

Embedded Systems – Mid Sem Solutions

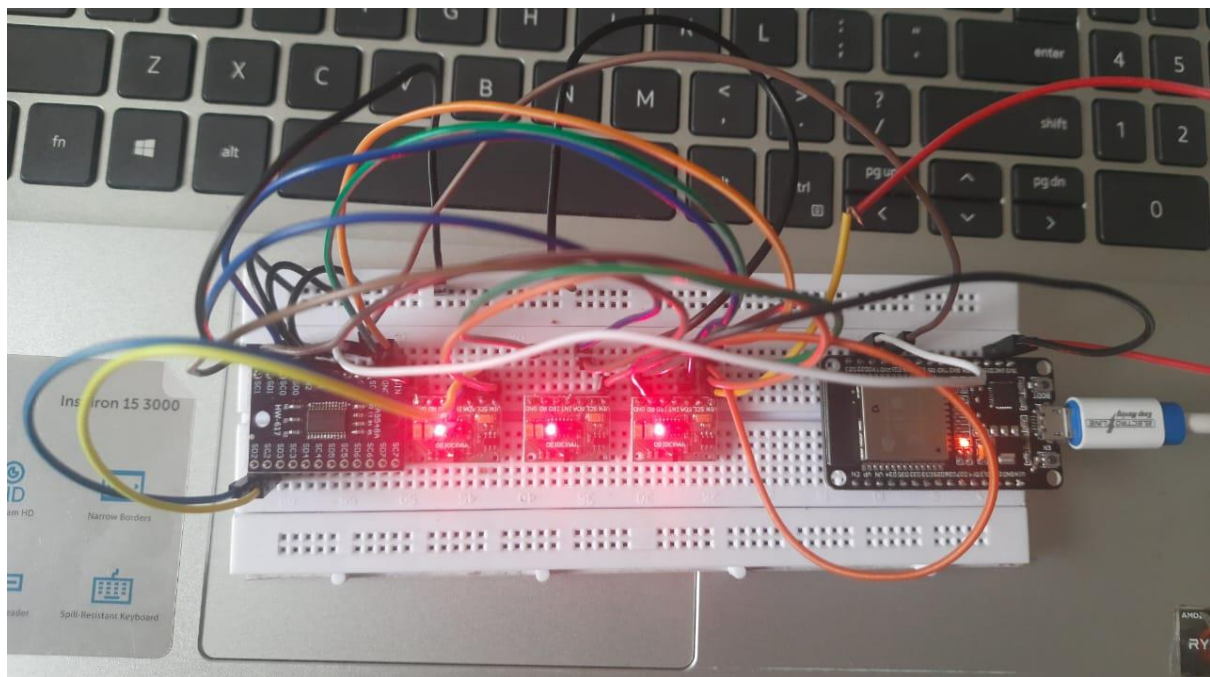
Name: Drishti Diwani

Enrolment Number: BT21ECE120

Course Name: ECL403 Embedded Systems

Instructor: Dr. Ankit Bhurane sir

Q2. Pulse Sensor Project

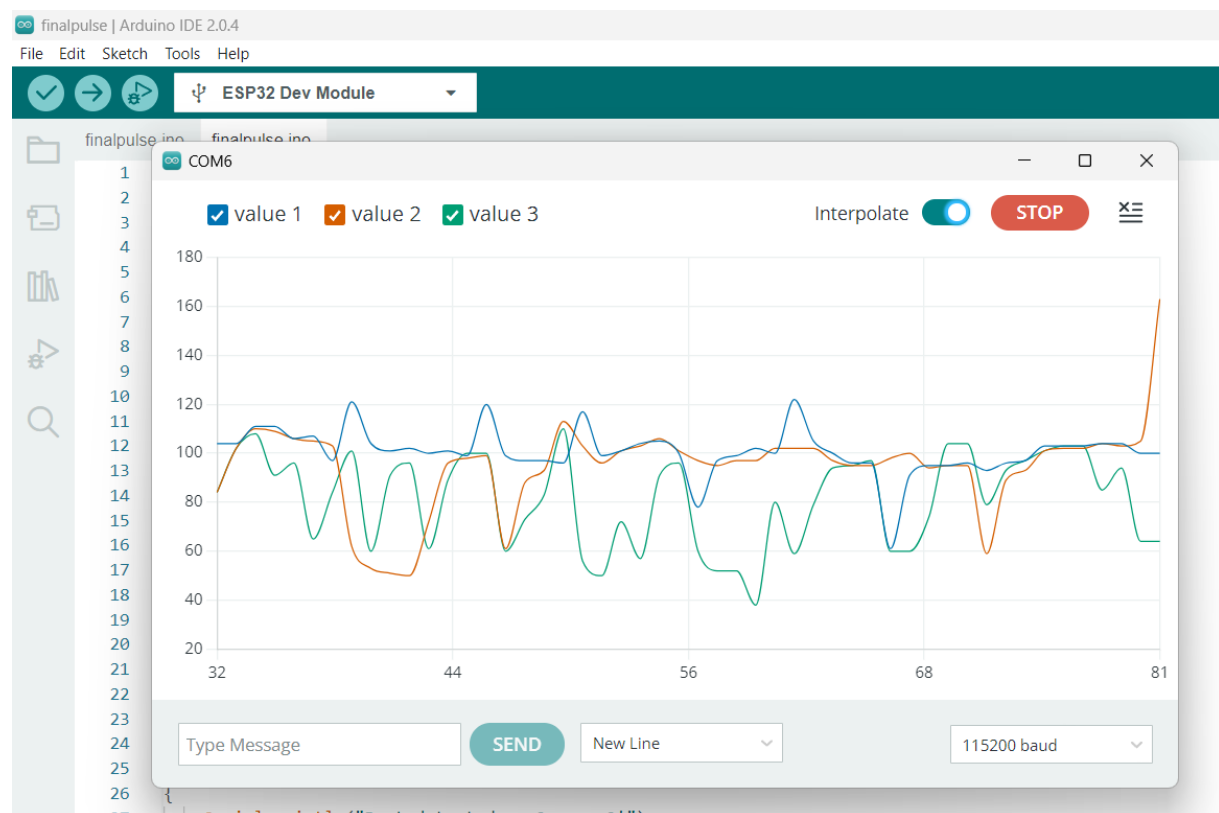


Explanation:

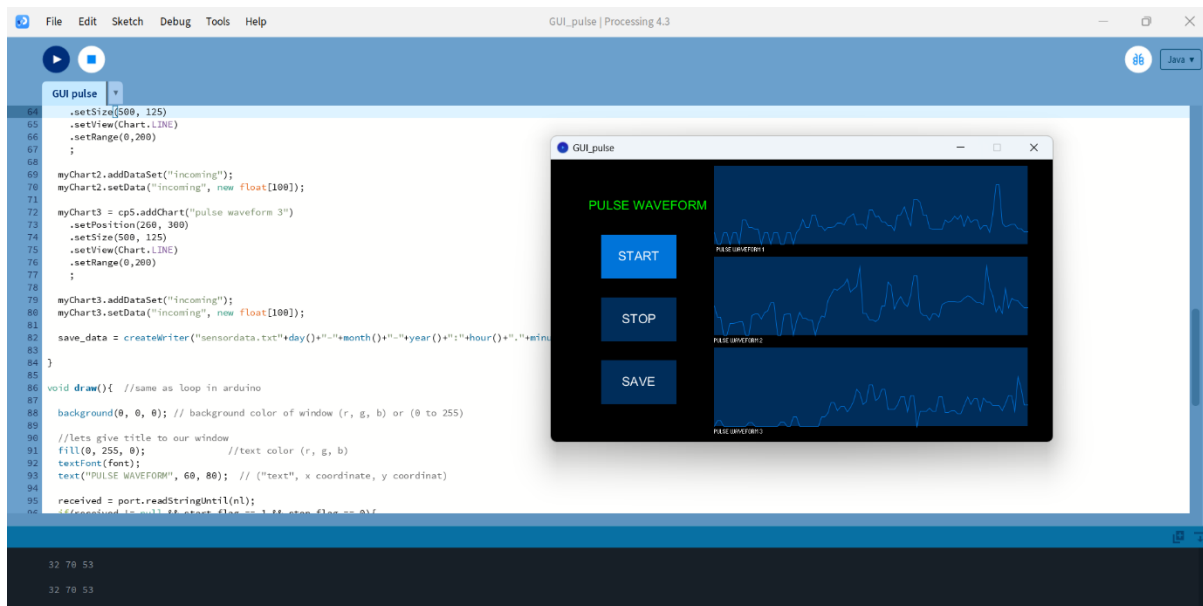
In this project we have used 3 MAX30100 pulse sensors and a I2C multiplexer TCA9548A which is used for using 3 serial communication devices at the same time. After trying out our code with one sensor we had to connect 2 more sensors but we got some problems with the code. After a lot of research, we realised that the function that is use to run 3 sensors together is not available with the MAX30100 sensor but with MAX30102 sensors. So, we bought TCA9548A multiplexer with which we serially connected our 3 sensors and they worked properly.

In the code provided below, we have included libraries Wire.h and MAX30100_PulseOximeter.h as we have used MAX30100 pulse sensor and I2C multiplexer TCA9548A for serial communication of heart beat data. We defined

a constant `REPORTING_PERIOD_MS` with a value of 1000 milliseconds (1 second). This constant is used to control how often the sensor readings are reported. Then we define `TCA9548A` function is used to switch between the sensors by selecting the appropriate I2C multiplexer channel before reading data from the sensor. We then define objects for each sensor and checked the initialisation of each sensor if they are connected properly and working fine in void setup function. In the void loop we used the `getHeartRate` method to get the BPM for each sensor. It repeatedly reads heart rate data from each sensor, stores it in an array (`bpm`), and prints the data to the serial monitor at regular intervals specified by `REPORTING_PERIOD_MS`. We also get 3 different plots in serial plotter as shown below.



After the code is uploaded in ESP32 we opened the GUI interface and wrote the code as provided. We changed the port where the ESP was connected and set the range for each sensor as required. As we run the code on the GUI interface, we got three waveform plots as shown below.



Code:

```
#include <Wire.h>
#include "MAX30100_PulseOximeter.h"

#define REPORTING_PERIOD_MS 1000

// Function to select the I2C multiplexer channel
void TCA9548A(uint8_t bus) {
    Wire.beginTransmission(0x70); // TCA9548A address
    Wire.write(1 << bus);         // send byte to selected bus
    Wire.endTransmission();
}

// PulseOximeter instances for each sensor
PulseOximeter sensor1;
PulseOximeter sensor2;
PulseOximeter sensor3;

float BPM1 = 0, BPM2 = 0, BPM3 = 0;

uint32_t tsLastReport = 0;

// Callback functions for beat detection
void onBeatDetected1() {
    Serial.println("Beat detected on Sensor 1!");
}

void onBeatDetected2() {
    Serial.println("Beat detected on Sensor 2!");
}
```

```

void onBeatDetected3() {
    Serial.println("Beat detected on Sensor 3!");
}

void setup() {
    Serial.begin(115200);

    Wire.begin();

    // Initializing the PulseOximeter instances for each sensor
    TCA9548A(1);
    if (!sensor1.begin()) {
        Serial.println("Sensor 1 initialization FAILED");
        for (;;);
    } else {
        Serial.println("Sensor 1 initialization SUCCESS");
    }
    // Uncomment the line below if you want to set a beat detection callback
    for Sensor 1
    // sensor1.setOnBeatDetectedCallback(onBeatDetected1);

    TCA9548A(0);
    if (!sensor2.begin()) {
        Serial.println("Sensor 2 initialization FAILED");
        for (;;);
    } else {
        Serial.println("Sensor 2 initialization SUCCESS");
    }
    // Uncomment the line below if you want to set a beat detection callback
    for Sensor 2
    // sensor2.setOnBeatDetectedCallback(onBeatDetected2);

    TCA9548A(2);
    if (!sensor3.begin()) {
        Serial.println("Sensor 3 initialization FAILED");
        for (;;);
    } else {
        Serial.println("Sensor 3 initialization SUCCESS");
    }
    // Uncomment the line below if you want to set a beat detection callback
    for Sensor 3
    // sensor3.setOnBeatDetectedCallback(onBeatDetected3);
}

void loop() {
    // Select the first sensor
    TCA9548A(1);
    sensor1.update();
}

```

```

BPM1 = sensor1.getHeartRate();

// Select the second sensor
TCA9548A(0);
sensor2.update();
BPM2 = sensor2.getHeartRate();

// Select the third sensor
TCA9548A(2);
sensor3.update();
BPM3 = sensor3.getHeartRate();

// Store heart rates in an array
int bpm[3];
bpm[0] = BPM1;
bpm[1] = BPM2;
bpm[2] = BPM3;

// Check if it's time to report
if (millis() - tsLastReport > REPORTING_PERIOD_MS) {
    // Print the heart rates
    Serial.print(bpm[0]);
    Serial.print(",");
    Serial.print(bpm[1]);
    Serial.print(",");
    Serial.print(bpm[2]);
    Serial.println();

    // Update the timestamp for the next report
    tsLastReport = millis();
}
}

```