



# **Visvesvaraya National Institute of Technology (VNIT), Nagpur**

## **Embedded Systems (ECP403)**

Submitted by :

**N SRI SAI NITYA (BT21ECE121)**

**PEDURI NIGAM (BT21ECE103)**

**VULLENGALA KARAN CHANDRA (BT21ECE091)**

**VYTTLA VENKATA SAI VARSHIT (BT21ECE094)**

**V BHAVANA (BT21ECE030)**

Semester 5

Submitted to:

**Dr. Ankit A. Bhurane**

(Course Instructor)

Department of Electronics and Communication Engineering,  
VNIT Nagpur

## AIM:

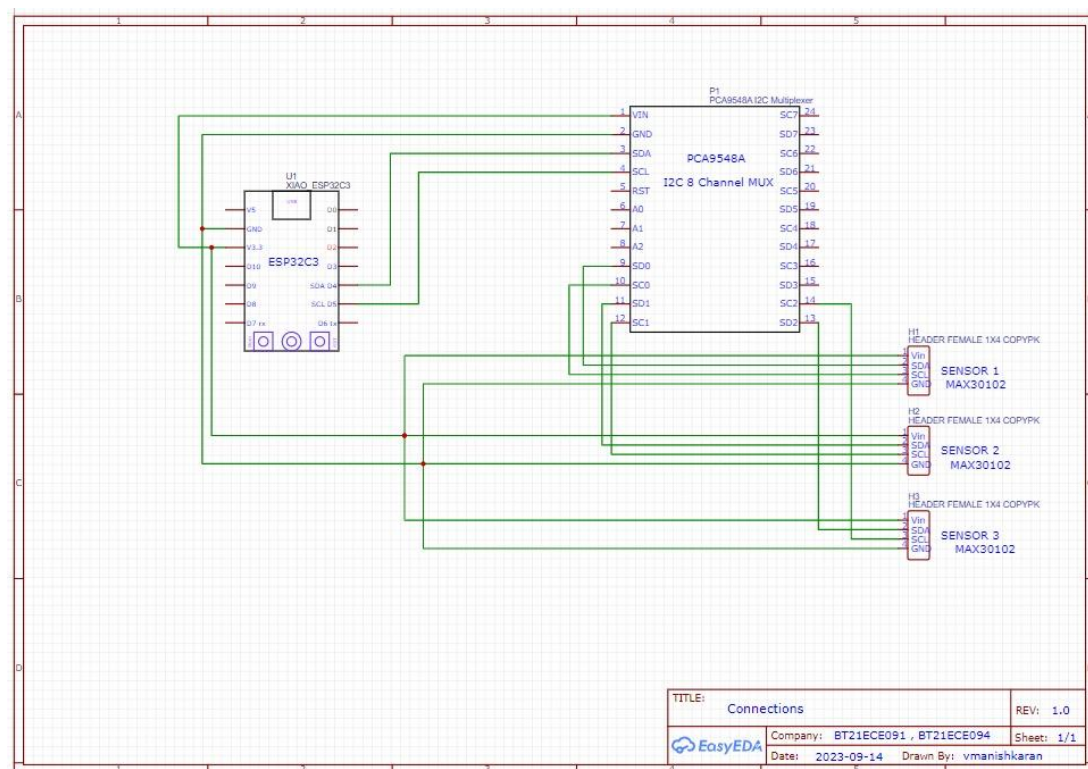
Demonstrate a PCB-level prototype (say, Technology Readiness Level (TRL-6)) of a wireless wrist pulse recording platform as discussed in the class. A broad guideline for the evaluation:

- Battery-operated standalone (5-10)
- Module-level or SMD level (5-10)
- PCB Compactness/ uniqueness of PCB (3-5-7-10)
- Group size (5-3-1)
- Viva/Add-on (5-8-12-15)
- Bonus: Recorded stable waveform of at least 10 people (preferably 5 boys and 5 girls data for at least 10 sec). (5)

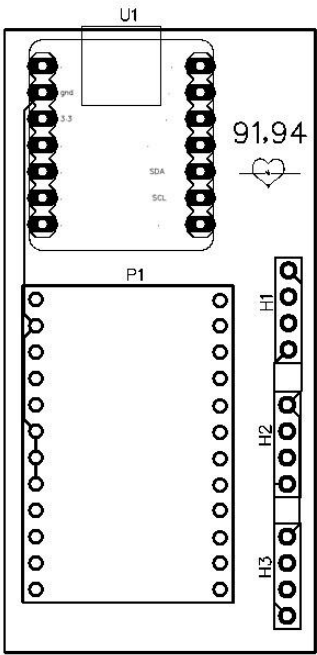
**COMPONENTS:** esp32c3, max30102 pulse sensors, tca9548a, lipo 250 mah battery, PCB.

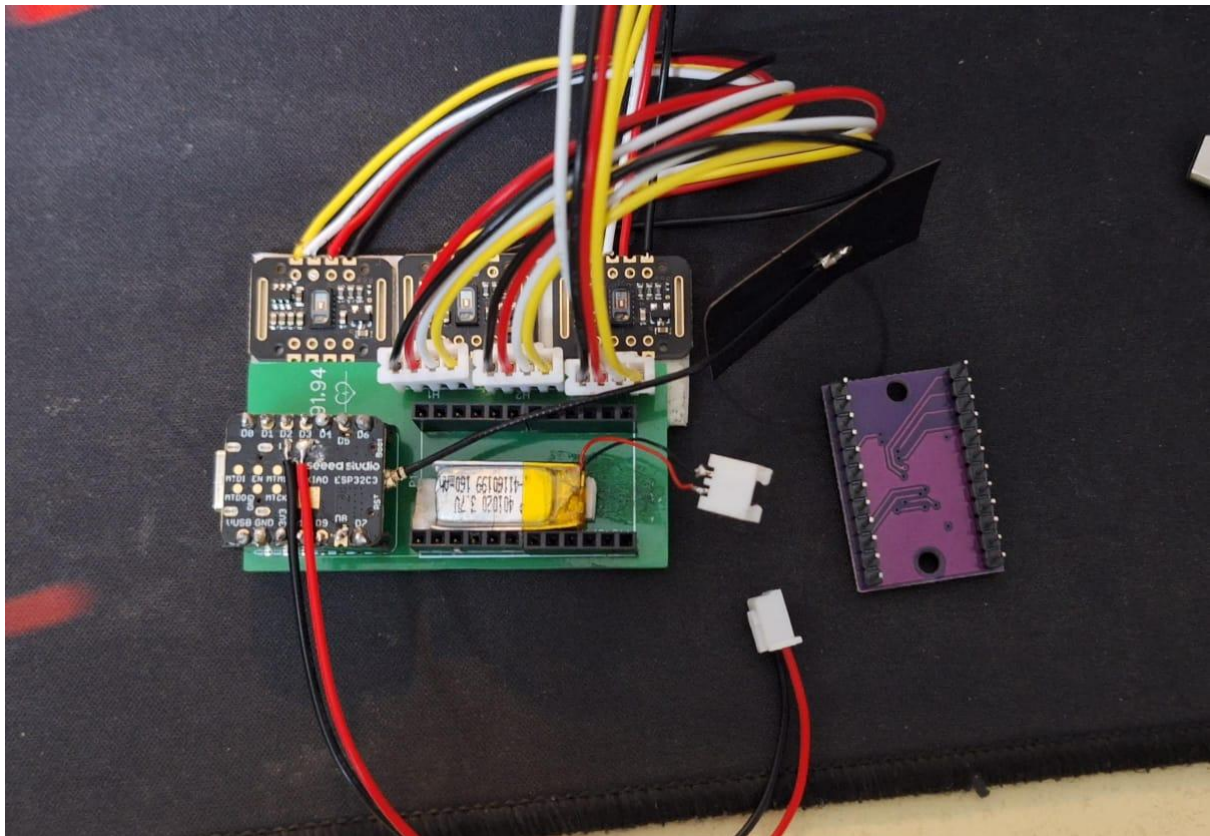
**SOFTWARE:** Arduino IDE, Python

## CIRCUIT DIAGRAM:



To
Cc
Bcc
Subject





### CODE:

```
#include <Wire.h>
#include "MAX30105.h"
#include "heartRate.h"
#include <WiFi.h>
#include <ESPAsyncWebServer.h>

MAX30105 particleSensors[3];

int irvalues[3];

const int numSensors = 3;
const int multiplexerAddress = 0x70;

const char* ssid = "Galaxy454";
const char* password = "xxxxxxxx";
AsyncWebServer server(80);
```

```

void selectSensor(int sensorNumber) {
  Wire.beginTransmission(multiplexerAddress);
  Wire.write(1 << sensorNumber);
  Wire.endTransmission();
}

void connectToWiFi() {
  Serial.println("Connecting to WiFi...");
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.println("Connecting to WiFi...");
  }

  Serial.println("Connected to WiFi");
  Serial.println(WiFi.localIP());
}

void setup() {
  Serial.begin(115200);
  Serial.println("Initializing...");

  connectToWiFi();

  Wire.begin();

  for (int i = 0; i < numSensors; ++i) {
    selectSensor(i);
    if (!particleSensors[i].begin(Wire, I2C_SPEED_FAST)) {
      Serial.print("MAX30102 ");
      Serial.print(i);
      Serial.println(" was not found. Please check wiring/power. ");
      while (1);
    }
    Serial.print("MAX30102 ");
    Serial.print(i);
    Serial.println(" found.");
    particleSensors[i].setup();
  }
}

```

```
    particleSensors[i].setPulseAmplitudeRed(0x0A);  
    particleSensors[i].setPulseAmplitudeGreen(0);  
}
```

```
Serial.println("Place your index finger on the sensors with steady  
pressure.");
```

```
server.on("/", HTTP_GET, [](AsyncWebServerRequest *request){  
    request->send(200, "text/plain", readSensorData().c_str());  
});
```

```
// Start server  
server.begin();  
}
```

```
String readSensorData() {  
    // Read data from your sensors  
    double data1 = particleSensors[0].getIR(); // Replace with the actual  
value from sensor 1  
    double data2 = particleSensors[1].getIR(); // Replace with the actual  
value from sensor 2  
    double data3 = particleSensors[2].getIR(); // Replace with the actual  
value from sensor 3
```

```
    // Create a string in the format "data1, data2, data3"  
    String result = String(data1, 2) + ", " + String(data2, 2) + ", " +  
String(data3);
```

```
    return result;  
}
```

```
void loop() {  
    // Add a delay to avoid overwhelming the server with requests  
    delay(10);  
    String sensorData = readSensorData();  
    server.on("/", HTTP_GET, [sensorData](AsyncWebServerRequest  
*request){  
        request->send(200, "text/plain", sensorData.c_str());
```

```
});  
Serial.println(sensorData.c_str());  
}
```