International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue:2 Volume No.10 March - April – 2026
DOI:10.46647/ijetms.2026.v10i02.033 ISSN: 2581-4621

RTOS & IoT-based Health Monitoring System

Sreepada Ravi Kumar[1],Kommidi Manish Reddy[2],Shaik Musthaq[3],Bobbili Asini Reddy[4]
[1]Assistant Professor, Dept. of Electronics and Communication Engineering, MVSR Engineering College(A), Nadergul (P.O.), Telangana, India
[2,3,4]BE Students, Dept. of Electronics and Communication Engineering, MVSR Engineering College(A), Nadergul (P.O.), Telangana, India

Abstract – The increasing prevalence of lifestyle-related health conditions, particularly cardiovascular and respiratory disorders, has created a strong demand for continuous and reliable patient monitoring systems beyond traditional clinical environments. Conventional healthcare methods, which rely on periodic examinations and manual observation, are often insufficient for timely diagnosis and early intervention. To address these limitations, this paper presents a real-time physiological monitoring framework that integrates Internet of Things (IoT) technology with a real-time operating system (RTOS) to enable efficient, scalable, and low-latency healthcare monitoring.

The proposed system utilizes an ESP32 microcontroller as the core processing unit, interfaced with multiple biomedical sensors including ECG, blood oxygen saturation (SpO_2), and temperature sensors. These sensors continuously acquire physiological data, which is processed using FreeRTOS-based task scheduling. By assigning priority to critical tasks such as ECG signal acquisition, the system ensures timely processing while maintaining overall system stability. Synchronization mechanisms are implemented to prevent resource conflicts during concurrent task execution, enabling efficient multitasking and reliable performance.

The processed data is transmitted to the HiveMQ cloud platform via Wi-Fi, allowing real-time remote monitoring by both users and healthcare professionals. Experimental results demonstrate an accuracy of approximately 98% under continuous operating conditions, with stable performance over extended durations. The proposed framework provides a cost-effective, scalable, and energy-efficient solution for applications such as remote patient monitoring, elderly care, and telemedicine, making it a promising approach for next-generation smart healthcare systems.

I. INTRODUCTION

Over the past decade, IoT integrated with embedded systems has transformed the way healthcare services are delivered [2]. In practice, conventional healthcare methods still depend largely on periodic examinations and manual observation. This limitation becomes more serious in the case of chronic diseases, especially cardiovascular problems, where continuous monitoring is essential for effective diagnosis [1].

To address these limitations, a real-time physiological monitoring system was designed and implemented that can track important health parameters in real time. The system focuses on monitoring ECG signals, blood oxygen saturation (SpO_2), and body temperature, as these parameters provide valuable insights into a person's cardiovascular and respiratory health.

The ESP32 microcontroller was selected as the core component due to its low power consumption, built-in Wi-Fi capability, and dual-core processing. These features make it well-suited for IoT-based healthcare applications.

One of the core features of the proposed system is the integration of FreeRTOS. Instead of handling tasks sequentially, the system manages multiple operations simultaneously. Critical tasks, such as ECG signal acquisition, are given higher priority to ensure timely processing. At the same time, synchronization mechanisms are used to avoid conflicts between sensors.

By transmitting the collected data to the cloud, the system enables remote monitoring, allowing doctors and users to access health data anytime. This makes the solution particularly useful for home healthcare and telemedicine applications.

Key Contributions:

1. Design and implementation of a multi-sensor physiological monitoring system using ESP32 and FreeRTOS.
2. Efficient task prioritization using RTOS to ensure real-time ECG signal processing.
3. Integration with HiveMQ cloud platform for reliable real-time data transmission.

II. LITERATURE SURVEY

Several IoT-based healthcare monitoring systems have been proposed in recent years, focusing on remote data acquisition and real-time analysis. Ke Qi [1] introduced a multilayer machine learning-based approach that improves monitoring accuracy; however, the increased computational complexity makes it less suitable for low-power embedded systems. Similarly, Rashid and Nemati [2] emphasized human-centered healthcare monitoring in the Healthcare 5.0 paradigm, highlighting the importance of real-time responsiveness, but challenges related to latency and scalability remain unresolved.

Baby Shalini [3] developed a sensor-based IoT monitoring system capable of continuous data transmission, although it lacks efficient task scheduling when handling multiple physiological parameters simultaneously. Gutte and Vadali [4] implemented a Raspberry Pi-based solution, which offers flexibility but results in higher power consumption and reduced portability.

Despite these advancements, most existing systems do not efficiently manage concurrent sensor operations or prioritize critical physiological signals such as ECG. In addition, latency and synchronization issues are often overlooked in real-time embedded implementations. To address these limitations, the proposed work adopts an RTOS-based architecture using FreeRTOS, enabling prioritized task scheduling, reduced processing delay, and improved system reliability for continuous health monitoring.

III. METHODOLOGY

A.System Overview

In this system, physiological parameters are continuously acquired and processed using an IoT-based approach. It consists of an ESP32 microcontroller connected to biomedical sensors such as:

- AD8232 (ECG sensor)
- MAX30102 (SpO_2 sensor)
- LM75 (temperature sensor)

These sensors collect health data and send it to the ESP32 for processing.

In our implementation, FreeRTOS is used to manage multiple tasks efficiently. Individual sensors are assigned independent tasks within the RTOS environment, and priority is assigned based on the importance of the data. Synchronization techniques, such as mutex locks, are used to avoid resource conflicts when accessing shared resources.

Once processed, the data is transmitted to the HiveMQ cloud platform via Wi-Fi, enabling real-time monitoring. The use of RTOS-based scheduling reduces latency in signal processing compared to conventional loop-based execution. By prioritizing ECG data acquisition, the system minimizes delay in capturing critical cardiac signals. Additionally, careful task synchronization ensures stable communication over shared resources such as the I2C bus, preventing data loss and timing conflicts during continuous operation.

B. Block Diagram

The system is divided into three main layers: sensing, processing, and communication. In the sensing layer, sensors capture physiological parameters such as ECG, SpO_2, and temperature. These signals are sent to the ESP32, which acts as the processing unit.

In the processing layer, FreeRTOS manages multiple tasks using prioritization and synchronization techniques, ensuring smooth and conflict-free operation. In the communication layer, the processed data is transmitted through Wi-Fi to the cloud platform, where it can be stored and visualized for remote monitoring.

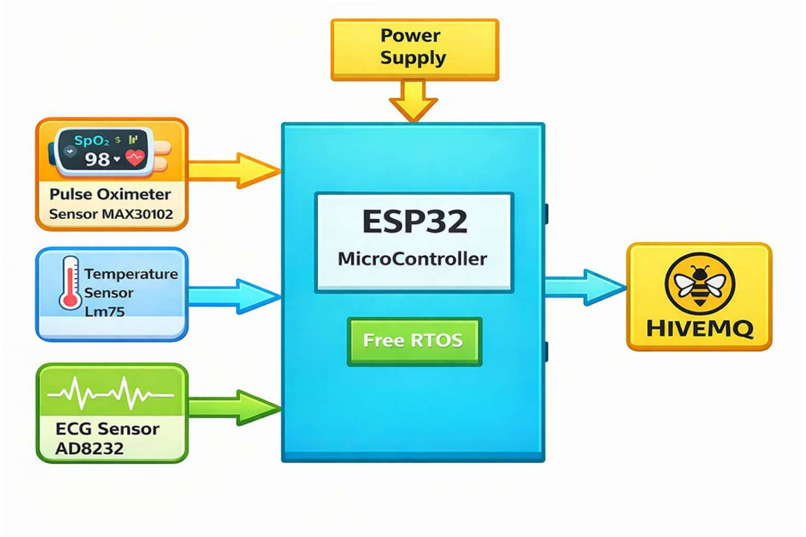

Fig.1. Block diagram illustrating the overall system architecture

The system operation begins with initialization, where the ESP32 sets up communication protocols and connects to a Wi-Fi network. After establishing the connection, the sensors are initialized, and multiple tasks are created using FreeRTOS.

Each task is responsible for collecting specific data such as ECG, SpO_2, or temperature. Mutex mechanisms are used to ensure proper synchronization when accessing shared resources like the I2C bus. The collected data is then processed and transmitted to the cloud for real-time monitoring. Error-handling mechanisms are included to maintain system reliability during failures.

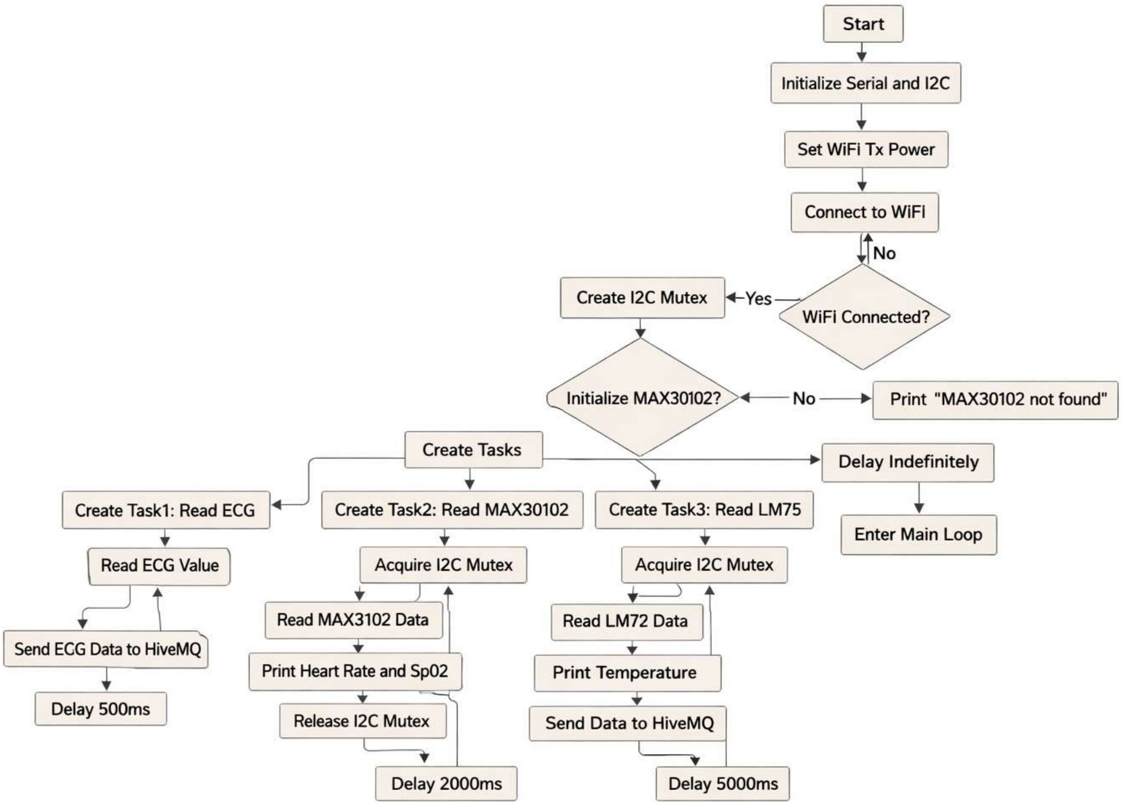

Fig.2. Flowchart of the Proposed System

efficient task scheduling, real-time data acquisition and continuous health monitoring.

D. Schematic Diagram

The schematic diagram shows the hardware connections between the ESP32 and the sensors. The AD8232 ECG sensor is connected to an analog pin for signal acquisition, while the MAX30102 and LM75 sensors communicate through the I2C protocol using shared SDA and SCL lines.

Pull-up resistors are used to ensure stable communication on the I2C bus. Proper grounding and circuit design are followed to reduce noise and improve signal accuracy. The overall design is compact, efficient, and suitable for continuous monitoring applications.

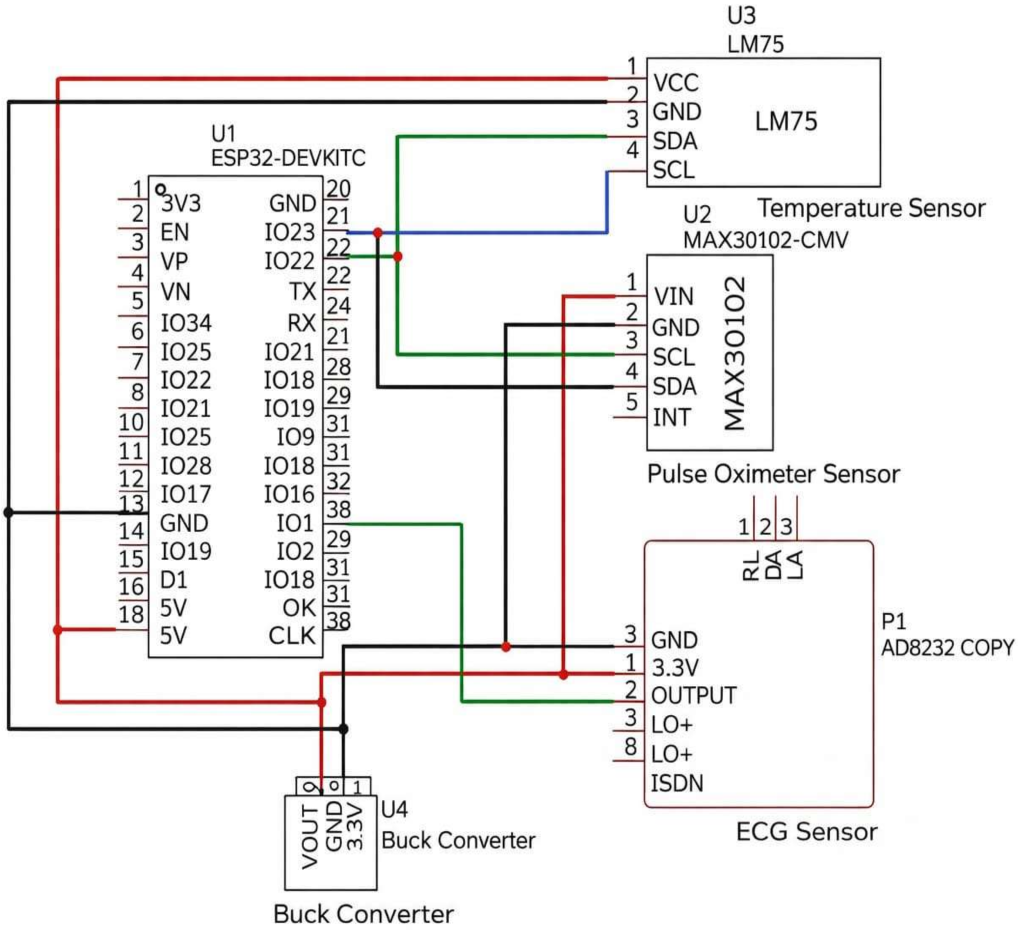

Fig.3. Schematic Diagram of the Proposed System

A regulated 3.3V power supply is provided to all components, either through an external voltage regulator or an onboard power module, ensuring stable operation and protecting the devices from voltage fluctuations. Proper grounding is maintained across all components to minimize noise and ensure accurate sensor readings, particularly important for sensitive signals such as ECG. The circuit design also considers efficient wiring and minimal interference between analog and digital components to improve signal integrity. Once the sensors acquire the physiological data, the ESP32 processes the signals and organizes them for transmission. With built-in Wi-Fi capability, the ESP32 sends the data to a cloud platform, such as HiveMQ, using standard communication protocols. This enables real-time monitoring and remote access to health parameters. Overall, the circuit is designed to be compact, energy-efficient, and reliable, supporting continuous operation while maintaining accuracy and scalability for healthcare applications.

IV. RESULTS AND ANALYSIS

The prototype was tested under real-time operating conditions to assess its performance, processing efficiency, and transmission reliability. The prototype consistently captured key physiological parameters with stable performance, including electrocardiogram (ECG) signals, blood oxygen saturation (SpO_2) and temperature under continuous operating conditions. The system maintained consistent performance even under continuous operation exceeding several hours, demonstrating its suitability for long-term monitoring applications.

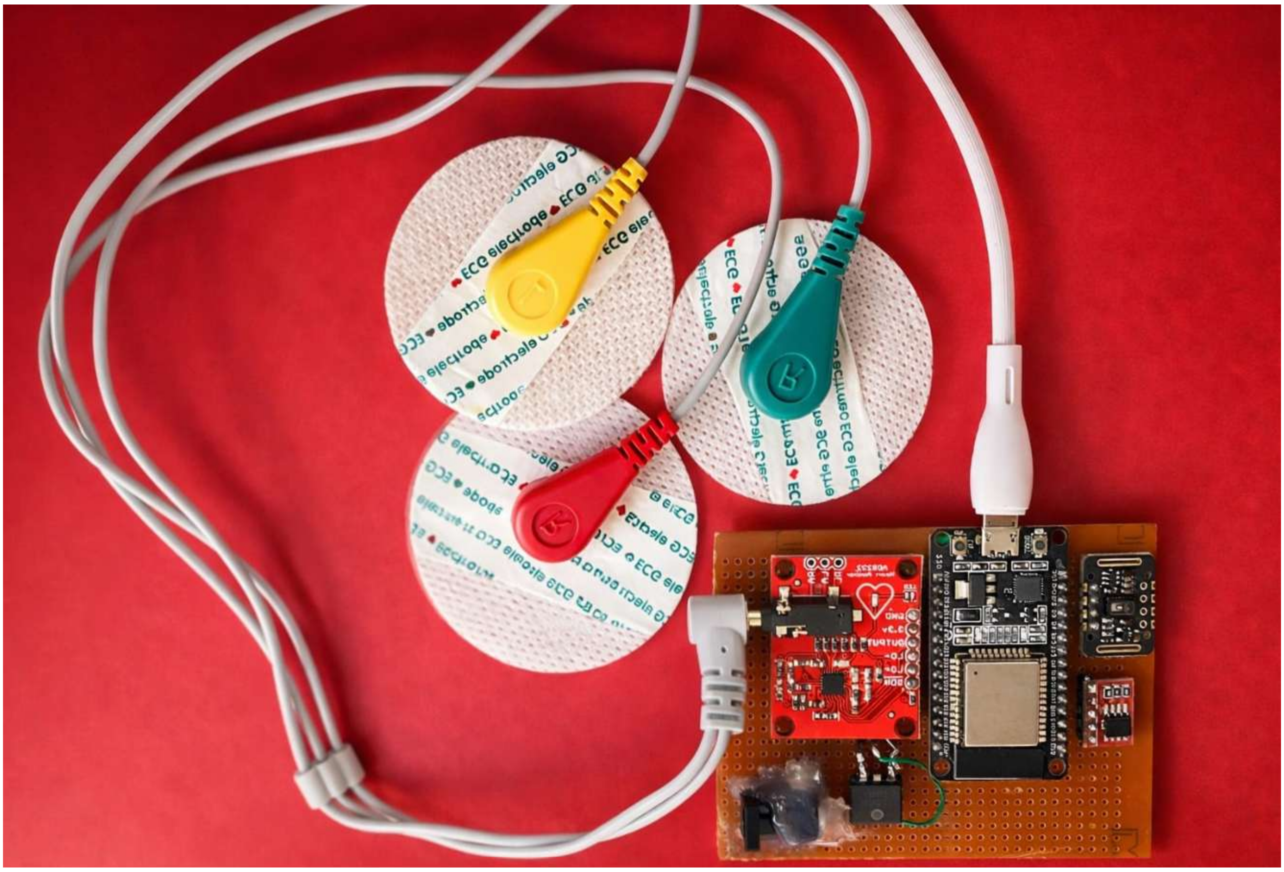

Fig.4. Hardware Prototype

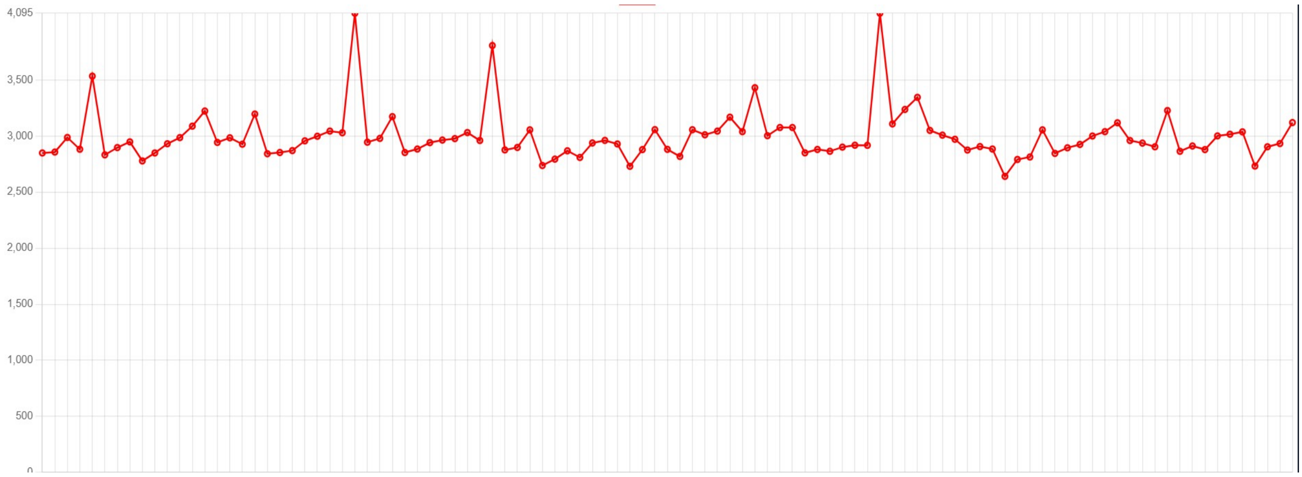

Fig.5. ECG Sample output in HiveMQ

A. System Performance Evaluation

The integration of ESP32 with FreeRTOS enabled efficient multitasking and smooth system operation. Critical tasks such as ECG signal acquisition were processed with minimal delay, while other tasks were handled without affecting system performance.

The use of mutex synchronization ensured reliable communication between sensors, especially over the I2C bus. During testing, reliable Wi-Fi connectivity was observed throughout the testing phase and transmitted data to the cloud without packet loss, making it suitable for real-time applications.

B. ECG Signal Analysis

The AD8232 sensor produced stable ECG signals with clear waveform patterns, including P, QRS, and T waves. Basic filtering techniques were applied to improve signal quality.

The system operates at an ECG sampling rate of approximately 250 Hz, ensuring accurate waveform representation while maintaining low processing latency. Comparison with a commercial ECG device showed high accuracy with minimal variation.

Table I: ECG Accuracy Comparison

Parameters	Commercial Device	Proposed System	Accuracy(%)
ECG Amplitude (mV)	1.2 mV	1.18 mV	98.3%

C. Pulse Oximeter Performance

The MAX30102 sensor was used to measure blood oxygen levels and heart rate. The system achieved an accuracy of around 97–99%, which is comparable to commercial devices.

Some small variations were observed due to factors like finger placement and external light. To improve accuracy, the sensor adjusted LED intensity automatically. Heart rate measurements were also consistent, with very small deviations.

Table 2: SpO2 Accuracy

Parameter	Commercial Oximeter	Proposed System	Accuracy (%)
SpO2 (%)	98%	97%	98.9%

D. Temperature Sensor Performance

The LM75 temperature sensor demonstrated stable and accurate temperature measurements throughout the testing period. The system achieved an accuracy of approximately **±0.5°C**, with a response time of less than **2 seconds**, making it suitable for continuous physiological monitoring.

Comparative analysis with a calibrated digital thermometer showed minimal deviation, confirming the reliability of the sensor for real-time applications.

Table 3: Temperature Sensor Accuracy

Parameter	Digital Thermometer	Proposed System	Accuracy (%)
Temperature (°C)	37.5°C	37.3°C	98.6%

E. Cloud Integration and Data Visualization

The system successfully sent all collected data to the HiveMQ cloud platform using Wi-Fi. The data was displayed in real time with minimal delay.

Users were able to access the data remotely, which makes the system useful for applications like telemedicine and home healthcare monitoring.

F. Limitations and Observations

During testing, minor variations were observed due to sensor placement and environmental noise, which required manual adjustment.

One practical challenge encountered was maintaining stable readings during continuous operation, especially for ECG signal acquisition.

Although the system performed well, some limitations were observed. The quality of ECG signals depended on proper electrode placement and was sometimes affected by external noise.

The system also depends on a stable internet connection. If the network is unstable, data transmission may be affected.

Minor variations in SpO_2 readings were also noticed due to environmental conditions and user handling. These issues indicate that further improvements can be made in calibration and system reliability.

V. CONCLUSION

This work presents a real-time physiological monitoring framework that combines IoT connectivity with RTOS-based task scheduling to enhance system responsiveness and reliability. The system integrates multiple biomedical sensors with an ESP32 microcontroller to track important health parameters such as ECG, blood oxygen levels (SpO_2), and temperature.

By using FreeRTOS, the system is able to handle multiple tasks efficiently, ensuring that critical data is processed on time without delays. The integration of IoT enables the collected data to be transmitted to a cloud platform, making it possible to monitor patient health remotely and in real time.

The proposed model exhibited consistent and dependable performance during evaluation. Its simple design and low cost make it suitable for practical applications like home healthcare and remote patient monitoring.

However, a few limitations were observed during testing. The system depends on stable internet connectivity, and small variations in sensor readings can occur due to environmental factors. Power consumption is another factor that can be improved.

The proposed framework establishes a robust foundation for scalable healthcare monitoring by improving task efficiency, reducing latency, and ensuring reliable long-term operation.

With further improvements, it can be made more accurate, efficient, and suitable for wider real-world applications.

VI. FUTURE SCOPE

Although the current system performs well, there is still room for further development. One possible improvement is the addition of more sensors, such as blood pressure and respiration sensors, to provide more complete health monitoring.

The system can also be enhanced by using machine learning techniques to analyze the collected data and detect potential health issues at an early stage. This would make the system more intelligent and helpful for preventive care.

Another important area is data security. Applying encryption and secure communication methods can help protect sensitive health information from unauthorized access.

Improving power efficiency is also important, especially if the system is to be used in wearable or portable devices. Optimizing both hardware and software can help reduce energy consumption.

In addition, developing a mobile or web-based application can improve user experience by providing easy access to data and real-time alerts. Integration with advanced technologies like edge computing can further reduce delay and improve system performance.

With these improvements, the system can become more reliable, secure, and suitable for large-scale healthcare applications such as telemedicine and remote monitoring.

REFERENCES

[1] Ke Qi, “Advancing hospital healthcare: achieving IoT-based secure health monitoring through multilayer machine learning,” Journal of Big Data, 2025.

[2] Samad Rashid and Arash Nemati, “Human-centered IoT-based health monitoring in the Healthcare 5.0 era,” 2024.

[3] V. Baby Shalini, “Smart Health Care Monitoring System based on Internet of Things (IoT),” IEEE, 2021.

[4] Ashwini Gutte and Ramkrishna Vadali, “IoT Based Health Monitoring System Using Raspberry Pi,” ICCUBEA, IEEE, 2018.

[5] https://ijetms.in/Vol-10-issue-2/Vol-10-Issue-2-6.pdf