EMBEDDED SYSTEMS PROJECT

Department of Electronics and Communication Engineering Indian Institute of Information Technology Design and Manufacturing Kurnool-518008, Andhra Pradesh (India)

PROJECT REPORT

Sign Language Glove

Submitted by:

- **K Shyam Prasad (122EC0006)**

Overview:

This project focuses on developing a wearable glove that recognizes sign language gestures and translates them into text or speech using Arduino and Bluetooth. The goal is to create an accessible tool for individuals with hearing impairments, enabling easier communication through sign language.

Importance:

Sign language is the primary mode of communication for many individuals with hearing disabilities. However, not everyone is familiar with it. This project aims to bridge that communication gap by creating an automated system to translate sign language into readable text or audible speech.

Technology:

The system utilizes Arduino for hardware control, flex sensors to detect finger movements, and Bluetooth for wireless communication to send data to a receiving device (smartphone, PC, etc.). The gesture data is then translated into text or speech.

Objectives:

- Develop a Wearable Sign Language Recognition Glove
- Integrate Bluetooth for Wireless Communication
- Convert Gestures into Digital Text or Speech
- User-Friendly Interface

System Architecture:

- Arduino UNO/NANO
- Flex Sensors
- Bluetooth Module (HC-05)
- Accelerometer ADXL
- Jumper Wires, Glove
- Bread Board, 9V Battery
- Bluetooth Communication Protocol
- Arduino IDE and Gesture Recognition Algorithm**.**

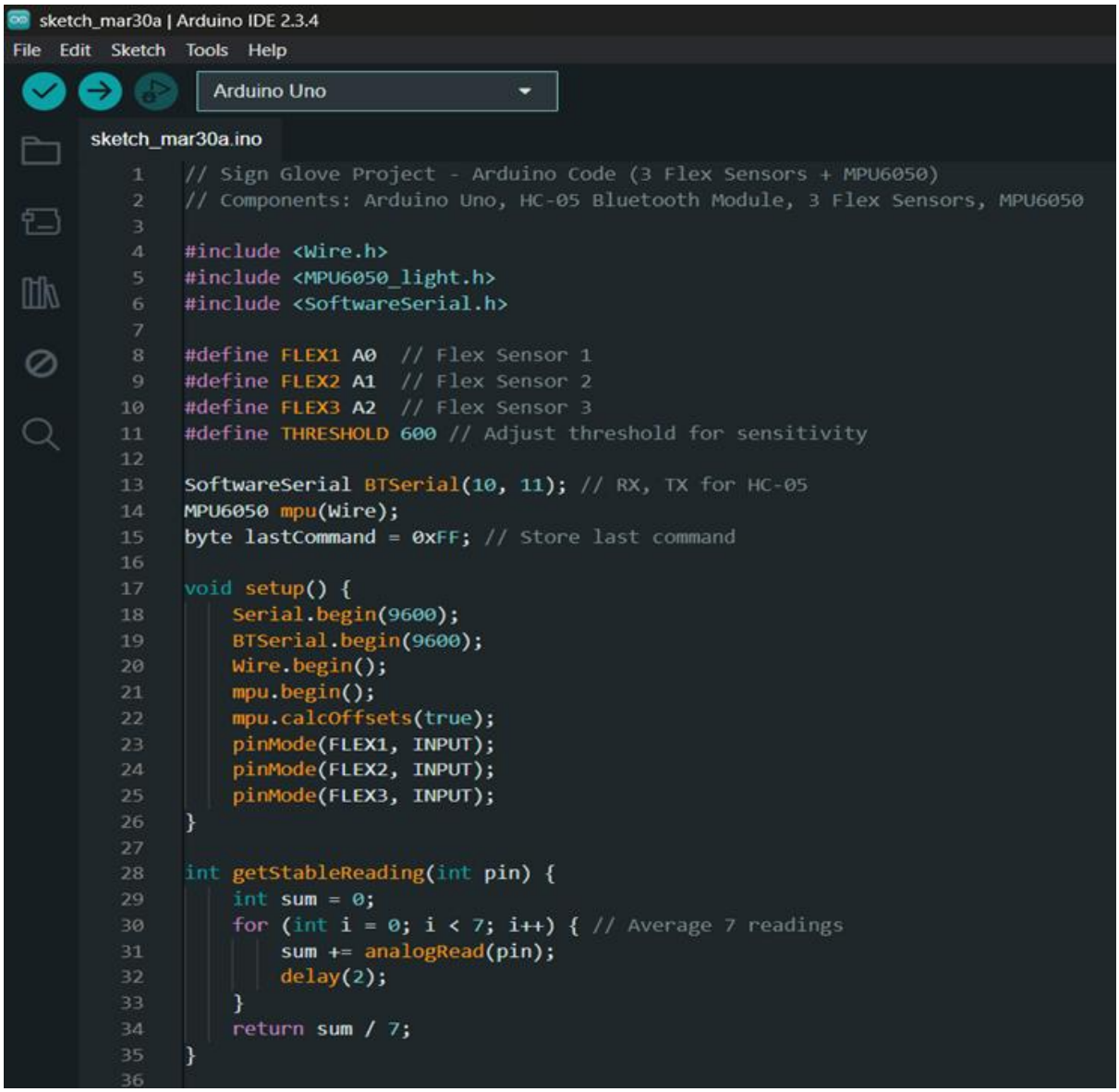
Working Mechanism:

- Step 1: The user wears the glove and makes a sign language gesture.
- Step 2: The flex sensors on the glove detect the bending of the fingers, generating corresponding data.
- Step 3: The Arduino processes the data and identifies the gesture based on predefined patterns.
- Step 4: The identified gesture is transmitted to a smartphone or computer via Bluetooth.
- Step 5: The receiving device translates the gesture into text or speech, providing real-time feedback to the user.

TRUTH TABLE FOR COMMANDS USING 3-BITS (8 variations):

Flex3 (A2)	Flex2 (A1)	Flex1 (A0)	Command Sent
0	0	0	Emergency
0	0	1	I need water
0	1	0	I need help
0	1	1	I am hungry
1	0	0	Call a doctor
1	0	1	I am in pain
1	1	0	I am okay
1	1	1	(No command defined)

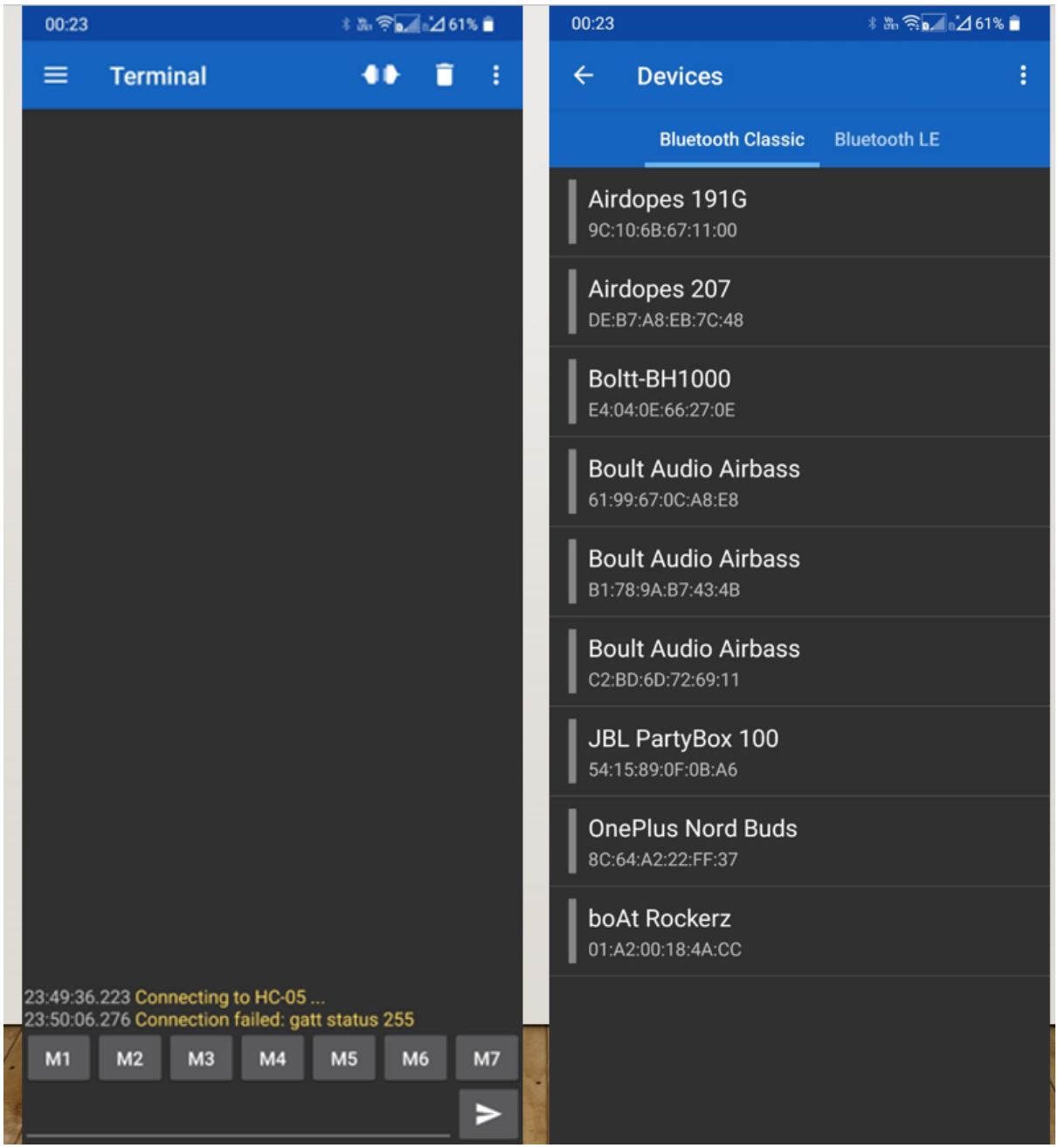
CPP CODE FOR GESTURE RECOGNITION:



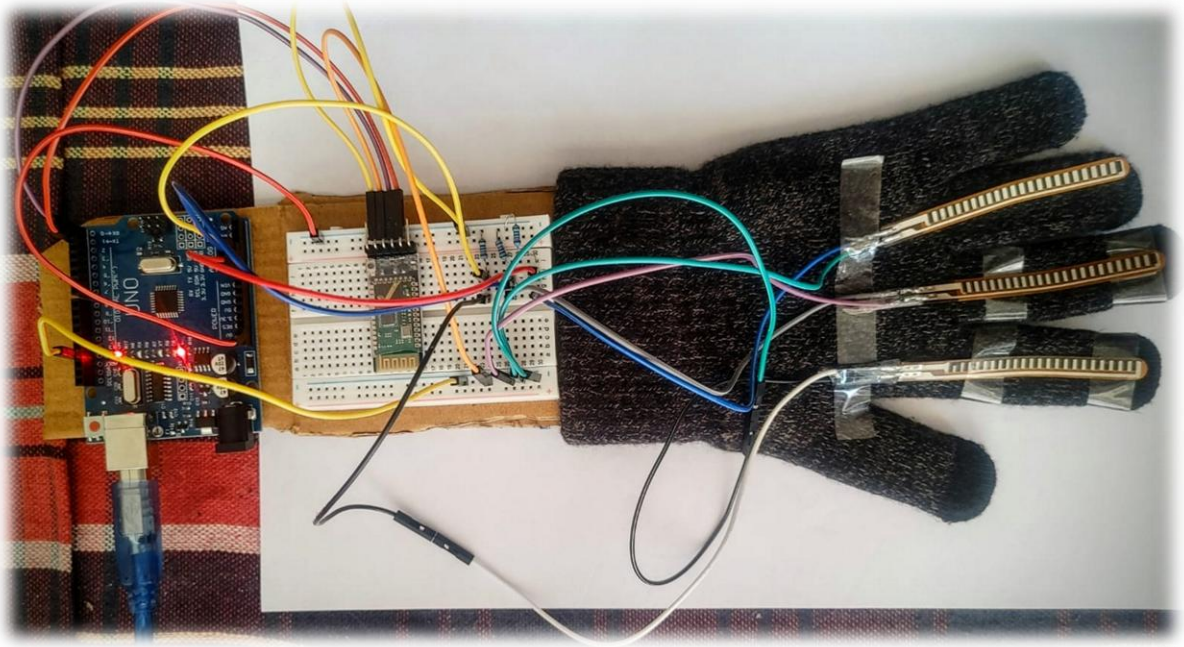

```cpp
// Sign Glove Project - Arduino Code (3 Flex Sensors + MPU6050)
// Components: Arduino Uno, HC-05 Bluetooth Module, 3 Flex Sensors, MPU6050

#include <Wire.h>
#include <MPU6050_light.h>
#include <SoftwareSerial.h>

#define FLEX1 A0  // Flex Sensor 1
#define FLEX2 A1  // Flex Sensor 2
#define FLEX3 A2  // Flex Sensor 3
#define THRESHOLD 600 // Adjust threshold for sensitivity

SoftwareSerial BTSerial(10, 11); // RX, TX for HC-05
MPU6050 mpu(Wire);
byte lastCommand = 0xFF; // Store last command

void setup() {
    Serial.begin(9600);
    BTSerial.begin(9600);
    Wire.begin();
    mpu.begin();
    mpu.calcOffsets(true);
    pinMode(FLEX1, INPUT);
    pinMode(FLEX2, INPUT);
    pinMode(FLEX3, INPUT);
}

int getStableReading(int pin) {
    int sum = 0;
    for (int i = 0; i < 7; i++) { // Average 7 readings
        sum += analogRead(pin);
        delay(2);
    }
    return sum / 7;
}
```

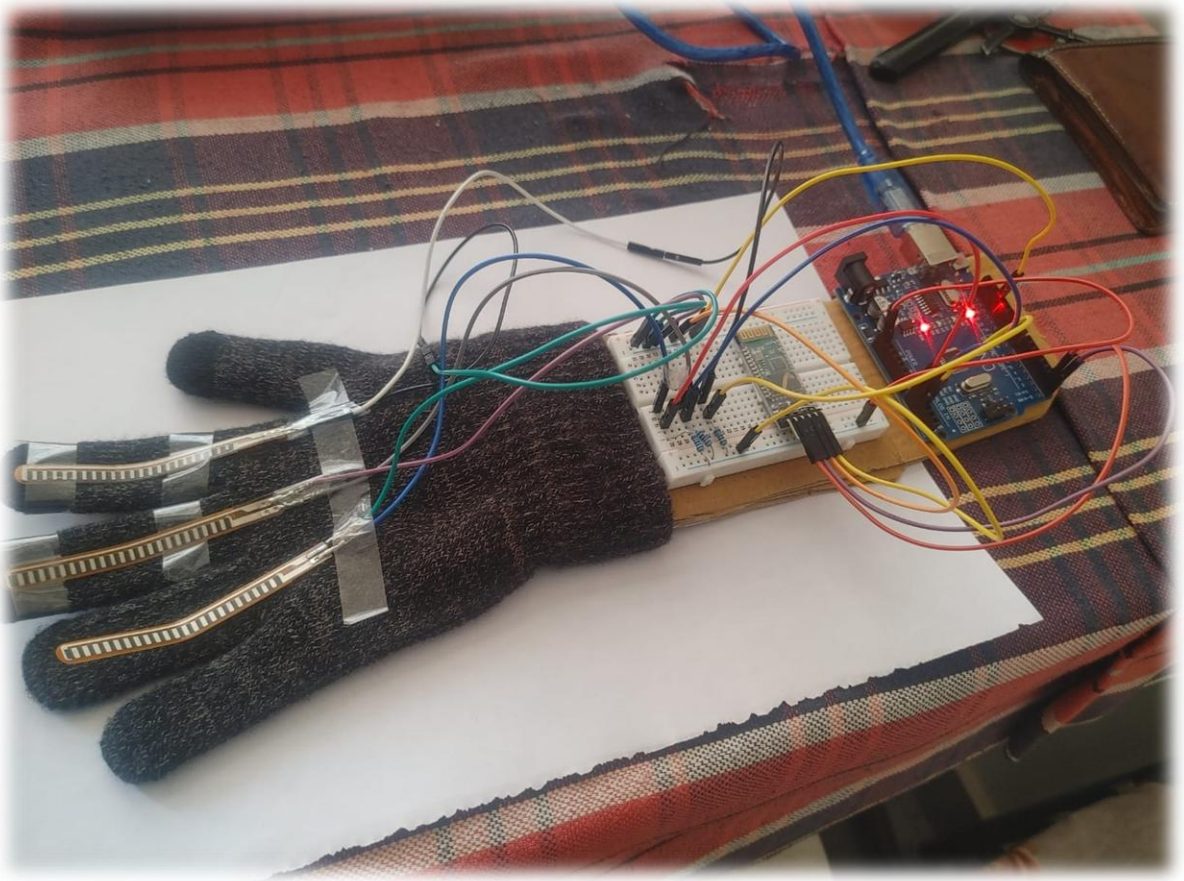
```
void loop() {
    int flex1 = getStableReading(FLEX1);
    int flex2 = getStableReading(FLEX2);
    int flex3 = getStableReading(FLEX3);

    byte flex1State = (flex1 > THRESHOLD) ? 0b1 : 0b0;
    byte flex2State = (flex2 > THRESHOLD) ? 0b1 : 0b0;
    byte flex3State = (flex3 > THRESHOLD) ? 0b1 : 0b0;
    byte command = (flex3State << 2) | (flex2State << 1) | flex1State;

    String gesture = "";
    switch (command) {
        case 0b000: gesture = "Emergency"; break;
        case 0b001: gesture = "I need water"; break;
        case 0b010: gesture = "I need help"; break;
        case 0b011: gesture = "I am hungry"; break;
        case 0b100: gesture = "Call a doctor"; break;
        case 0b101: gesture = "I am in pain"; break;
        case 0b110: gesture = "I am okay"; break;
    }

    mpu.update();
    float ax = mpu.getAccX();
    float ay = mpu.getAccY();
    float az = mpu.getAccZ();

    if (ax > 1.5) gesture = "Call a nurse";
    else if (ax < -1.5) gesture = "I feel dizzy";
    else if (ay > 1.5) gesture = "Increase room temperature";
    else if (ay < -1.5) gesture = "Decrease room temperature";
    else if (az > 1.5) gesture = "Turn on lights";
    else if (az < -1.5) gesture = "Turn off lights";

    if (gesture != "") {
        Serial.print("COMMAND SENT -> ");
        Serial.println(gesture);
        BTSerial.println(gesture);
    }

    delay(200);
}
```

Designed Bluetooth Terminal Application:

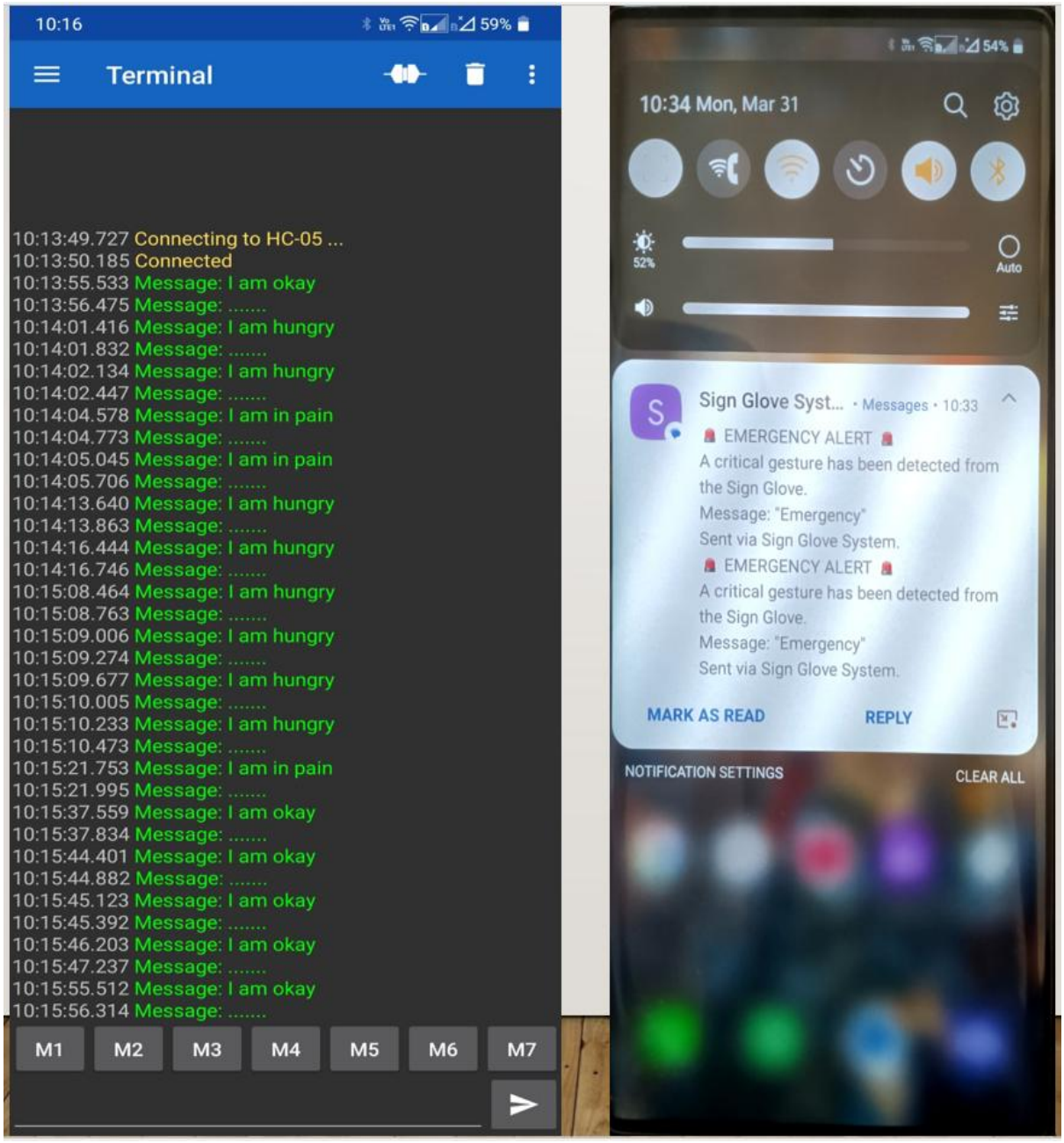


PROTOTYPE:





Results:


10:16
59%
Terminal
10:13:49.727 Connecting to HC-05 ...
10:13:50.185 Connected
10:13:55.533 Message: I am okay
10:13:56.475 Message: .....
10:14:01.416 Message: I am hungry
10:14:01.832 Message: .....
10:14:02.134 Message: I am hungry
10:14:02.447 Message: .....
10:14:04.578 Message: I am in pain
10:14:04.773 Message: .....
10:14:05.045 Message: I am in pain
10:14:05.706 Message: .....
10:14:13.640 Message: I am hungry
10:14:13.863 Message: .....
10:14:16.444 Message: I am hungry
10:14:16.746 Message: .....
10:15:08.464 Message: I am hungry
10:15:08.763 Message: .....
10:15:09.006 Message: I am hungry
10:15:09.274 Message: .....
10:15:09.677 Message: I am hungry
10:15:10.005 Message: .....
10:15:10.233 Message: I am hungry
10:15:10.473 Message: .....
10:15:21.753 Message: I am in pain
10:15:21.995 Message: .....
10:15:37.559 Message: I am okay
10:15:37.834 Message: .....
10:15:44.401 Message: I am okay
10:15:44.882 Message: .....
10:15:45.123 Message: I am okay
10:15:45.392 Message: .....
10:15:46.203 Message: I am okay
10:15:47.237 Message: .....
10:15:55.512 Message: I am okay
10:15:56.314 Message: .....
M1
M2
M3
M4
M5
M6
M7
54%
10:34 Mon, Mar 31
52%
Auto
Sign Glove Syst... • Messages • 10:33
EMERGENCY ALERT
A critical gesture has been detected from the Sign Glove.
Message: "Emergency"
Sent via Sign Glove System.
EMERGENCY ALERT
A critical gesture has been detected from the Sign Glove.
Message: "Emergency"
Sent via Sign Glove System.
MARK AS READ
REPLY
NOTIFICATION SETTINGS
CLEAR ALL

Uses of the Sign Glove for Patients 🏥🤖:

The Sign Glove project using flex sensors + MPU6050 accelerometer provides an innovative way for patients to communicate effectively, especially for those with limited speech or mobility.

💡 Key Benefits for Patients:

1. Non-Verbal Communication
2. Quick Emergency Alerts
3. Motion-Based Assistance
4. Reduced Caregiver Workload
5. lmproved Patient Comfort
6. Usable for Multiple Medical Conditions

BILL OF MATERIALS:

Component	Approx. Cost (INR)
Arduino Uno Clone	₹350
HC-05 Bluetooth Module	₹150
3 Flex Sensors	₹300 (₹100 each)
MPU6050 IMU Sensor	₹100
Glove (base material)	₹100
Wires, connectors, misc	₹100
Assembly labor	₹100
Total BOM Cost	**~₹1,200**

IDEAL SELLING PRICE:

Strategy	Price Range (INR)	Target Group	Notes
Social Impact / NGO	₹1,500 – ₹2,000	Needy patients, hospitals	Break-even or slight profit
Retail / Market Rate	₹2,500 – ₹3,500	Mass market	Includes packaging, profit, support
Premium Smart Version	₹4,000 – ₹5,000	Hospitals, accessibility startups	Add mobile app, SMS alerts, battery

Conclusion:

The **Sign Glove** successfully bridges the communication gap for **patients with speech or mobility impairments**, allowing them to convey their needs effortlessly. By integrating **flex sensors** and an **MPU6050 accelerometer**, the glove translates hand gestures into predefined commands, which are then transmitted via **Bluetooth** to a connected device.

This project has significant **applications in healthcare**, especially for **bedridden patients, individuals with paralysis, and elderly patients** who face difficulty in verbal communication. The system ensures **quick response times** from caregivers, enhancing patient care and comfort.