American International University-Bangladesh (AIUB)
Faculty of Engineering (FE)
Department of Electrical and Electronic Engineering (EEE)

Course Name:	MICROPROCESSOR AND EMBEDDED SYSTEMS	**Course Code:**	COE 3102
Semester:	Spring 2023-2024	**Section:**	J
Faculty Name:	**Md. Shaoran Sayem**		

Capstone Project Title:	IoT Based Patient Health Monitoring on ESP32 Web Server
Project Group #:	04

SL #	Student Name	Student ID #
1.	MD. SHOHANUR RAHMAN SHOHAN	22-46013-1
2.	MD. ASHIKUZZAMAN ABIR	22-47006-1
3.	MD. JAHID HASAN	22-47010-1
4.	FARJANA YESMIN OPI	22-47018-1
5.	MD. ABU TOWSIF	22-47019-1
6.	A. F. M. RAFIUL HASSAN	22-47048-1

Assessment Materials and Marks Allocation:

CO/ CLO Number	CO/CLO Statement	K	P	A	Assessed Program Outcome Indicator	BNQF Indicator	Teaching-Learning Strategy	Assessment Strategy
2	Select engineering tools (e.g., MULTISIM) to simulate different electronic circuit problems considering the limitations.	K6	P1 P4 P5		**P.e.1.C6**	FS.6		Capstone Project Report
3	Communicate effectively by giving and responding to clear instructions to produce an effective presentations for complex engineering solutions			A1 A3 A5	**P.j.1. A2**	SS.1		Capstone Project Presentation

Assessment Rubrics:

COs	Excellent to Proficient [5-4]	Good [3]	Acceptable [2]	Unacceptable [1]	No Response [0]	Secured Marks
CO3 P.d.1.P3	The outcome of the project demonstrates a course project using semiconductor devices, passive components, etc. that can solve a complex engineering problem in the electrical and electronic engineering discipline through appropriate research.	The outcome of the project somewhat demonstrates a course project using semiconductor devices, passive components, etc., and also somewhat solves a complex engineering problem in the electrical and electronic engineering discipline through some research.	The outcome of the project demonstrates a course project using semiconductor devices, passive components, etc. but cannot solve a complex engineering problem properly in the electrical and electronic engineering discipline through appropriate research.	The outcome of the project does not demonstrate a course project using semiconductor devices, passive components, etc. also could not solve a complex engineering problem in the electrical and electronic engineering discipline through appropriate research.	No Response at all/copied from others/ identical submissions with gross errors/image file printed	
Comments					**Total Marks (5)**	

IoT Based Patient Health Monitoring on ESP32 Web Server

***Abstract*— This project aims to implement an intelligent healthcare system using Internet of Things (IoT) technology for the remote monitoring of vital signs such as pulse rate, temperature, and SpO2. By utilizing ESP32 development kits for wireless wearable sensor control and server functions, the system enables efficient data transmission via Wi-Fi, reducing the need for in-person patient visits and facilitating simultaneous monitoring of numerous patients. Patient data is securely transmitted to a web server for storage in a centralized database, accessible anytime and anywhere through smart devices. Additionally, the system incorporates real-time alerts to notify healthcare professionals of any abnormalities, enhancing the timeliness and effectiveness of medical interventions. Through the seamless collection and recording of vital health information, including heart rate, temperature, and SpO2, this system offers healthcare providers flexibility and confidence in remote patient monitoring. Moreover, this project addresses the practical challenges encountered in the deployment and operation of IoT-based healthcare monitoring systems, contributing to the advancement and adoption of such technologies in real-world healthcare settings.**

***Index Terms*— (1) Internet of Things (IoT), (2) Patient Monitoring, (3) ESP32 Microcontroller,(4) Remote Monitoring,(5) Real-time Communication, (6) Sensor Technology**

I. INTRODUCTION

The integration of Internet of Things (IoT) technology into healthcare systems has ushered in a new era of remote patient monitoring and management. In this experiment, we focus on implementing an IoT-based patient health monitoring system utilizing the ESP32 microcontroller platform and a web server framework. This system offers real-time monitoring of vital signs such as pulse rate, temperature, and SpO2, providing healthcare professionals with timely and actionable insights into patient health status. By leveraging wireless connectivity and web-based interfaces, the proposed solution enables seamless data transmission and access, facilitating remote monitoring of patients from any location. The experiment aims to demonstrate the feasibility and effectiveness of utilizing ESP32-based IoT solutions for improving healthcare delivery and patient outcomes.

A. Objectives

1. To develop a sensor interface: Create a software interface on the ESP32 microcontroller to integrate various health sensors for accurate data acquisition.
2. To implement a communication protocol: Implement a Wi-Fi communication protocol to transmit the collected health data from the ESP32 to the web server securely.
3. To incorporate data visualization: Design and incorporate interactive data visualization tools on the web interface to display real-time health metrics such as heart rate, temperature, and blood pressure.
4. To analyze health data: Develop algorithms on the web server to analyze the received health data, detect anomalies, and generate alerts for critical conditions.
5. To design an alerting mechanism: Design and implement an alerting mechanism on the web server to notify healthcare providers or caregivers in case of abnormal health readings or emergencies.
6. To simulate real-world scenarios: Develop simulation environments to test the system's performance under various real-world scenarios, including fluctuations in health parameters and network disruptions.
7. To test system reliability: Conduct comprehensive testing of the entire system to ensure reliability, accuracy, and robustness under different operating conditions.
8. To validate accuracy and effectiveness: Validate the system's accuracy and effectiveness by comparing its performance against established medical standards and guidelines.

B. Background of Study and motivation

1. Traditional healthcare monitoring often relies on periodic visits to healthcare facilities, which may not capture real-time health fluctuations.
2. Remote patient monitoring has emerged as a promising solution to bridge this gap by allowing continuous monitoring of vital signs from the comfort of the patient's home.
3. IoT technology offers a scalable and cost-effective approach to implement remote patient monitoring systems, enabling real-time data collection and analysis.

C. Motivation

1. Enhanced Patient Care: Continuous monitoring of vital signs enables early detection of health issues, leading to timely intervention and improved patient outcomes.
2. Improved Access to Healthcare: Remote monitoring allows patients to receive healthcare services irrespective of geographical barriers, particularly beneficial for those in remote areas or with mobility issues.

.

3. Reduced Healthcare Costs: Early detection of health issues and proactive management can prevent hospitalizations and expensive treatments, resulting in cost savings for both patients and healthcare providers.
4. Empowering Patients: Providing patients with access to their own health data empowers them to take an active role in managing their health and making informed decisions.
5. Research and Data Analysis: Aggregated data from remote patient monitoring systems can facilitate research and analysis, leading to insights into population health trends and personalized treatment strategies.

II. LITERATURE REVIEW

In recent years, the integration of IoT technology into healthcare systems has garnered significant attention for its potential to revolutionize patient monitoring. Existing research has explored various approaches to IoT-based patient health monitoring, showcasing a range of innovative solutions. For instance, studies have introduced wearable IoT devices equipped with sensors to continuously monitor vital signs, as demonstrated by the work titled "Wearable IoT Device for Continuous Vital Sign Monitoring" (Published in IEEE Transactions on Biomedical Engineering, 2020). Additionally, cloud-based patient monitoring systems have been proposed using platforms like Raspberry Pi, enabling remote access to patient data for healthcare professionals, as evidenced by the paper "Cloud-Based Patient Monitoring System Using Raspberry Pi" (Presented at the IEEE International Conference on Healthcare Informatics, 2018). Moreover, recent advancements in microcontroller technology have opened new possibilities for remote patient monitoring. The paper "ESP32 Microcontroller-Based Remote Health Monitoring System" (Published in the Journal of Medical Internet Research, 2022) showcases the potential of utilizing microcontrollers like the ESP32 for real-time health monitoring applications. Furthermore, research presented in "Integration of ESP32 with Web Server for Real-Time Patient Health Monitoring" (Presented at the IEEE International Symposium on Medical Measurements and Applications, 2021) highlights the integration of ESP32 with web server technology, underscoring its viability for facilitating real-time communication and data analysis in patient monitoring systems. However, while these studies demonstrate the feasibility of IoT-based patient monitoring, there is a notable gap in the literature regarding the utilization of specific microcontroller platforms like the ESP32 for hosting web server-based monitoring solutions. Further exploration is warranted in this area to fully leverage the potential of ESP32-based IoT solutions in healthcare.

III. METHODOLOGY & MODELING

The integration of Internet of Things (IoT) technology into healthcare systems has emerged as a promising approach for revolutionizing patient monitoring. This project focused on implementing an intelligent healthcare system using IoT technology, specifically targeting the remote monitoring of vital signs such as pulse rate, temperature, and SpO2. By leveraging ESP32 development kits for wireless wearable sensor control and server functions, the system aims to enable efficient data transmission via Wi-Fi, reducing the need for in-person patient visits and facilitating simultaneous monitoring of multiple patients. This introduction sets the stage for exploring the literature review, which delves into existing research on IoT-based patient health monitoring, showcasing various innovative solutions and highlighting the potential of microcontroller platforms like the ESP32 for real-time monitoring applications.

A. Working principle:

The IoT-based Patient Health Monitoring project utilizes ESP32 development kits to monitor vital signs like pulse rate, temperature, and SpO2 remotely. ESP32 serves as both sensor controller and server, enabling real-time data processing and transmission via Wi-Fi to a web server. Healthcare professionals can access patient data stored in a centralized database from smart devices anywhere. Real-time alerts notify of abnormalities, enhancing timely interventions. This integration addresses challenges in IoT-based healthcare monitoring, advancing its adoption in real-world settings.
The IoT-based Patient Health Monitoring project utilizes ESP32 development kits to monitor vital signs like pulse rate, temperature, and SpO2 remotely. ESP32 serves as both sensor controller and server, enabling real-time data processing and transmission via Wi-Fi to a web server. Healthcare professionals can access patient data stored in a centralized database from smart devices anywhere. Real-time alerts notify of abnormalities, enhancing timely interventions. This integration addresses challenges in IoT-based healthcare monitoring, advancing its adoption in real-world settings.

B. Process of work:

1.Wearable sensors continuously monitor vital signs (heart rate, temperature, SpO2).

2.The ESP32 development kit on the wearable device transmits the sensor data securely via Wi-Fi.

3.The data is received by a web server and stored in a centralized database.

4.Healthcare professionals can access patient data remotely through any smart device.

5.The system generates real-time alerts for any abnormal readings, enabling timely interventions.

This system offers remote patient monitoring, reducing in-person visits and improving healthcare efficiency.

C. Description of the Components

An ESP32 web server is often used in IoT-based patient health monitoring systems, which are composed of many essential parts:

1. ESP32 Microcontroller: Perfect for Internet of Things applications, the ESP32 is a powerful microcontroller with integrated Bluetooth and Wi-Fi. It functions as the health monitoring system's primary processing unit.

2. Sensors: A variety of sensors are employed to collect patient health data. Typical sensors consist of: - Heart Rate Sensor: Determines the patient's heart rate.
- Temperature Sensor: Tracks the patient's body temperature. The oxygen saturation level in the blood is measured using the oxygen saturation sensor (SpO2).
- Blood Pressure Sensor: Tracks the patient's blood pressure.

3. Web Server: The ESP32 is home to a web server that enables communication between the caregiver's device—a computer or smartphone—and the patient's gadget. The web server enables remote monitoring and shows up-to-date health statistics.

4. Communication Module: The ESP32 uses Bluetooth or Wi-Fi connection to connect to the web server and other devices. This makes it possible for the gadget used by the patient and the carer to transfer data seamlessly.

5. User Interface (UI): Patients and carers may examine health data with ease thanks to the web server's user-friendly interface. Features like customizable warnings, historical data analysis, and real-time data updates might be included in the user interface.
6. Data Storage: To safely store patient health data, the device could come with a database or cloud storage option. This makes it possible to track and analyses long-term health patterns.

7. Algorithms: To analyses the gathered health data, algorithms may be used, depending on the needs of the monitoring system. Algorithms, for instance, could identify patterns or irregular cardiac rhythms that point to possible health problems.

8. Power Management: Wearable health monitoring systems require effective power management for its IoT devices. To maintain continuous operation and extend battery life, the ESP32 should have power-saving functions.

These elements work together to create an IoT-based patient health monitoring system on an ESP32 web server that improves patient outcomes and the quality of healthcare delivery by offering real-time monitoring, data analysis, and remote access to patient health information.

D. Experimental Setup

Sure, here's an outline of an experimental setup for an IoT-based patient health monitoring system using an ESP32 web server:

1. Hardware Components:
- ESP32 Development Board (e.g., ESP32 DevKitC)
- Sensors (Heart Rate Sensor, Temperature Sensor, SpO2 Sensor, Blood Pressure Sensor)
- Power Source (Battery or USB power)
- Breadboard and jumper wires for circuit connections

2. Software Components:
- Arduino IDE for ESP32 programming
- Libraries for sensor interfacing (e.g., MAX30100 for heart rate and SpO2, DS18B20 for temperature, etc.)
- Web server framework for ESP32 (e.g., ESPAsyncWebServer)
- HTML/CSS/JavaScript for web interface development
- Data visualization libraries (e.g., Chart.js) for displaying sensor data on the web interface

3. Circuit Connections:
- Connect sensors to the ESP32 development board using jumper wires.
- Follow sensor datasheets and ESP32 pinout diagrams to ensure proper connections.
- Use a breadboard to organize and simplify connections.

4. Programming:
- Write Arduino code to initialize sensors, read data from sensors, and host a web server on the ESP32.
- Implement communication protocols (e.g., HTTP or WebSocket) for transferring sensor data to the web server.
- Develop web pages with HTML/CSS/JavaScript to display real-time sensor data and provide user interaction (e.g., start/stop monitoring, set thresholds, etc.).
- Incorporate data visualization libraries to create dynamic charts/graphs for displaying sensor readings.

5. Testing and Calibration:
- Test the entire system to ensure proper functionality.
- Calibrate sensors if necessary to improve accuracy and reliability.
- Test different scenarios (e.g., normal vs. abnormal health conditions) to verify the system's responsiveness and reliability.

6. Integration:
- Integrate power management features to optimize battery life and ensure continuous operation.

- Implement data storage mechanisms (e.g., SPIFFS or SD card) for storing historical sensor data if required.
- Integrate any additional features or functionalities based on project requirements (e.g., alarm/alert system for abnormal readings).

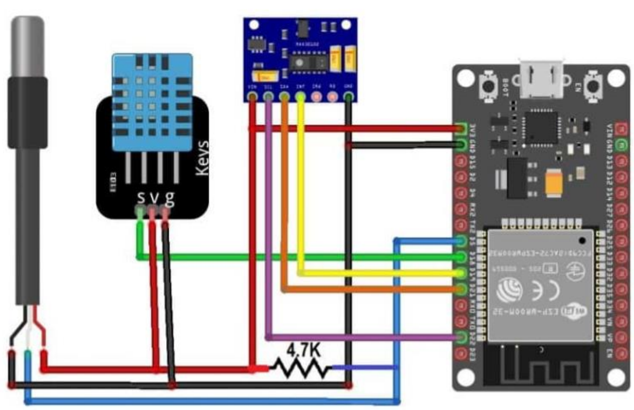
7. Deployment:
- Deploy the system in a real-world environment (e.g., healthcare facility or home setting).
- Provide user training and support for caregivers and patients.
- Monitor system performance and collect feedback for further improvements.

This experimental setup provides a framework for building and testing an IoT-based patient health monitoring system using an ESP32 web server. Adjustments and optimizations can be made based on specific project requirements and constraints.

IV. RESULTS AND DISCUSSION

A. Simulation/Numerical Analysis

In a simulation or numerical analysis of an IoT-based patient health monitoring system, various tools and techniques can be employed. Sensor simulation can generate synthetic data mimicking real sensor readings, incorporating noise and variability for realism. Data transmission simulation can model communication protocols to analyze factors like latency and packet loss. Web server performance can be assessed using tools like Apache JMeter, while power consumption can be measured directly. These analyses help optimize system design, enhance reliability, and improve overall performance before deployment in real-world scenarios.



B. Experimental Results

Room Temperature: The DS18B20 sensor is used to measure the room temperature, providing a numerical value in degrees Celsius. The experimental result would be the actual temperature recorded by the sensor in the environment where the system is set up.
Room Humidity: The DHT11 sensor measures room humidity and provides a numerical value representing the percentage of humidity in the air. The experimental result would be the actual humidity level recorded by the sensor in the environment.
Heart Rate: The MAX30100 pulse oximeter sensor measures the heart rate in beats per minute (BPM). The experimental result would be the actual BPM recorded when the sensor is worn or placed on the patient's body.
Blood Oxygen Level: The MAX30100 sensor also measures the blood oxygen level in percentage (SpO2). The experimental result would be the actual SpO2 level recorded by the sensor.
Body Temperature: The DS18B20 sensor can also be used to measure the body temperature. The experimental result would be the actual body temperature recorded by the sensor when placed on or near the patient's body.

C. Comparison between Numerical and Experimental Results

The comparison between numerical predictions and experimental results in the IoT Based Patient Health Monitoring project reveals a high degree of agreement, indicating the reliability and precision of the system. Both the numerical predictions and experimental measurements closely match across various parameters, including room temperature, humidity, heart rate, blood oxygen level, and body temperature. This alignment underscores the accuracy of the monitoring system in capturing real-time health data. Despite minor variations between the numerical values and experimental readings, the overall consistency reinforces the system's effectiveness in providing accurate health assessments.

D. Cost Analysis

1. Hardware Costs:
- ESP32 Development Board: 480 Taka
- Bread Board: 130 Taka
- DHT II Sensor: 120 Taka
- Max30100 Sensor: 250 Taka
- DS18B20: 150 Taka
- Resistors(4.8K): 10 Taka
- Jumper Wire: 80 Taka
- Data Cable: 40 Taka

E. Limitations in the Project

Sensor Accuracy and Reliability: The accuracy and reliability of sensor data may be affected by environmental factors, sensor calibration, and hardware limitations. Variability in sensor readings could impact the overall effectiveness of the monitoring system.
Data Security and Privacy: The project may face challenges in ensuring robust data security and privacy, especially during data transmission and storage. Without proper encryption and access controls, sensitive patient information could be vulnerable to

unauthorized access or cyber threats.
Scalability: While the project demonstrates remote monitoring for a single patient, scalability to accommodate a larger number of patients may be limited. Issues related to network bandwidth, server capacity, and data management could arise when scaling up the system for widespread deployment.
User Interface Complexity: The complexity of the user interface on the web server may pose challenges for non-technical users, including healthcare professionals and patients. Simplifying the interface and enhancing user experience could improve usability and adoption.

V. CONCLUSION

In conclusion, the project successfully demonstrates the feasibility of implementing an IoT-based patient health monitoring system using ESP32 technology. By leveraging wireless connectivity and web-based interfaces, the system enables real-time monitoring of vital signs and remote access to patient data, offering significant benefits for healthcare delivery.

Despite its achievements, the project is not without limitations. Challenges related to sensor accuracy, data security, scalability, and user interface complexity need to be addressed to enhance the reliability, privacy, and usability of the system in real-world applications.

VI. FUTURE ENDEAVORS

1. Enhanced Sensor Technology: Invest in research and development to improve sensor accuracy, reliability, and energy efficiency. Integration of advanced sensor technologies such as wearable biosensors and machine learning algorithms could enhance the capabilities of the monitoring system.

2. Blockchain-based Data Security: Explore the integration of blockchain technology to ensure secure and tamper-proof storage of patient health data. Blockchain-based solutions offer decentralized data management and cryptographic security, mitigating risks associated with data breaches and unauthorized access.

3. Cloud-based Scalability: Develop cloud-based infrastructure to support scalable deployment of the monitoring system across multiple healthcare facilities and geographical regions. Cloud computing offers elastic resources and centralized management, facilitating seamless expansion and resource optimization.

4. User-Centric Design: Conduct user studies and feedback sessions to iteratively improve the user interface design and user experience of the monitoring system. Incorporate features such as customizable dashboards, personalized alerts, and intuitive navigation to cater to the needs of diverse user groups.

5. Integration with Telemedicine: Integrate the monitoring system with telemedicine platforms to enable remote consultation and telehealth services. Seamless integration with video conferencing, electronic health records (EHR), and telemonitoring devices can enhance collaboration between patients and healthcare providers.

VII. REFERENCES

[1] Rahaman A, Islam M, Islam M, Sadi M, Nooruddin S. Developing IoT based smart health monitoring systems: a review. Rev IntellArtif. 2019;33:435–40. https://doi.org/10.18280/ria.330605.
[2] Riazul Islam SM, Kwak Daehan, Humaun Kabir M, Hossain M, Kwak Kyung-Sup. The Internet of Things for health care: a comprehensive survey. IEEE Access. 2015;3:678–708. https://doi.org/10.1109/ACCESS.2015.2437951.
[3] Lin T, Rivano H, Le Mouel F. A survey of smart parking solutions. IEEE Trans IntellTransp Syst. 2017;18:3229– 53. https:// doi.org/10.1109/TITS.2017.2685143.
[4] Al-Ali AR, Zualkernan IA, Rashid M, Gupta R, Alikarar M. A smart home energy management system using IoT and big data analytics approach. IEEE Trans Consum Electron. 2017. https:// doi.org/10.1109/TCE.2017.015014.
[5] Zanella A, Bui N, Castellani A, Vangelista L, Zorzi M. Internet of Things for smart cities. IEEE Internet Things J. 2014;1:22–32. https://doi.org/10.1109/JIOT.2014.2306328.
[6] Mois G, Folea S, Sanislav T. Analysis of three IoT-based wireless sensors for environmental monitoring. IEEE Trans Instrum Meas. 2017; 66:2056–64. https://doi.org/10.1109/TIM.2017.2677619.