

Smart Health Monitoring System Using IOT and AI

¹Ms. Ayesha Shaikh, ²Ms. Swarangi Aundhekar, ³Ms. Palak Kumari,
⁴Ms. Reshma Bakshi ⁵Mr. Ali Karim Sayed

^{1,2,3}Student AIML, ⁴Lecturer AIML

Anjuman-I-Islam's A. R. Kalsekar Polytechnic, New Panvel

¹ayeshashaikh28007@gmail.com

²aundhekarswarangi@gmail.com

³palakkumari8834@gmail.com

⁴reshma.bakshi@aiarkp.ac.in

⁵alikarim.sayed@gmail.com

Abstract--- The rapid advancement of the Internet of Things (IoT) has enabled continuous and remote monitoring of patient health. This paper presents a Smart Health Monitoring System that tracks vital parameters such as heart rate, blood oxygen saturation (SpO₂), and body temperature using sensors interfaced with an ESP32 microcontroller. The collected data is transmitted in real time to a web-based dashboard developed using Python and Flask, where it is displayed in tabular and graphical formats for effective analysis. An Artificial Intelligence (AI)-based alert module is integrated to analyze incoming data, detect abnormal health patterns, and generate early warnings when parameters deviate from normal ranges. The system aims to reduce human error, enable timely medical intervention, and provide a low-cost, portable, and efficient solution for continuous patient monitoring and improved healthcare delivery.

Keywords--- Smart Health Monitoring, IoT, Artificial Intelligence, ESP32, SpO₂, Real-Time Monitoring, Remote Healthcare, Flask, Arduino, Raspberry Pi.

I. Introduction

The Internet of Things (IoT) refers to the interconnection of physical devices such as industries, buildings, and objects embedded with sensors, software, and communication technologies. These devices can collect and exchange data continuously, enabling automation and intelligent decision-making.

IoT allows remote sensing and control over network infrastructure, reducing human intervention while improving system efficiency and reliability. In healthcare, IoT plays a vital role by enabling continuous monitoring of patient health.

A Raspberry Pi, a Linux-based single-board computer, can interface with sensors and transmit data to web applications.

It acts as a central hub that collects data from sensors such as heart rate, temperature, and SpO₂ sensors.

In this system, a wearable smart bracelet equipped with sensors is used to monitor patient vitals. The collected data is transmitted to a web dashboard, allowing doctors to monitor patients remotely. If any abnormality is

detected, alerts are generated to ensure timely medical attention.

II. Background and Related Work

With the growth of IoT technologies, healthcare systems are transitioning from traditional hospital-based monitoring to continuous remote monitoring solutions.

Patients with chronic conditions and elderly individuals require constant monitoring of vital parameters such as heart rate, oxygen levels, and body temperature. Traditional monitoring methods may fail to detect sudden changes.

IoT-based systems overcome these limitations by continuously collecting and transmitting physiological data. Sensors like MAX30102 are used for heart rate and

- Perception Layer: Sensors (MAX30102, DS18B20)

SpO₂ measurement, while DS18B20 is used for temperature sensing. These sensors are typically connected to microcontrollers such as ESP32.

Recent systems also use Raspberry Pi as a gateway device to process and forward data to web dashboards.

Working Methodology:

1. Sensor Module:

Sensors such as the LM35 temperature sensor and SEN11574 pulse sensor are used to collect physiological data from the patient continuously. These sensors are placed carefully to ensure proper contact and accurate readings. Since sensor values can sometimes fluctuate due to slight movements or environmental factors, continuous monitoring helps in capturing more stable and reliable data over time.

2. ESP32 Microcontroller:

The ESP32 acts as the main controller of the system. It reads the data from the sensors, performs basic processing, and prepares it for transmission. The microcontroller ensures that the data is handled efficiently and sent without delay. It also maintains a continuous connection with the Wi-Fi network, allowing real-time data transfer. Due to its built-in Wi-Fi capability and fast processing, ESP32 is well-suited for IoT-based health monitoring applications.

These dashboards enable remote monitoring and visualization.

Furthermore, AI and Machine Learning techniques are now integrated into such systems to detect abnormal patterns and provide early alerts, making healthcare more proactive and efficient.

III. System Design and Methodology

The Smart Health Monitoring System is designed to monitor vital health parameters in real time using IoT.

5. AI Alert Module:

The system analyzes incoming data to detect abnormal values and generates alerts whenever parameters deviate from predefined normal ranges.

IV. Block Diagram and Description

The system follows a multi-layer IoT architecture

3. Communication Layer:

The processed data is transmitted from the ESP32 to the Flask-based web server using a Wi-Fi network. This communication happens continuously so that updated health data is always available. The system ensures that the data is sent in a structured format, making it easier for the server to process and display it correctly. A stable network connection is important here, as it directly affects the speed and reliability of data transmission.

4. Web Dashboard:

The web dashboard displays the received data in both tabular and graphical formats for better understanding. Users can view real-time values of heart rate, temperature, and oxygen levels, along with trends over time. Graphs help in easily identifying sudden changes or abnormal patterns in the data.

- Edge Layer: ESP32 microcontroller

- Gateway Layer: Raspberry Pi

- Application Layer: Web dashboard + AI module

The ESP32 collects sensor data and sends it to the Raspberry Pi, which processes and forwards it to the dashboard

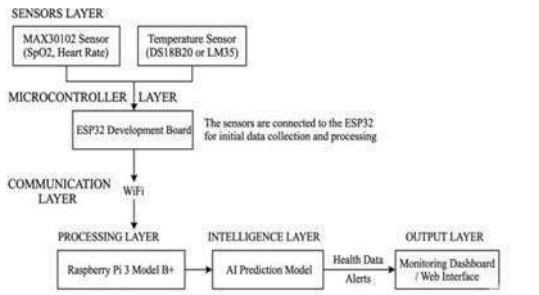


Fig. 1. Block Diagram of Smart Bracelet

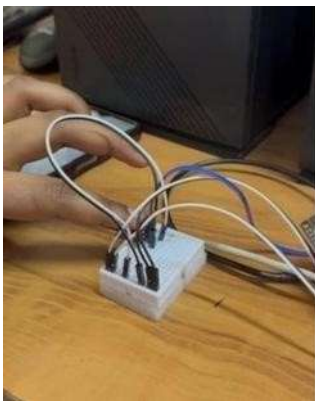


Fig. 2. Hardware Prototype - Sensor Connections

V. Proposed Hardware

A. Raspberry Pi 3

The Raspberry Pi 3 is a Linux-based single-board computer that functions as the central processing unit of the system. It receives sensor data from the ESP32, runs the Flask web server, and hosts the monitoring dashboard. It is equipped with a Broadcom BCM2837 64-bit ARMv7 quad-core processor operating at 1.2 GHz, 1 GB RAM, built-in Wi-Fi, Bluetooth Low Energy (BLE), and multiple GPIO pins, making it suitable for IoT applications.

B. Arduino Uno

The Arduino Uno is an open-source microcontroller board based on the ATmega328. It is used for reading analog sensor values and converting them into digital signals. These signals are then transmitted via a communication module. The board features 14 digital I/O pins, 6 analog input pins, and operates at a clock speed of 16 MHz.

C. HC-05 Bluetooth Module

The HC-05 Bluetooth module enables wireless serial communication between devices. It operates in the 2.4 GHz ISM band and supports a baud rate of up to 9600. This module facilitates data transfer between the Arduino and Raspberry Pi in certain configurations.

D. LM35 Temperature Sensor

The LM35 is a precision temperature sensor that provides output voltage linearly proportional to temperature in degrees Celsius. It operates within a voltage range of 4–20 V and produces an output of 10 mV per degree Celsius, ensuring reliable temperature measurement.

E. SEN-11574 Pulse Sensor

The pulse sensor works based on infrared light detection. It measures heart rate by detecting variations in blood flow through the skin. The sensor produces a filtered analog signal, which is then processed by the microcontroller to calculate pulse rate accurately.

VI. Results and Discussion

The Smart Health Monitoring System was successfully implemented and tested under different conditions. The system effectively measured and displayed vital parameters such as heart rate, oxygen saturation (SpO₂), and body temperature in real time.

The ESP32 transmitted sensor data to the Flask server consistently without packet loss under stable Wi-Fi conditions. The web dashboard updated instantly, presenting readings along with accurate timestamps. This ensured reliable tracking of patient data over time.

The response time of the system was observed to be between 500 milliseconds and 2 seconds when detecting abnormal readings.

This latency is acceptable for real-time health monitoring applications.

Graphical visualization of the data allowed easy identification of trends and anomalies. For instance, simulated drops in SpO₂ levels were clearly reflected in the graphs, making abnormal patterns easy to detect.

Overall, the system demonstrated efficient performance in continuous monitoring and early alert generation,

indicating its suitability for real-world healthcare applications.

VII. Advantages

The system enables continuous real-time monitoring of vital parameters, eliminating the need for frequent hospital visits and allowing early detection of health issues.

It provides remote accessibility through a web-based dashboard, allowing doctors and caregivers to monitor patients from any location within the network.

The design is affordable and portable, with a total estimated cost of ₹6,000–6,500, making it accessible for home healthcare use.

Additionally, the interface is user-friendly, with clear data representation through tables and graphs, ensuring ease of use even for non-technical users.

VIII. Limitations

The system depends on a stable Wi-Fi connection for data transmission. In areas with poor connectivity, performance may be affected.

Sensor accuracy may vary due to environmental factors, improper placement, or motion artifacts, as consumer-grade sensors are used.

There may be minor delays in processing and alert generation under high network load, which can slightly impact real-time responsiveness.

Although relatively affordable, the initial hardware cost may still be a limitation for some users.

IX. Cost Estimation

Table I shows the estimated cost of all hardware components required to build the Smart Health Monitoring System.

Component	Estimated Cost
ESP32 Development Board	₹800
Pulse Sensor (SEN-11574)	₹400
SpO ₂ Sensor (MAX30102)	₹700
Temperature Sensor (LM35)	₹300
Raspberry Pi 3	₹4,000
Connecting Wires & Misc.	₹300
Total	₹~6,000 – 6,500

IX. Conclusion

The Smart Health Monitoring System presents an effective solution for continuous patient monitoring using IoT and AI technologies. By integrating sensors with the ESP32 and a Flask-based dashboard, the system enables real-time tracking of vital health parameters.

The inclusion of an AI-based alert system enhances its capability by identifying abnormal conditions and providing early warnings. This can significantly improve response time in critical situations.

Despite certain limitations such as network dependency and sensor accuracy, the system offers a practical and cost-effective approach for remote healthcare. Future enhancements may include cloud integration, mobile application support, and more advanced machine learning models to improve prediction accuracy.

Acknowledgment

The authors sincerely thank Anjuman-I-Islam's A. R. Kalsekar Polytechnic, New Panvel, for providing the necessary support and resources for this project. Special gratitude is extended to Ms. Reshma Bakshi for her valuable guidance throughout the development of this system. The authors also acknowledge the support of faculty members and peers who contributed to this work.

References

- [1] Khan Adil Parvez. "Smart Health Monitoring and Controlling System using raspberry pi". Available online at www.ijariit.com
- [2] J. Ferdous, B. Roy, M. Hossen, and M. M. Islam, "Implementation of IoT-Based Patient Health Monitoring System Using ESP32 Web Server," 2023. Available: https://www.researchgate.net/publication/372268729_IMPLEMENTATION_OF_IOT_BASED_PATIENT_HEALTH_MONITORING_SYSTEM_USING_ESP32_WEB_SERVER
- [3] U. A. Contardi *et al.*, "MAX30102 Photometric Biosensor Coupled to ESP32 for Continuous Monitoring," *Engineering Proceedings*, 2021. Available: <https://www.mdpi.com/2673-4591/16/1/9>
- [4] P. Jha *et al.*, "IoT-Based Real-Time Multi-Node Health Monitoring System," 2025. Available: <https://www.ijreet.com/wp-content/uploads/2025/05/IoT-Based-Real-Time-Multi-Node-Health-Monitoring-System-.pdf>
- [5] A. Saw *et al.*, "IoT-Based Health Monitoring System Using Arduino and ESP8266," 2025. Available: <https://ijsrem.com/download/iot-basedhealthmonitoring-system-using-arduino-and-genericesp8266/>
- [6] S. Thakur *et al.*, "ER-ADENN: Design and Implementation of EEG-based Emotion Recognition Using Adaptive Dropout Enabled Neural Network," *InCACCT*, 2025.
- [7] "IoT-Based Patient Health Monitoring System Using NodeMCU," 2025. Available: https://www.researchgate.net/publication/389779345_IoTBased_Patient_Health_Monitoring_System_Using_NodeM_CU
- [8] M. I. P. Pratama *et al.*, "Design and Evaluation of an IoT-Enabled ESP32 Photoplethysmography Monitor," 2025. Available: <https://journal.trunojoyo.ac.id/triac/article/view/31989>
- [9] J. P. Jose *et al.*, "IoT Enabled Health Monitoring System Using Sensors and AI," *Scientific Reports*, 2025. Available: <https://www.nature.com/articles/s4159802510199-4>