

Heart Rate Monitoring and Emergency Alert System Using Iot

Mrs. Chethan Kumar T¹, Devaraj V M², Akash L³, Thimmegowda.G⁴, Dhananjay M M⁵

Department Of Ise Kalpataru Institute Of Technology Tumkur, India Dhanudhananjaya1024@Gmail.Com

ABSTRACT- The increasing need for continuous health monitoring has encouraged the development of Internet of Things (IoT)-based healthcare systems. This paper presents an IoT-based heart rate monitoring and alert system prototype using an ESP32 microcontroller and a MAX30102 optical sensor. The proposed system estimates heart rate and peripheral oxygen saturation (SpO₂) from sensor signals and displays the readings locally through an LCD or serial monitor. The ESP32 supports wireless data transmission over Wi-Fi to a Python Flask server, where readings can be stored in an SQLite database and accessed through a web dashboard. A configurable threshold-based mechanism is proposed to flag readings that fall outside predefined limits for further observation. The system aims to provide an affordable and modular approach to collecting, displaying, and reviewing physiological measurements. Its practical performance depends on sensor placement, motion, signal quality, network connectivity, and validation of the measurements. The prototype is intended for educational and preliminary monitoring purposes and is not designed to diagnose heart attacks or replace professional medical assessment.

KEYWORDS- Internet of Things (IoT), Heart Rate Monitoring, ESP32, MAX30102, SpO₂, Wireless Health Monitoring, Flask, SQLite, Emergency Alert System, Remote Patient Monitoring.

I. INTRODUCTION

Continuous monitoring of physiological parameters can help individuals observe changes in their health status. Heart rate is an important physiological measurement, while peripheral oxygen saturation provides an estimate of the oxygen saturation of blood. Conventional spot-check measurements provide readings at particular moments and may not offer continuous monitoring, centralized data storage, or convenient remote access.

The Internet of Things provides a way to connect sensors and embedded devices to software platforms through wireless communication. By integrating sensing hardware with web

technologies, physiological readings can be transmitted, stored, and reviewed remotely. This paper presents a prototype heart rate monitoring system based on the ESP32 microcontroller and MAX30102 optical sensor.

The sensor collects pulse-related optical signals from which heart rate and SpO₂ estimates can be obtained. The ESP32 processes the sensor data and displays the measurements locally. Through Wi-Fi connectivity, the

system can transmit readings to a Python Flask API for storage in an SQLite database and presentation on a web dashboard.

The system also incorporates configurable threshold-based alert logic to identify readings that may require rechecking. The principal objective is to demonstrate a low-cost, modular IoT monitoring architecture that combines physiological sensing, wireless communication, data storage, and remote visualization. The system is an educational prototype and requires independent validation before use in any clinical setting.

The main contributions of this project are:

- Integration of the MAX30102 sensor with an ESP32 for heart rate and SpO₂ estimation.
- Local visualization of physiological readings.
- An IoT communication approach for transmitting data to a server.
- A proposed database and dashboard for maintaining and reviewing timestamped readings.
- Configurable threshold-based flagging to support preliminary monitoring.

II. RELATED WORK

Existing technologies for physiological monitoring provide the foundation for the proposed IoT-based system. The project presentation identifies sensor documentation, embedded-controller documentation, communication libraries, and web-storage technologies as the principal sources relevant to its design.

1. Optical Heart Rate And SpO_2 Sensing

The MAX30102 is an integrated optical sensor designed for heart rate monitoring and pulse oximetry. It uses optical components to collect signals associated with blood-volume changes and communicates with a microcontroller through an I²C-compatible interface. Its suitability for compact monitoring devices makes it relevant to embedded healthcare prototypes. However, measurement quality can be affected by sensor placement, movement, and signal quality.

2. Esp32-Based Iot Monitoring

The ESP32 provides embedded processing and wireless connectivity suitable for IoT applications. In the proposed system, it interfaces with the MAX30102, obtains sensor estimates, displays readings, and supports transmission to a remote server. The use of a single microcontroller can reduce hardware complexity, although operation depends on power availability and network connectivity.

3. Web-Based Data Collection And Storage

A web API can provide an interface between the embedded device and the monitoring dashboard. Flask is a lightweight Python web framework suitable for building such an API, while SQLite can be used to maintain timestamped records locally. These technologies support a modular design in which data collection, storage, and visualization are handled by separate software components.

III. RESEARCH GAP

The project addresses the practical integration of sensing, local display, wireless communication, data storage, and threshold-based flagging in a single prototype. Compared with a standalone device that provides only local readings, this architecture is designed to support historical data review and remote access. Nevertheless, the project presentation does not establish clinical accuracy, comparative performance, or

improved patient outcomes. These aspects require further experimentation and validation.

IV. METHODOLOGY

The methodology consists of hardware integration, sensor-data acquisition, local visualization, wireless transmission, server-side processing, database storage, and threshold-based monitoring.

1. Hardware Components

- ESP32 microcontroller: Processes sensor readings and provides Wi-Fi connectivity.
- MAX30102 sensor: Acquires optical signals used to estimate heart rate and SpO_2 .
- I²C LCD or serial monitor: Displays the latest measurements locally.
- Power supply: Provides the required operating power to the hardware.

2. Sensor Interfacing And Data Acquisition

The MAX30102 sensor is connected to the ESP32 through the I²C interface. After initialization, the controller checks sensor availability and reads the data required by the selected sensor library. The library processes the optical signal and provides estimated heart rate and SpO_2 values. Reliable readings require appropriate sensor placement and sufficient signal quality. Measurements affected by movement or poor contact should not automatically be treated as valid physiological readings.

3. Local Display

The latest estimated heart rate and SpO_2 values are presented through an I²C LCD or the serial monitor. This provides a local means of observing the measurements independently of the web interface.

4. Iot Communication

The ESP32 connects to a Wi-Fi network and sends the collected readings to a Python Flask API. The transmitted data can include measurement values and timestamps. Network failures or server unavailability may interrupt remote monitoring, so communication status should be considered during system evaluation.

5. Data Storage And Dashboard

The Flask server receives incoming measurements and can store them in an SQLite database. Timestamped

records enable the dashboard to display current readings and historical observations. This separation of acquisition, transmission, storage, and presentation allows the components to be developed and tested independently.

6. Alert Mechanism

A configurable threshold-based mechanism is proposed to flag measurements that fall outside selected demonstration limits. To reduce unnecessary flags, a practical implementation should check signal validity and consider whether an abnormal value persists across multiple readings. A flag indicates that a reading should be rechecked; it is not evidence of a heart attack or a confirmed medical emergency.

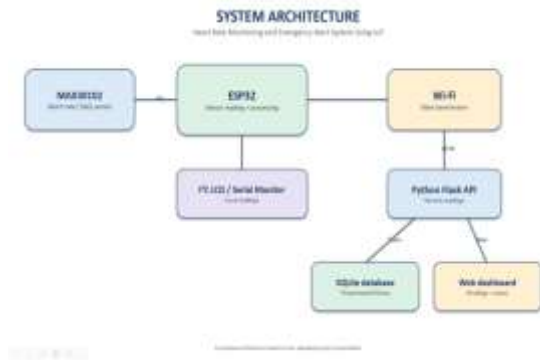


FIG 1 System Architecture

V. RESULTS AND DISCUSSION

Time	Heart Rate (bpm)	SpO2 (%)	Status
14:32:10	72	98	[Normal]
14:32:15	72	98	[Normal]
14:32:20	71	98	[Normal]
14:32:25	72	98	[Normal]
14:32:30	73	97	[Normal]
14:32:35	72	98	[Normal]
14:32:40	74	97	[Normal]
14:32:45	72	98	[Normal]

FIG 2:Serial MONITOR OUTPUT

The displayed output shows heart rate estimates of approximately 71–74 bpm and SpO₂ estimates of 97–98%. The software labels these readings as [Normal], indicating that they fall within the system's configured acceptable limits. The changing IR values demonstrate that the sensor is providing varying infrared measurements to the processing algorithm.

However, these displayed values alone do not establish measurement accuracy. Further testing against validated reference devices would be needed to assess the accuracy and reliability of the system.



FIG 3:DASHBOARD

The figure shows the Heart Rate Monitoring Web Dashboard, which displays real-time health measurements collected by the IoT-based monitoring system. It shows a heart rate of 72 bpm and an SpO₂ level of 98%, with the status indicated as "Normal" according to the system's configured thresholds. The latest readings table presents timestamped heart rate, SpO₂, and status information, while the trend graph visualizes changes in these parameters over time. The dashboard provides a user-friendly interface for remotely monitoring measurements, reviewing historical data, and observing health parameter trends through a web browser.



FIG 4 : LCD OUTPUT

The image shows the output displayed on a 16×2 LCD (Liquid Crystal Display) used in the IoT-based heart rate monitoring system. The LCD displays two important physiological parameters: heart rate and peripheral oxygen saturation (SpO₂).

VI. DISCUSSION

The proposed architecture combines sensing and embedded processing with wireless communication and server-side data management. Local visualization allows users to inspect measurements without depending entirely on the network, while the server and database components are intended to support remote review and historical analysis.

The MAX30102 provides a compact sensing solution, and the ESP32 offers the processing and Wi-Fi capabilities required for this design. The Flask API and SQLite database provide a relatively simple software architecture for receiving and

Several limitations must be considered. Optical readings can be affected by motion, poor finger placement, and signal quality. Network interruptions can prevent timely server updates. Threshold-based alerts can also produce false positives or fail to identify a problem if the readings are inaccurate or the thresholds are inappropriate.

A complete experimental evaluation should measure heart rate and SpO₂ error against suitable reference instruments, successful data transmission rate, end-to-end latency, database reliability, and alert response time. Tests should be repeated under different operating conditions, and the results should be reported using actual observations rather than assumed performance values.

VII. CONCLUSION

This paper presents an IoT-based heart rate monitoring and alert system prototype using an ESP32 microcontroller and MAX30102 optical sensor. The design combines physiological signal acquisition, heart rate and SpO₂ estimation, local display, Wi-Fi communication, server-side data handling, and timestamped storage. A web dashboard and configurable threshold-based flagging mechanism are proposed to facilitate remote observation and review of measurements.

The architecture demonstrates how embedded hardware and web technologies can be integrated into a modular monitoring application. Its usefulness depends on reliable sensor readings, stable network communication, appropriate data handling, and validation against reference instruments. Future work

includes completing and testing the remote dashboard and alert delivery, evaluating measurement accuracy, improving data security, and conducting systematic performance experiments.

The prototype is intended for educational and preliminary monitoring purposes only. It is not a clinically validated medical device and cannot diagnose a heart attack or replace professional medical assessment.

REFERENCES

1. Analog Devices, "MAX30102: High-Sensitivity Pulse Oximeter and Heart-Rate Sensor for Wearable Health." Publication: Technical Datasheet, Analog Devices, Inc.
1. Espressif Systems, "ESP32 Series Datasheet." Publication: Technical Datasheet, Espressif Systems.
2. Espressif Systems, "Arduino-ESP32 Programming Documentation." Publication: Official Technical Documentation, Espressif Systems.
3. SparkFun Electronics, "SparkFun MAX3010x Pulse and Proximity Sensor Library." Publication: Open-Source Software Library Documentation, SparkFun Electronics.
4. Pallets Projects, "Flask Documentation." Publication: Flask Web Framework Documentation.
5. Python Software Foundation, "SQLite Database Interface Documentation." Publication: Python Standard Library Documentation.
6. .Azzola, F., "ESP32-Powered Heart Rate and Pulse Oximetry Monitoring with MAX30102." Publication: Espressif Systems Technical Article, 2021.
7. Khaleel Khan Mohammed. "A Survey on Digital Health Care Data Analysis Techniques forDeveloping Machine Learning Models