

DAFTAR PUSTAKA

- Abougarair, A. J. (2022). Smart glove for sign language translation. *International Robotics & Automation Journal*, 8(3), 109–117.
<https://doi.org/10.15406/iratj.2022.08.00253>
- Agus Faudin. (2022, August 23). *Protocol Komunikasi I2C pada ESP32*.
<https://www.nyebarilmu.com/protocol-komunikasi-i2c-pada-esp32/>
- Bang, J., You, J., & Lee, Y. (2019). *A Prototype of Flex Sensor Based Data Gloves to Track the Movements of Fingers*.
- Electrical4U. (2020, October 28). *Sensor | Type of Sensor*.
<https://www.electrical4u.com/sensor-types-of-sensor/>
- Haupt, R. L. (2020). Bluetooth. In *Wireless Communications Systems* (pp. 413–414). [https://doi.org/https://doi.org/10.1002/9781119419204.app4](https://doi.org/10.1002/9781119419204.app4)
- Hemmanur, K. (2009). *Inter-Integrated Circuit (I2C)*.
- Kamus SIBI. (2023). <https://pmpk.kemdikbud.go.id/sibi/profil>
- KBBI. (2023). *Kamus Besar Bahasa Indonesia Edisi VI*.
<https://kbbi.kemdikbud.go.id/>
- Khamdi, N., & Raja Adrafi, M. (2022). *Sarung Tangan Cerdas Sebagai Translator Bahasa Isyarat untuk Tuna Wicara* (Vol. 8, Issue 2).
<https://jurnal.pcr.ac.id/index.php/elementer>
- Made Bayu, I., Yudha, K., Afwani, R., & Zafrullah, A. (2022). *Aplikasi Augmented Reality Berbasis Android Sebagai Media Pembelajaran Mengenal Bahasa Isyarat SIBI Dengan Metode Based Marker Multi Target (Android-Based Augmented Reality Application as a Learning Media to Recognize SIBI Sign Language Using Multi-Target Marker-Based Method)*.
<http://jtika.if.unram.ac.id/index.php/JTIKA/>
- Monique Bernice H. Flores. (2014). *User-Oriented Finger-Gesture Glove Controller with Hand Movement Virtualization Using Flex Sensors and a Digital Accelerometer*.
- Pranoto, D., Handojo, A., & Satiabudhi, G. (2022). *Aplikasi War Game Pada Mobile Device Menggunakan Sensor Gyroscope dan Accelerometer*.

- Rahman Alma, V., & Setiawan, I. (2011). *Makalah Seminar Tugas Akhir Aplikasi Sensor Accelerometer Pada Deteksi Posisi*.
- Reynolds, C. R. & M. L. (1983). *Encyclopedia of Special Education*. A Wiley-Interscience Publication.
- Ri Adi, A., & Aditia, P. (2017). *Buku Ilustrasi Bahasa Isyarat Indonesia (BISINDO) Bagi Anak Tunarungu Illustrated Book Of Indonesian Sign Language For Deaf Children*.
- SBG Systems. (2018). *IMU - Inertial Measurement Unit*. <https://www.sbg-systems.com/inertial-measurement-unit-imu-sensor/>
- Sinukun, R. S., & Darise, Y. (2017). Aplikasi Bahasa Isyarat Sederhana Berbasis Android. In *JTII* (Vol. 2, Issue 1).
- Tania, F. (2014). *Pola Komunikasi Guru Penyandang Tunarungu Terhadap Siswa Penyandang Autis Pada Pembelajaran Seni Lukis (Studi Kasus di Kelas Menengah SLB Autisme Pelita Hafizh Bandung)*. <https://api.semanticscholar.org/CorpusID:142232863>
- Wiranda, N., & Putro, A. E. (2019). Model Identifikasi Kata Ucapan Tuna Wicara. *IJEIS (Indonesian Journal of Electronics and Instrumentation Systems)*, 9(2), 131. <https://doi.org/10.22146/ijeis.47609>

Lampiran 1 Kode Program

```

#include "BluetoothSerial.h"
#include <Wire.h>
#include <MPU6050.h>
BluetoothSerial SerialBT;
MPU6050 mpu;

int FLEX_PIN1 = A0;
int flexADC1 = 0;
int sensorMin1 = 510;
int sensorMax1 = 950;
int FLEX_PIN2 = A3;
int flexADC2 = 0;
int sensorMin2 = 400;
int sensorMax2 = 840;
int FLEX_PIN3 = A6;
int flexADC3 = 0;
int sensorMin3 = 480;
int sensorMax3 = 955;
int FLEX_PIN4 = A7;
int flexADC4 = 0;
int sensorMin4 = 930;
int sensorMax4 = 1200;
int FLEX_PIN5 = A4;
int flexADC5 = 0;
int sensorMin5 = 540;
int sensorMax5 = 960;

char letter = ' ';
char previousLetter = ' ';
unsigned long previousMillis = 0;
const long interval = 500; // 0.5 second interval

void setup() {
  SerialBT.begin("ESP32-Bluetooth");
  Serial.begin(57600);
  // Initialize MPU6050
  Wire.begin();
  mpu.initialize();

  // Calibrate MPU6050
  calibrateMPU6050();

  // Initial read to set min and max values correctly
  flexADC1 = analogRead(FLEX_PIN1);
  if(flexADC1 < sensorMin1) { sensorMin1 = flexADC1; }

```

```

    if(flexADC1 > sensorMax1) { sensorMax1 = flexADC1; }

    flexADC2 = analogRead(FLEX_PIN2);
    if(flexADC2 < sensorMin2) { sensorMin2 = flexADC2; }
    if(flexADC2 > sensorMax2) { sensorMax2 = flexADC2; }

    flexADC3 = analogRead(FLEX_PIN3);
    if(flexADC3 < sensorMin3) { sensorMin3 = flexADC3; }
    if(flexADC3 > sensorMax3) { sensorMax3 = flexADC3; }

    flexADC4 = analogRead(FLEX_PIN4);
    if(flexADC4 < sensorMin4) { sensorMin4 = flexADC4; }
    if(flexADC4 > sensorMax4) { sensorMax4 = flexADC4; }

    flexADC5 = analogRead(FLEX_PIN5);
    if(flexADC5 < sensorMin5) { sensorMin5 = flexADC5; }
    if(flexADC5 > sensorMax5) { sensorMax5 = flexADC5; }
}

void loop() {
    unsigned long currentMillis = millis();

    if (currentMillis - previousMillis >= interval) {
        previousMillis = currentMillis;

        // Read accelerometer data
        int16_t accelX, accelY, accelZ;
        mpu.getAcceleration(&accelX, &accelY, &accelZ);

        // Check if MPU6050 is in horizontal state
        if (accelZ > 800 && accelZ < 1200) {
            Serial.println("Horizontal state");
        }
        // Check if MPU6050 is in vertical state
        else if (accelX > 800 && accelX < 1200) {
            Serial.println("Vertical state");
        }
        // Check if MPU6050 is in equilibrium state
        else if (accelY > 800 && accelY < 1200) {
            Serial.println("Equilibrium state");
        }
        // Reading and mapping flex sensor values
        flexADC1 = analogRead(FLEX_PIN1);
        flexADC1 = constrain(flexADC1, sensorMin1, sensorMax1);
        int angle1 = map(flexADC1, sensorMin1, sensorMax1, 0, 90);

        flexADC2 = analogRead(FLEX_PIN2);
    }
}

```

```

flexADC2 = constrain(flexADC2, sensorMin2, sensorMax2);
int angle2 = map(flexADC2, sensorMin2, sensorMax2, 0, 90);

flexADC3 = analogRead(FLEX_PIN3);
flexADC3 = constrain(flexADC3, sensorMin3, sensorMax3);
int angle3 = map(flexADC3, sensorMin3, sensorMax3, 0, 90);

flexADC4 = analogRead(FLEX_PIN4);
flexADC4 = constrain(flexADC4, sensorMin4, sensorMax4);
int angle4 = map(flexADC4, sensorMin4, sensorMax4, 0, 90);

flexADC5 = analogRead(FLEX_PIN5);
flexADC5 = constrain(flexADC5, sensorMin5, sensorMax5);
int angle5 = map(flexADC5, sensorMin5, sensorMax5, 0, 90);

// Logic to determine the letter based on sensor readings
if((angle1 <= 60) && (angle1 >= 45) && (angle2 <= 56) && (angle3
<= 42) && (angle3 >= 10) && (angle4 <= 42) && (angle5 <= 22)) {
    letter = 'A';
}
else if((angle1 <= 30) && (angle2 >= 80) && (angle3 >= 80) &&
(angle4 >= 80) && (angle5 >= 80)) {
    letter = 'B';
}
else if((angle1 <= 80) && (angle1 >= 60) && (angle2 <= 80) &&
(angle2 >= 50) && (angle3 <= 80) && (angle3 >= 50) && (angle4 <= 80)
&& (angle4 >= 50) && (angle5 <= 80) && (angle5 >= 45)) {
    letter = 'C';
}
else if((angle1 <= 75) && (angle1 >= 50) && (angle2 >= 87) &&
(angle3 <= 60) && (angle3 >= 35) && (angle4 <= 60) && (angle4 >= 35)
&& (angle5 <= 55) && (angle5 >= 20)) {
    letter = 'D';
}
else if((angle1 <= 30) && (angle2 <= 60) && (angle2 >= 10) &&
(angle3 <= 42) && (angle3 >= 25) && (angle4 <= 42) && (angle4 >= 25)
&& (angle5 <= 22)) {
    letter = 'E';
}
else if((angle1 <= 50) && (angle2 <= 40) && (angle3 >= 80) &&
(angle4 >= 80) && (angle5 >= 80)) {
    letter = 'F';
}
else if((angle1 <= 85) && (angle1 >= 60) && (angle2 >= 65) &&
(angle3 <= 40) && (angle4 <= 40) && (angle5 <= 30) && (accely > 800
&& accely < 1200)) {
    letter = 'G';
}

```

```

    }
    else if((angle1 <= 65) && (angle1 >= 10) && (angle2 >= 80) &&
(angle3 >= 80) && (angle4 <= 75) && (angle5 <= 45) && (accelY > 800
&& accelY < 1200)) {
        letter = 'H';
    }
    else if((angle1 <= 50) && (angle2 <= 50) && (angle3 <= 45) &&
(angle4 <= 65) && (angle5 >= 80)) {
        letter = 'I';
    }
    else if((angle1 >= 66) && (angle1 <= 85) && (angle2 >= 75) &&
(angle3 >= 75) && (angle3 <= 90) && (angle4 <= 70) && (angle5 <= 45)
&& (accelZ > 800 && accelZ < 1200)) {
        letter = 'K';
    }
    else if((angle1 >= 80) && (angle2 >= 70) && (angle3 <= 40) &&
(angle4 <= 40) && (angle5 <= 40)) {
        letter = 'L';
    }
    else if((angle1 <= 35) && (angle2 <= 49) && (angle3 <= 49) &&
(angle4 <= 60) && (angle4 >= 35) && (angle5 <= 27)) {
        letter = 'M';
    }
    else if((angle1 <= 35) && (angle2 <= 55) && (angle2 >= 11) &&
(angle3 <= 55) && (angle3 >= 26) && (angle4 >= 40) && (angle5 <= 22))
{
        letter = 'N';
    }
    else if((angle1 <= 70) && (angle1 >= 35) && (angle2 <= 49) &&
(angle2 >= 20) && (angle3 <= 55) && (angle3 >= 20) && (angle4 <= 55)
&& (angle4 >= 20) && (angle5 <= 44) && (angle5 >= 10)) {
        letter = 'O';
    }
    else if((angle1 >= 66) && (angle1 <= 85) && (angle2 >= 75) &&
(angle3 >= 75) && (angle3 <= 90) && (angle4 <= 70) && (angle5 <= 45)
&& (accelX > 800 && accelX < 1200)) {
        letter = 'P';
    }
    else if((angle1 <= 85) && (angle1 >= 60) && (angle2 >= 65) &&
(angle3 <= 40) && (angle4 <= 40) && (angle5 <= 30) && (accelX > 800
&& accelX < 1200)) {
        letter = 'Q';
    }
    else if((angle1 <= 40) && (angle2 >= 80) && (angle3 <= 80) &&
(angle3 >= 70) && (angle4 <= 80) && (angle5 <= 40)) {
        letter = 'R';
    }
}

```

```

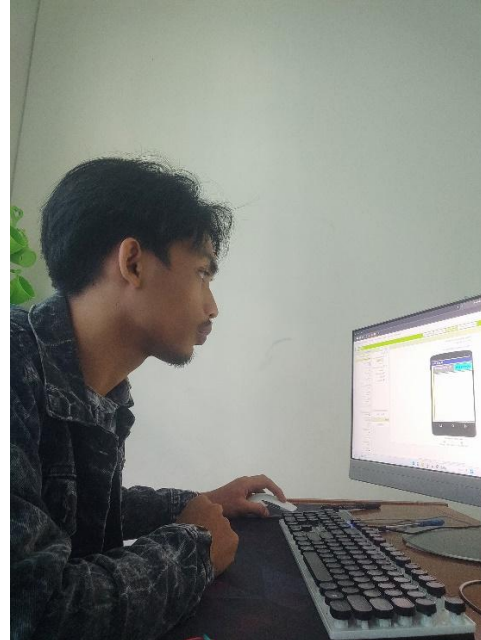
        else if((angle1 <= 25) && (angle2 <= 15) && (angle3 <= 25) &&
(angle4 <= 40) && (angle5 <= 27)) {
            letter = 'S';
        }
        else if((angle1 <= 85) && (angle1 >= 65) && (angle2 <= 31) &&
(angle2 >= 11) && (angle3 <= 30) && (angle3 >= 10) && (angle4 <= 50)
&& (angle4 >= 9) && (angle5 <= 20)) {
            letter = 'T';
        }
        else if((angle1 <= 65) && (angle1 >= 10) && (angle2 >= 80) &&
(angle3 >= 80) && (angle4 <= 75) && (angle5 <= 45) && (accelZ > 800
&& accelZ < 1200)) {
            letter = 'U';
        }
        else if((angle1 <= 65) && (angle1 >= 10) && (angle2 >= 80) &&
(angle3 >= 80) && (angle4 <= 75) && (angle5 <= 45)) {
            letter = 'V';
        }
        else if((angle1 <= 40) && (angle2 >= 80) && (angle3 >= 80) &&
(angle4 >= 80) && (angle5 <= 40)) {
            letter = 'W';
        }
        else if((angle1 <= 40) && (angle1 >= 20) && (angle2 <= 70) &&
(angle2 >= 45) && (angle3 <= 42) && (angle4 <= 54) && (angle5 <= 22))
{
            letter = 'X';
        }
        else if((angle1 >= 70) && (angle2 <= 45) && (angle3 <= 45) &&
(angle4 <= 65) && (angle5 >= 80)) {
            letter = 'Y';
        }
        else {
            letter = ' ';
        }

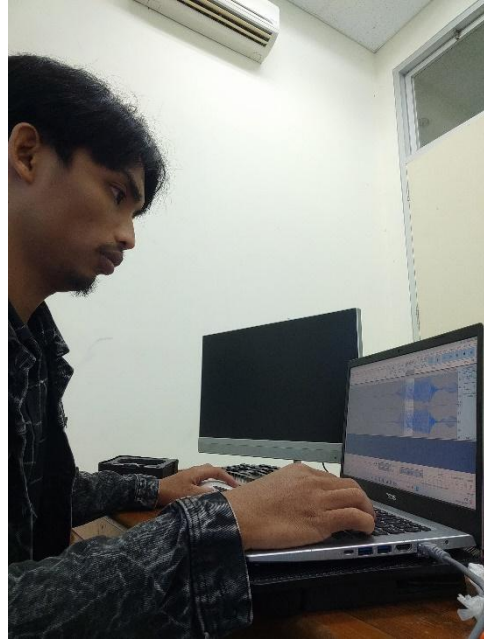
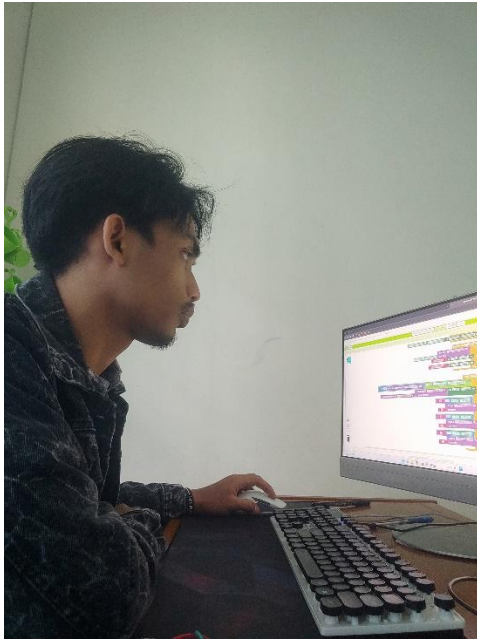
        // Only send if the letter has changed
        if (letter != previousLetter) {
            Serial.printf("Letter: %c | Angles: %d, %d, %d, %d, %d\n",
letter, angle1, angle2, angle3, angle4, angle5); // Print for
debugging
            SerialBT.print(letter); // Send the letter via Bluetooth
            previousLetter = letter;
        }
    }
    delay(20);
}
void calibrateMPU6050() {

```

```
// Put your calibration logic here  
Serial.println("Calibrating MPU6050...");  
// Example: mpu.calibrateGyro();  
}
```

Lampiran 2 Dokumentasi Perancangan





Lampiran 3 Pengujian Alat

