

Biomedical Monitoring System - Code Implementation

1. Introduction

This report details the implementation of the **Biomedical Monitoring System** using **ESP32**, **Arduino IDE**, and **Blynk IoT App**. The system continuously monitors **body temperature, heart rate, blood oxygen levels (SpO2), and humidity**. Data is transmitted via Wi-Fi and displayed on the **Blynk** app for real-time remote monitoring.

2. Development Environment & Tools

2.1 Software & Tools

- **Arduino IDE** – Used for programming the ESP32.
- **Blynk IoT Platform** – Cloud-based application for remote data visualization.
- **ESP32 Board Package** – Required for ESP32 support in Arduino IDE.
- **Libraries:**
 - **WiFi.h** – Enables Wi-Fi communication.
 - **WebServer.h** – Allows web-based access to sensor data.
 - **Wire.h** – For I2C communication with sensors.
 - **MAX30100_PulseOximeter.h** – Reads heart rate and SpO2 levels.
 - **OneWire.h & DallasTemperature.h** – For DS18B20 temperature sensor.
 - **DHT.h** – Reads humidity and temperature.
 - **BlynkSimpleEsp32.h** – For Blynk IoT integration.

2.2 Blynk IoT App Overview

Blynk is an **IoT platform** that allows users to monitor and control IoT devices remotely. It provides:

- **Cloud Connectivity** – Secure data transmission and storage.
- **Mobile Dashboard** – Real-time visualization via the Blynk app.
- **Virtual Pins** – Easy communication between ESP32 and the app.
- **Widgets** – Customizable UI for displaying sensor data.
- **Event Notifications** – Alerts for abnormal sensor readings.

2.3 Hardware Components

- **ESP32 Microcontroller** – Wi-Fi-enabled processing unit.

- **MAX30100 Pulse Oximeter Sensor** – Measures heart rate and SpO2.
- **DS18B20 Temperature Sensor** – Reads body temperature.
- **DHT11 Sensor** – Monitors ambient temperature and humidity.
- **Wi-Fi Network** – Enables data transmission to Blynk cloud.

3. Setting Up the Development Environment

3.1 Installing Required Libraries

1. Open **Arduino IDE**.
2. Navigate to **Sketch > Include Library > Manage Libraries**.
3. Install the following libraries:
 - **Blynk** (BlynkSimpleEsp32)
 - **Pulse Oximeter** (MAX30100_PulseOximeter)
 - **Dallas Temperature** (DallasTemperature)
 - **DHT Sensor Library** (DHT.h)

3.2 Connecting ESP32 to Blynk

1. Install the **Blynk IoT app** from Google Play or App Store.
2. Create a new project and select **ESP32** as the device.
3. Add widgets to display heart rate, SpO2, temperature, and humidity.
4. Obtain the **Blynk Authentication Token** from the app.
5. Replace BLYNK_AUTH_TOKEN in the code with the obtained token.

4. Setting Up Sensors

4.1 MAX30100 Pulse Oximeter Setup

- Connect **VCC** to **3.3V** and **GND** to **GND**.
- Connect **SCL** to **GPIO22** and **SDA** to **GPIO21**.
- Ensure **finger placement** on the sensor for accurate readings.

4.2 DS18B20 Body Temperature Sensor Setup

- Connect **VCC** to **3.3V** and **GND** to **GND**.
- Connect **Data Pin** to **GPIO5**.
- Use a **4.7kΩ pull-up resistor** between **VCC** and **Data Pin**.

4.3 DHT11 Temperature & Humidity Sensor Setup

- Connect **VCC** to **3.3V** and **GND** to **GND**.
- Connect **Data Pin** to **GPIO18**.

4.4 Wi-Fi and Blynk Integration

- Ensure ESP32 is connected to Wi-Fi before transmitting data.
- Link virtual pins in the Blynk app to corresponding sensor readings.

5. Code Implementation

The ESP32 code is written in C++ and is uploaded using the **Arduino IDE**. The program reads data from sensors, processes the readings, and transmits the information to the **Blynk IoT App** over Wi-Fi. The code includes:

- **Wi-Fi Connection Handling** – Ensures ESP32 connects to the network.
- **Sensor Data Acquisition** – Reads values from **MAX30100**, **DS18B20**, and **DHT11** sensors.
- **Blynk Integration** – Sends real-time data to the Blynk Cloud using virtual pins.
- **Serial Monitoring** – Prints sensor readings for debugging.

The complete code is structured to run continuously in the `loop()` function, updating sensor values every second and sending data to the Blynk app.

6. Uploading and Running the Code

1. **Open Arduino IDE** and paste the ESP32 sketch.
2. **Select Board:** Go to **Tools > Board > ESP32 Dev Module**.
3. **Set COM Port:** Go to **Tools > Port > Select the correct port**.
4. **Install Required Libraries** if not installed previously.
5. **Upload the Code** by clicking the **Upload** button.
6. **Monitor Serial Output:** Open **Tools > Serial Monitor** and set baud rate to **115200**.
7. **Open Blynk App:** Ensure the ESP32 appears online and is transmitting sensor data.

Once the upload is complete, the ESP32 will start transmitting real-time sensor data to the Blynk Cloud, making it accessible through the mobile app.

7. Troubleshooting & Debugging

- **Wi-Fi Connection Issues:**

- Verify **SSID and password** in the code.
- Ensure the **Wi-Fi network is stable**.
- Check if the **ESP32 is in range of the router**.
- **Blynk Not Updating:**
 - Ensure **Blynk Authentication Token** is correct.
 - Check if **ESP32 is online in the Blynk app**.
 - Restart the ESP32 if data is not appearing.
- **Sensor Readings are NaN or Zero:**
 - Verify **sensor connections and wiring**.
 - Ensure **proper power supply** to sensors.
 - Check **library dependencies** for compatibility.
- **Code Upload Failure:**
 - Ensure the **correct ESP32 board is selected**.
 - Check if another application is using the **serial port**.
 - Press and hold the **BOOT button** on ESP32 while uploading.

8. Conclusion

This project successfully integrates **ESP32, Blynk IoT, and biomedical sensors** to create a real-time **health monitoring system**. The data is displayed remotely via **Blynk Cloud**, allowing continuous tracking of patient vitals. Future enhancements may include **cloud data storage, SMS alerts, and AI-based anomaly detection**.

9. Resources & References

- **Blynk Documentation:** <https://docs.blynk.io>
- **ESP32 Technical Documentation:** <https://docs.espressif.com/projects/esp-idf/en/latest/>
- **Arduino Libraries:** <https://www.arduino.cc/en/Reference/Libraries>
- **MAX30100 Pulse Oximeter Guide:** <https://lastminuteengineers.com/max30100-pulse-oximeter-arduino-tutorial/>
- **DS18B20 Temperature Sensor Guide:** <https://www.analog.com/en/parametricsearch/11094#/>

- **DHT11 Sensor Datasheet:** <https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT11.pdf>
- **Blynk Cloud API Reference:** <https://blynkapi.docs.apiary.io/>