

Remote Patient Health Monitoring System

Bhaskar Jyoti Das,
M.Tech in Computer Technology,
2023EET2180

Abstract— The Real-Time Remote Patient Health Monitoring System is an IoT-based solution that enables healthcare professionals to monitor patients' health records remotely and in real-time. This system uses smart IoT devices, including an esp8266 microcontroller and a pulse sensor, to record several health parameters such as heart rate, temperature, and SpO2. The system also includes a user-friendly Android application developed using Kotlin, which allows patients and healthcare providers to efficiently analyze and access comprehensive health records. Additionally, the system features an emergency call system that enables patients to inform healthcare providers of critical situations. This project aims to establish a reliable and efficient remote patient monitoring system that enhances healthcare delivery and improves patient outcomes.

Index Terms— IoT, ESP8266, Android, Firebase

I. INTRODUCTION

The healthcare industry is experiencing significant challenges due to the increasing patient numbers, which strain doctors' ability to conduct periodic in-person health inspections. Analyzing and keeping track of patients' health records for assessing their health improvement is crucial for healthcare professionals. Delayed treatment may result from the lack of timely and accurate collection of health data. Moreover, treating patients with contagious diseases remotely is necessary to avoid frequent in-person visits. To address these challenges, this project proposes a Real-Time Remote Patient Health Monitoring System using esp8266 microcontroller, pulse sensor, Firebase, Android, and Kotlin. The system aims to establish a reliable and efficient remote patient monitoring system that enhances healthcare delivery and improves patient outcomes. The esp8266 microcontroller is a low-cost Wi-Fi microchip with full TCP/IP stack and microcontroller capability, making it an ideal choice for IoT-based applications. The pulse sensor measures the patient's heart rate, and the Firebase cloud platform provides scalable solutions for data storage, analysis, and accessibility. The Android application, developed using Kotlin, enables efficient health record analysis and accessibility, while the emergency call system allows patients to inform healthcare providers of critical situations. The project also incorporates I2C communication for sensor data transmission, adding a layer of efficiency and accuracy to the remote patient monitoring system. I2C (Inter-Integrated Circuit) is a communication protocol that facilitates digital communication between microcontrollers and peripheral devices like sensors.

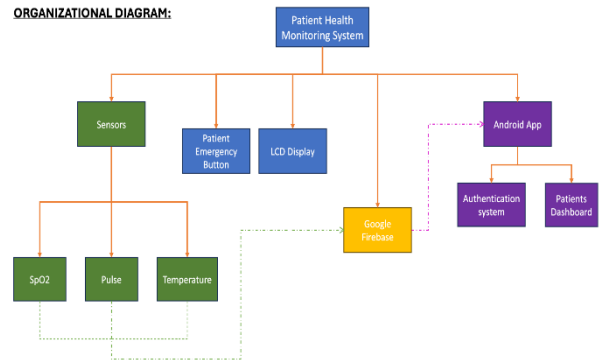
II. LITERATURE REVIEW

Several studies have explored the integration of IoT devices in healthcare applications. For instance, a study by Ahmed et al. (2023) developed an IoT-based remote monitoring system for patients with chronic diseases. The system collected health data such as blood pressure, glucose

levels, and heart rate and transmitted the data to a cloud server for analysis. The results showed that the system improved patients' quality of life and reduced hospital readmissions. Similarly, a study by Khairnar et al. (2019) developed an IoT-based remote monitoring system for patients with cardiovascular diseases. The system collected health data such as ECG, heart rate, and blood pressure and transmitted the data to a healthcare provider's dashboard. The results showed that the system improved patients' health outcomes and reduced healthcare costs. In terms of mobile applications, a study by Sarker et al. (2023) developed a mobile application for remote monitoring of patients with diabetes. The application collected health data such as blood glucose levels, physical activity, and medication intake and provided personalized recommendations to patients. The results showed that the application improved patients' glycemic control and quality of life.

III. WORKFLOW

ORGANIZATIONAL DIAGRAM:



The Patient Health Monitoring System is a comprehensive solution for monitoring a patient's health in real-time. The system consists of various components that work together to collect, transmit, store, and display health data. The system uses sensors to collect health data from the patient, including SpO2, pulse rate, and temperature. The collected data is then transmitted to an ESP8266 MCU (Microcontroller Unit) for processing and transmission to the Google Firebase cloud platform. The ESP8266 MCU is a low-cost, low-power Wi-Fi microchip that is capable of transmitting data over Wi-Fi networks. The MCU receives data from the sensors and processes it before transmitting it to the cloud for storage and analysis. The Google Firebase cloud platform provides scalable solutions for data storage, analysis, and accessibility. The health data collected by the sensors and transmitted via the ESP8266 MCU is stored in the Firebase Realtime Database, which stores the data in real-time. The data is then analyzed and processed using Firebase Cloud Functions, which enables the development of server-side logic for the application. The Android App is used to authenticate patients and provide them with access to their health data. The app also allows healthcare providers to view the health data of their

patients on a dashboard. The dashboard provides a user-friendly interface for healthcare providers to efficiently analyze and access comprehensive health records. The authentication system ensures that only authorized users have access to the patient's health data, ensuring privacy and security. The system is designed to provide real-time monitoring of a patient's health, enabling healthcare providers to take prompt action in case of any emergency. In summary, the Patient Health Monitoring System is a comprehensive solution for monitoring a patient's health in real-time. The system uses sensors to collect health data, which is then transmitted to an ESP8266 MCU for processing and transmission to the cloud. The Google Firebase cloud platform is used for data storage, analysis, and accessibility. The Android App is used for authentication and providing patients and healthcare providers with access to health data. The system also includes an emergency button for patients to use in case of an emergency. The system is designed to ensure accuracy, privacy, and security, making it an ideal solution for remote patient monitoring.

IV. METHODOLOGY

A. . INTERFACING HARDWARE

The system utilizes the ESP8266 NodeMCU microcontroller and the MAX30102 heart rate and SpO2 sensor. This sensor, operating on the I2C protocol with SDA connected to D2 and SCL connected to D1 of the ESP8266, is an optimal choice for wearables due to its ultra-low power consumption (600 μ A in measurement mode, 0.7 μ A in standby mode). Its high sample rate and fast data output capabilities make it ideal for real-time health monitoring, while integrated ambient light cancellation enhances accuracy. The combination of these components ensures efficient and accurate measurement of vital health parameters, facilitating seamless integration into the Remote Patient Health Monitoring System.

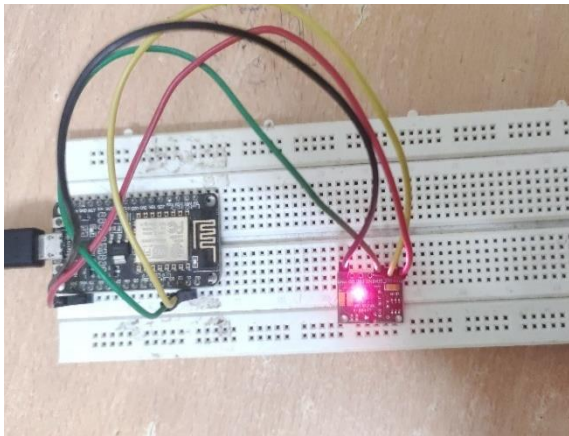
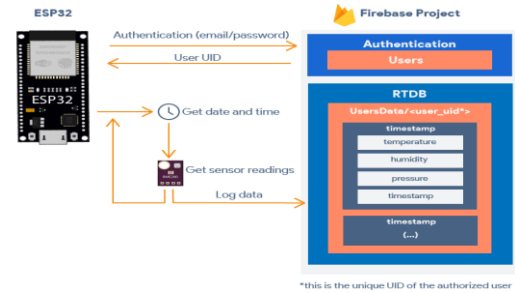


Fig. 1. Interfacing MAX30102 pulse rate sensor with esp8266

To program the ESP8266 MCU, the Arduino IDE is connected to the MCU through the UART (Universal Asynchronous Receiver/Transmitter) protocol. UART is a serial communication protocol that allows for the transmission of data between devices over a single communication line.

B. SETUPING FIREBASE

The necessary settings for authentication, such as the API key, user email, and password, have been configured in Firebase. The database URL has been set, and WiFi reconnection has been enabled to ensure a stable connection. Additionally, the token generation callback and retry count have been configured to handle any errors that may occur during the token generation process. Overall, it is essential for establishing a secure and reliable connection to a Firebase Realtime Database, allowing developers to store and retrieve data in real-time.



C. DEVELOPMENT OF THE APP

The application for real-time patient monitoring has been developed using Android Studio and the Kotlin programming language. The application features a login screen, which healthcare professionals can utilize to sign in and access the system to monitor their registered patients. Upon successful authentication, the user is presented with a screen displaying a list of their registered patients. By selecting a patient from the list, the user is then directed to a screen displaying real-time data for the patient's pulse rate, SpO2 levels, and temperature. This information is updated in real-time, providing healthcare professionals with up-to-date and accurate data to monitor their patients' health status.

V. RESULT

The Firebase Realtime Database serves as a dynamic platform showcasing current sensor readings in real-time, essential for ongoing monitoring. Incorporating timestamp functionality, it enables easy tracking and comparison with previous readings, fostering a comprehensive understanding of health trends. Leveraging the Firebase API, the Android application seamlessly fetches this data from the database, facilitating its visualization through a real-time chart. This integration not only ensures the timely availability of critical health information but also provides healthcare professionals with valuable insights into patient conditions, aiding in informed decision-making and proactive interventions.

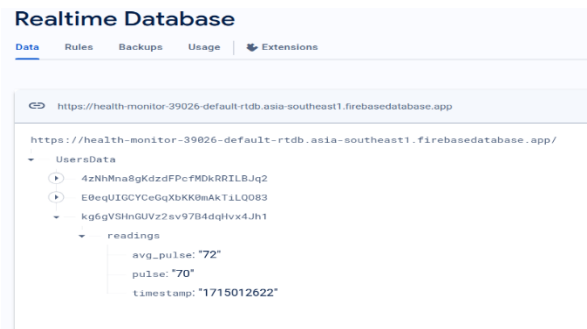


Fig. 2. Firebase Realtime Database dashboard

[illegible]

Fig. 3. Android Studio Logcat showing received values

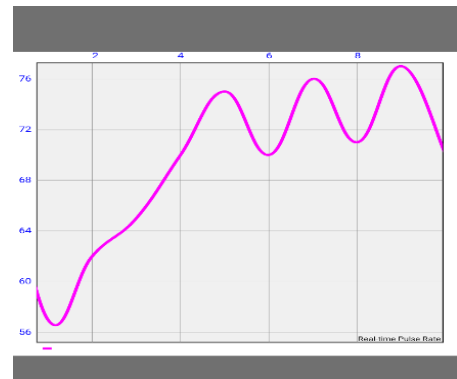


Fig. 4. Custom developed chart showing pulse in real-time

This chart serves as a pivotal component of the user experience, providing a clear and intuitive representation of the pulse rate trends over time. The implementation of this chart not only enhances the aesthetic appeal of the application but also fulfills a critical functional role by presenting vital health information in a comprehensible format. Users can monitor their pulse rate dynamics effortlessly, allowing for informed self-assessment and enabling healthcare professionals to make data-driven decisions.

VI. CONCLUSION

In conclusion, the Real-Time Remote Patient Health Monitoring System represents a significant advancement in healthcare technology, addressing the challenges faced by healthcare professionals in monitoring and managing patients remotely. By leveraging IoT devices such as the esp8266 microcontroller and pulse sensor, along with the Firebase cloud platform and Kotlin-based Android application, the system enables real-time monitoring of vital health parameters and efficient analysis of comprehensive health records. The inclusion of an emergency call system further enhances patient safety and enables timely interventions. Overall, this project has successfully established a reliable and efficient remote patient monitoring system that not only enhances healthcare delivery but also improves patient outcomes, paving the way for more effective and accessible healthcare solutions in the future.

REFERENCES

- [1] <https://www.arduino.cc/reference/en/iot/api/>
- [2] <https://developer.android.com/kotlin>
- [3] <https://esp32tutorials.com/category/esp32-esp-idf-tutorials>
- [4] T M Kadarina and R Priambodo Monitoring heart rate and SpO2 using Thingsboard IoT platform for mother and child preventive healthcare