EIRA Journal of Computational Intelligence and Engineering Technologies (EIRAJCIET)
Journal homepage: https://eirapublisher.com/eirajciet/
Email: office.eirapublisher@gmail.com
ISSN: XXXX-XXXX (Online)
Volume 1, Issue 1, (September-October) 2026
DOI:

Design of Health Monitoring System and Health Collection Data

ENE NIKECHUKWUM

Department of Electrical & Information Engineering, Covenant University

Article history:
Received: 15/06/2026
Accepted: 07/082026
Published: 02/09/2026

Keywords: *Internet of Things (IoT), Patient health monitoring.*

***Corresponding Author**:
ENE NIKECHUKWUM

Abstract **Original Research Article**

The study examined design of health monitoring system and health collection data . With inadequate and inconsistent health monitoring techniques and technology, the issue of sudden deaths has become of great concern. In this study, the issue of inconsistent patient health monitoring was expatiated upon. To address the above-mentioned issue, an IoT based patient health monitoring system was proposed to aid medical practitioners to monitor their patients beyond the four walls of their medical firms. The proposed solution was a device capable of attaining health data from its users and transmitting the data to an online database where they can be accessed, through a website, from any location. By incorporating a heart rate and pulse oximetry sensor, a body temperature sensor, an environmental temperature and humidity sensor, and a Wi-Fi-enabled microcontroller unit, health data was able to be sensed and transferred to a database over the internet where they could easily be retrieved. Various tests of the proposed system's health sensors were performed and compared with results from standard medical equipment, showing minimal variations between the two sets of results and thus giving the system viability. Patient health monitoring plays a crucial role in medical procedures and through the implementation of this system, monitoring can be made more frequent for patients with less demanding underling health conditions.

How to cite this article: ENE NIKECHUKWUM. (2026). Design of health monitoring system and health collection data. EIRA Journal of Computational Intelligence and Engineering Technologies (EIRAJCIET), 1(1). 01-11.

Introduction

Health care has been one of the necessities of humanity for time immemorial as we need to be in sustainable health to perform daily tasks and survive. With the passing of time technology has progressed and evolved from one state to another, each state more powerful and useful than the last. It has enabled us to perform tasks that would be near impossible to carry out humanly, especially in the field of health and medicine. The everlasting desire for good health has brought about the rapid development of medical techniques and technology, making the healthcare industry one of the largest and fastest growing industries in the global economy [1].

Many innovative medical inventions have been made over the past centuries which have utilized the advancements in science and engineering to perform medical operations more efficiently and precise than ever before with popular examples being the electrocardiograph machine and the magnetic resonance imaging (MRI) scanner. Also, there have been a multitude of scientific breakthroughs in medicine such as the development of vaccines in the 18th century and the discovery of penicillin in 1929 [2] [3].

Health institutions depend on various forms of technology to properly administer and monitor patients in their firms, thus the medicine and health sector indisputably benefits from technological advancements. One of the more modern advances in technology is the internet of things (IoT) which is a system of interrelated computing devices, capable of transferring data over networks without the need for human intervention. Such devices or 'things' usually contain sensors that collect data and transmit the data to an IoT gateway which then transmits the collected data over a network such as the internet. The Internet of Things is a rapidly expanding technological trend that enables devices or gadgets that are connected through a network (such as the internet) to gain the ability to acquire, process, and share information. The internet of things is known as a technology that has demonstrated its existence as a network of machines or gadgets that can communicate with one another [4].

Problem Statement

In many countries of the world, especially the developing countries, the lack of proper patient health monitoring has become a terror. Statistics show in [7] that more than 40% of adults in Nigeria are estimated to have manageable diseases like high blood pressure, heart diseases and hypertension, but most of these patients have met with various disabilities and in some cases untimely death due to lack of adequate patient health monitoring and insidious manual system of monitoring utilized in the region, as observed in [8]. This project proffers a solution which is an IoT based patient health monitoring system that is capable of monitoring patient's vitals beyond the walls of a health firm.

Aim and Objectives

Aim

This project aims to successfully design of health monitoring system and health collection data

Objectives

The objectives of the project include:

i. Design a health monitoring system able to collect health data from users.

Literature Review

As science and technology have greatly advanced over the past centuries, so have medicine and health practices but despite the advancements, consistent availability of adequate health care still remains a global challenge. With the inclusion of certain technologies, such as the Internet of Things (IoT), patient health monitoring can become wireless, thus becoming more convenient and accessible than traditional health monitoring.

Patient Health Monitoring

Patient health monitoring is a process of observing the vital signs of patients with the aim of prognosing ailments to treat them. Traditionally when patients seek medical attention from medical firms, their vitals are first checked by the nurses and then they proceed to the doctors for a diagnosis. The doctors use their discretion to determine whether the patient is to be hospitalized or not. The inpatients (hospitalized patients) are continually monitored by medical personnel of the medical firm while the outpatients are given prescriptions and medical advice then leave. In modern-day hospitals, there are various facilities and equipment that medical personnel use to continually monitor their patients' health. Patient health monitoring can be classified based on target parameters to be monitored/measured in the patients, some of which include:

i. **Respiratory monitoring:** These are the processes that involve monitoring the patients' respiratory system to determine if the blood cells are properly oxygenated and carbon dioxide (CO_2) is adequately passed out of the body. Some respiratory monitoring processes involve: Pulse oximetry, which is the use of an oximeter to measure the saturated percentage of oxygen in the blood (SpO_2) and Capnography, which involves using a capnometer to measure the partial pressure of CO_2 in patients' airways during inspiration and expiration.

ii. **Cardiac monitoring:** These are the monitoring processes that involve the measurement of patients' heart rate and cardiac rhythm. One of the most commonly used means of cardiac monitoring is through electrocardiography (ECG) where electrodes are placed on different parts of the patient's chest to measure/record electrical signals generated from the heart, thus producing an electrocardiograph [9]. Another common means of cardiac monitoring can be through the use of a stethoscope which is an acoustic medical device used to listen for internal sounds in a patient's body, like the heartbeats.

iii. **Hemodynamic monitoring:** Hemodynamics is the study of blood flow. Hemodynamic monitoring, simply put, are the processes used to measure a patient's general blood circulation. Hemodynamics provides vital physiological information such as arterial blood pressure, tissue perfusion, and oxygenation of tissues and organs. Hemodynamic monitoring systems use catheters which are long medical tubes inserted into the patient, allowing for blood flow observations to take place [10] [11]. Cardiac monitoring and hemodynamic monitoring are usually performed simultaneously, especially for critically ill patients.

iv. **Neurological monitoring:** Also known as intraoperative neuropsychological monitoring or simply, neuromonitoring. It involves assessments of the integrity of various neural structures during surgical operations to prevent unintentional damage to the patient's nervous system. Usually, a neuromonitoring technician is assigned with special monitoring software which he/she uses to detect potential nerve damage that could go unnoticed during the operations. Parameters such as the patient's cranial blood pressure, cerebral perfusion, and blood flow, as well as electric potentials from various nerves of the patient's nervous system and brain, are monitored.

v. **Blood glucose monitoring:** Blood glucose monitoring is the testing and observation of the concentration of glucose in the blood. Traditionally, blood glucose tests are performed through invasive means such as piercing the patient's skin to attain the blood sample to be tested but there are non-invasive means such as Continuous Glucose Monitoring (CGM) which uses sensors that measure the amount of glucose in the interstitial fluid around the patient's skin cells [12]. Glucose monitoring is especially useful in preventing or reversing diabetes and hyperglycemia (high blood sugar). Monitored health parameters of inbound patients are usually displayed on a bedside monitor such as the one seen below in figure 2.1.

Figure 2.1: A bedside monitor displaying health data of patient being monitored Source: Adapted from [11]

Sensors

A sensor is a device that detects or measures physical quantities from its surroundings and converts it into data that can be interpreted by humans or machines. Electronic sensors are a kind of sensors that convert the physical inputs they measure to electronic data. There are two main types of electronic sensors:

i. Analog sensors: These types of electronic sensors measure external parameters and produce a continuous voltage (signal) as output. Their outputs can range from 0v – 5v; 0v – 3.5v (Low logic) and 3.5v – 5v (High logic).

ii. Digital sensors: On the other hand, digital sensors measure their specified external parameters and produce a discrete pulsating voltage as output. Their outputs are binary (0s and 1s).

Microcontrollers

A microcontroller is a compact integrated circuit that controls one or more operations in an embedded system. On a single chip, a microcontroller contains a processor, memory, and input/output (I/O) peripherals. Also referred to as an embedded

controller or microcontroller unit (MCU), they can be found in a variety of devices, including automobiles, robots, office equipment, medical devices, mobile wireless transceivers, vending machines, and home appliances. Some examples of MCUs are the Arduino Uno and the ESP32 as seen in figure 2.2.

A microcontroller accomplishes its tasks by evaluating data received from its I/O peripherals using its core CPU. Temporary data is stored in the microcontroller's data memory, from which the processor retrieves it using program memory instructions to decode and apply the incoming data. Then it interacts with its I/O peripherals and performs the necessary operation. The principal components of a microcontroller are as follows:

i. **The processor (CPU):** A processor can be thought of as the device's brain. It processes and responds to a variety of commands that control the operation of the microcontroller. This involves fundamental arithmetic, logic, and input/output (I/O) operations. It also sends commands to other embedded system components via data transfer operations.

ii. **Memory:** The memory of a microcontroller is used to retain the data that the processor receives and uses to respond to instructions that it's been programmed to carry out. There are two main memory types, the program memory which is used to store long-term data on the instructions to be carried out, and the data memory which stores temporary data.

iii. **I/O peripherals:** The processor's interface with the outside world is provided through its input and output devices. The processor receives data from the input ports in binary format. The output devices, which carry out functions not under the control of the microcontroller, get the data from the processor and are given the necessary instructions.

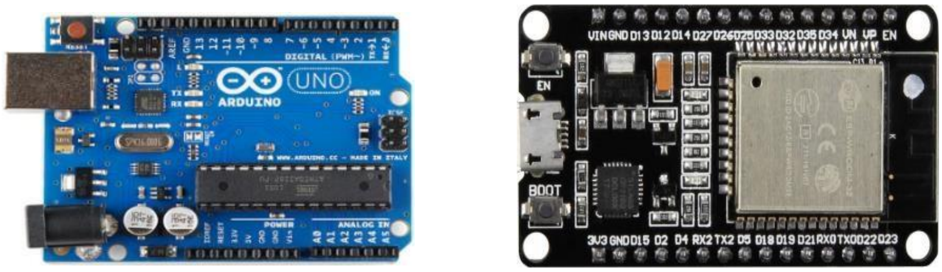


Figure 2.2: Examples of microcontroller units - Arduino Uno (left), ESP32 (right)

Methodology

Design Specifications

A design specification is a detailed engineering document that provides explicit information regarding a product or a process. Design specifications are aimed at providing clear and unambiguous definition of the functions, features and performance of a product or process. The proposed system design of this project is divided into two:

A. Hardware design: The specifications involving the principal hardware components; components of which include:

1. MAX30100 heart rate and pulse oximetry sensor
2. DS18B20 temperature sensor
3. DHT11 temperature and humidity sensor
4. OLED display module
5. ESP32 DEV KIT V1 MCU

B. Software design: The specifications of the programs, programming tools and environments used in the project. The software tools and environment used include:

1. Arduino IDE
2. Google's Firebase online database
3. Test website for displaying collected data

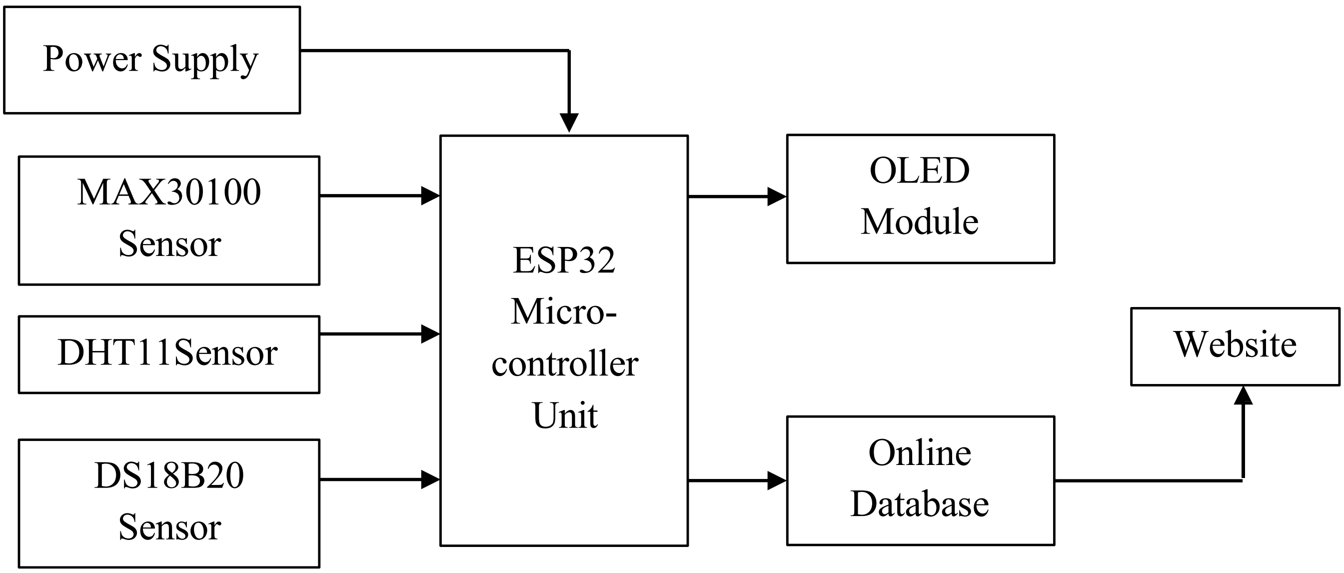


Figure 3.1: Block diagram of the project

Hardware design

Observing the block diagram provided in figure 3.1, the operations of the hardware components used in the project can be categorized into four stages – power stage, sensing stage, control stage and display stage.

Power Stage

The system is powered by anyone of two means: through a battery supply or through a micro-USB connection to an external power source. A 3.7v battery was chosen as a means to supply power and based on the 3.3v requirement of the ESP32 MCU, both means of supplying power to the system are regulated by the ESP32 by an internal Linear dropout (LDO) voltage regulator.

Voltage regulation: This is the process of providing a near constant voltage over a wide range of loads. As seen in figure 3.2, the ESP32 possesses an AMS1117 Linear dropout voltage regulator that regulates the supplied voltage to about 3.3v. The output voltage from the voltage regulator can be gotten by,

$$V_{OUT} = V_{IN} - V_{DROP} \quad (3.1)$$

where VIN is the input voltage, VDROP is the voltage drop of the LDO and VOUT is the resulting output voltage.

The designed VDROP for an AMS1117 voltage regulator is 1.3v, therefore let VDROP = 1.3v,

Since the desired output is 3.3v, let VOUT = 3.3v,

From equation (3.1) stated above, make VIN the subject and substitute the values for VOUT and VDROP,

VIN = VOUT + VDROP

VIN = 3.3v + 1.3v (3.2)

Solving equation (3.2) will result in VIN = 4.6v

From this result a 3.7v battery was chosen since its voltage is within a safe range and yet high enough to power the system.

Also, the same voltage regulation will be performed on power received through the micro USB port, ensuring proper voltages for the microcontroller's operations.

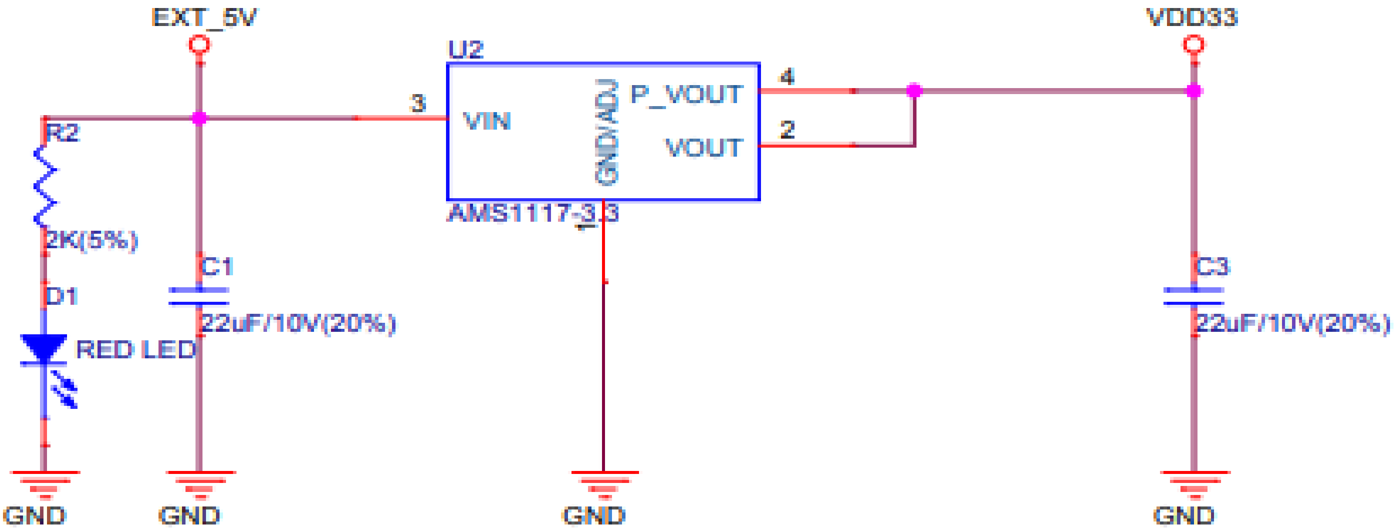


Figure 3.2: Schematic diagram of the power supply segment of the ESP32 MCU

Soldering

This is the connection of a conductor or component to a circuit board using a soldering iron and a soldering lead wire. This part was done after the initial testing was conducted on the beard board.

PCB Manufacturing

The PCB used in this project was manufactured using the Toner transfer method while making use of the following equipment:

1. LaserJet printer
2. Electric iron
3. 4" x 6" sticker
4. Piece of copper coated sheet
5. Liquid Ferric chloride (CI3Fe)
6. Mini drill machine
7. Liquid rosin

The following steps are taken when using the Toner transfer method;

1. Take a sticker, and peel off its backside and paste off its backside and past its opposite side to the stickers.
2. Now open the circuit layout and change some printing settings to match the paper and print it.
3. Sticker paper is very slippery, and it is difficult for printer to pick it, so push a little inside while printing job is being started.
4. Now cut a piece of copper coated sheet according to your circuit size and clean it using scotch brite and water.
5. Use steam iron to transfer the circuit layout from the sticker paper to the copper coated board. Apply a lot of weight on it for 5 to 10 mins.
6. Now peel off carefully.
7. Dip the board in the ferric chloride; be sure to wear eye protection, rubber gloves and protective suit when handling chemicals.
8. Stir for 15 to 20 mins or until the exposed copper is removed
9. Wash with water and scratch it using scotch brite
10. Make the holes using the mini drill
11. Apply a thin layer of liquid rosin on it will keep it safe from rusting.
12. The PCB is ready for use.

Assembling of components

The number of components used will determine the design of the PCB and the size.

1. Begin by fitting the components into their appropriately designed spaces.
2. Make sure polarized pieces are oriented accordingly.
3. When designing the PCB ensure there are no intersection of cables in the circuit
4. Do no put components at different angles from 0 to 90 degrees.
5. Ensure the excess metal under the board is cut off to a reasonable length after soldering.
6. Ensure to connect the 3.7V voltage source at the right polarity.

Results And Discussions

Hardware Implementation

The implementation of the hardware involved the connection of a MAX30100 sensor, a DS18B20 temperature sensor, a DHT11 sensor, a 0.96 inch OLED screen module, the push button, the slide switch and the battery to an ESP32 as already designed. Before the complete circuit was constructed, each component was tested with the ESP32 to look for any underlying faults that may cause issues later on. This was achieved by attaining each components sample program from the internet, uploading the programs to the ESP32 and testing each one with the microcontroller consecutively. This test was carried out on a breadboard and after proving no underlying faults in the components; the entire circuit was then constructed and tested again before transferring the circuit to a printed circuit board.

Software implementation

The Arduino IDE was used to write the code that instructs the microcontroller on the processes to perform.

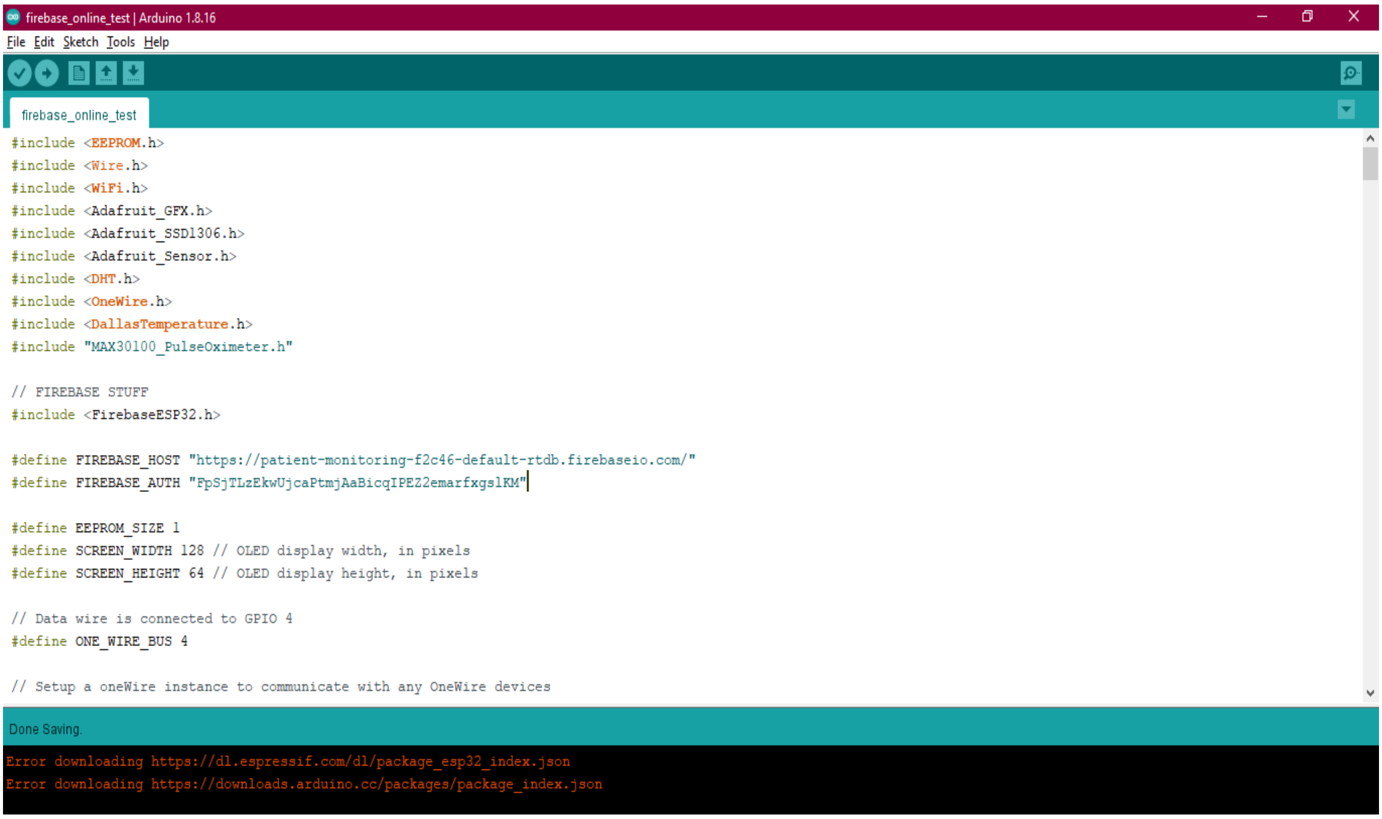


Figure 4.1: A snippet of the code

Explanation of code

The code starts off by calling the necessary header files that contains the functions that are integral to its successful run. This is done with the '#include' preprocessor directive. The header files called include:

i. Wire.h: Contains functions that allow for I2C device communications.

ii. WiFi.h: Contains functions that enable the ESP32's Wi-Fi capabilities.

iii. EEPROM.h: Contains functions that allow for the utilization of the ESP32's flash memory.

iv. DHT.h: Contains functions that allow for the basic functioning of the DHT11 sensor.

v. Adafruit_GFX.h: Contains functions for the proper functioning of the graphical display for the OLED.

vi. Adafruit_SSD1306.h: This contains the functions for the operating of components that use the SSD1306 integrated circuit chip, like the OLED.

vii. OneWire.h: This contains functions that enable components that utilize the one wire protocol in their means of operation.

viii. DallasTemperature.h: Contains the necessary functions for the retrieval of readings from all Dallas made components, such as the DS18B20.

ix. MAX30100_PulseOximeter.h: This contains functions necessary for the operation of the MAX30100 sensor.

x. FirebaseESP32.h: This header file contains the functions that allow for the ESP32 to transmit sensed data to Firebase's online database.

Immediately after the header files are called, some constant values are defined with the '#define' preprocessor directive. These values will not change throughout the entirety of the programs running. These constant values include the particular GPIO's the components connect with the ESP32, the display height and width for the OLED, the size of the ESP32 flash memory to be used and the details for connecting to the Firebase online database. Some other variables are declared in this section as well. These variables are global in scope.

The main part of the code begins at the 'void setup()' function block. It starts off by setting up the ESP32 flash EEPROM memory and a conditional statement to check the memory if the user had set the system to offline or online mode – 0 for online and 1 for offline. The OLED's SSD1306 chip is started with the 'display.begin(SSD1306_SWITCHCAPVCC, 0x3C)' statement and then the OLED displays my name and matriculation number as instructed.

After the display, the code then states that the ESP32, if in online mode, should try to connect to the wifi connection whose details (SSID and password) have already been stated in the code earlier. This is done with the 'WiFi.begin(ssid, password);' statement. Following the code, if the connection is successful, the OLED displays 'Connected to server', and the Firebase is started and setup. If the connection was not successful, the OLED displays, 'Could not connect to the internet', then the ESP carries on like it was in offline mode. The rest of the 'void setup()' block just starts up the sensors, using the 'dht.begin();' and the 'pox.begin();' functions for the DHT11 and the MAX30100 respectively.

After the 'void setup()' block ends the 'void loop()' begins. The 'void loop()' ensures continual reiteration in the code unless deliberately prevented from doing so by the programmer. All that lies within the 'void loop()' block are things that need be perpetually checked or executed. Inside the block the GPIO pin 5 of the ESP32 is constantly being checked to know if the push button has been pressed to switch between offline and online mode of operation. The conditional statement around this is set so that if the push button allow current to enter the GPIO pin 5, the variable in which stores the offline-online mode state in the stored is changed either from 0 to 1 (offline to online) or from 1 to 0 (online to offline). The next lines of code updates the MAX30100 sensor and store the readings from the various sensors in their respective and appropriately named variables. A block of code which sets up a timing function is used to time the display of sensor data onto the OLED module. This is done with the use of the millis function which is a function that starts counting at the startup of the microcontroller and does not stop till the MCU is disconnected from power or reset. Coupling the millis function with a variable, timing control is achieved in the code. The 'n' variable is what is checked to know what the OLED should be displaying at a time. Following the timing function, after three seconds, the 'n' variable changes and so does what the OLED displays. When n = 1, it OLED shows a request to prompt the user to put his/her finger over the MAX30100 sensor to take readings, when n = 2, it displays the DHT11 sensor readings, when n = 3, it displays the DS18B20 temperature readings and then it goes back to when n = 1. There is an added condition that is checked when displaying the sensor readings and that is if the MAX30100 is not taking any readings. If the MAX30100 is taking any readings, all other sensor readings are removed from the display and the MAX30100 readings are shown on the OLED. This is done to prioritize the heart rate and SpO2 because if the user is placing his/her finger over the MAX30100, he/she is truly eager to know those particular set of data at the moment. After three seconds of no activity from the MAX30100, the OLED resumes displaying the other sensor data.

Lastly, still within the 'void loop()' block, a counter variable is used to control the timing of transmission of the sensed data to the Firebase database. This block is only considered if connection to the internet was successful. The 'count' variable increases every time the 'void loop()' block runs. When this variable reaches 100 (starting from 0), data is sent to the database in the form of JSON files. The JSON files are created with the 'JSON .set();' function before they are sent to the database. Also in this block, a condition is executed to allow for stored data when the system was offline to be sent to the database when the system next connects to the database.

Database setup

The database used was the online realtime database service from Google's Firebase. A user account was created and the specifications of the desired database were specified through the forms provided on their website. After the account was created and the realtime database was established, the server and database details were added to the ESP32's code so it can be able to access the newly created database and transmit the sensor data.

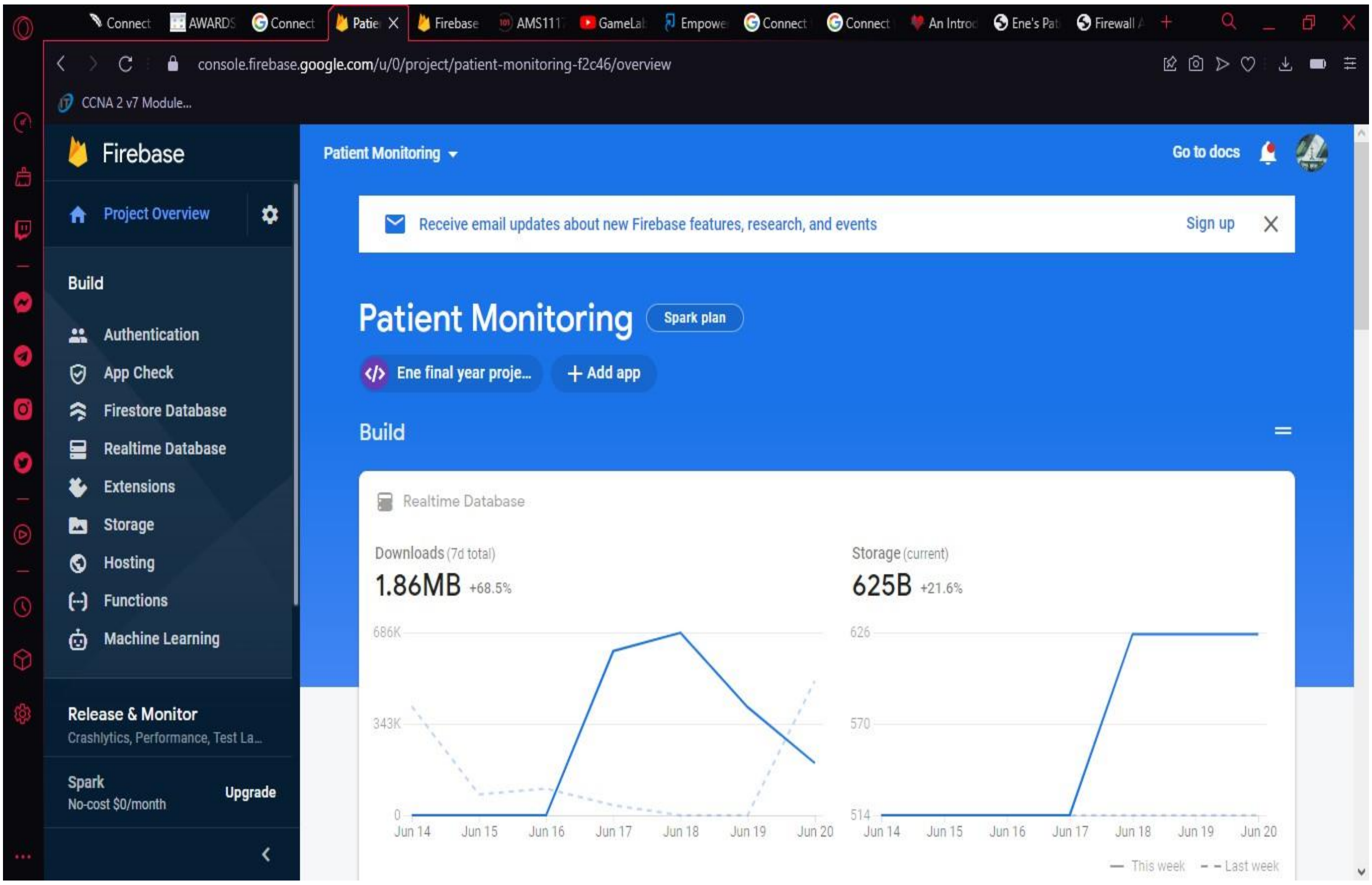


Figure 4.2: A screenshot of the Firebase console

4.1 Packaging

Packaging of the system was performed for convenient mobility and professionalism. The packaging was done using a white plastic box referred to as a double gang pattress box. The pattress box came with some openings but not in usable locations, as openings were needed for the OLED display module to be viewable, the MAX30100 to be accessible, the DS18B20 to be accessible and the push button to be usable while the whole circuit remained in the box. Thus, openings were created in the box at strategic locations allowing for the components listed above to be accessible to the user. Also, openings were created for the slide switch and the micro USB port of the ESP32 so as to allow the user to be able to turn on/off battery power supply and to allow the system to be powered through a micro USB connection to an external power source.

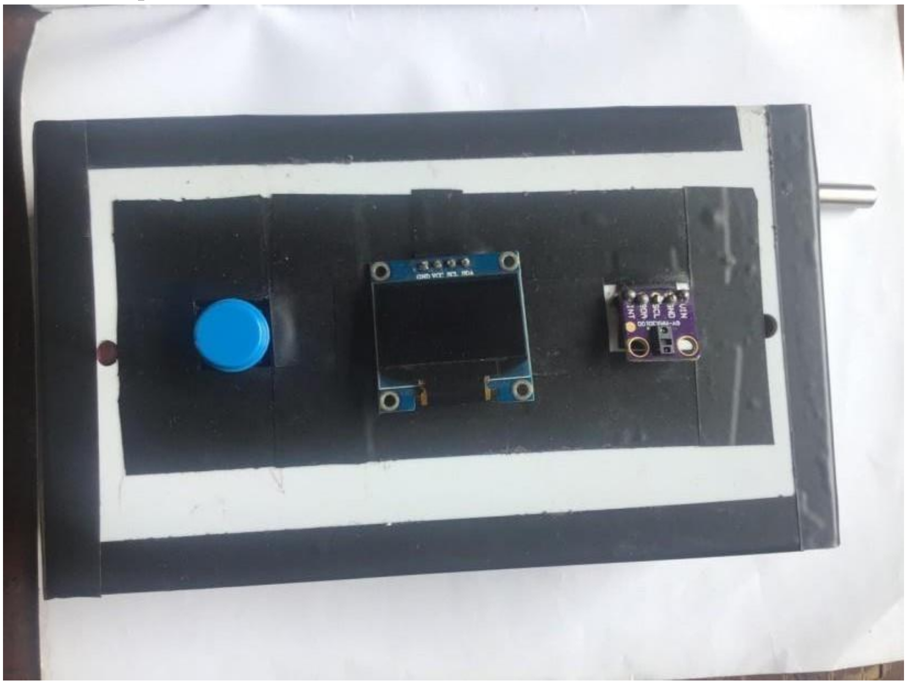


Figure 4.3: The fully constructed proposed system

Testing and Results

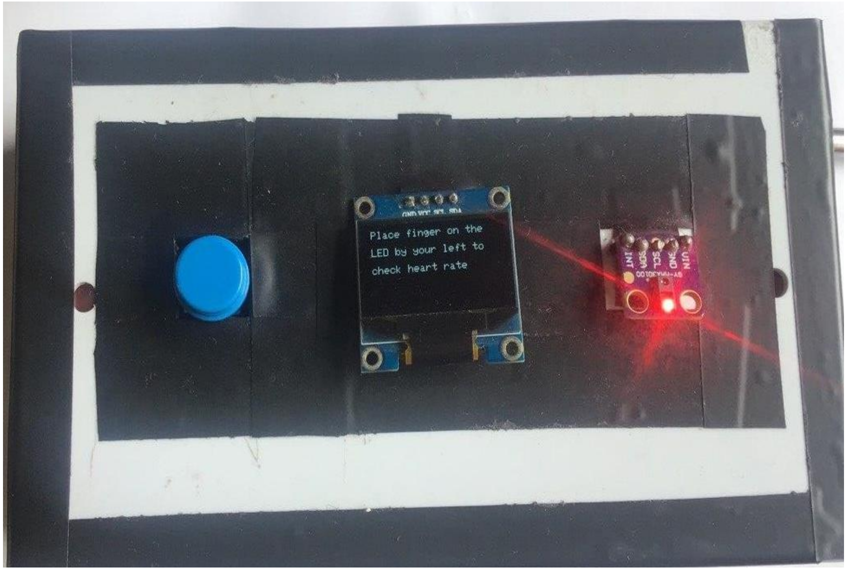
The hardware components, after being put together and packaged were tested with the code written for the ESP32 through the Arduino IDE. As the microcontroller received power, it first displayed my name and matriculation number, and then it prompts the user to place a finger over the MAX30100 to check his/her heart rate as shown in figure 4.4. After three seconds of idleness, it then displays the room temperature and humidity. After another three seconds of idleness, it displays the temperature readings from the DS18B20 sensor. To test the proper functioning of the program, I placed my finger on the MAX30100 sensor, and just as programmed, it displayed my heart rate and blood oxygen saturation level on the OLED screen. I also held the DS18B20 probe for a short while, giving me a temperature reading of 35ºC. I compared the averages of the data attained by the system to those of standard medical devices which are shown in the table below.

Table 4.1 Attained Data Comparison

	Data from system	Data from standard medical devices
Heart/Pulse rate	81.5	89.2
SpO2	94%	94%
Body temperature	35ºC	36.4ºC

Carrying out an analysis of the data comparison it is seen the heart rate values and the temperature values attained by the system average slightly lower than those of the standard medical thermometers and pulse oximeter sensors. Regardless, the margins of variation are not too significant and can be accounted for by the users and medical practitioners.

Figure 4.4: The system in operation



As seen in figure 4.5, the data collected by the system was successfully sent to the database. Since the database operates in real-time, the data was sent almost instantly as long as the internet connection was stable.

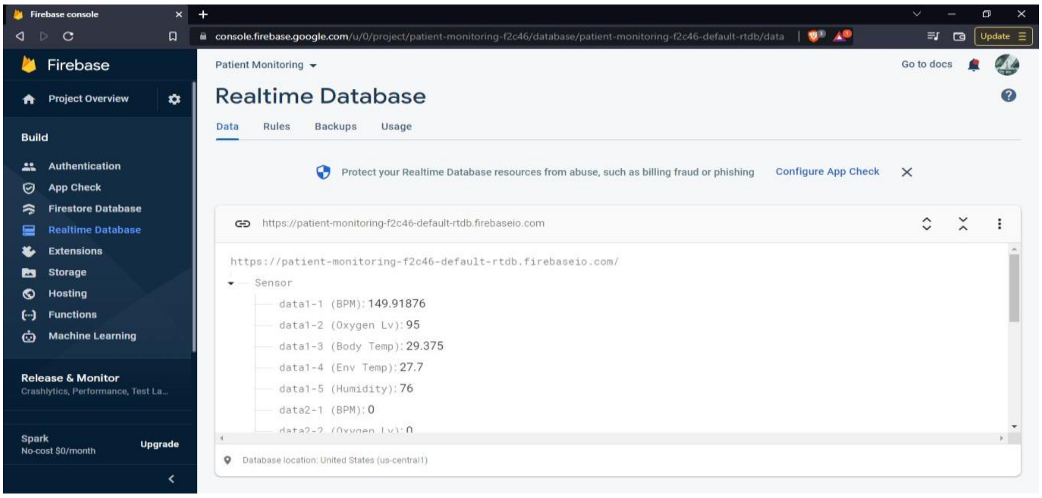


Figure 4.5: Sensor data in the database transferred from the system

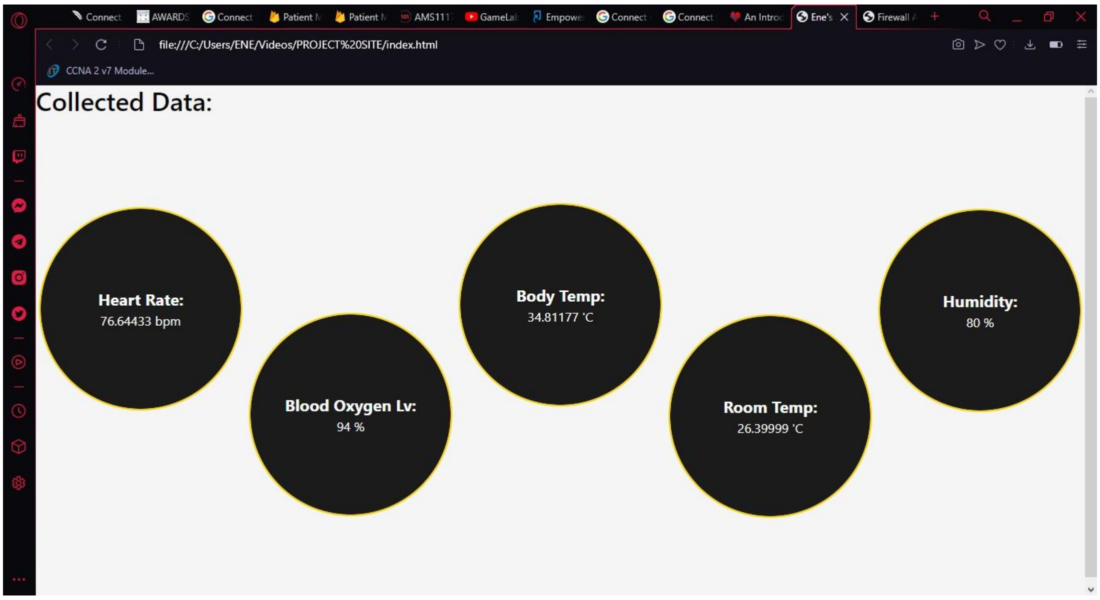


Figure 4.6: A screenshot of a test website with attained data from the database

As shown in figure 4.6, a live webpage was able to display data collected from the realtime online database which was sent from the implemented system. The webpage shows the user's heart rate, blood oxygen level, body temperature and the temperature and humidity of the environment around the user.

4.2 Bill of Engineering Measurements and Evaluation

This bill as shown in table 4.1 shows the components used for this project, the number of units, the cost per unit and the total cost respectively. It is as follows:

Table 4.2 Bill of Engineering Measurements and evaluation

S/N	ITEMS	QUANTITY	COST/UNIT (₦)	TOTAL COST (₦)
1	Resistors: 1kΩ	1	10	10
2	Resistors: 2.2kΩ	1	10	10
3	MAX30100 Heart rate and pulse oximetry sensor	1	3000	3000
4	DS18B20 Waterproof Temperature sensor	1	4000	4000
5	DHT11 Temperature and humidity sensor module	1	1500	1500
6	ESP32 micro- controller board	1	12500	12500
7	OLED display	1	4600	4600
	module			
8	Tactile push button	1	50	50
9	Batteries: AA 1.5v	2	500	1000
10	AA Battery connector	1	500	500
11	Slide switch	1	100	100
12	Plastic casing: Double gang pattress box	1	500	500
13	Jumper wires	23		3000
14	Printed circuit board	1	15000	15000
15	Miscellaneous			5000
	TOTAL			50770

Conclusion

The main purpose of the project was to design a patient health monitoring system that could monitor patients' health vitals outside the hospital, to allow for continual monitoring regardless of the patients geographical location. Three sensors were utilized, the MAX30100, the DS18B20, and the DHT11 which collected heart rate and blood oxygen levels, body temperature, and environmental temperature and humidity respectively. With the aid of the ESP32 microcontroller, the attained data from the sensors were able to be transferred to an online database. The database that was utilized was the Firebase which enables easy retrieval of data from the ESP32 and easy access to the data through a website, from anywhere in the world.

Recommendations

i. More health parameters to be monitored to allow for a more effective patient monitoring system.

ii. Increased number of health data to be stored for better analysis of patient by medical personnel.

References

1. World Health Organisation, "New perspectives on global health spending for universal health coverage," World Health Organisation, Geneva, WHO/HIS/HGF/HFWorkingPaper/18.2, 2017.

2. Inaya Hajj Hussein et al., "Vaccines Through Centuries: Major Cornerstones of Global Health," Frontiers in public health, November 2015.

3. Andrew W. Truman, Barrie Wilkinson and Matthew I. Hutchings, "Antibiotics: past, present and future," Current Opinion in Microbiology, vol. 51, pp. 72-80, November 2019.

4. Amrit Thapa Magar, Harsh Verma, Meeta Singh, Pinki Sagar and Abhishek Malik, "A Detailed Study Of An Internet Of Things (Iot)," INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH, vol. 8, no. 12, pp. 2989-2994, December 2019.

5. Scott Eldridge, Lyman Chapin and Karen Rose, THE INTERNETOF THINGS: AN OVERVIEW Understanding the Issues and Challenges of a More Connected World.: The Internet Society (ISOC), 2015.

6. IoT Analytics. (2021, September) Global IoT market forecast. [Online]. https://iot-analytics.com/wp/wp-content/uploads/2021/09/Global-IoT-market-forecast-in-billion-connected-iot-devices-min

7. Alile Solomon O. and Otokiti Kareem O., "IoT Based Patient Health Monitoring System: The Way Forward for Patient Health Monitoring in Nigeria," Journal of Science and Technology Research, pp. 200-212, 2020.

8. Ibrahim Oreagba, Catherine O. Falade, Ambrose Isah and Olayinka Ogunleye, "Medication errors among health professionals in Nigeria: A national survey," The International journal of risk & safety in medicine, vol. 2, no. 28, pp. 77-91, August 2016.

9. Erik Fung et al., "Electrocardiographic patch devices and contemporary wireless cardiac monitoring," Frontiers in Physiology, vol. 6, no. 149, March 2015.

10. Joachim Boldt, "Clinical review: Hemodynamic monitoring in the intensive care unit," Critical Care, vol. 6, no. 1, February 2002.

11. Terry P. Clemmer, R. Scott Evans and Reed M. Gardner, "Patient Monitoring Systems," in Biomedical Informatics, James J. Cimino Edward H. Shortliffe, Ed. London, England: Springer-Verlang, 2014, ch. 19, pp. 561-591.

12. Gérard Reach and George S. Wilson, "Can Continuous Glucose Monitoring Be Used for the Treatment of Daibetes?," ANALYTICAL CHEMISTRY, vol. 64, no. 6, pp. 381-386, March 1992.

13. C. Doukas, V. P. Plagianakos, I. Maglogiannis and V. Pigadas, "Enabling Constant Monitoring of Chronic Patient Using Android Smart Phones," in Proceedings of the 4th International Conference on Pervasive Technologies Related to Assistive Environments: PETRA'11, Crete, May 2011, pp. 1-2.

14. J. Neuhaeuser and L. D'Angelo, "Collecting and distributing wearable sensor data: An embedded personal area network to local area network gateway server," in 35th Annual International Conference of the IEEE EMBS, Osaka, 2013, pp. 4650-4653.

15. Urs Anliker et al., "AMON: A Wearable Multiparameter Medical Monitoring and Alert System," IEEE Transactions on Information Technology in Biomedicine, vol. 8, no. 4, pp. 415-427, December 2004.

16. [16] Jeonggil Ko et al., "MEDiSN: Medical emergency detection in sensor networks," ACM Transactions on Embedded Computing Systems, vol. 10, no. 1, pp. 1-29, August 2010.

17. Mars Lan et al., "WANDA: an end-to-end remote health monitoring and analytics system for heart failure patients," in WH '12: Proceedings of the conference on Wireless Health, New York, 2012, pp. 1-8.

18. Ana Gotter. (2021, October) Healthline. [Online]. https://www.healthline.com/health/pulse-oximetry

19. Bruce M. Lo. (2021, January) Medscape. [Online]. https://emedicine.medscape.com/article/2116433-overview

20. Maxim Integrated. (2014, September) [Online]. https://datasheets.maximintegrated.com/en/ds/MAX30100.pdf