NITTE EDUCATION TRUST | NMAM INSTITUTE OF TECHNOLOGY

INTERNET OF THINGS LAB

Report

Heart Rate and SpO2 Monitor

Project Report submitted by

Amshula R Shetty (4NM21IS013)

Amulya S (4NM21IS015)

Anagha Rao (4NM21IS016)

Under the Guidance of

Dr. Jason Elroy Martis

Associate Professor

Information Science and Engineering Department

In partial fulfillment of the requirements for the award of the

Degree of

Bachelor of Engineering in Information Science and Engineering

from

Visvesvaraya Technological University, Belagavi

Department of Information Science and Engineering

NMAM Institute of Technology, Nitte - 574110

(An Autonomous Institution affiliated to VTU, Belagavi)

NOVEMBER 2023

NMAM INSTITUTE OF TECHNOLOGY

N.M.A.M. INSTITUTE OF TECHNOLOGY

(An Autonomous Institution affiliated to Visvesvaraya Technological University, Belagavi)

Nitte – 574 110, Karnataka, India

Accredited with 'A' Grade by NAAC

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the project work entitled

"Heart rate and SpO2 Monitor"

is a bonafide work carried out by

Amshula R Shetty (4NM21IS013)

Amulya S (4NM21IS015)

Anagha Rao (4NM21IS016)

in partial fulfilment of the requirements for the award of

Bachelor of Engineering Degree in Information Science and Engineering

prescribed by Visvesvaraya Technological University, Belagavi

during the year 2022-2023.

It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library.

The project report has been approved as it satisfies the academic requirements in respect of the project work prescribed for the Bachelor of Engineering Degree.

Signature of the Guide **Signature of the HOD** **Signature of the Principal**

Contents

Abstract

This report presents the design, development, and evaluation of a comprehensive monitoring system capable of measuring heart rate and oxygen saturation (SpO2) levels using the MAX30100/30102 sensor, Arduino Adafruit Metro ESP32-S2 Module, and an SSD1306 OLED display.

The project aimed to create a user-friendly and accurate monitoring system for healthcare applications. The report outlines the hardware components' integration, including sensor interfacing. It describes the software implementation, including the integration of the Blynk IoT platform, enabling real-time display and remote monitoring through the Blynk app on mobile devices. Experimental validation involved testing the system's accuracy, reliability, and response time under various conditions. Results demonstrated the system's capability to accurately measure and display real-time heart rate and SpO2 levels.

The motivation behind undertaking this project stems from the growing significance of merging technology with healthcare for improved monitoring and diagnosis. With the increasing prevalence of wearable devices, there exists a tremendous opportunity to develop a sophisticated yet accessible monitoring system capable of providing real-time data on vital health metrics. The project aspires to address the need for accurate, accessible, and efficient health monitoring systems in the contemporary digital age.

Chapter 1: Introduction

In recent years, the fusion of technology and healthcare has revolutionized medical monitoring and diagnosis. One such area of advancement is the development of wearable devices capable of measuring vital signs, such as heart rate and oxygen saturation (SpO2), in real time. These metrics serve as crucial indicators of an individual's overall health status and are integral in various medical contexts, including clinical diagnostics, fitness tracking, and wellness monitoring.

This report centers on the design and assessment of an all-encompassing monitoring system that employs the MAX30100/30102 sensor, Arduino Adafruit Metro ESP32-S2 Module, an SSD1306 OLED display, and the Blynk IoT platform for the precise measurement and display of heart rate and SpO2 levels. Renowned for its precision and effectiveness in capturing vital signs, the MAX30100/30102 sensor is seamlessly integrated with the Arduino Adafruit Metro ESP32-S2 Module for data acquisition and processing. The SSD1306 OLED display serves as the user interface, providing real-time information. Furthermore, integration with the Blynk platform facilitates remote monitoring and data visualization through the Blynk app on mobile devices, contributing to the development of a portable, cost-effective, and user-friendly monitoring system.

This report delivers a thorough exploration of both the hardware and software integration processes, highlighting the instrumental role of the Blynk platform in enabling real-time data access. Additionally, it highlights the significance of this innovative integration by emphasizing how the Blynk platform empowers users with seamless access to live updates, enhancing the system's practicality and user engagement. The inclusion of the Blynk platform not only broadens the scope of applications but also aligns with the evolving landscape of digital health solutions, fostering a new era of accessible and personalized healthcare monitoring.

Chapter 2: Description of Components

1. Adafruit Metro ESP32 S2 Module:

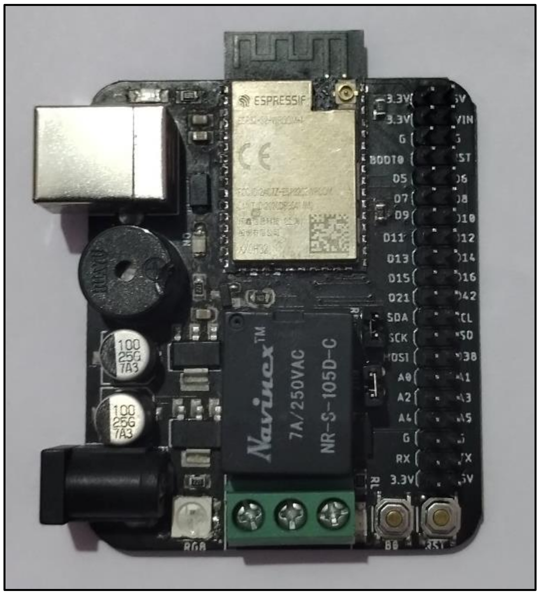


Figure 1: Arduino Adafruit Metro ESP32 S2

The Adafruit Metro ESP32 S2 Module is a development board based on the ESP32 microcontroller. It's manufactured by Adafruit and designed to offer an accessible platform for creating IoT (Internet of Things) projects and other embedded applications. This board includes features such as built-in Wi-Fi and Bluetooth connectivity, making it suitable for wireless communication and IoT-based projects. With its compatibility with the Arduino IDE, it provides a familiar programming environment for enthusiasts and developers to easily prototype and create diverse electronic projects leveraging the power and versatility of the ESP32 microcontroller. In addition to its wireless capabilities, the Arduino Adafruit Metro ESP32-S2 Module also comes equipped with a variety of I/O pins, enabling users to interface with a range of sensors, actuators, and other external devices. Its compact form factor and user-friendly design make it an ideal choice for both beginners and experienced makers alike.

2. MAX30100/30102 sensor:

Figure 2 : MAX30100/30102

The MAX30100/30102 sensor is a versatile integrated pulse oximetry and heart-rate monitoring module. It combines red and infrared LEDs with a photodetector to measure the absorption of light by the blood, allowing it to determine heart rate and blood oxygen saturation non-invasively. This sensor is commonly used in wearable health devices, fitness trackers, and medical equipment due to its accuracy, low power consumption, and small form factor. Its ability to measure vital signs makes it a valuable component in various applications focused on health monitoring and assessment.

The MAX30100 chip requires two different supply voltages: 1.8V for the IC and 3.3V for the RED and IR LEDs. So the module comes with 3.3V and 1.8V regulators. One of the most important features of the MAX30100 is its low power consumption: the MAX30100 consumes less than 600µA during measurement.

The MAX30100 works by shining both lights onto the finger or earlobe (or essentially anywhere where the skin isn't too thick, so both lights can easily penetrate the tissue) and measuring the amount of reflected light using a photodetector. This method of pulse detection through light is called Photoplethysmogram. A photoplethysmogram (PPG) is an optically obtained plethysmogram that can be used to detect blood volume changes in the microvascular bed of tissue.

The oxygenated hemoglobin (HbO2) in the arterial blood has the characteristic of absorbing IR light. The redder the blood (the higher the hemoglobin), the more IR light is absorbed. As the blood is pumped through the finger with each heartbeat, the amount of reflected light changes, creating a changing waveform at the output of the photodetector. Pulse oximetry is based on the principle that the amount of RED and IR light absorbed varies depending on the amount of oxygen in your blood.

3. SSD1306 OLED display:

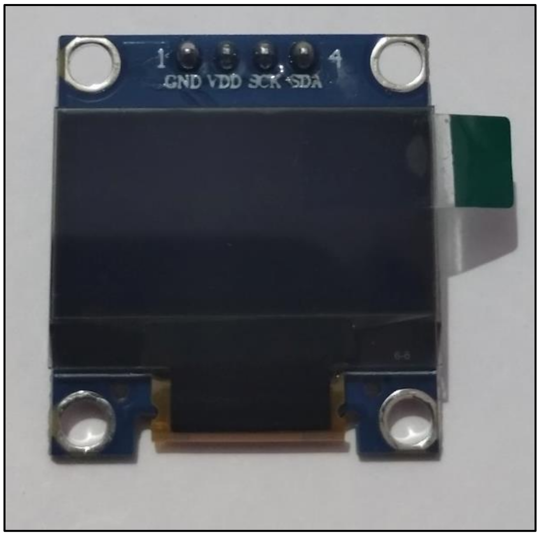


Figure 3: SSD1306 OLED front

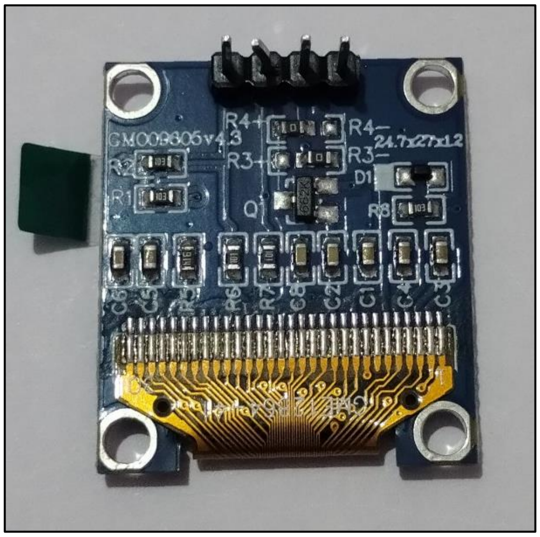


Figure 4:SSD1306 OLED back

The SSD1306 OLED display is a compact, high-contrast screen known for its clarity and efficiency. Utilizing organic light-emitting diodes (OLEDs), it offers vivid, sharp visuals and operates without a backlight, enabling it to consume less power compared to traditional LCDs. The oled used here is a monocolor, 0.96-inch display with 128×64 pixels.This display is popular in electronics projects for its crisp readability and compatibility with a wide range of microcontrollers, making it ideal for showcasing data, graphics, and text in diverse applications from wearable devices to IoT projects.

4. Jumper Wire:

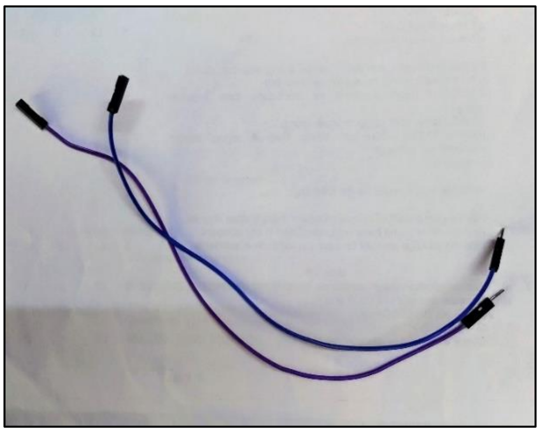


Figure 5: Male to Female Jumper wires

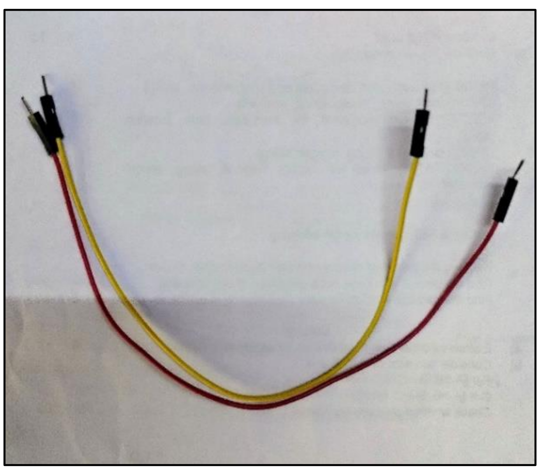


Figure 6 : Male to Male Jumper wires

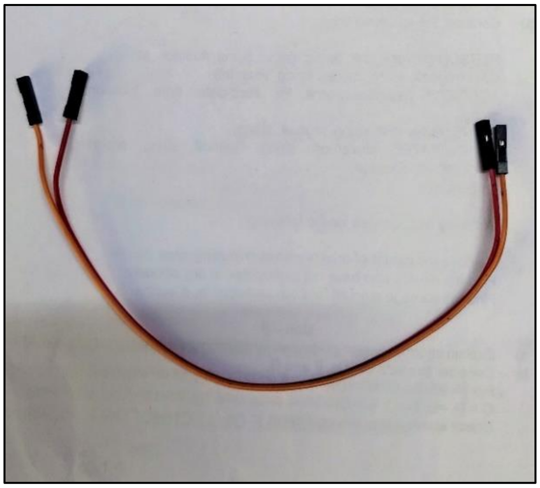


Figure 7: Female to Female Jumper wires

Jumper wires are flexible cables with connectors at both ends, typically used to establish electrical connections on a breadboard or between various electronic components. They come in different lengths, colors, and connector types (such as male-to-male, male-to-female, and female-to-female) to suit different circuit arrangements. These wires facilitate easy prototyping and circuit building in electronics projects by allowing quick and temporary connections without the need for soldering, enabling rapid experimentation and testing of electronic designs.

5. Blynk Application

The Blynk app is a versatile mobile application designed for IoT (Internet of Things) projects, offering a user-friendly interface for controlling and monitoring connected devices. In the context of a heart and SpO2 (blood oxygen saturation) project, Blynk supports widgets such as Superchart and Gauge to visualize and display vital health metrics. The Superchart widget allows users to graphically represent real-time data over a specific time period, making it ideal for tracking the dynamic fluctuations of heart rate and SpO2 levels. On the other hand, the Gauge widget provides a visual representation of numerical data, enabling users to view and assess health parameters in a concise and easily understandable format. These widgets not only enhance the user experience by presenting health data intuitively but also empower developers to create comprehensive and visually appealing interfaces for health monitoring applications using the Blynk platform. Moreover, the Superchart widget in the Blynk app offers customization options, allowing users to adjust the time range, scale, and appearance of the charts to meet their specific project requirements. This flexibility is particularly valuable in health monitoring applications, where users may want to analyze trends and patterns in heart rate and SpO2 data over different time intervals. Additionally, the Gauge widget can be configured to display threshold values, providing instant visual cues if vital signs fall outside the desired range. This feature is crucial for early detection of potential health issues.

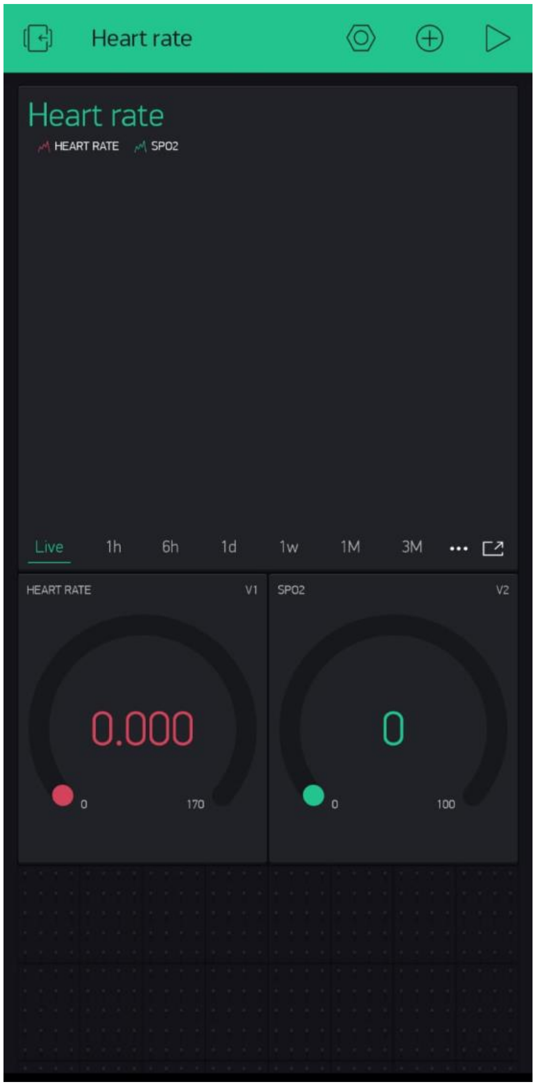
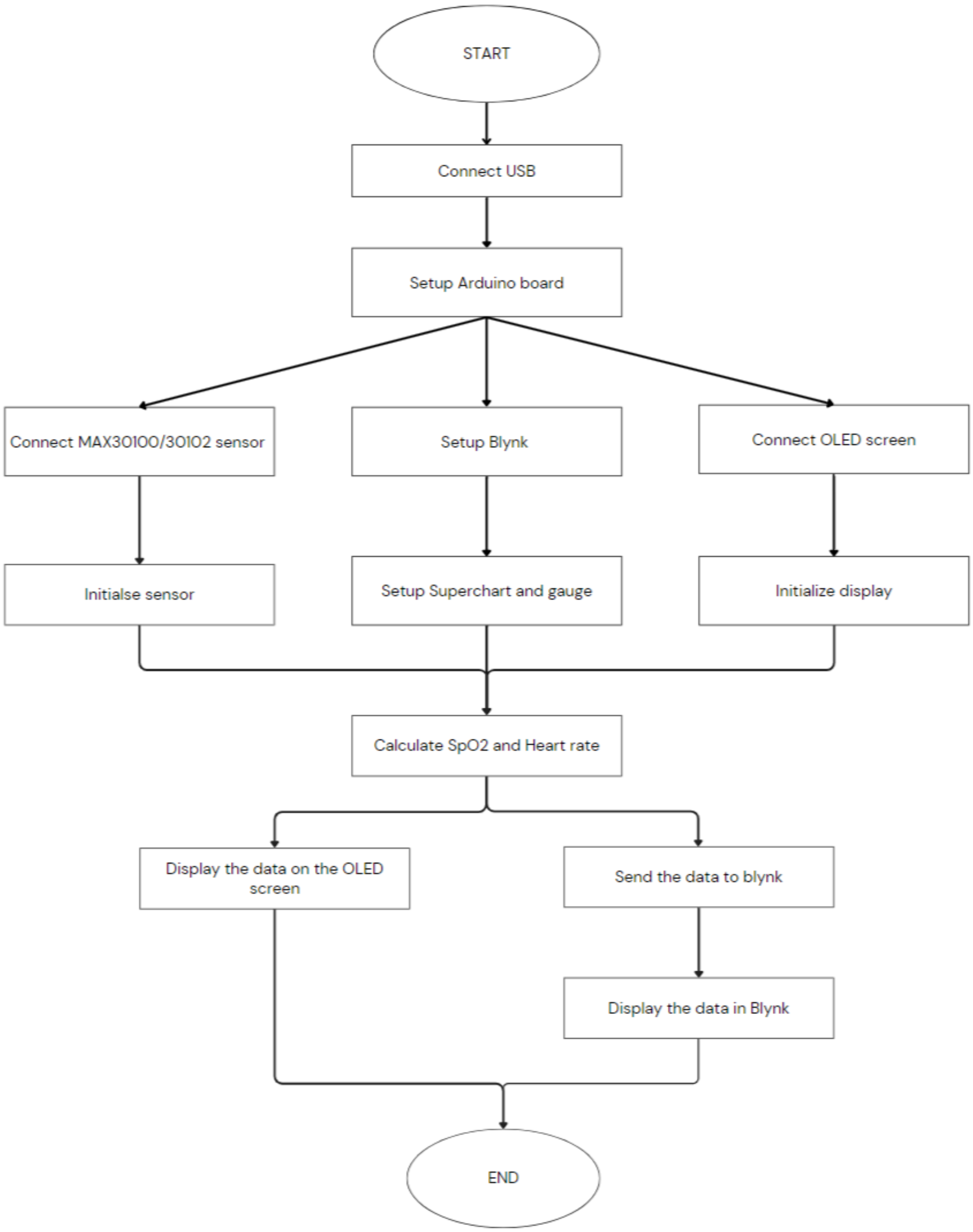


Figure 8: Blynk App Interface

Chapter 3: Flow Chart



To initiate the project, begin by connecting the Arduino via USB to both the Arduino board and the CPU, using the Arduino IDE—a platform tailored for coding IoT projects. Establish a connection with the MAX30100/30102 sensor and integrate it with the Arduino, along with an OLED screen. Additionally, connect the Arduino to the Blynk server and upload the respective codes.

As you delve into the coding process, ensure to initialize the sensor setup, incorporate a superchart and gauge in the Blynk app, and set up the display for the OLED screen.the sensor calculates both SPO2 and heart rate. the outcomes will be displayed not only on the OLED screen but also the real-time data being transmitted with live readings on the Blynk app.

Chapter 4: Circuit Diagram

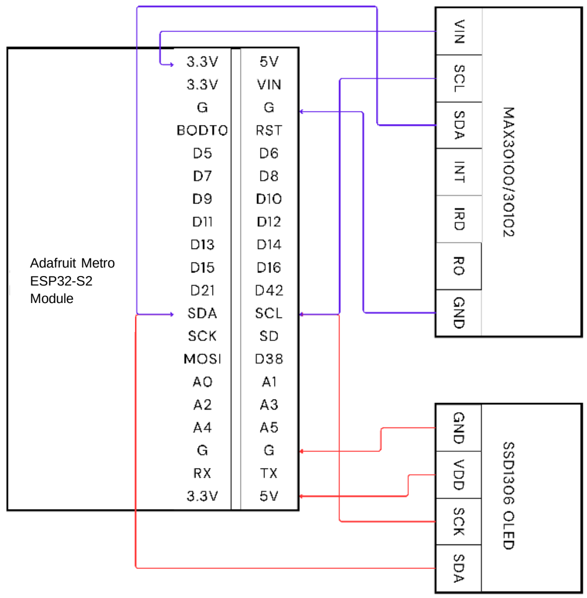


Figure 9: Circuit Diagram

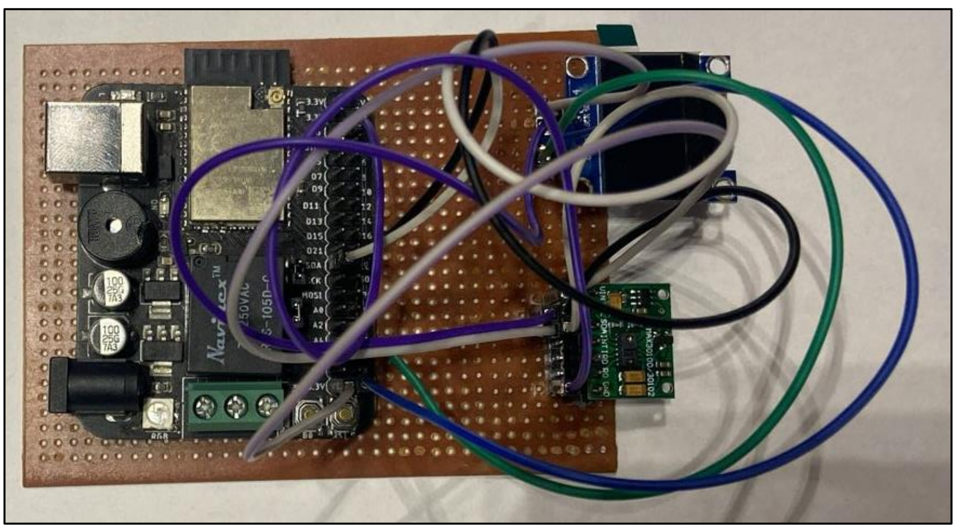


Figure 10a: Overall Circuit Design

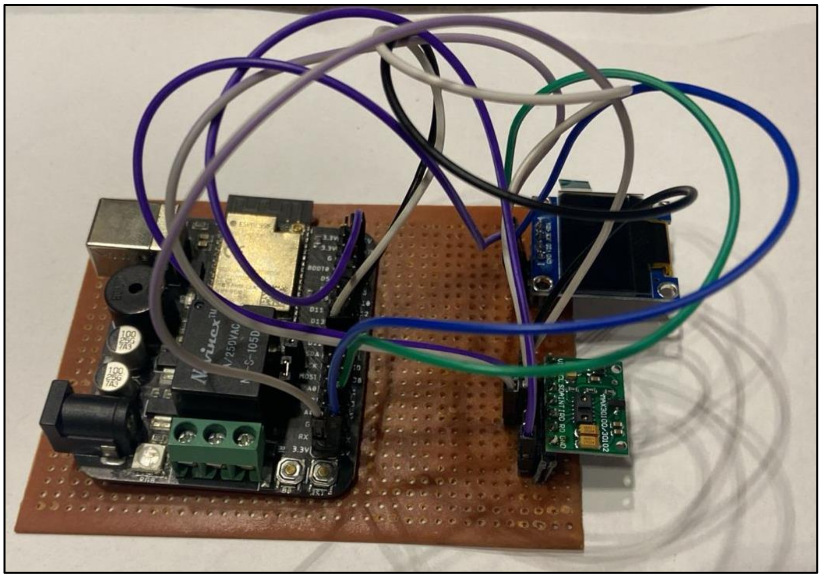


Figure 10b: Overall Circuit Design

Connect the Arduino to the MAX30100 by linking VIN to 3.3V, SCL to SCL, SDA to SDA, and GND to GND.

For the SSD1306 OLED screen, connect GND to GND, VDD to 5V, SCK to SCL, and SDA to SDA.

Importantly, individually short SCL and SDA from the Arduino side. This ensures proper communication pathways without interconnection. This straightforward arrangement facilitates seamless interaction among the components, establishing a reliable circuit for effective functionality.

Chapter 5: Code

This Arduino code integrates the Blynk IoT platform with a pulse oximeter (MAX30100) and OLED display (SSD1306) on an ESP32, enabling real-time monitoring of heart rate and SpO2 levels, with data visualization through Serial communication, Blynk app, and an onboard display.

```
#include <BlynkSimpleEsp32.h>

#include <Wire.h>

#include "MAX30100_PulseOximeter.h"

#include <Adafruit_GFX.h>

#include <Adafruit_SSD1306.h>

#include <WiFi.h>

// Define reporting period and screen dimensions

#define REPORTING_PERIOD_MS 1000

#define SCREEN_WIDTH 128

#define SCREEN_HEIGHT 32

#define OLED_RESET -1

#define SCREEN_ADDRESS 0x3C

// Initialize OLED display with specified dimensions

Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

// Initialize PulseOximeter object
```

```
PulseOximeter pox;

// Variables to store heart rate and SpO2 readings

float BPM, SpO2;

// Timestamp for the last data report

uint32_t tsLastReport = 0;

// Blynk authentication token and Wi-Fi credentials

char auth[] = "3aZk_EPSKybgsNBSYF2R6ET3qTJbF9LH";

char ssid[] = "Realme";

char pass[] = "hi22me03";

// Callback function for beat detection

void onBeatDetected()

{

   Serial.println("Beat Detected!");

}

// Setup function runs once at the beginning

void setup()

{

   Serial.begin(115200);

   // Initialize Blynk with authentication token and Wi-Fi credentials

   Blynk.begin(auth, ssid, pass, IPAddress(117, 236, 190, 213), 8080);
```

```
// Initialize Wire library for I2C communication
Wire.begin();
// Initialize OLED display
display.begin();
display.clearDisplay();
display.setTextSize(1);
display.setTextColor(1);
display.setCursor(0, 0);
display.println("Initializing pulse oximeter..");
display.display();
Serial.print("Initializing pulse oximeter..");
// Check if PulseOximeter initialization is successful
if (!pox.begin())
{
   Serial.println("FAILED");
   display.clearDisplay();
   display.setTextSize(1);
   display.setTextColor(1);
   display.setCursor(0, 0);
   display.println("FAILED");
```

```
        display.display();
        for (;;);
    }
    else
    {
        display.clearDisplay();
        display.setTextSize(1);
        display.setTextColor(1);
        display.setCursor(0, 0);
        display.println("SUCCESS");
        display.display();
        Serial.println("SUCCESS");
    }
    // Set callback for beat detection
    pox.setOnBeatDetectedCallback(onBeatDetected);
}
// Loop function runs repeatedly
void loop()
{
    // Update PulseOximeter data
```

```
pox.update();

// Run Blynk

Blynk.run();

// Get heart rate and SpO2 readings

BPM = pox.getHeartRate();

SpO2 = pox.getSpO2();

// Check if it's time to report data

if (millis() - tsLastReport > REPORTING_PERIOD_MS)

{

   // Print data to Serial and update Blynk virtual pins

   Serial.print("Heart rate:");

   Serial.print(BPM);

   Serial.print("SpO2:");

   Serial.print(SpO2);

   Serial.println(" %");

   Blynk.virtualWrite(V1, BPM);

   Blynk.virtualWrite(V2, SpO2);

   // Update OLED display with heart rate and SpO2

   display.clearDisplay();

   display.setTextSize(1);
```

```
        display.setTextColor(1);

        display.setCursor(0, 8);

        display.print("BPM: ");

        display.print(BPM);

        display.setTextSize(1);

        display.setTextColor(1);

        display.setCursor(0, 24);

        display.print("SpO2: ");

        display.print(SpO2);

        display.display();

        // Update timestamp for the last report

        tsLastReport = millis();

    }

}
```

This code is for an ESP32-based project using the MAX30100 Pulse Oximeter sensor to measure heart rate (BPM) and blood oxygen saturation (SpO2). Here's a breakdown:

- Libraries: It includes various libraries required for the project, such as Blynk for IoT connectivity, libraries for the MAX30100 sensor, OLED display, Wire library for I2C communication, and Wi-Fi for network connectivity.
- Initialization: The setup() function initializes serial communication for debugging, connects to a Wi-Fi network using the provided credentials, and initializes the

OLED display and the MAX30100 sensor. It displays initialization status on the OLED screen.

- Callbacks: It sets a callback function ‘onBeatDetected()’ to be executed when a heartbeat is detected by the MAX30100 sensor.
- Main Loop: The main loop continuously updates the MAX30100 sensor readings using ‘pox.update()’ and runs the Blynk functions ‘Blynk.run()’ for IoT communication. It retrieves the BPM and SpO2 values from the sensor and reports them to both the serial monitor and the Blynk app every one second. Additionally, it updates and displays these values on the OLED screen.
- Reporting: When the reporting period (1 second in this case) elapses, it collects BPM and SpO2 values, sends them to Blynk using ‘Blynk.virtualWrite()’ to update widgets (V1 and V2), and displays these values on the OLED screen.

Overall, this code sets up the ESP32 to read data from the MAX30100 sensor, transmit it to the Blynk app for monitoring, and display real-time values on the OLED screen.

Chapter 6: Result Snapshot

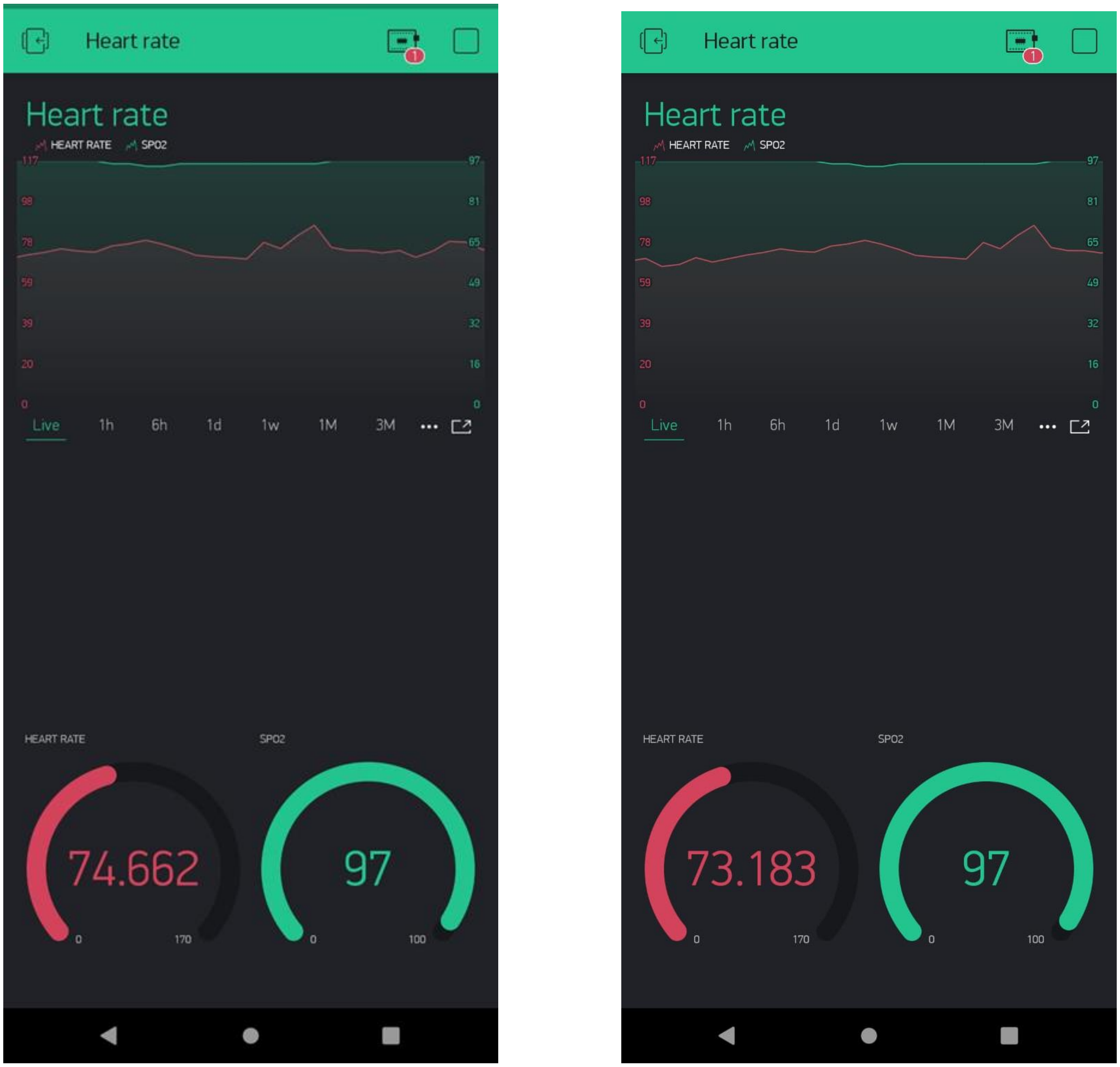


Figure 11: Live Heartrate and SpO2 in Blynk App

Chapter 7: Conclusion

Creating a heart rate and SpO2 monitor with the Arduino Adafruit Metro ESP32-S2 Module, MAX30100/30102 sensor, and SSD1306 OLED display, and then connecting it with the Blynk server, results in a versatile health monitoring system.

This hardware configuration enables seamless wireless communication, accurate biometric data collecting, and immediate visual feedback via the OLED display. The connectivity with the Blynk server allows for remote monitoring and data visualization on a smartphone or computer, improving accessibility and allowing for continuous health tracking. This comprehensive solution meets a wide range of user requirements, from personal health management to telemedicine applications, by providing a dependable and user-friendly platform for real-time health parameter monitoring and analysis.

Future iterations may incorporate additional sensors, utilize machine learning for enhanced accuracy, and expand the user interface for detailed analytics. Collaborations with healthcare professionals and considerations for scalability could refine the system. Continuous research and development efforts aim to evolve this health monitoring system into a dynamic solution for diverse healthcare needs, ensuring compatibility with emerging technologies and meeting the evolving standards of healthcare

References

1. Google, Interfacing MAX30100 Pulse Oximeter and Heart Rate Sensor with Arduino:

   https://lastminuteengineers.com/max30100-pulse-oximeter-heart-rate-sensor-arduino-tutorial/

2. Google, Guide for I2C OLED Display with Arduino:

   https://randomnerdtutorials.com/guide-for-oled-display-with-arduino/

3. Google, Blood Oxygen & Heart Rate Monitor with MAX30100 & Arduino:

   https://how2electronics.com/blood-oxygen-heart-rate-monitor-max30100-arduino/