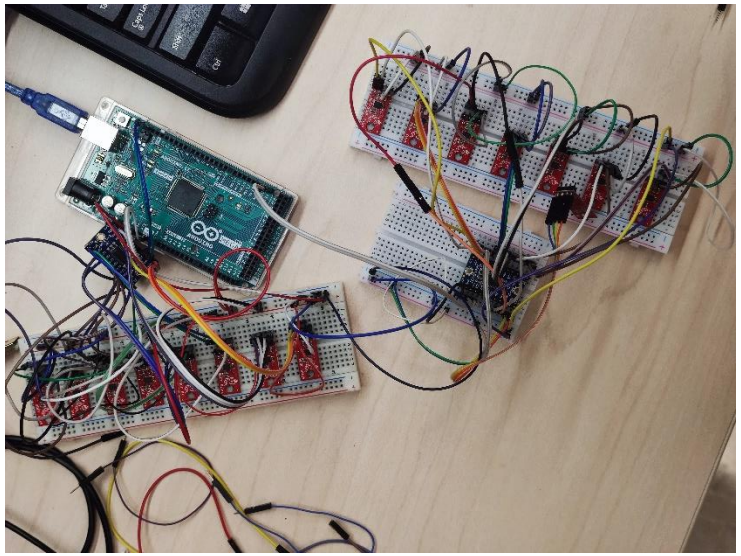
The data read from the sensors will be processed by the Arduino mega2560. Then we can get the acceleration, angular rate, magnetic field, direction data(from IMU) and angle data(from flex sensors)

3 Transmission

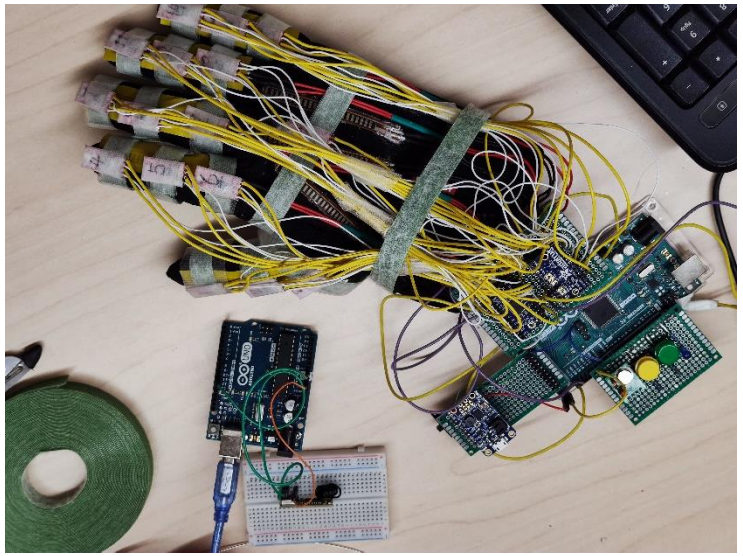
After the Arduino mega2560 get the data, it will work as a transmitter and send the data to the receiver(Arduino Uno), which is connected to the laptop. Then we can see the data from the laptop and do further analysis.



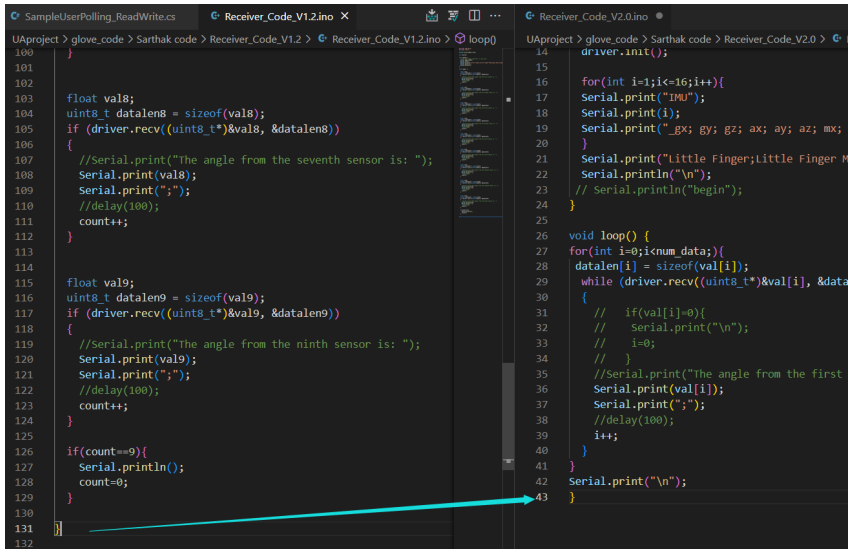
My Contribution to this Project



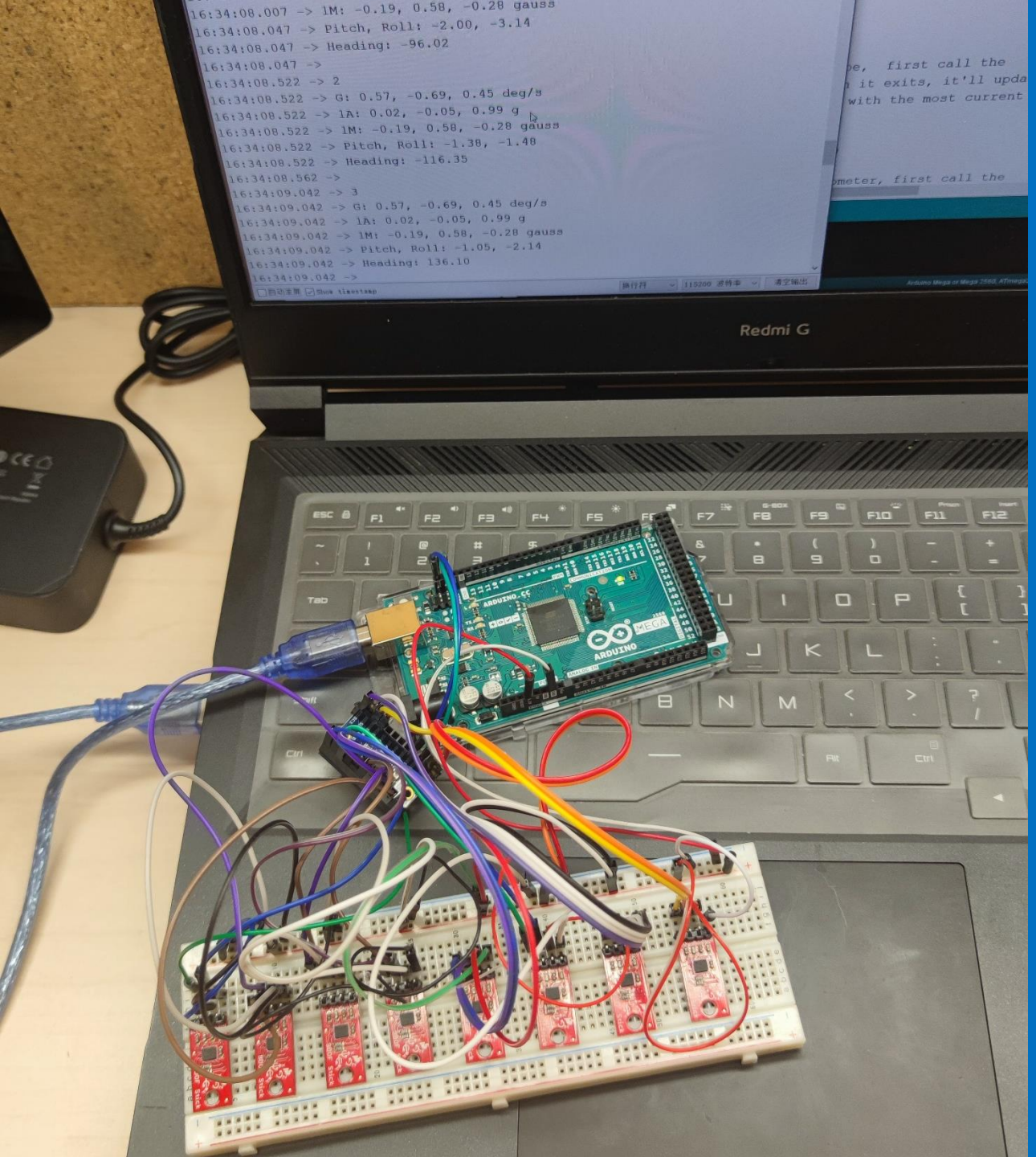
IMU module test



Overall assembly and debugging



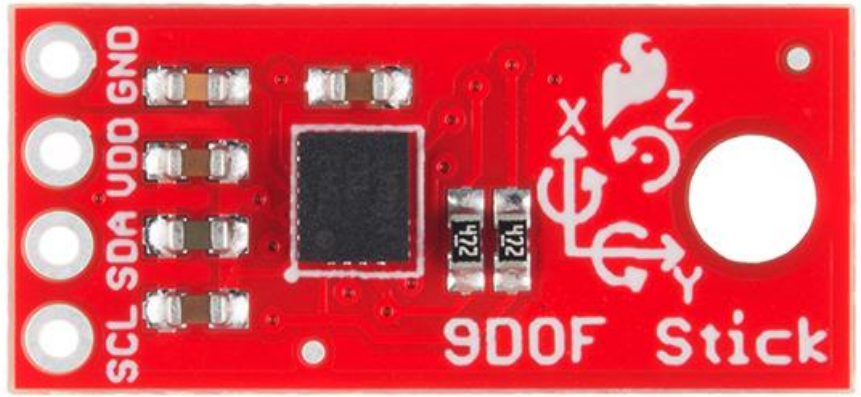
Code improvement



1.1 IMU Module Test

1.1 IMU Module Test

Hardware Overview



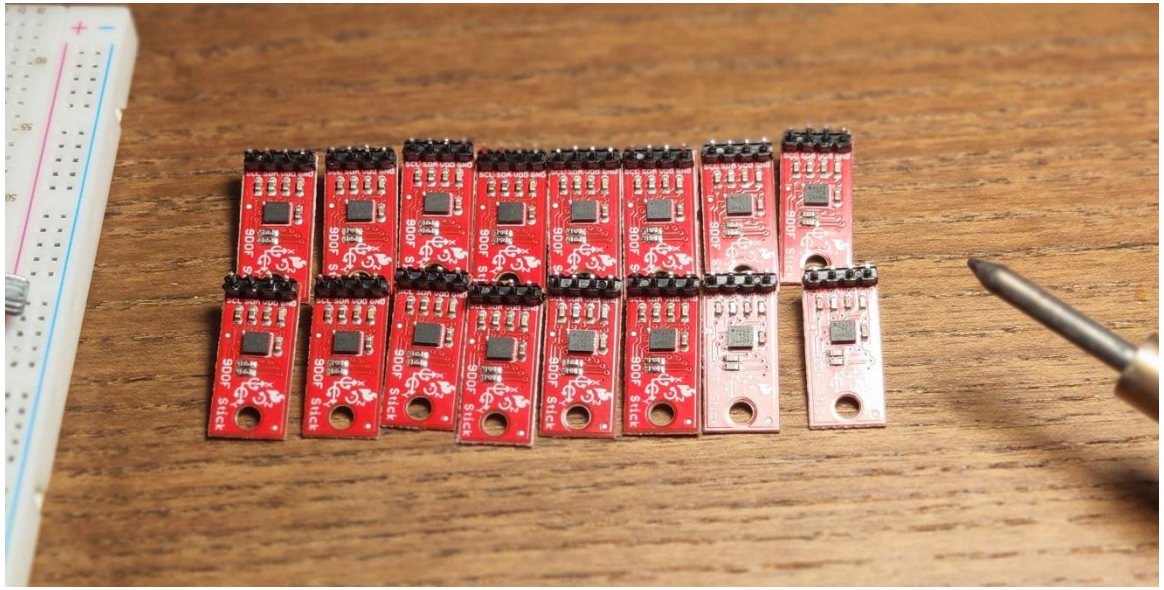
9DoF Sensor Stick (LSM9DS1)

LSM9DS1 Details:

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

1.1 IMU Module Test

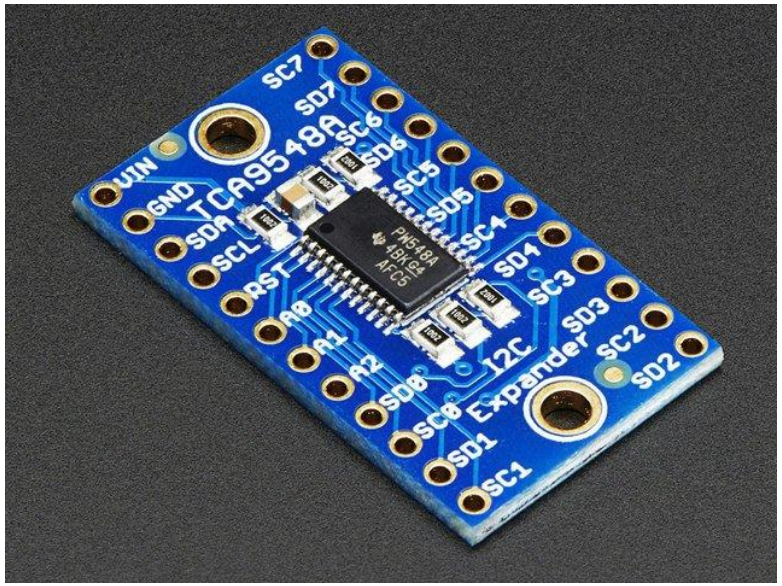
Problem: How to get the data from the 16 Sensors?



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



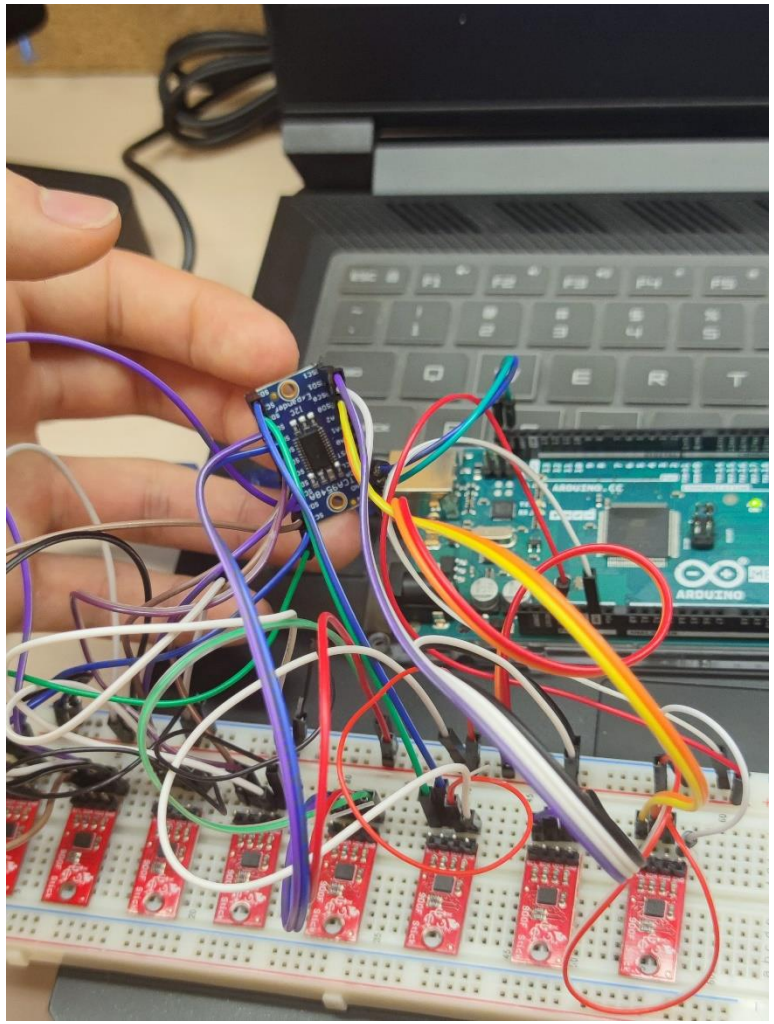
The TCA9548A multiplexer is interesting in that it has an I2C address (0x70 by default) - and we basically send it a command to tell it which I2C multiplexed output we want to talk to, then we can address the board we want to address.

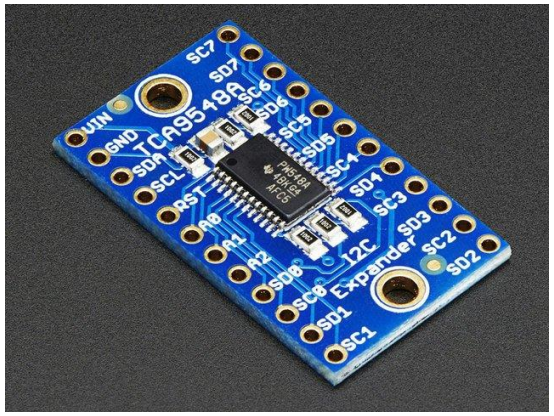
```
#define TCAADDR  0x70
#define TCAADDR2 0x71   //enable the A0 Pin on the second TCA
#define num 16     //the number of the IMUs
void tcaselect (uint8_t i) {

  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.

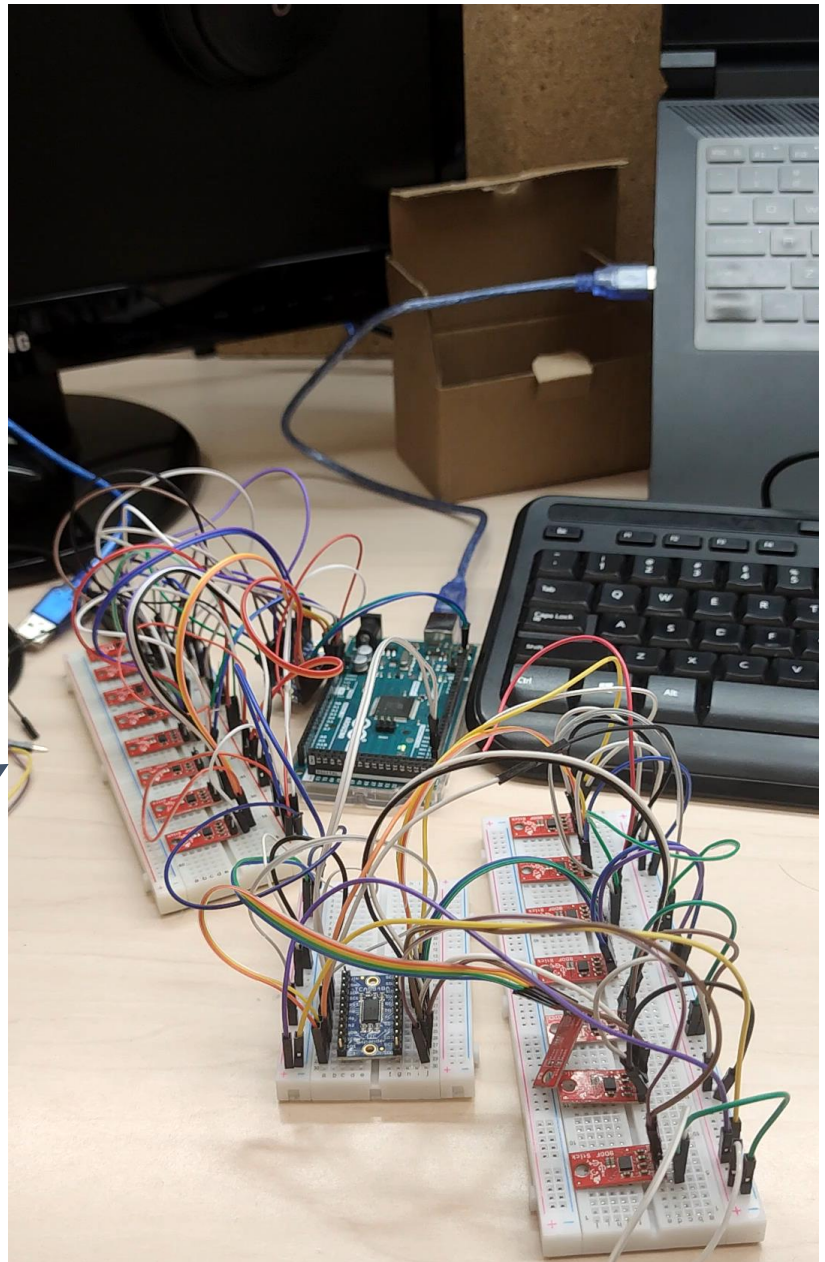
1.1 IMU Module Test

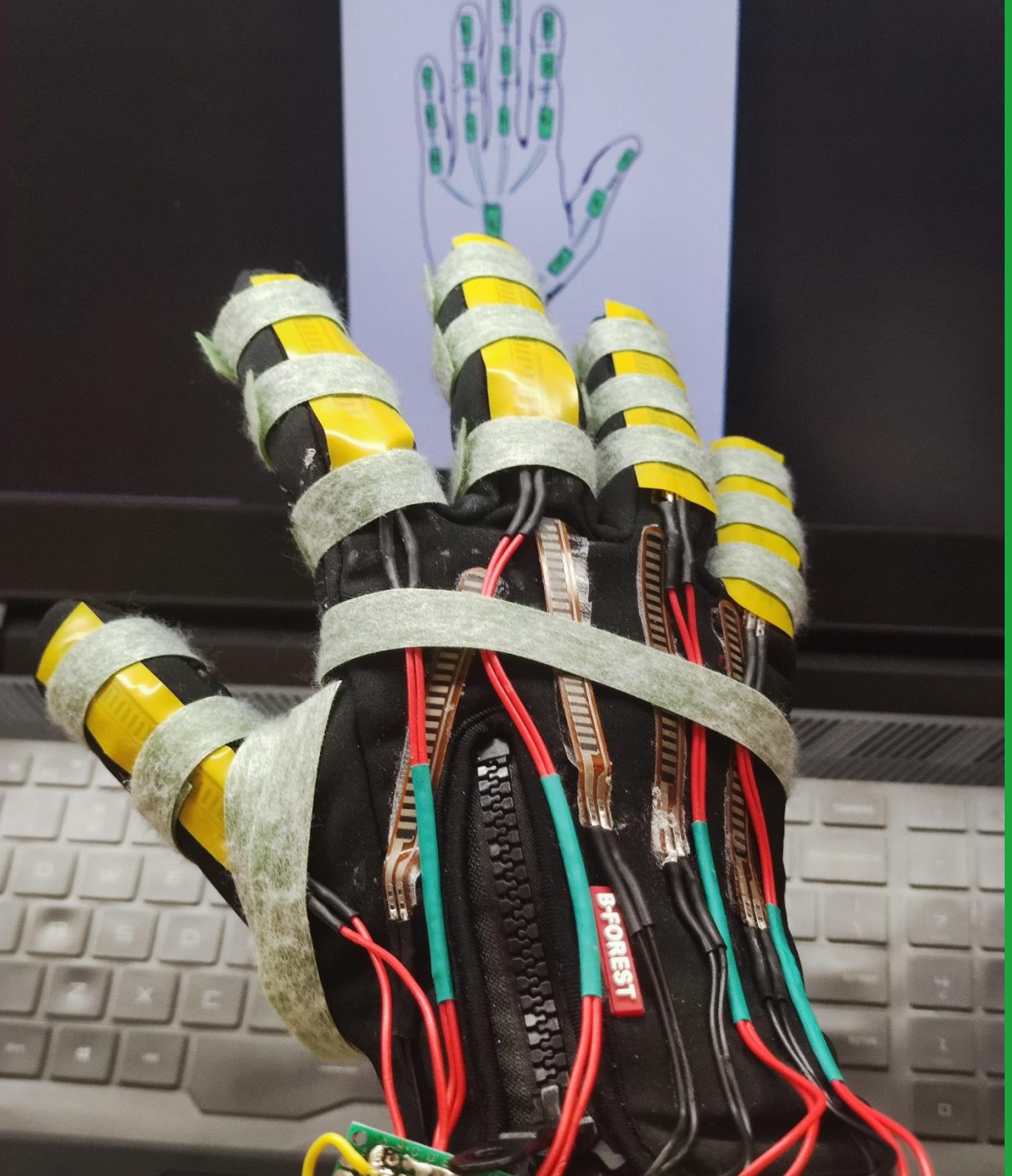
Test Result





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

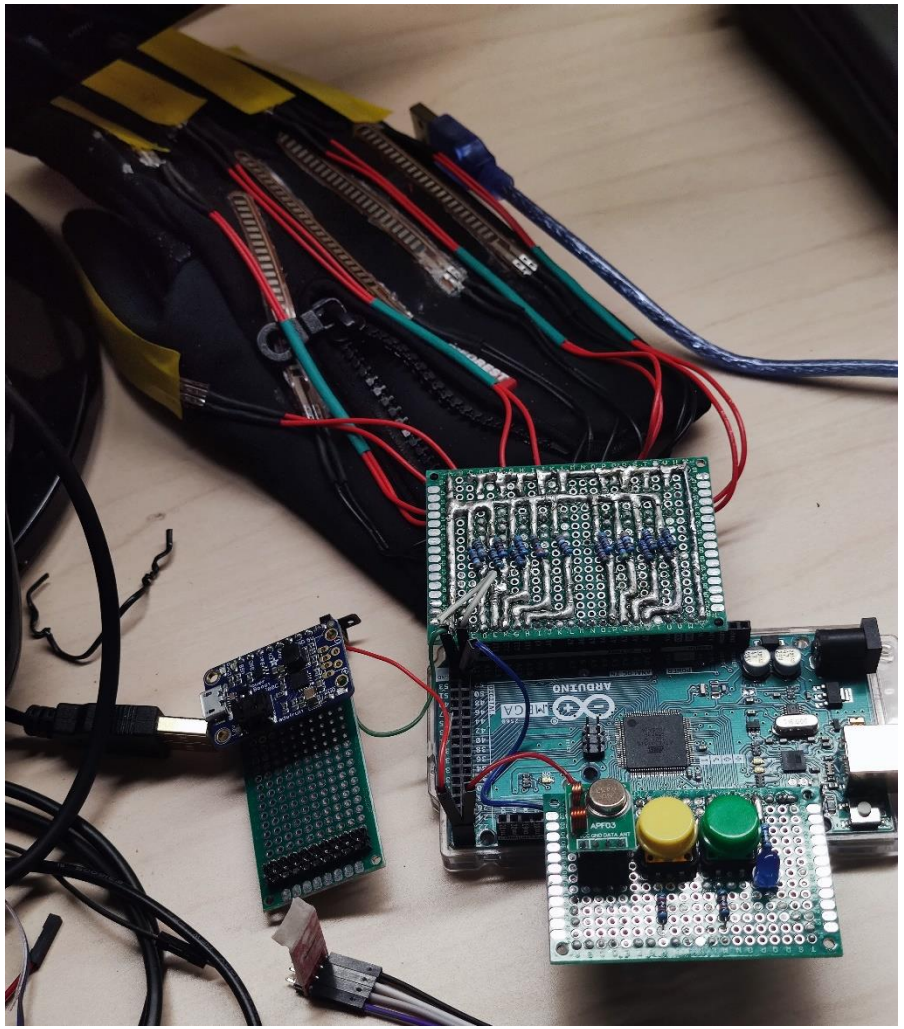




1.2 Overall Assembly and Debugging

1.2 Overall Assembly and Debugging

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

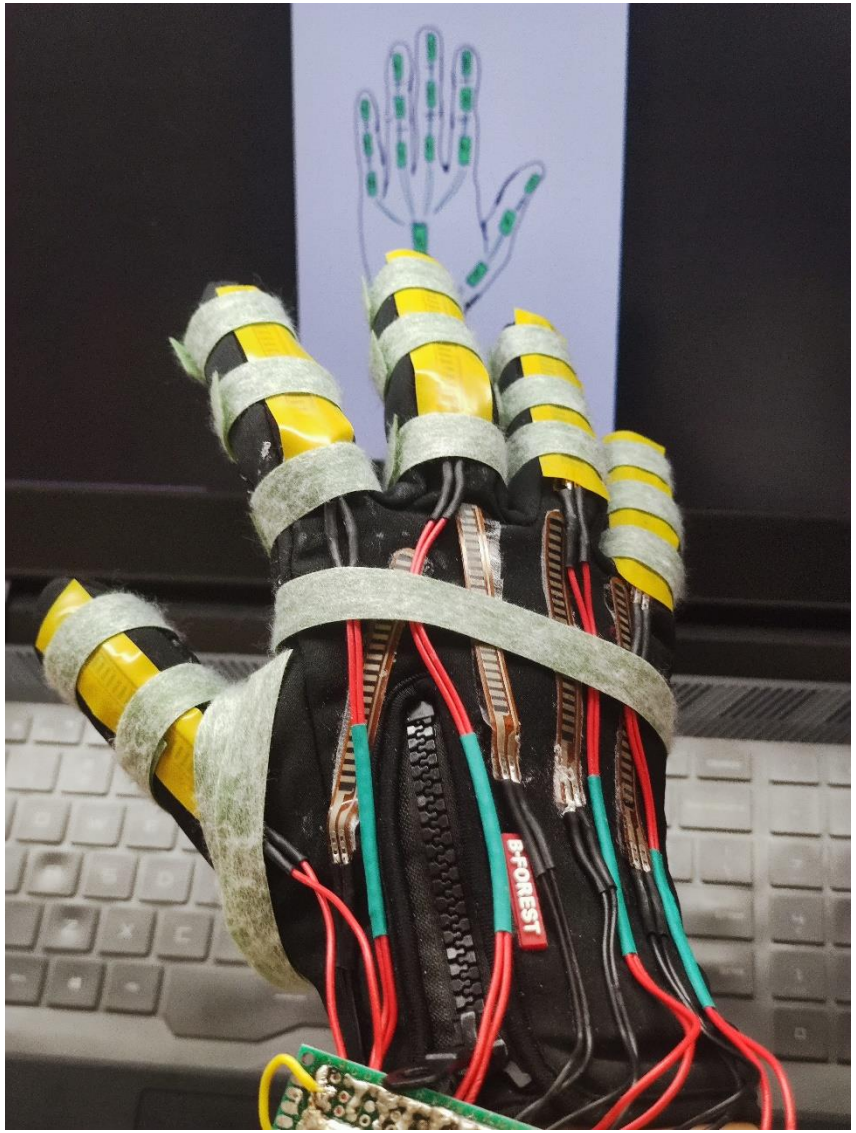


When I got the glove, it looked like in the left picture. The glove already had several parts which had some impact on the movement of user's hand. That means I should be careful when I add the IMU module to the glove so that I can let the restriction on movement reduce to minimum as possible.

1.2 Overall Assembly and Debugging

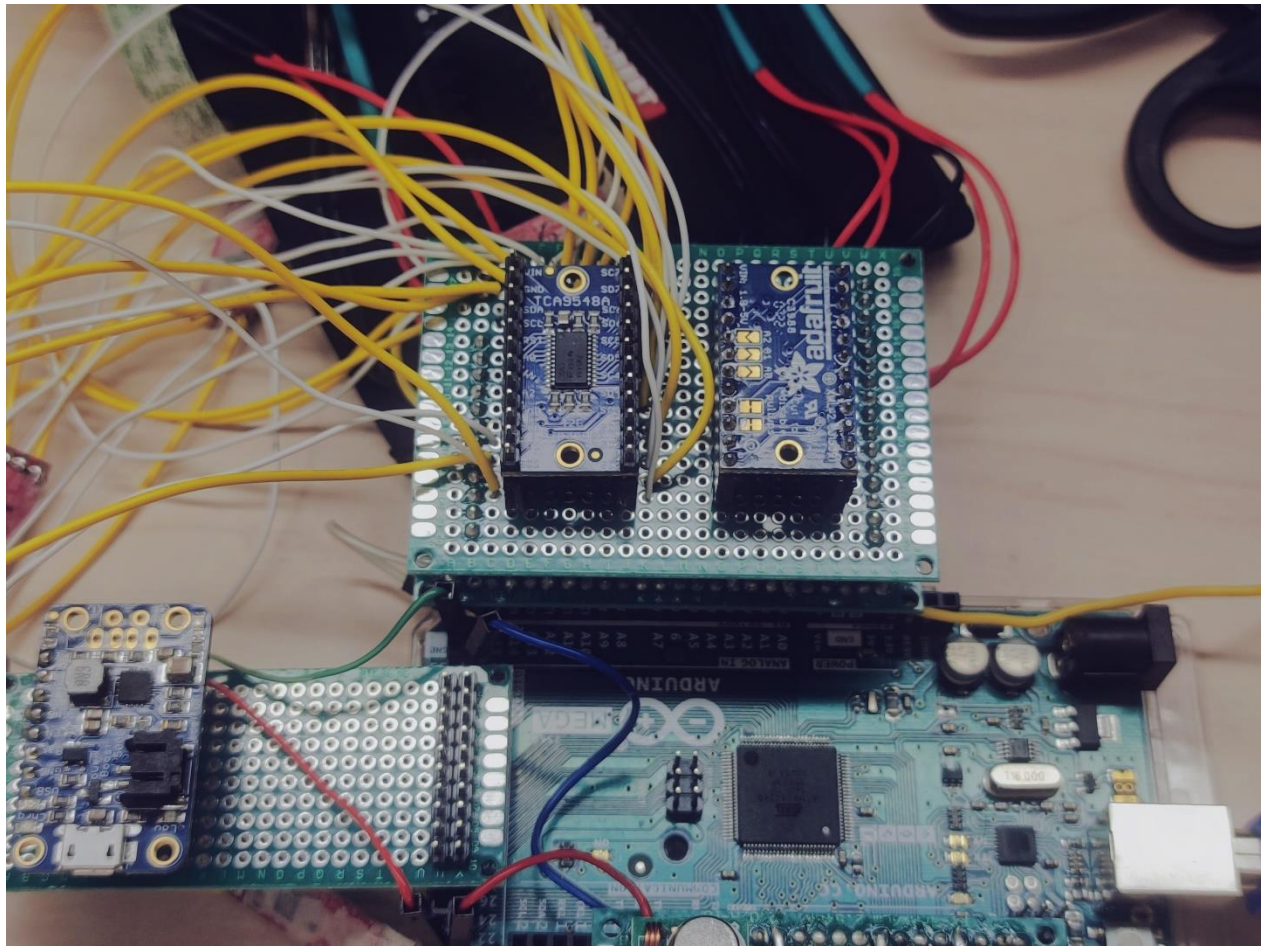
Solution1: Put IMU on Velcro

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

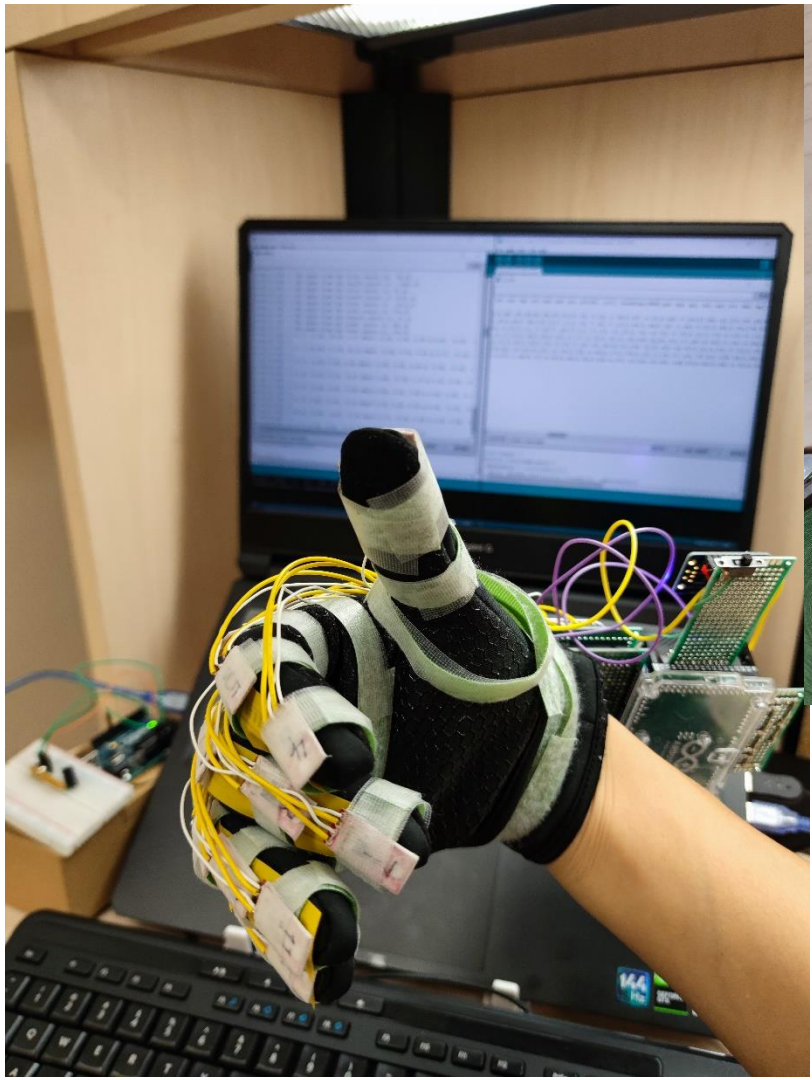
Solution2: Mount the multiplex board on the existing circuit board

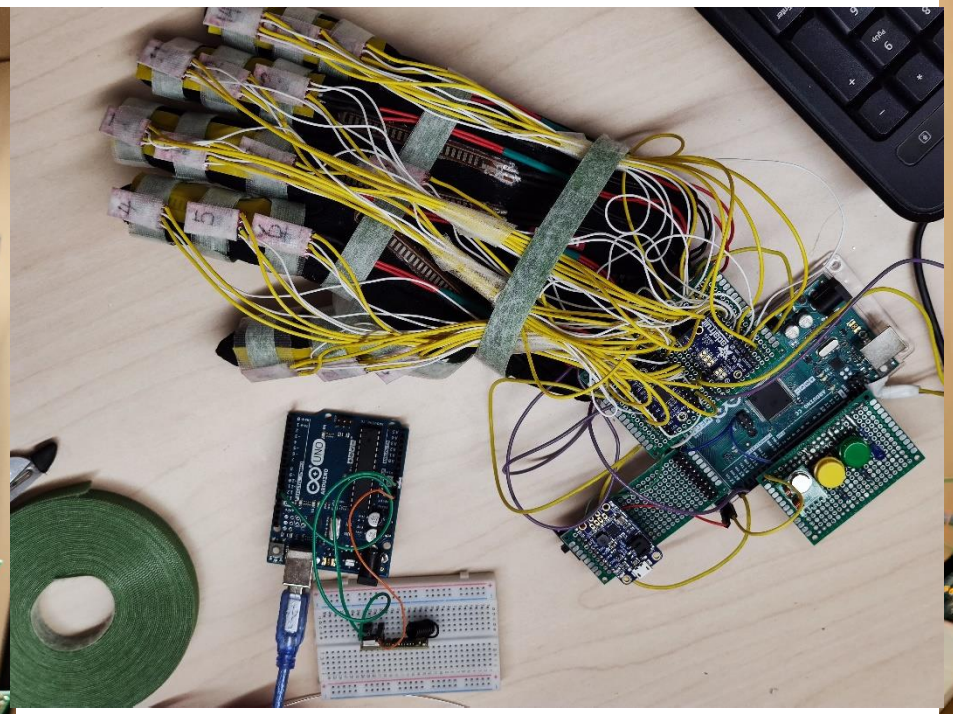


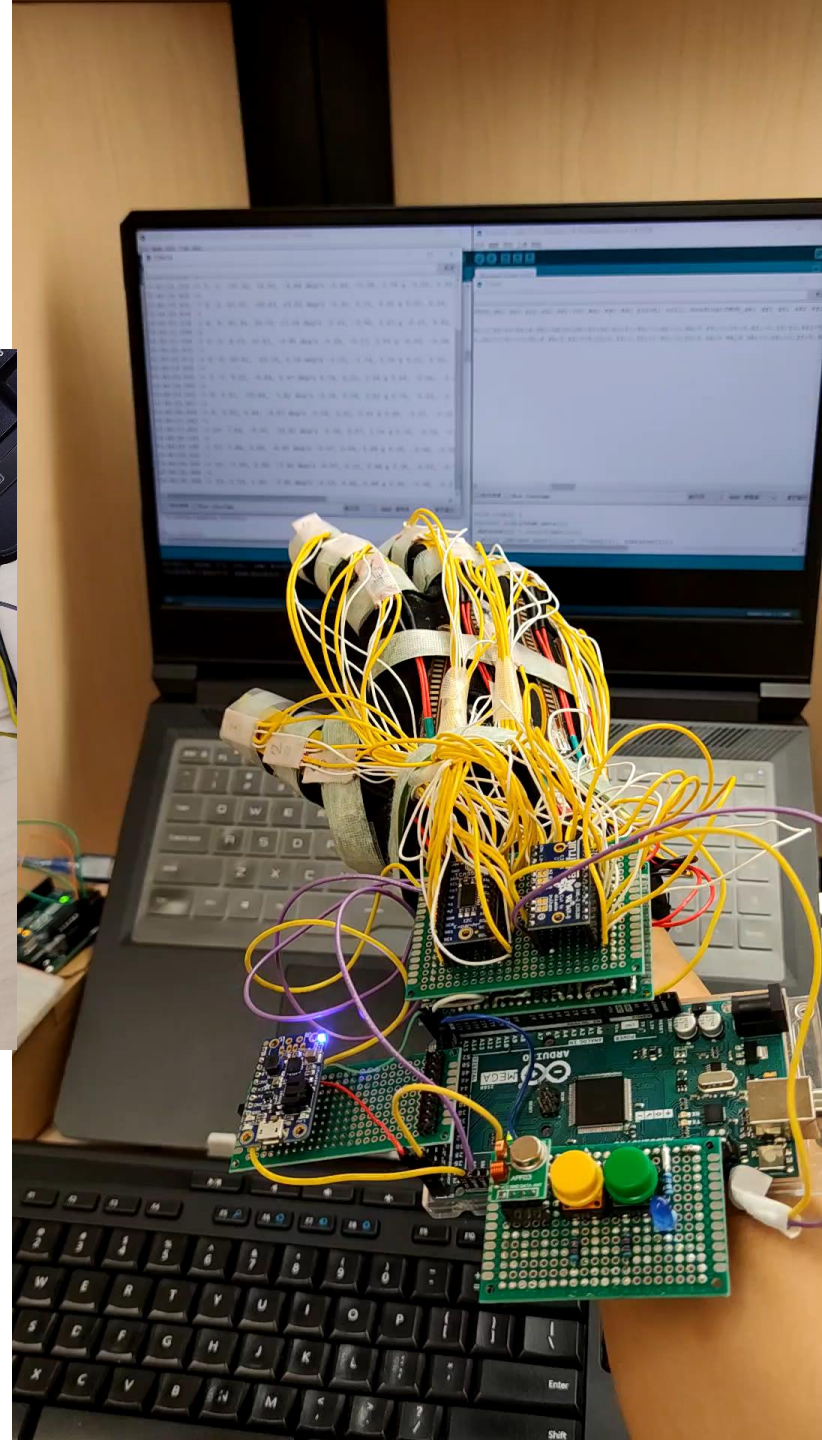
To avoid adding more restriction on movement, I decided to put the multiplex board on the existing board. I soldered some pins on the two boards so they can put together, easy for disassembly as well.

1.2 Overall Assembly and Debugging

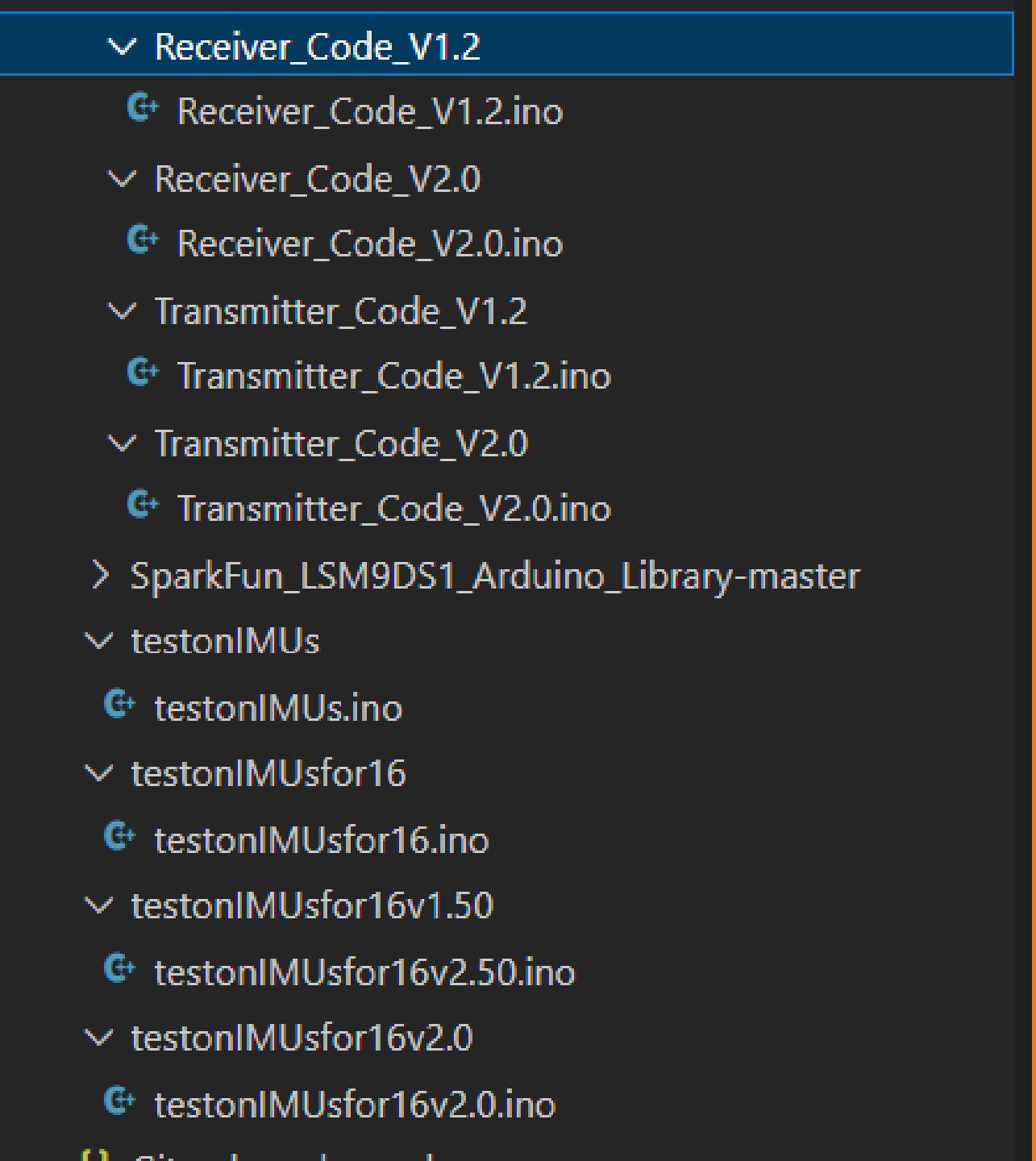
Test Result







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



1.3 Code Improvement

1.3 Code Improvement

Improvement1: Code shortening

Before

```
void loop() {

  float val1;
  uint8_t datalen1 = sizeof(val1);
  if (driver.recv((uint8_t*)&val1, &datalen1))
  {
    //Serial.print("The angle from the first sensor is: ");
    Serial.print(val1);
    Serial.print(";");
    //delay(100);
    count++;
  }


  float val2;
  uint8_t datalen2 = sizeof(val2);
  if (driver.recv((uint8_t*)&val2, &datalen2))
  {
    //Serial.print("The angle from the second sensor is: ");
    Serial.print(val2);
    Serial.print(";");
    //delay(100);
    count++;
  }


  float val3;
  uint8_t datalen3 = sizeof(val3);
  if (driver.recv((uint8_t*)&val3, &datalen3))
  {
```

```
void loop() {
for(int i=0;i<num_data;){
 datalen[i] = sizeof(val[i]);
  while (driver.recv((uint8_t*)&val[i], &datalen[i]))
  {
    Serial.print(val[i]);
    Serial.print(";");
    //delay(100);
    i++;
  }
}
Serial.print("\n");
}
```

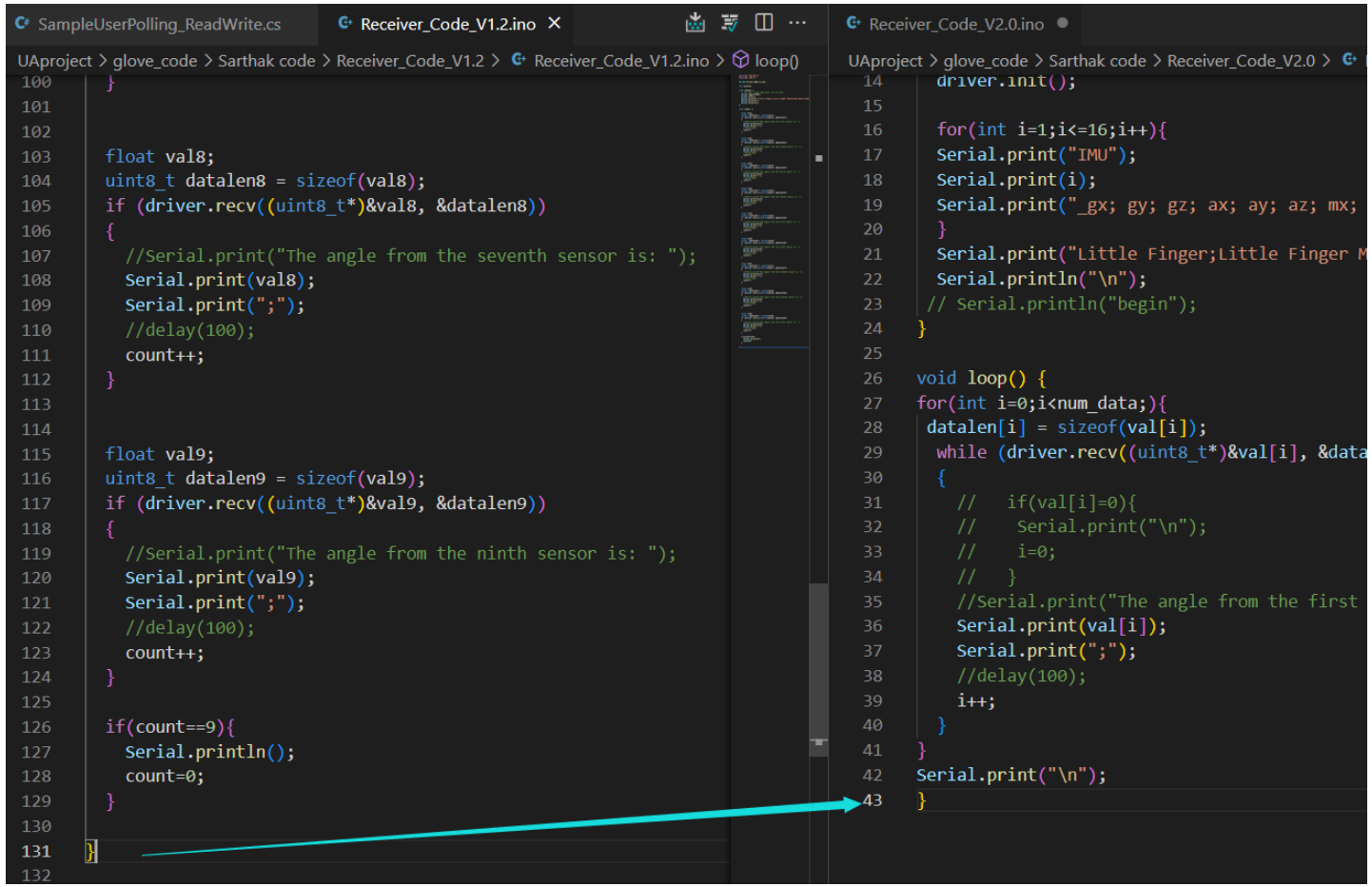
By using loops and arrays, I shortened the code significantly.

After

```
void loop() {
int t=0;
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);
}
}
```

1.3 Code Improvement

Result



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

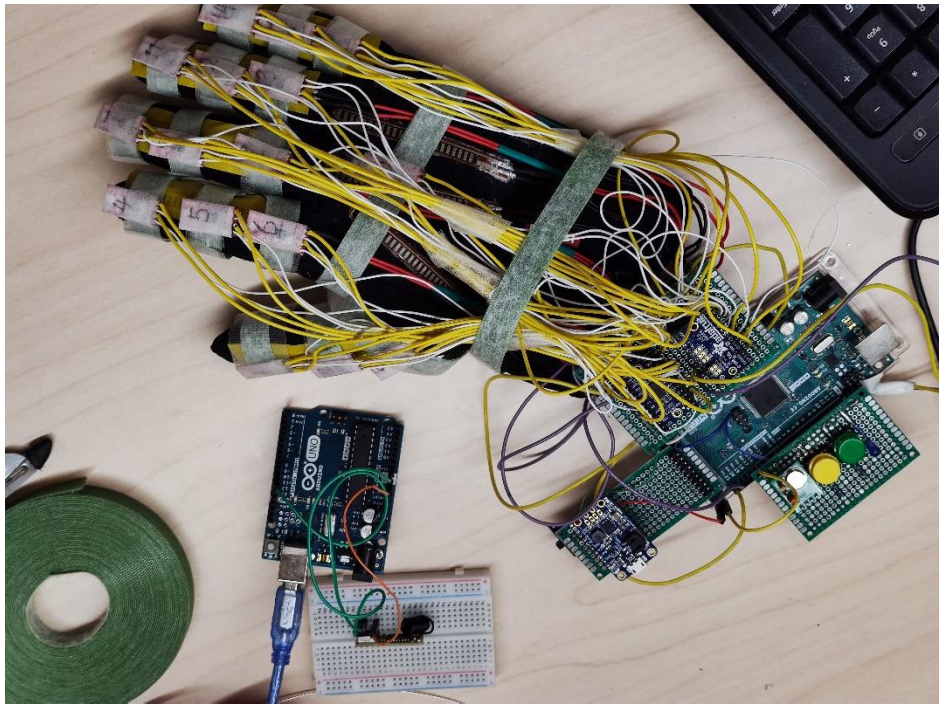
Result

```
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 eigth 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
```

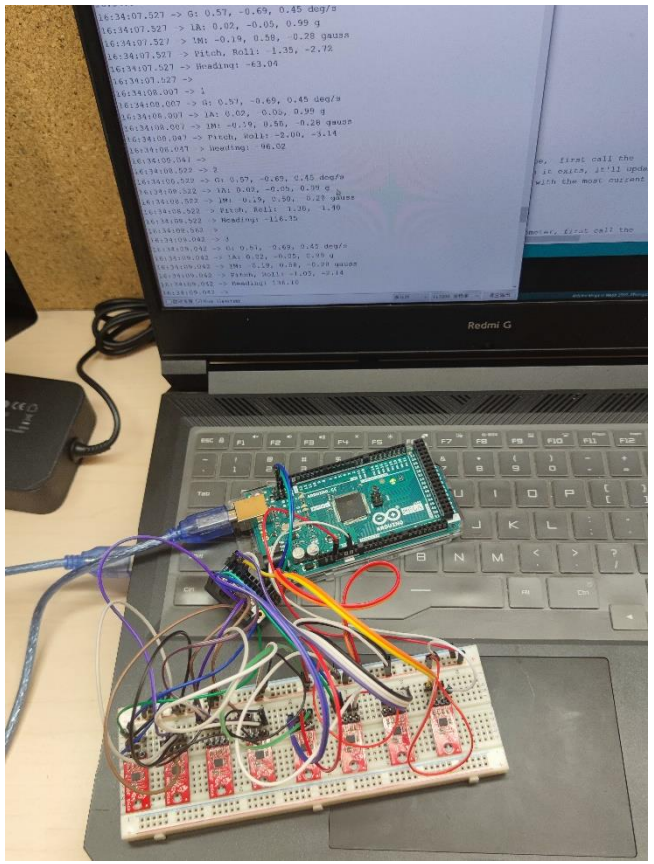
The data printing frequency is improved from 0.1Hz to 7.5Hz（to be improved）

1.4 Further Improvement of the Glove

Structure Simplification

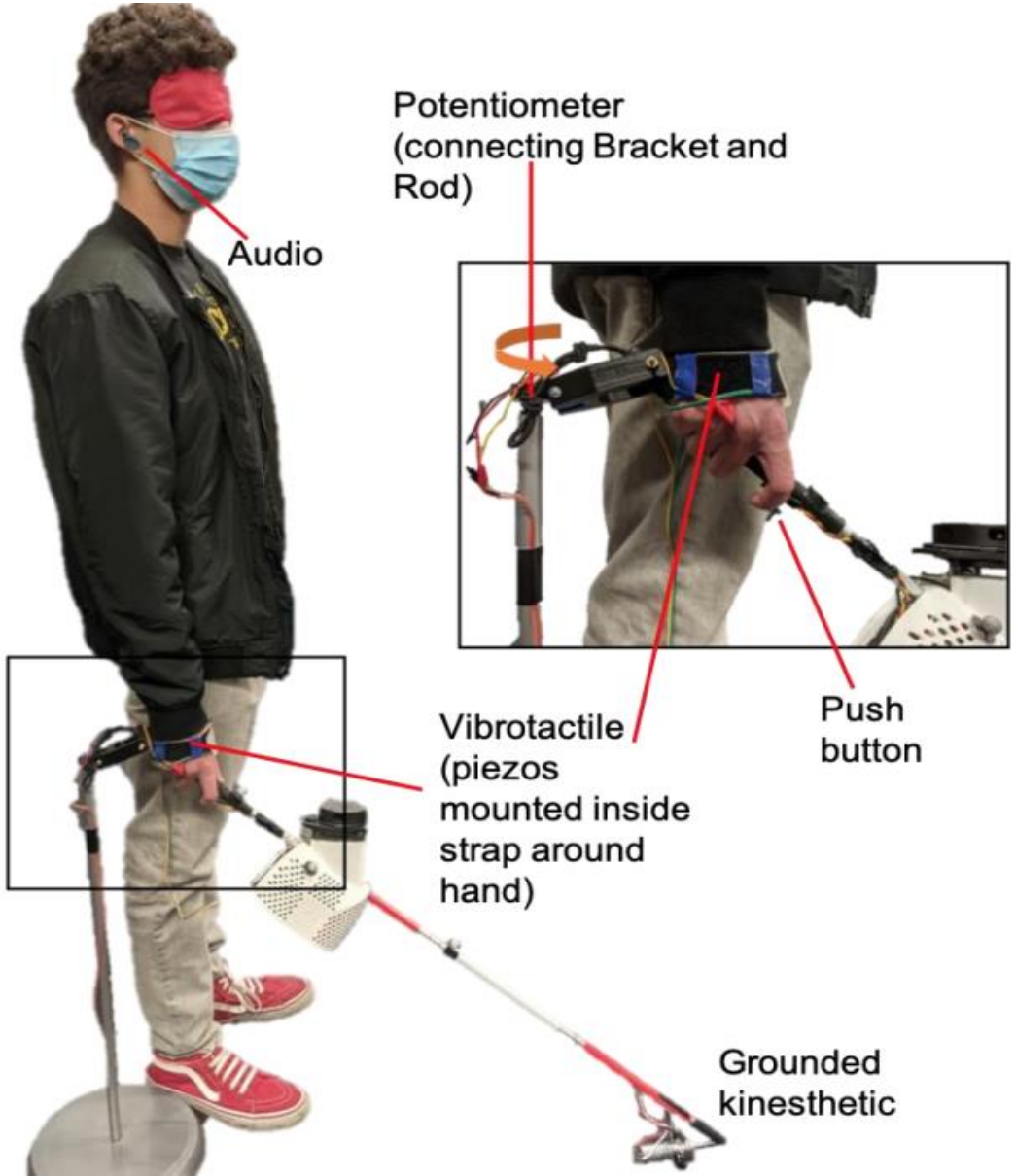


Improve Data Printing Speed



Augmented Cane

Augmented Cane

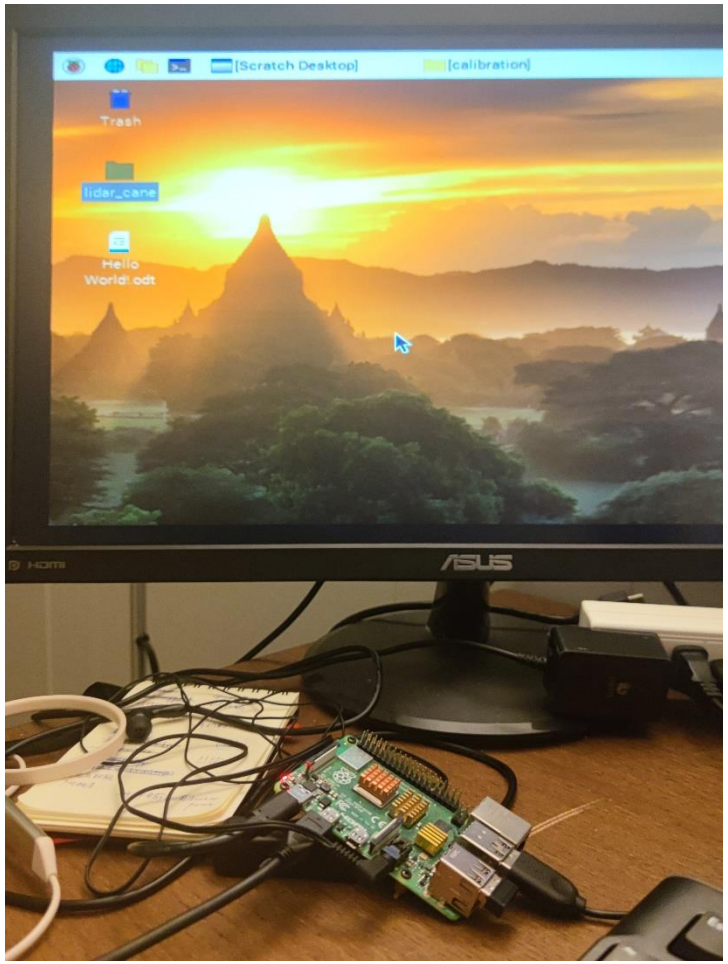


Augmented Cane is a white cane with a comprehensive set of sensors and an intuitive feedback method to steer the user, which addresses navigation challenges and improves mobility for people with impaired vision. It is a cane developed by the students in Stanford University.

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								

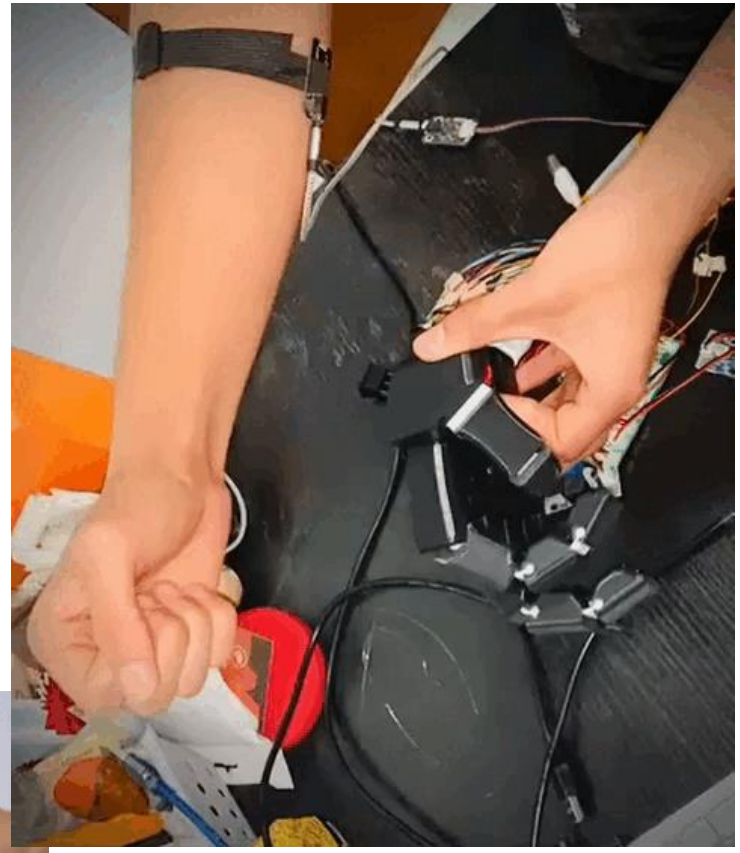
Made the bill of materials for the project



Learned to use Raspberry Pi

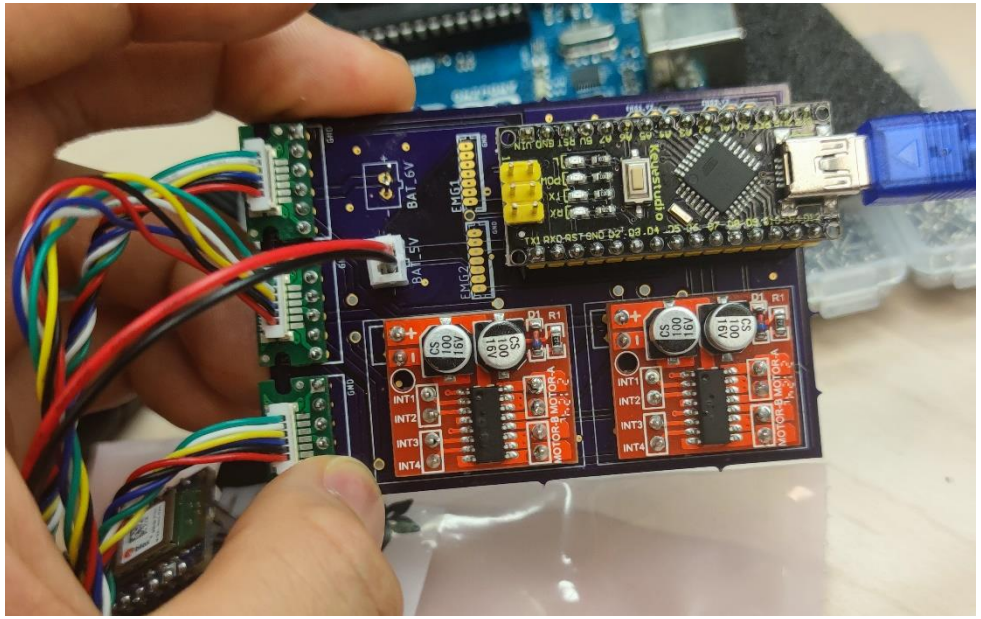
Prosthetic Hand

Prosthetic Hand

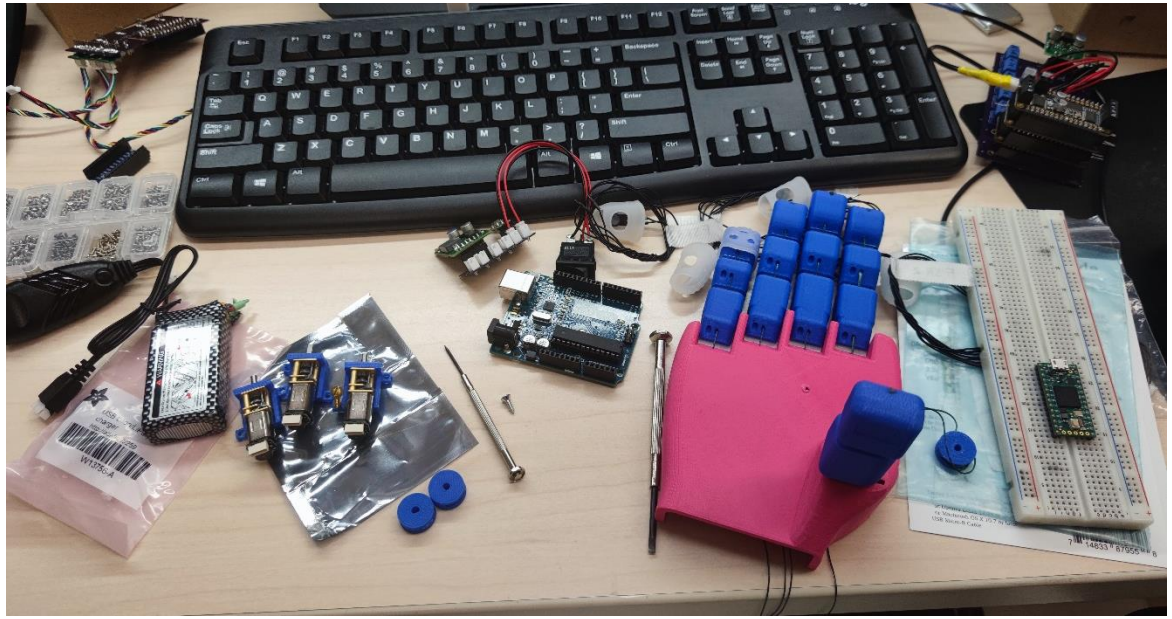


The project proposal was to develop a prosthetic hand that would be accessible with an affordable implantation cost. Functional with an EMG-controlled system, and scalable to customize EMG processing to each user.

What I have done for this Project



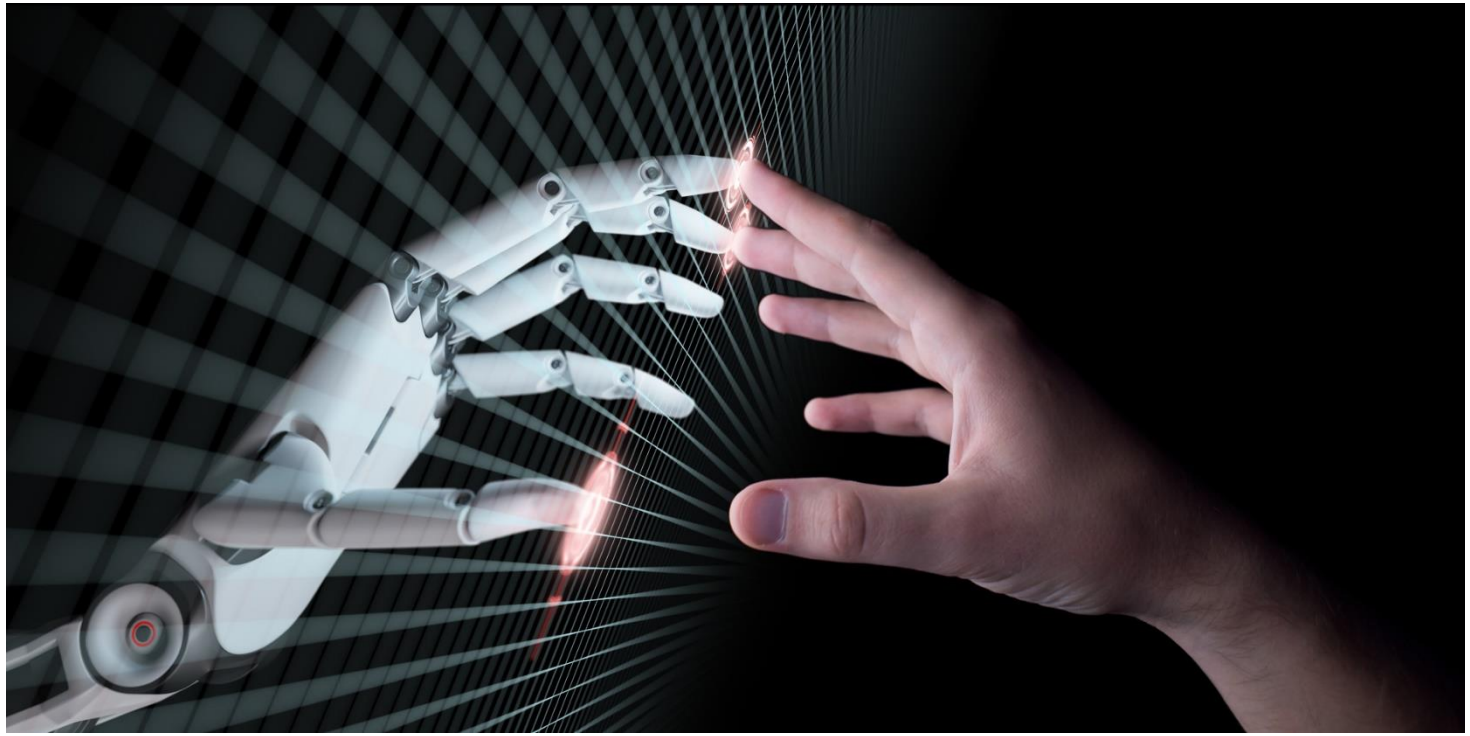
Test on the DC motor driver



Some assemblage of the prosthetic hand

Further Development of the Prosthetic Hand

Digital Twin/Prosthetic Mimic

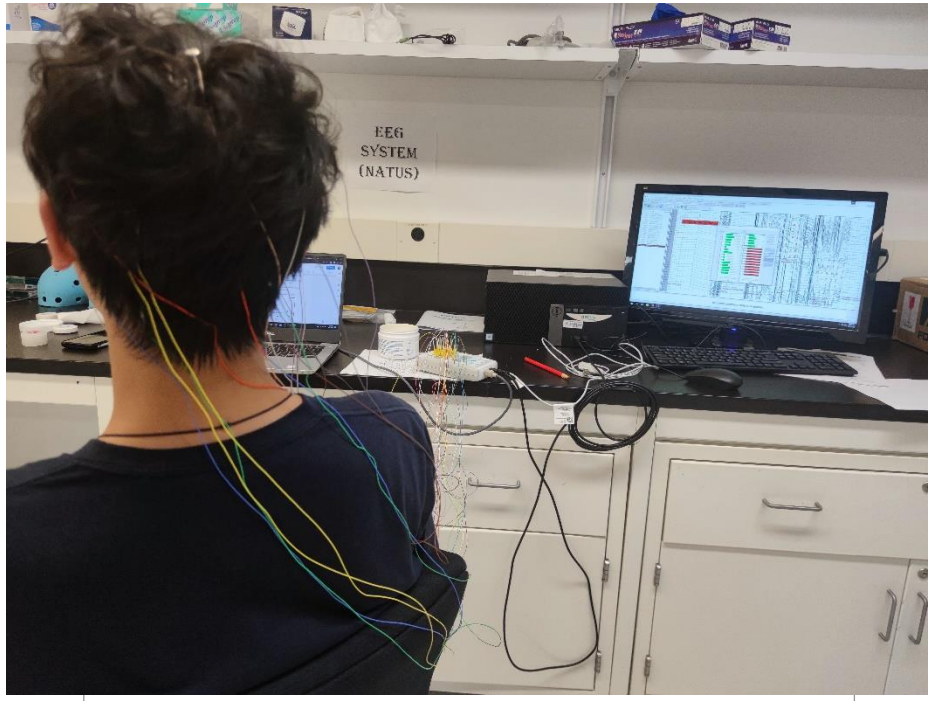


After finishing the Assembly and debugging of the hand, what we can do with the Prosthetic Hand is creating its digital twin on the computer. Connecting Arduino with Unity, we can assign data to the model in unity and drive the model. Then we can get the digital twin of the hand in the computer.

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