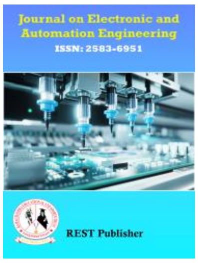
Journal on Electronic and Automation Engineering

Vol: 4(2), June 2025

REST Publisher; ISSN: 2583-6951 (Online)

Website: https://restpublisher.com/journals/jeae/

DOI: https://doi.org/10.46632/jeae/4/2/5



IoT Based Remote Health Monitoring System Enhancing Two Way Communication

***N. Usha Rani, S. Roshini, M. Lakshmi Harshitha, S. Jaffer Hussain, A. Maheswara Reddy T**

Annamacharya Institute of Technology & Science (AITK), Kadapa, Andhra Pradesh, India.

**Corresponding Author Email*: *usharaninare880@gmail.com*

Abstract: *The integration of the Internet of Things (IoT) in healthcare has significantly improved remote patient monitoring by enabling continuous real-time tracking of vital signs and movement. This paper presents an IoT-based Real-Time Remote Patient Monitoring System enhanced with two-way communication to facilitate improved patient-caregiver interaction. The system is built around an ESP32 micro controller, which collects data from multiple sensors, including an ADXL345 accelerometer for fall detection, a DHT11 sensor for temperature and humidity monitoring, a 16×2 LCD provides real-time local feedback, while Wi-Fi connectivity ensures seamless remote data transmission. A key innovation of this system is its bidirectional communication capability, allowing caregivers to send alerts, emergency instructions, or medical guidance directly to the patient's device. This interaction is facilitated through a buzzer and display, ensuring timely responses and reducing delays in critical situations. The proposed system enhances traditional remote monitoring by integrating real-time sensor data acquisition with an interactive response mechanism. Experimental evaluations demonstrate its effectiveness in continuous health tracking, fall detection, and emergency alert transmission, making it a valuable solution for telemedicine, elderly care, and chronic disease management.*

Keywords: *IoT, Remote Patient Monitoring, Two-Way Communication, ESP32, Healthcare, Wireless Sensors, Telemedicine.*

1. INTRODUCTION

The rapid evolution of healthcare technologies has created an urgent need for innovative patient monitoring solutions that extend beyond traditional manual methods. With the rise of chronic diseases, an aging population, and the expansion of home-based care, remote patient monitoring (RPM) has emerged as a critical tool for healthcare providers. By leveraging modern IoT technologies, RPM enables continuous tracking of vital signs and health-related data, minimizing the need for frequent hospital visits while allowing proactive management of medical conditions. Real-time patient monitoring not only reduces hospital readmission rates but also enhances the quality of life for individuals receiving care at home. This is particularly significant for elderly patients, post-operative individuals, and those with chronic illnesses, who require consistent health supervision. Furthermore, the integration of wireless communication, IoT-based data analytics, and automated alert systems has the potential to revolutionize healthcare by enabling immediate responses to critical health events. One of the most vital aspects of RPM systems is their ability to generate timely alerts for abnormal conditions. In critical scenarios, rapid detection of medical emergencies such as a high fever, sudden drop in blood pressure, or an unexpected fall—can be lifesaving. The capability to issue instant notifications not only reassures patients but also empowers caregivers and medical professionals to deliver timely interventions. The incorporation of such real-time alert mechanisms into RPM solutions marks a significant advancement in healthcare delivery, ensuring that no critical event goes unnoticed.

2. OBJECTIVE

The primary objective of the IOT Real Time Remote Patient Monitoring System is to design and implement a robust solution that captures critical sensor data and uses it to monitor patient health continuously. To achieve this, the project is guided by several specific objectives:

Sensor Data Acquisition:

Develop a system capable of interfacing with multiple sensors. These include an accelerometer (ADXL345) for movement and fall detection, a temperature and humidity sensor (DHT11) for monitoring vital signs and environmental conditions, and a GPS module for tracking the patient's location. The integration of these sensors will enable a multi-dimensional assessment of the patient's status.

Local Display and Feedback:

Implement a user-friendly local interface using a 16×2 LCD display, which will provide real-time visualization of the sensor data. This immediate feedback ensures that patients and caregivers can quickly interpret vital parameters such as temperature, estimated heart rate or blood pressure, and movement status (stable or fall).

Remote Data Transmission:

Equip the system with Wi-Fi connectivity to facilitate the transmission of real-time data to a remote server. This remote data logging and analysis is critical for healthcare providers to monitor the patient's condition continuously, review historical data, and take timely actions in case of emergencies.

Real-Time Alerts:

Incorporate a mechanism for generating immediate alerts, such as triggering an audible buzzer, when the sensor data indicates abnormal conditions. Timely alerts ensure that any deviation from normal health parameters, such as a high fever or an unanticipated fall, is promptly communicated to caregivers.

System Reliability and Scalability:

Ensure that the system is both reliable and scalable. The hardware and software architecture should be designed to handle continuous operation and allow for future expansion, such as integrating additional sensors or communication protocols as healthcare requirements.

3. SYSTEM DESIGN AND ARCHITECTURE

The Internet of Things (IoT)-based Real-Time Remote Patient Monitoring System is designed to seamlessly integrate hardware and software components to enable continuous health monitoring. The system architecture ensures efficient data acquisition, processing, display, and wireless transmission of vital health parameters.

3.1. Hardware Architecture:

The system incorporates various hardware components to facilitate real-time patient monitoring:

- **ESP32 Development Board:** Serves as the central processing unit, leveraging its dual-core micro controller, integrated Wi-Fi, and Bluetooth capabilities to manage sensor data and ensure real-time wireless communication.
- **ADXL345 Accelerometer**: A three-axis MEMS accelerometer used for motion tracking and fall detection. It operates over the I2C protocol and provides high-resolution motion analysis with low power consumption.
- **DHT11 Temperature and Humidity Sensor**: Measures ambient temperature and humidity, providing crucial health indicators. It is selected for its compatibility with the ESP32 and moderate precision.
- **GPS Module**: Enables real-time tracking of the patient's geographical location, ensuring prompt intervention in case of emergencies.
- **16x2 LCD Display**: Provides local real-time data visualization, displaying critical patient health parameters such as temperature, movement status, and alerts.
- **Buzzer**: Functions as an alert mechanism to notify caregivers in case of abnormal readings, such as elevated temperature or detected falls.

3.2. System Operation and Data Flow:

1. **Data Acquisition**: The ESP32 continuously collects sensor data from the accelerometer, temperature sensor, and GPS module.
2. **Processing & Analysis**: The microcontroller processes the received data, detecting anomalies such as sudden movement (fall), abnormal temperature, or patient displacement.
3. **Data Visualization**: Processed information is displayed on the LCD screen for local monitoring.
4. **Alert Mechanism**: If predefined thresholds are exceeded, the buzzer is activated to alert caregivers.
5. **Wireless Transmission**: The ESP32 transmits the data to a remote server or cloud platform via Wi-

Fi, enabling real-time monitoring by healthcare providers. The proposed architecture ensures a reliable, scalable, and efficient patient monitoring solution, leveraging IoT capabilities for enhanced healthcare management.

3.3. System Architecture and Interconnections:

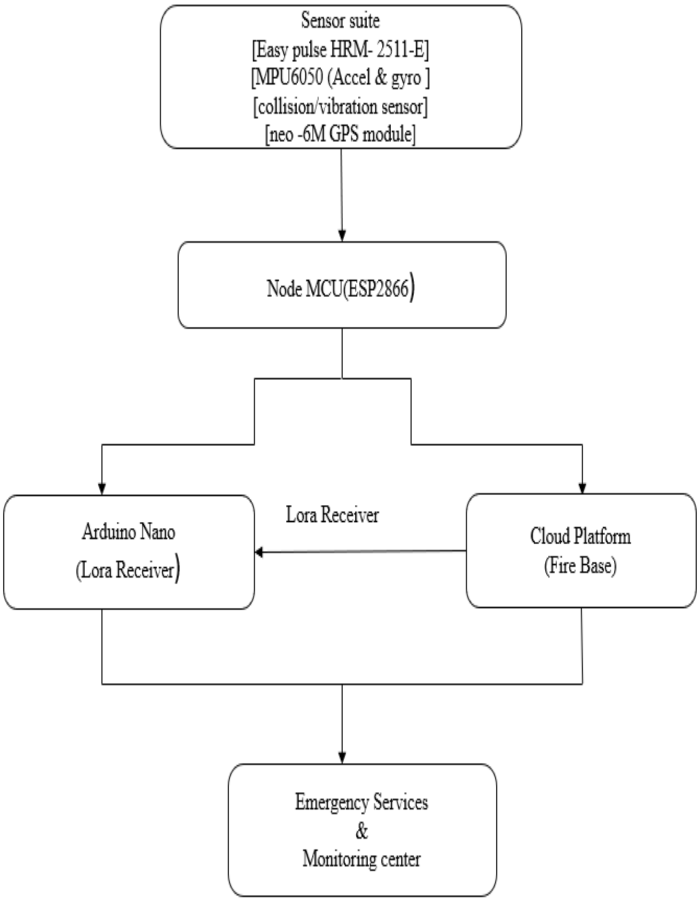


FIGURE 1. Block diagram for System Architecture and Interconnection

The system follows a modular architecture where all components are interconnected to the ESP32 micro controller:

1. **ESP32 Power and Ground:**
   - Provides 3.3V to low-power sensors (ADXL345, I2C LCD) and 5V to modules that require higher voltage.
   - All sensor ground pins are connected to the ESP32 ground.

2. **Sensor and Peripheral Connections**:

ADXL345 Accelerometer (I2C Interface):

VCC → 3.3V, GND → Ground, SDA → GPIO21, SCL → GPIO22.

16×2 LCD (I2C Interface):

VCC → 3.3V, GND → Ground, SDA → GPIO21, SCL → GPIO22.

DHT11 Temperature Sensor:

VCC → 3.3V or 5V, GND → Ground, Data → GPIO4 (with a pull-up resistor if necessary).

GPS Module:

VCC → 3.3V or 5V, GND → Ground, TX → GPIO16 (ESP32 RX2), RX → GPIO17 (ESP32 TX2).

Buzzer: Positive Terminal → GPIO23, Negative Terminal → Ground (with series resistor for current limitation if required).

4. EXISTING SYSTEM

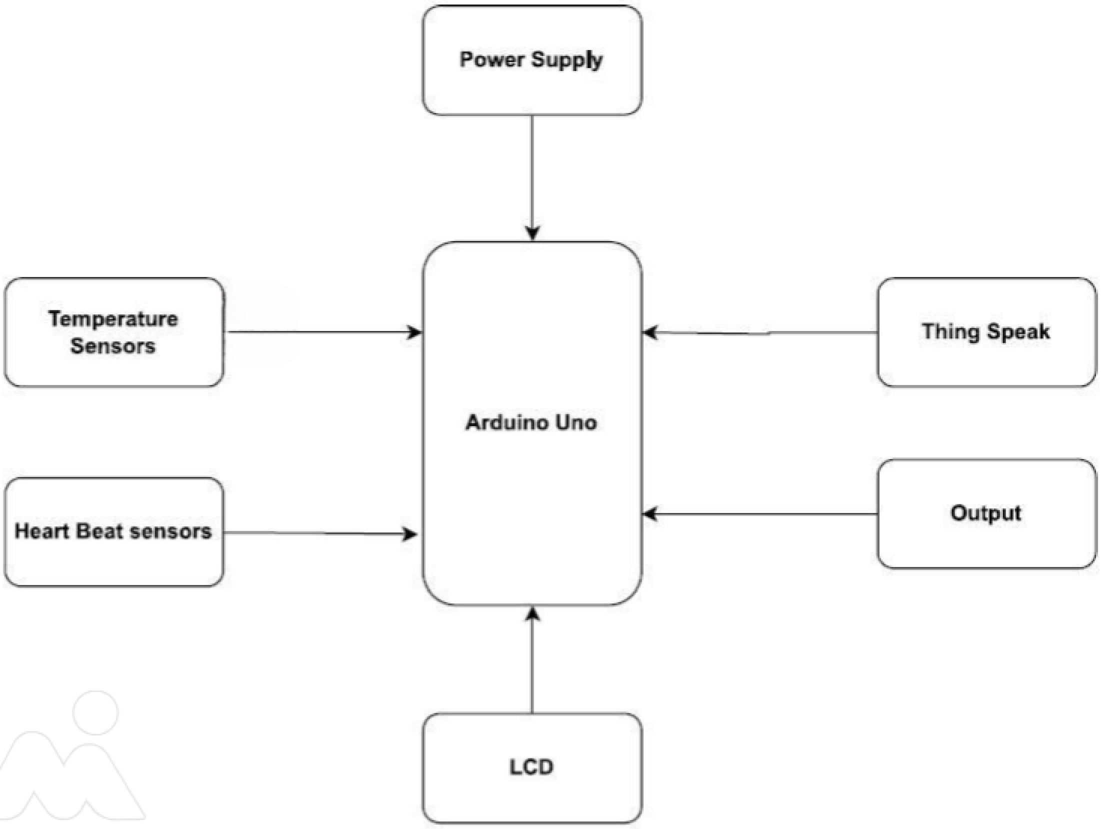


FIGURE 2. Existing System

ABOUT EXISTING SYSTEM:
Manual monitoring
Standalone devices
Limited remote monitoring
Emergency detection delays
No centralized data storage

5. PROPOSED SYSTEM

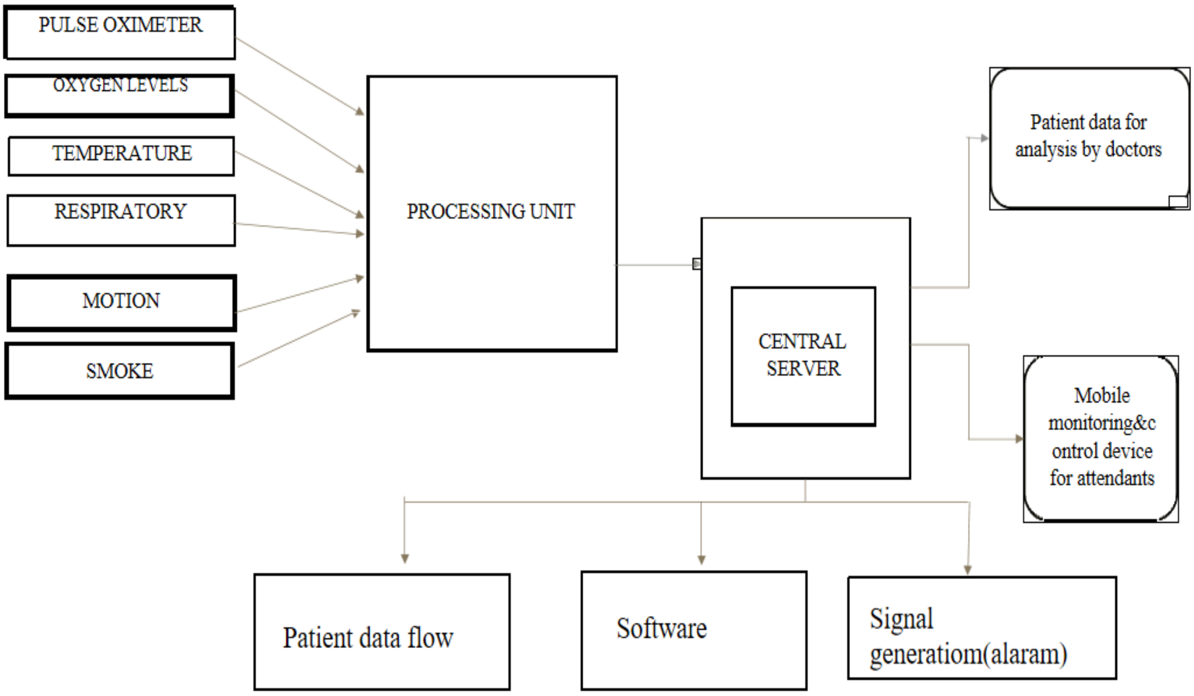


FIGURE 3. Block Diagram for Proposed System

5.1. ABOUT PROPOSED SYSTEM

- Wearable sensors & IoT devices
- Data transmission
- Cloud & data processing
- Health care provider interface
- Patient mobile application

6. SOFTWARE ARCHITECTURE

The server used in this project plays a crucial role in enabling remote patient monitoring by acting as the central repository and processing unit for the sensor data transmitted from the ESP32. Here’s an in-depth explanation of its function, architecture, and significance:

Role and Functionality:

- **Data-reception:**
  The server is designed to receive HTTP GET requests sent from the ESP32-based monitoring system. Each request includes various sensor readings—such as encode as URL query parameters.
- **Data-processing-and-logging:**
  Once the server receives these requests, it extracts the sensor data from the query parameters. The server-side script (for example, a PHP script such as stored data or upload data PHP) processes the incoming data and logs it into a database. This logging allows for historical tracking of patient conditions, enabling retrospective analysis and trend identification.
- **Real-Time-monitoring:**
  By storing the data centrally, the server facilitates real-time remote monitoring. Healthcare providers or caregivers can access this data through a web-based dashboard, which can display live updates on the patient’s status. This is essential for ensuring that any abnormalities or emergencies are addressed promptly.

- **Alert-Generation-and-Notification:**
  In more advanced configurations, the server could also be programmed to analyse incoming data in real time. For instance, if a sudden drop in acceleration (indicating a fall) or a temperature spike is detected, the server can trigger additional notifications—such as sending SMS or email alerts—to the appropriate medical personnel.

7. SIGNIFICANCE FOR REMOTE PATIENT MONITORING

- **Centralized-Data-Repository:**
  By aggregating data from one or multiple devices, the server provides a centralized platform where healthcare providers can monitor patient conditions from anywhere. This is particularly beneficial in home-care scenarios or remote regions where continuous supervision is not feasible.
- **Historical-Data-Analysis:**
  The data logged on the server can be used for historical analysis to detect trends or patterns. For example, repeated fall incidents or gradual increases in body temperature can signal deteriorating health, prompting further medical investigation.
- **Scalability:**
  The server-based architecture is inherently scalable. As the number of monitored patients increases, the system can be expanded by optimizing the database or deploying additional servers to handle higher loads. This scalability ensures that the monitoring system can grow alongside the needs of a healthcare facility.
- **Integration-with-Other-Systems:**
  A server-centred design allows for integration with other healthcare systems and data analytics tools. Data from the monitoring system could be combined with electronic health records (EHR) or integrated into hospital information systems (HIS) to provide a comprehensive view of patient health.

8. LITERATURE REVIEW

Over the years, health monitoring systems have evolved significantly, leveraging advancements in sensor technology and wireless communication. Early systems primarily focused on periodic health checks, where data such as heart rate, blood pressure, and oxygen. Another approach involved the use of wearable devices, such as smartwatches and fitness trackers, which continuously monitored vital signs. These devices utilized Bluetooth or Wi-Fi to transmit data to a paired smartphone or directly to a cloud server. While these systems provided more frequent data updates, they were still limited by their dependency on external devices and networks. Moreover, they lacked the ability to detect falls or other physical emergencies, which are critical for elderly or vulnerable individuals. A significant advancement in this field is the adoption of *Lo-Ra (Long Range) * technology, which enables long-range, low-power wireless communication. LoRa is particularly well-suited for health monitoring systems, as it allows for continuous data transmission over long distances without draining the device's battery. This makes it ideal for remote or rural environments where cellular networks are unreliable. Furthermore, LoRa's cost-effectiveness and scalability make it an attractive option for widespread deployment.

9. SOFTWARE IMPLEMENTATION

The software implementation is equally detailed and crucial for transforming raw sensor data into actionable information. The code, developed using the Arduino IDE or Platform IO, is structured to initialize the system, acquire and process sensor data, provide user feedback, and communicate with remote servers.

Initialization:

- **Setting-Up-Sensors-LCD-and-WIFI-Connectivity:**
  In the initialization phase (the setup function), all sensors and peripherals are configured for operation.
  - The ESP32 begins by initializing the ADXL345 accelerometer, confirming its connection and setting its sensitivity range (e.g., 16G). Simultaneously, the DHT11 sensor is set up to ensure that

it can accurately sample temperature and humidity data.

- **LCD-Initialization:**
  The LCD is initialized using the Liquid Crystal library, which configures the data and control pins based on the previously defined pin assignments. A welcome message is displayed, indicating that the system is starting up.
- **WIFI-Connectivity:**
  The ESP32 attempts to connect to a pre-configured Wi-Fi network. This connection is critical because it enables the remote data transmission functionality. The setup code includes a loop that waits until a stable connection is established, ensuring that the system does not proceed without network connectivity.
- **GPS-Data-Acquisition:**
  The system also initiates the GPS module during setup. It starts reading serial data, parses NMEA sentences, and converts the raw location data into human-readable latitude and longitude values, setting a baseline location for the system.

10. HARDWARE IMPLEMENTAION

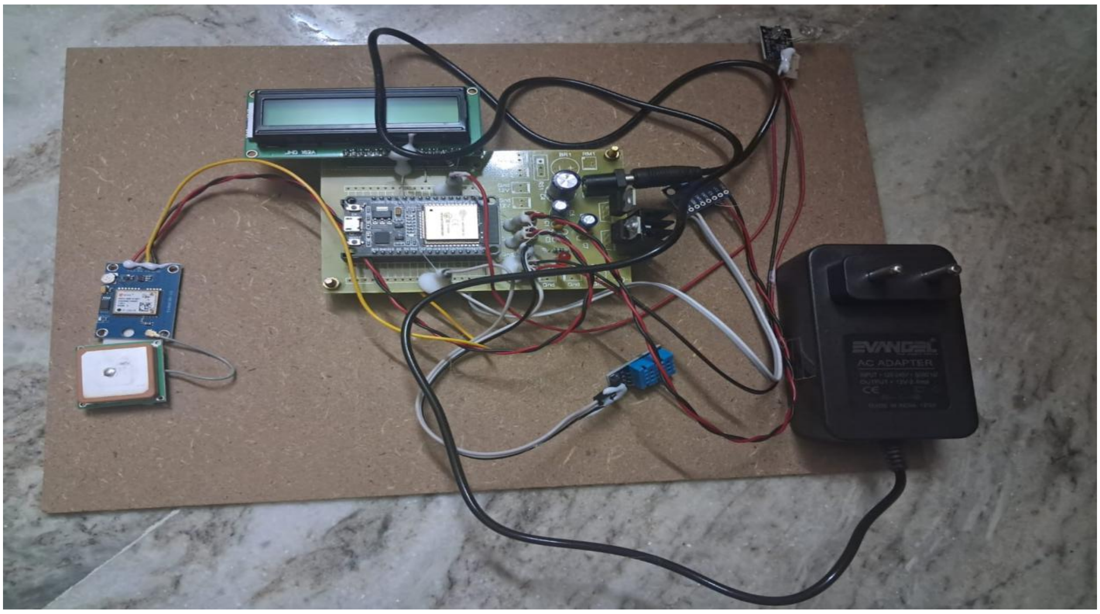


FIGURE 4. Circuit Kit

11. CONCLUSION

The IOT Realtime Remote Patient Monitoring System represents a significant step forward in leveraging affordable sensor technology and wireless communication to enhance patient care and safety. This project integrates an ESP32 micro controller with multiple sensors—including the ADXL345 accelerometer for fall detection, the DHT11 for temperature and humidity measurements, and a GPS module for location tracking into a cohesive system that delivers both local feedback via a 16×2 LCD display and remote notifications through Wi-Fi connectivity. In this conclusion, we recap the project's key functionalities and contributions, discuss the lessons learned, and outline future work to further enhance the system**.**

12. ACKNOWLEDGMENT

We sincerely express our gratitude to our mentors and faculty members for their invaluable guidance, technical insights, and continuous support throughout the research on the IOT based remote patient health monitoring system. Their expertise played a crucial role in refining our approach and overcoming technical challenges. We also extend our appreciation to the research community and open-source contributors whose pioneering work in esp32 technology and GPS systems provided a strong foundation for our development.

REFERENCES

[1]. Electronic wings, "Adxl345 accelerometer interfacing with esp32," Available: https://www.electronicwings.com/esp32/adxl345-accelerometer-interfacing-with-esp32.

[2]. Analog devices, "Adxl345 datasheet," 2021. Available:https://www.analog.com/media/en/technical-documentation/data-sheets/ADXL345.pdf.

[3]. Sparkfun,"Dht11 temperature & humidity sensor datasheet," Available: https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT11.pdf.

[4]. M. Merlin, "DHTESP library documentation," GitHub, available: Https://github.com/marcmerlin/dhtesp.

[5]. ESPRESSI systems, "Esp32 technical reference manual" 2023. Available: Https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf.

[6]. Arduino, "Arduino esp32 guide," Available: Https://www.arduino.cc/en/guide/arduinoesp32.

[7]. Arduino "Liquid-crystal library documentation," Available: Https://www.arduino.cc/en/reference/liquidcrystal.

[8]. ESPRESSIF systems, "Wi-fi library esp32," GitHub, available: Https://github.com/espressif/arduino-esp32/tree/master/libraries/wifi.

[9]. ESPRESSIF systems, "HTTP client library for esp32," Git-hub available: Https://github.com/espressif/arduino-esp32/tree/master/libraries/httpclient.

[10]. U-BLOX, "Neo-6 GPS module datasheet," Available: Https://www.u-blox.com/en/product/neo-6-series.

[11]. Author unknown, "A review on IOT-based remote patient monitoring systems," In IEEE *explore*, available through academics databases.

[12]. Arduino forum, "Community discussions on esp32 and sensors," Available: Https://forum.arduino.cc/.