

Gest Key

Motivation

The motivation of building this project is to help people who are blind , deaf or can't speak so with the help of our device the recognised characters are translated from the gestures given by a human that can be understood from Arduino with accelerometer , by using pygarl library in python.

Pygarl is basically python gesture analysis and recognition library.

Introduction

- ▶ The Specially-Abled Community has a lot of issues regarding communication, so, in order to facilitate communication between them, we have designed a Gesture Keyboard.
- ▶ The project basically, is a device which will use Machine Learning Algorithms to convert hand motion gestures into characters which will be printed on the screen.
- ▶ The Programming Languages mostly used are C (for Arduino) and Python to create the Machine Learning Algorithm with SVM.

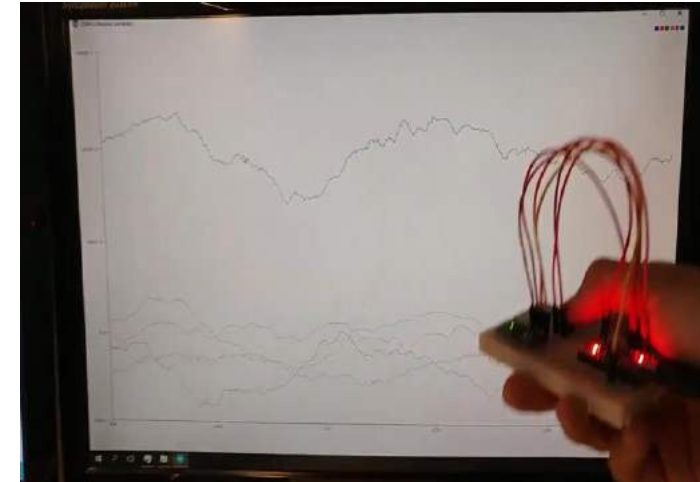
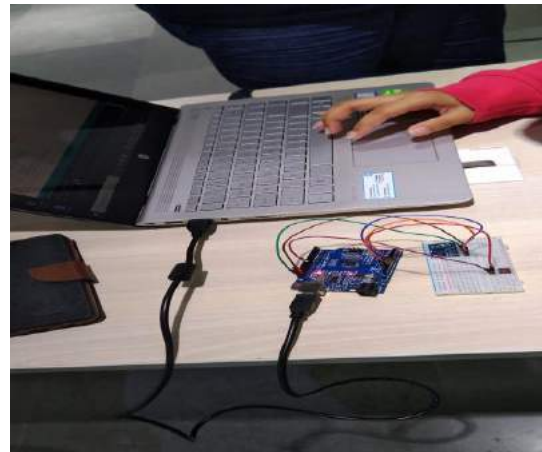
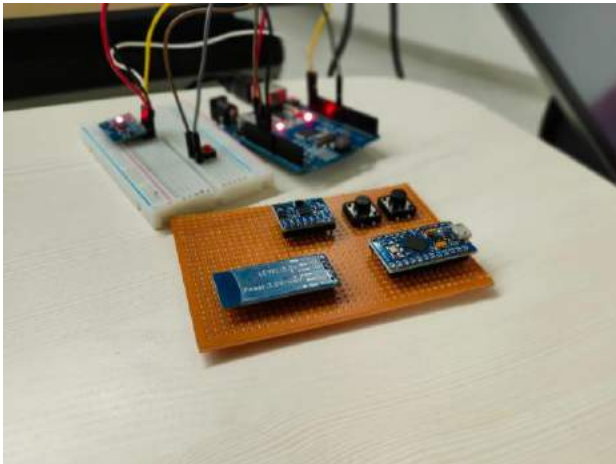
Basic idea

We have made a gesture keyboard using Arduino from which it will input some characters from device when we move it in air to draw some character . The general idea behind this is that the user would press or click a button and hold it down while dragging the pointer across the character before releasing the button to complete the it.

Dataset Used

The Dataset used for the Gesture Keyboard is created by us ourselves.

The content in our dataset for every gesture is getting saved in sample and temp.txt files.



Methodology / Model Used

- ▶ The components used for the project are:- an Arduino, an MPU 6050 Accelerometer, a breadboard, connecting wires .
- ▶ We are using the library Pygarl, which is written in Python itself and the library sklearn. These libraries are used to build gesture recognition systems.
- ▶ Algorithms of Machine Learning will be used to train the model so that it can identify the gesture accordingly .

Arduino Code

Project | Arduino 1.8.7

File Edit Sketch Tools Help

Upload

Project

```
#include <Wire.h>
// Pins used for I/O
int btnPin1 = 9;

// I2C address of the MPU-6050
const int MPU_addr=0x68;
// Variables that will store sensor data
int16_t AcX,AcY,AcZ,Tmp,GyX,GyY,GyZ;

// Status variables, used with buttons
int precBtn1 = HIGH;

void setup(){
  // Set the pin mode of the buttons using the internal pullup resistor
  pinMode(btnPin1, INPUT_PULLUP);

  // Start the communication with the MPU-6050 sensor
  Wire.begin();
  Wire.beginTransmission(MPU_addr);
  Wire.write(0x6B); // PWR_MGMT_1 register
  Wire.write(0);    // set to zero (wakes up the MPU-6050)
  Wire.endTransmission(true);

  // Start the serial communication
  Serial.begin(38400);
}

void loop(){
```

COM5

```
21:12:21.769 -> START 1784 3120 17240 -279 503 -465 END
21:12:21.769 -> START 1808 3168 17140 -123 398 -426 END
21:12:21.804 -> START 1760 3092 16896 17 328 -415 END
21:12:21.804 -> START 1868 3096 16948 191 240 -461 END
21:12:21.804 -> START 1988 3200 17236 164 236 -415 END
21:12:21.804 -> START 1960 3180 17248 51 418 -460 END
21:12:21.838 -> START 2008 3268 16908 73 734 -476 END
21:12:21.838 -> START 1892 3292 17128 191 965 -293 END
21:12:21.838 -> START 1788 3180 17484 68 1202 -295 END
21:12:21.872 -> START 1732 2696 17168 -133 1317 -503 END
21:12:21.872 -> START 1992 3048 17240 -120 1408 -754 END
21:12:21.872 -> START 2016 3128 17164 94 1346 -640 END
21:12:21.906 -> START 1828 3016 17524 230 1156 -765 END
21:12:21.906 -> START 1912 3128 17432 127 994 -847 END
21:12:21.906 -> START 1836 3028 17532 -25 764 -859 END
21:12:21.906 -> START 1892 3068 17332 78 461 -925 END
21:12:21.940 -> START 2016 3080 17236 73 408 -858 END
21:12:21.940 -> START 1844 3128 17164 138 426 -832 END
21:12:21.940 -> START 1876 3192 17160 342 483 -782 END
21:12:21.975 -> START 1700 3296 17432 635 482 -636 END
21:12:21.975 -> START 1808 3440 17572 668 650 -423 END
21:12:21.975 -> START 1448 3388 17768 756 617 -373 END
21:12:22.009 -> START 1376 3644 17756 799 445 -231 END
21:12:22.009 -> CLOSING BATCH
```

☒ Autoscroll ☒ Show timestamp

Newline

38400 baud

Clear output

Invalid library found in C:\Users\hp\Documents\Arduino\libraries\sketch_oct03a: no headers files (.h) found in C:\Users\hp\Documents\Arduino\libraries\sketch_oct03a

Invalid library found in C:\Users\hp\Documents\Arduino\libraries\sketch_oct05a: no headers files (.h) found in C:\Users\hp\Documents\Arduino\libraries\sketch_oct05a



Project

```
#include <Wire.h>
// Pins used for I/O
int btnPin1 = 9;

// I2C address of the MPU-6050
const int MPU_addr=0x68;
// Variables that will store sensor data
int16_t AcX,AcY,AcZ,Tmp,GyX,GyY,GyZ;

// Status variables, used with buttons
int precBtn1 = HIGH;

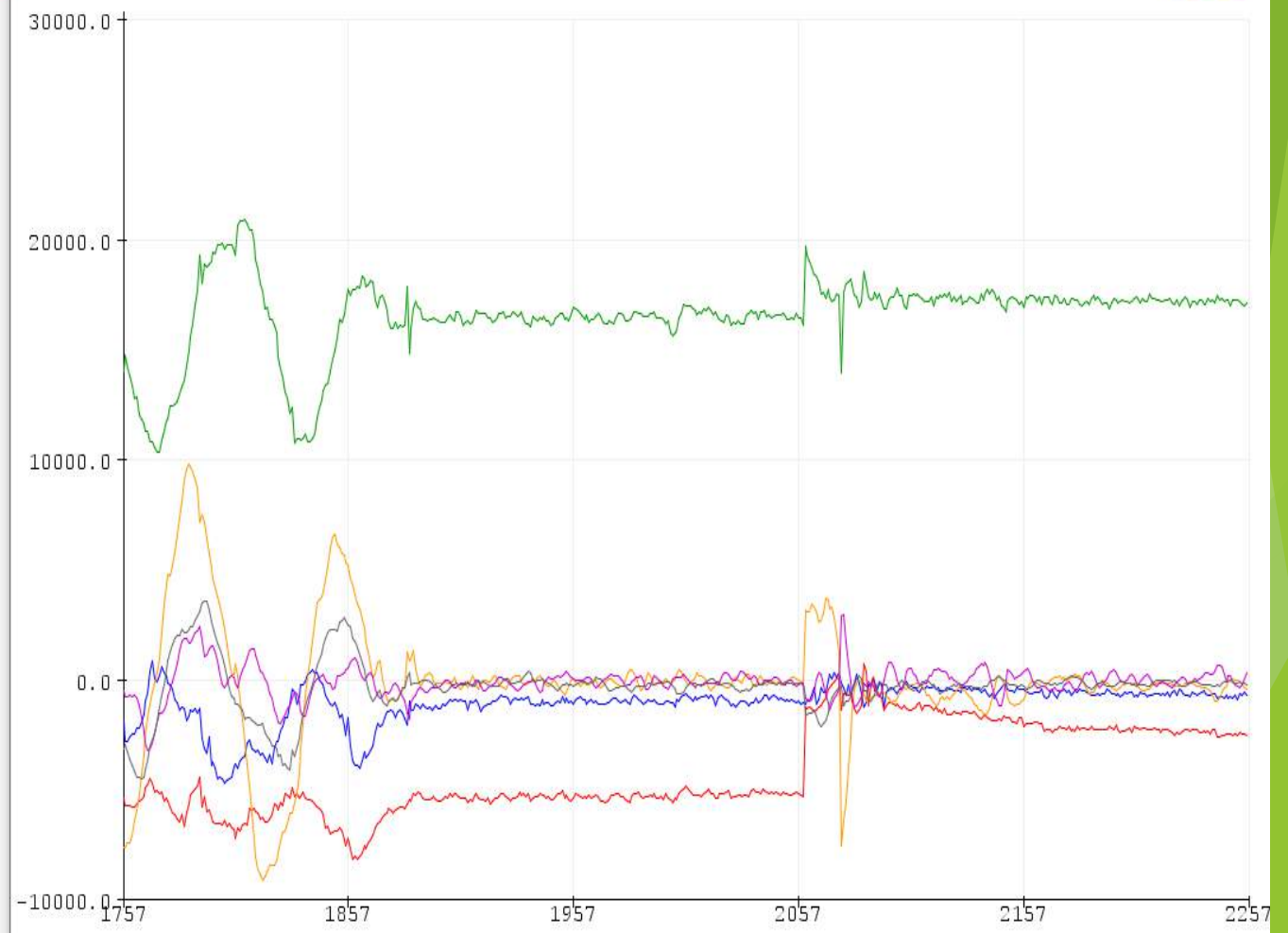
void setup(){
  // Set the pin mode of the buttons using the internal pullup resistor
  pinMode(btnPin1, INPUT_PULLUP);

  // Start the communication with the MPU-6050 sensor
  Wire.begin();
  Wire.beginTransmission(MPU_addr);
  Wire.write(0x6B); // PWR_MGMT_1 register
  Wire.write(0);    // set to zero (wakes up the MPU-6050)
  Wire.endTransmission(true);

  // Start the serial communication
  Serial.begin(38400);
}

void loop() {
```

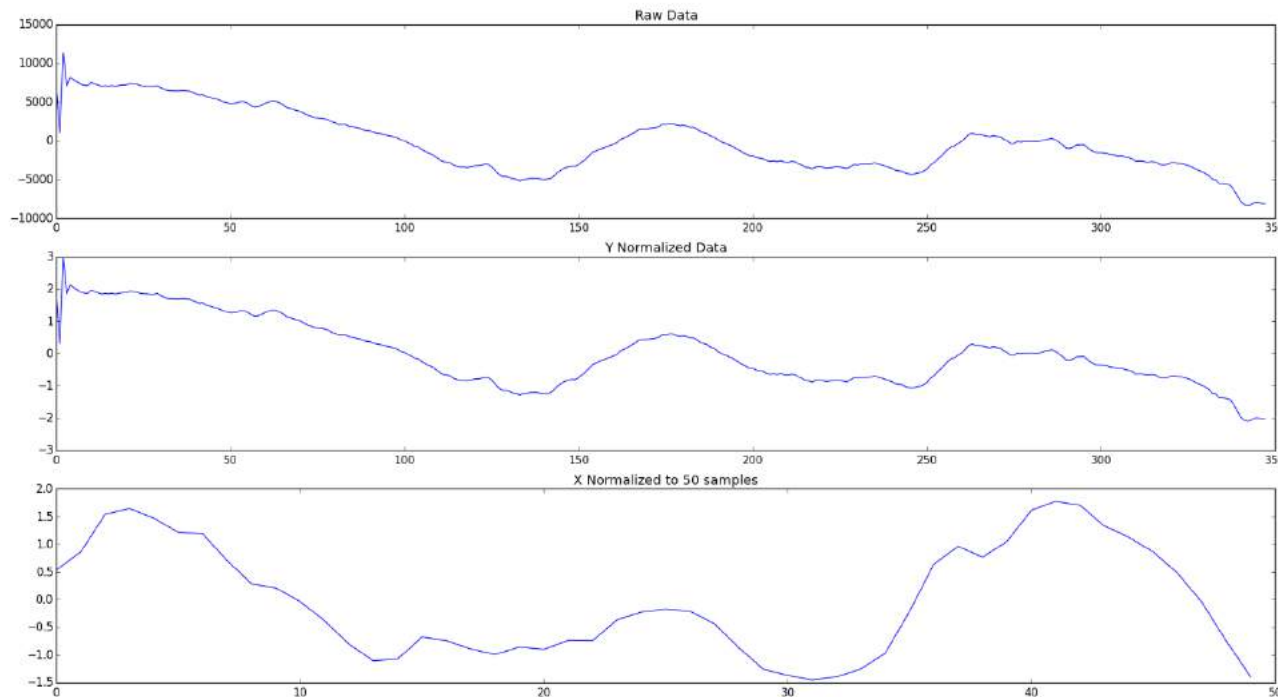
COM5



38400 baud

Results Achieved

- ▶ Finally, we came to a position, where the Gesture Keyboard is almost complete, and somewhat functional.
- ▶ graphical / tabular representation of the results.
- ▶ put proper titles



Conclusion

- ▶ Initially, we just had a little knowledge of Arduino and python, in the process of building the Gesture Keyboard system, we learnt new concepts, techniques, and improved our Computer Science knowledge. We learnt basics of SVM (support vector machine), and came to know how to develop, test and train the model.
- ▶ We got acquainted to the installation and worked on the library Pygarl. Enhanced our C and Python skills with the help of the project. Also, we made a step to enter the fascinating technology of Gesture Recognition.

limitations

Some **Limitations** of the project are:-

- ▶ 1. Gesture Keyboard doesn't involve sound, so the text printed on the screen couldn't be read out loud.
- ▶ 2. The system requires to be connected to a Power bank, so it's battery charging is an issue in case if we use Bluetooth module.
- ▶ The **Future Scope** of the project is:-
 - ▶ 1. Entering the world of controller driven games which also use Gesture Recognition.
 - ▶ 2. Creating the hands-free version of the Gesture Keyboard:- Presently, we need to hold the Arduino system in our hand and move it using our whole arm , later, we see to eliminate the need to hold the system .

References

- ▶ The **sources** that helped us for the project are:
- ▶ 1. GitHub:- This platform gave us the basic idea, and motivated us to chose the topic Gesture Keyboard for the 3rd Semester Project.
- ▶ 2. YouTube:- YouTube helped us working with Arduino, and build the system.
- ▶ 3. University Seniors and faculties:- The seniors helped us with the errors in our codes & lab faculty helped us with the part of soldering.