IoT Based Patient Health Monitoring System

Hagar Gaber, Duha Al-Meraghi, Noora Al-Qahtani, Marim Elhanafy, Fajer Robelah

***Abstract*—In today's world, health monitoring is a crucial issue as patients often face major health problems due to improper health monitoring. Our project aims to develop a real-time health monitoring system that utilizes Internet of Things (IoT) technology to track patients' health parameters, such as heart rate, blood oxygen level, and body temperature. The system comprises three microcontrollers functioning as masters, each connected to a specific sensor, and communicates with a slave microcontroller via ESP-NOW communication protocol (many-to-one configuration). The project was successfully implemented, and each microcontroller was tested to deliver accurate results. Also, the implementation and testing of the Quality of Service (QoS) metrics under varying distance scenarios is completed and we ensured that it delivers the best performance possible.**

***Index Terms*—Internet of Things (IoT), Healthcare devices, Slave and master microcontrollers, ESP-32S development board (ESP-WROOM-32), MAX30102 I2C pulse oximeter and heart rate sensor, DS18B20 One-Wire waterproof temperature sensor, DHT11 digital humidity temperature sensor, ESP-NOW communication protocol.**

I. Introduction

THE healthcare industry is experiencing a digital transformation through the integration of Internet of Things (IoT) technology. IoT refers to the connection of everyday devices and appliances to the internet, enabling them to communicate with each other and collect and share data. The use of wearable healthcare devices that continuously monitor patient's vital signs has become increasingly popular in recent years. Our goal is to develop a real-time health monitoring system that tracks patients' health parameters, such as heart rate, blood oxygen level, and body temperature, using a set of sensors.

The system comprises three microcontrollers that function as senders/masters, each connected to a specific sensor, and communicate with a slave microcontroller using ESP-NOW communication protocol (many-to-one configuration). The system can be used directly by patients or by nurses in health centers to monitor patients' health status. By providing healthcare professionals with real-time data, the project aims to improve the quality of healthcare and enable them to take appropriate actions as needed. The system aims to optimize QoS metrics to ensure efficient and reliable data transmission. The project will utilize the distance between microcontrollers as a key QoS metric to evaluate the system's performance under varying distance scenarios and optimize it to deliver the best performance possible. In this report, we will present the design, implementation, and evaluation of our system, highlighting the challenges and solutions we encountered during the development process.

II. System Model

A. High-Level Architecture

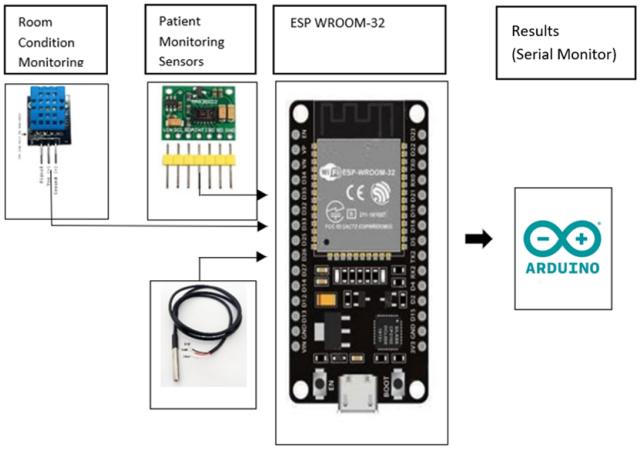

Fig. 1: High-level system architecture

B. Detailed System Design

For the detailed system design, we first started by connecting each of the three sensors, MAX30102 I2C Pulse Oximeter and Heart rate Sensor, DS18B20 One-Wire Waterproof Temperature Sensor, and DHT11 Digital Humidity Temperature Sensor, to one microcontroller in the hardware phase. Afterward, we wrote the code for each sensor's microcontroller in the software phase, followed by testing each one separately in the serial monitor. We will then evaluate the results to ensure they are accurate and represent real measurements.

Fig. 2: ESP-NOW communication protocol (many-to-one configuration)

Next, we established connections between each sensor and the receiver/slave using the ESP-NOW communication protocol in a many-to-one configuration, as illustrated in Fig. 2.

Each board was assigned a unique MAC address. To transmit messages via ESP-NOW, it is necessary to have the MAC address of the receiver board [1]. Please refer to the attached documents for the code related to obtaining the MAC address, as well as the sender and receiver code.

Subsequently, we conducted testing of the system at various distances to evaluate its performance. The distances tested were 1, 100, 200, and 300 meters, as illustrated in Fig. 3, using the link provided in [2].

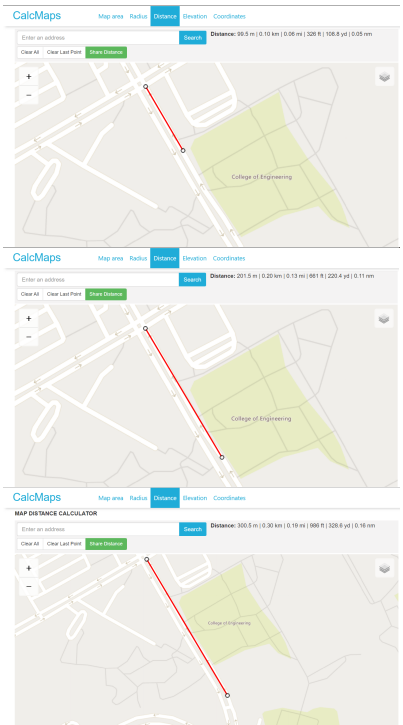

Fig. 3: 100, 200, and 300 meters inside Qatar University

At each distance, we measured and compared the delay and accuracy of the data between the slave and the three other microcontrollers. We also analyzed the performance of the ESP-NOW communication protocol. By comparing the results obtained at all distances, we were able to identify any differences in the system's performance and make adjustments as necessary. Our testing and analysis enabled us to optimize the system's design and performance, ensuring that it meets the needs of healthcare systems.

Finally, we integrated all sensors into one microcontroller to develop a multi-purpose device model that can be connected to the slave via ESP-NOW communication protocol (many-to-one configuration). This integration will ensure that the system is easily accessible to healthcare providers and patients alike. The real scenario for the system's use is when a nurse monitors all the patient's parameters from different rooms at varying distances.

1) Connectivity Diagram: The Connectivity diagram showcases the physical connections and wiring of the system as illustrated in Fig. 3, which shows all connections to one microcontroller to save space. This was done to avoid cluttering the diagram with individual images of each connection between one sensor and one microcontroller.

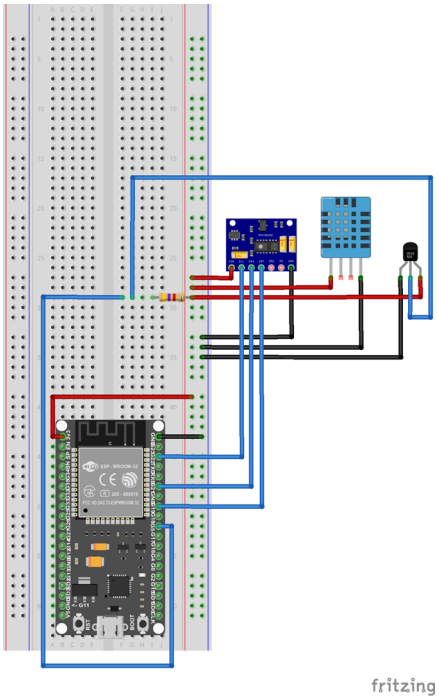

Fig. 4: Overall connectivity of the system

2) Proposed software architecture: In the overall system logic, as depicted in Fig. 4, the data flow starts from the sensors, which transmit data to the microcontrollers. Then, the data is transmitted from the microcontrollers to the serial monitor. However, it is important to note that the figure does not show the data being transmitted from microcontrollers to the slave microcontroller via ESP-NOW. Once received by the slave microcontroller, the data is then displayed on the serial monitor.

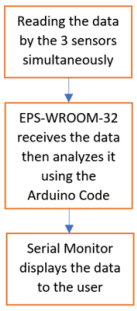

Fig. 5: Logic flow chart

3) Hardware Used: Table I contains a list of all hardware components used and their purpose in the project.

4) Software Used: In terms of software selection, we have utilized Fritzing software to design the circuit and create the connectivity diagram. Additionally, we have employed the Arduino IDE to write the code for the Arduino and display the results. The logos of both the Arduino IDE and Fritzing software can be seen in Fig. 5.

TABLE I: List of the components used in the project.

Component	Quantity	Usage
ESP-32S Development Board (ESP-WROOM-32) [3]	4	The ESP32 is a low-cost, low-power Microcontroller with Wi-Fi and built-in bluetooth (ESP-NOW communication protocol). ESP32 is used to implement the sensors used for tracking patients' health.
MAX30102 I2C Pulse Oximeter and Heart rate Sensor [4]	1	The MAX30102 pulse oximeter and heart rate sensor is an I2C-based low-power plug-and-play biometric sensor. It is used in two parts one is heart rate measurement and another is blood oxygen level measurement.
DS18B20 One-Wire Waterproof Temperature Sensor [5]	1	The DS18B20 sensor measures temperature with a minimal amount of hardware and wiring. It is used to send accurate temperature readings directly to the development board.
DHT11 Digital Humidity Temperature Sensor [6]	1	The DHT11 is a basic, ultra-low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air.
Resistor	1	The resistors are predominantly used to lower the flow of current, divide voltages, block transmission signals, and bias active elements.
Connecting Wires	7-10	Wires are used to connect sensors with ESP32 in breadboard.
Breadboard	4	Breadboard is used in making all the sensors connect with the ESP32 using connecting wires.

Fig. 6: Software IDEs used in the project

III. Implementation

For the implementation, we highly recommend referring to the comments provided in Appendix A, as they offer crucial details and tips for the hardware implementation and code execution. Furthermore, detailed illustrations depicting the connection of each sensor to the microcontroller can be found in Appendix B. First, we connected and tested each sensor individually, measuring the delay and accuracy. Then, we integrated all the sensors into a single microcontroller, as shown in Fig. 6. This integration aims to facilitate easy access to the system, particularly when a nurse monitors the patient's parameters from different rooms at varying distances.

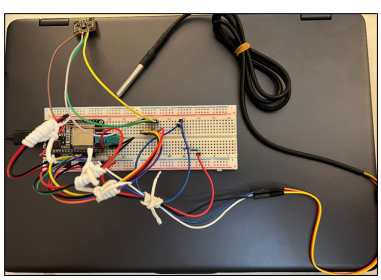

Fig. 7: Multi-Purpose Device Model Implementation

IV. Results and Evaluation

Here is the results of each sensor:

A. MAX30102

Fig. 8: Results of MAX30102 without finger

Fig. 9: Results of MAX30102 with finger

B. DHT11

Fig. 10: Result of DHT11 sensor

C. DS18B20

Fig. 11: Result of DS18B20 sensor

TABLE II: Comparison between the tolerable ranges and the practical measurements

	Tolerable Range	Practical Measurement
Heart Rate	56 – 104 beats per minute (BPM) [7]	75 BPM
SPO2	94 – 98% [8]	99.57%
Room Humidity	between approximately 25% RH and 60% to 70% RH [9]	59%
Room Temperature	25 – 28 °C [10]	25.8 °C
Body Temperature	35.5–37.0 °C [11]	34.56 °C

According to Table II, most of the readings fall within their respective tolerable ranges. While it is demonstrated that the heart rate and room temperature readings follow the expected range, the SPO2, room humidity, and body temperature measurements were found to be slightly in error.

Table III showcases the results of communication range testing between each two boards, and we were able to get a stable communication up to 200 meters in open field. In this experiment both ESP32 on-board antennas were pointing to each other.

As the distance between the boards increased, the delay in data transmission also increased. This suggests that the time required for data to travel between the boards becomes longer as the distance increases. The delay values varied

TABLE III: Comparison between the distance and the delay and accuracy of the data

MAX30102		
Distance (m)	Delay (sec)	Accuracy (%)
1	2	100
100	48	100
200	100	100
300	179	50
DHT11		
Distance (m)	Delay (sec)	Accuracy (%)
1	1	100
100	39	100
200	93	100
300	170	60
DS18B20		
Distance (m)	Delay (sec)	Accuracy (%)
1	1	100
100	31	100
200	90	100
300	163	70

across the different distances tested for each sensor, but in general, as the distance doubled, the delay tended to increase by approximately 2 to 3 times.

The accuracy of data transmission remained consistently high for shorter distances and then decreased at longer distances. For the tested sensors, the accuracy was 100% at 1 meter and 100 meters for all sensors. However, at distances of 200 meters and beyond, the accuracy began to decline. Specifically, at a distance of 300 meters, the accuracy dropped to 50% for the MAX30102 sensor, 60% for the DHT11 sensor, and 70% for the DS18B20 sensor.

It's important to note that these results were obtained in an open field with the antennas pointing towards each other, and the performance may vary in different environments or with different setups.

V. Conclusion

In conclusion, the implementation of the system was successful. It was observed that the delay in data transmission increased as the distance between the boards increased, while the accuracy remained high for shorter distances but decreased for longer distances. These findings highlight the importance of considering distance-related factors when designing and implementing communication systems between boards or devices. The experiments demonstrated that stable communication was achieved up to a distance of 200 meters for all the tested sensors. However, it should be noted that beyond this range, the accuracy of data transmission gradually declined.

The implemented system holds significant importance for healthcare providers, as it enables them to monitor patients' parameters wirelessly. However, it is crucial to consider the environment in which the system will be used, particularly within hospital settings where walls and rooms can introduce potential obstacles to signal stability. The tested system may not provide optimal performance under such conditions. Therefore, further experiments and evaluations are necessary to assess the system's stability and performance within a realistic hospital environment.

References

[1] "ESP-NOW: Receive Data from Multiple ESP32 Boards (many-to-one) — Random Nerd Tutorials." https://randomnerdtutorials.com/esp-now-many-to-one-esp32/.

[2] "CALCMAPS - Map Tools," CalcMaps. https://www.calcmaps.com/.

[3] ESP-WROOM-32 Datasheet: esp32_datasheet_en.pdf (espressif.com)

[4] MAX30102 Datasheet: MAX30102–High-Sensitivity Pulse Oximeter and Heart-Rate Sensor for Wearable Health (smart-prototyping.com)

[5] DS18B20 Datasheet: DS18B20.pdf (shopify.com)

[6] DHT11 Datasheet: DHT11 Humidity & Temperature Sensor (mouser.com)

[7] Umetani K, Singer D, McCraty R, et al. Twenty-Four Hour Time Domain Heart Rate Variability and Heart Rate: Relations to Age and Gender Over Nine Decades. J Am Coll Cardiol. 1998 Mar, 31 (3) 593– 601. https://doi.org/10.1016/S0735-1097(97)00554-8

[8] Aneman, A. (2012). SpO2 targets—How normal is normal?. Resuscitation, 83(10), 1175-1176. https://doi.org/10.1016/j.resuscitation.2012.07.026.

[9] Toftum, J., & Fanger, P. O. (1999). Air humidity requirements for human comfort. ASHRAE Transactions, 105, 641. Retrieved from http://0-search.proquest.com.mylibrary.qu.edu.qa/scholarlyjournals/air-humidity-requirements-human-comfort/docview/192573404/se-2

[10] Meng ZL. [Studies on the health standard for room temperature in cold regions]. Zhonghua Yu Fang Yi Xue Za Zhi. 1990 Mar;24(2):73-6. Chinese. PMID: 2364801.

[11] Sund-Levander, M., Forsberg, C., & Wahren, L. K. (2002). Normal oral, rectal, tympanic and axillary body temperature in adult men and women: a systematic literature review. Scandinavian journal of caring sciences, 16(2), 122-128. https://doi.org/10.1046/j.1471-6712.2002.00069.x

Appendix A

1. For MAX30102 please consider the following:

a. The sensor is used to show heart rate and SPO2 levels and it uses the Optical SP02 Detection (SPK Algorithm) using the MAX30105 Breakout.

b. When you press your finger against the sensor it varies enough to cause the blood in your finger to flow differently which causes the sensor readings to go wonky.

c. Hardware Connections (Breakout board to Arduino):

i. 5V = 5V (3.3V is allowed)

ii. GND = GND

iii. SDA = A4 (or SDA)

iv. SCL = A5 (or SCL)

2. For DHT11 please consider the following:

a. To read from the DHT sensor, we will use the DHT library from Adafruit.

b. Connect pin 1 (on the left) of the sensor to +5V

c. Connect pin 2 of the sensor to whatever your DHTPIN is.

d. Connect pin 3 (on the right) of the sensor to GROUND.

3. For DS18B20 please consider the following:

a. To interface with the DS18B20 temperature sensor, you need to install the One Wire library by Paul Stoffregen and the Dallas Temperature library.

b. New temperature readings are requested every 5 seconds.

APPENDIX B

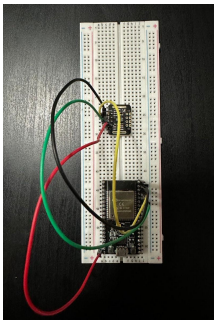

Fig. 12: MAX30102 sensor hardware implementation (Master)

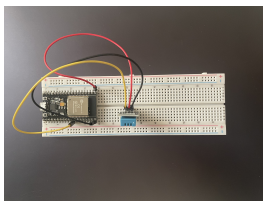

Fig. 13: DHT11 sensor hardware implementation (Master)

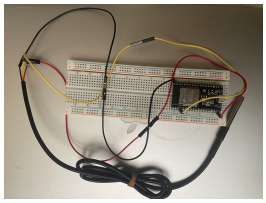

Fig. 14: DS18B20 sensor hardware implementation (Master)

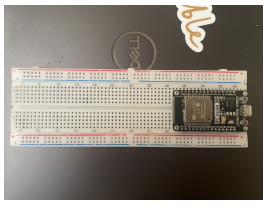

Fig. 15: Master sensor (Will be used in the final phase)