JSS MAHAVIDYAPEETHA
JSS SCIENCE AND TECHNOLOGY UNIVERSITY
SRI JAYACHAMARAJENDRA COLLEGE OF ENGINEERING
JSS Technical Institutions Campus, Mysuru – 570006

A mini project report on

"IOT BASED PATIENT MONITORING SYSTEM USING ESP8266 AND ARDUINO"

Submitted by

NIDHISHREE S
(01JST18EI026)

RAKSHA C
(01JST18EI037)

THEJASWINI L B
(01JST18EI050)

Under the Guidance of

Dr. Nanda S
Assistant Professor
Dept. of E&I, JSS S&TU
Mysuru

DEPARTMENT OF ELECTRONICS AND INSTRUMENTATION

June 2021

Contents

1. Introduction

Patient Health monitoring is a major problem in today's world. Due to the lack of proper health monitoring, patients suffer from serious health issues. With tons of new healthcare technology start-ups, IoT is rapidly revolutionizing the healthcare industry. Patient health monitoring can provide useful physiological information in the home. This monitoring is useful for elderly or chronically ill patients who would like to avoid a long hospital stay. Wireless sensors are used to collect and transmit signals of interest and a microcontroller is programmed to receive and automatically analyse the sensor signals.

The personal health monitoring of each individual is considered very important because of the rise in health problems in today's world. The increasing stressful lifestyle is taking its maximum toll on public health. With the ever-increasing queues at hospitals and ever-increasing number of patients, the doctor fees have sky-rocketed which is affecting especially those patients who cannot afford the fee or who are not suffering from major ailments but get to know so only after paying a hefty fee to the doctor. Using a single parameter monitoring system an approach to a remote health monitoring system was designed that extends healthcare from the traditional clinic or hospital setting to the patient's home. The system was to collect heartbeat detection system data, fall detection system data, temperature data and few other parameters. The data from the single parameter monitoring systems was then availed for remote detection.

In this project, IoT Based Patient Health Monitoring System using ESP8266 & Arduino has been designed. The IoT platform used in this project is ThingSpeak. ThingSpeak is an open-source Internet of Things (IoT) application and API to store and retrieve data from things using the HTTP protocol over the Internet or via a Local Area Network. This IoT device could read the pulse rate and measure the surrounding temperature. It continuously monitors the pulse rate and surrounding temperature and updates them to an IoT platform.

IoT or Internet of Things is usually considered as connecting things to the internet and using that connection to access individual objects. Otherwise, remote devices and objects with built in sensors are connected to the Internet of Things platform, which collects the information from different devices and stores it in the cloud and then transfers the data to that particular website.

Physiological monitoring hardware can be easily implemented using simple interfaces of the sensors with a Microcontroller and can effectively be used for healthcare monitoring. This will allow development of such low-cost devices based on natural human-computer interfaces. Monitoring and control refer to a field of industrial automation that is entering a new era with the development of wireless sensing devices.

2. Objectives

The main objective is to design and develop a Patient Health Monitoring System to continuously monitor the pulse rate and body temperature of the patient by transferring the data to a cloud platform with the help of IoT technology.

3. Methodology/ Design and Implementation

Heartbeat and temperature are major signs that are routinely measured by physicians. Heartrate refers to how many times a heart contracts and relaxes in a unit of time (usually per minute). Heartrate varies for different age groups. For a human adult of 18 years of age or more, the normal heartrate is 72 beats per minute (bpm). Like heartrate, the normal body temperature also varies from person to person and changes throughout the day. The body temperature is lowest in the early morning and highest in the early evening. Thus, the normal body temperature range is 97 to 100 degrees Fahrenheit or 36.1 to 37.8 degrees Celsius.

3.1 Block diagram

Fig.1 represents an IoT Based Patient Health Monitoring System using ESP8266 and Arduino. Pulse Sensor and LM35 Temperature Sensors measure pulse rate & body temperature respectively. The Arduino UNO processes the code and displays it on a 16*2 LCD Display. The ESP8266 Wi-Fi module connects to Wi-Fi and sends the fetched sensor data to the IoT device server. The IoT server used here is ThingSpeak. Finally, the data can be monitored from any part of the world by logging into the ThingSpeak channel.

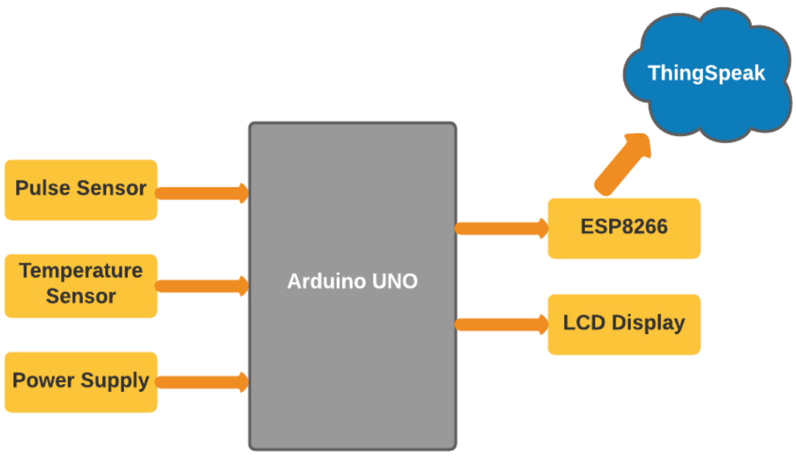


Fig.1: Block diagram representing IoT Based Patient Health Monitoring System

3.2 Circuit diagram

Fig.2 shows the circuit diagram of an IoT based patient monitoring system using

ESP8266 and Arduino.

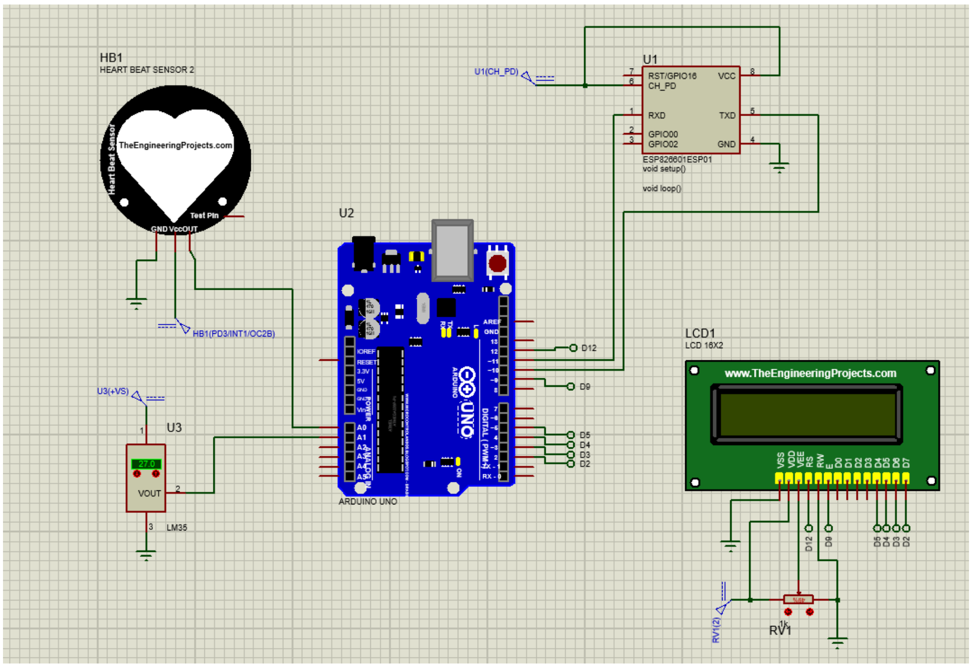


Fig.2: Circuit diagram of IoT Based Patient Health Monitoring System

3.3 Components' description

3.3.1 Arduino Uno:

Fig.3 shows Arduino UNO. Arduino UNO is based on ATmega328P Microcontroller, an 8-bit AVR Architecture based MCU from ATMEL. It is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It is an inexpensive, simple, cross-platform with a clear programming environment.

The essential part of Arduino is the microcontroller consisting of Random-Access Memory (RAM), Central Processing Unit (CPU) and memory storage.

Specifications of Arduino UNO:

- Microcontroller: ATmega328p
- Operating Voltage: 5V
- Input Voltage (recommended): 7-12V
- Input Voltage (limits): 6-20V

- Digital I/O Pins: 14 pins (of which 6 are PWM output pins)
- Analog Input Pins: 6
- DC Current per I/O Pin: 40 mA
- DC Current for 3.3V Pin: 50 mA
- Flash Memory: 32 KB (of which 0.5 KB is taken by bootloader)
- SRAM: 2 KB (ATmega328)
- EEPROM: 1 KB (ATmega328)
- Clock Speed: 16 MHz
- Length:6 mm
- Width:4 mm
- Weight: 25 g

Fig.3: Arduino UNO

3.3.2 ESP8266-01 Wi-Fi Module:

Fig.4 shows ESP8266 (also called ESP8266 Wireless Transceiver) is a cost-effective, easy-to-operate, compact-sized low-powered Wi-Fi module, designed by Espressif Systems, supports both TCP/IP and Serial Protocol. The ESP8266 ESP is a Wi-Fi module that allows microcontrollers access to a Wi-Fi network. It is a low-cost standalone wireless transceiver that can be used for end-point IoT developments.

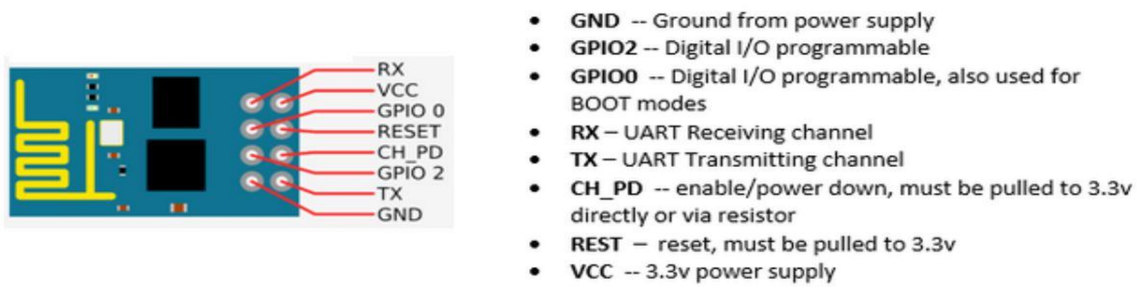


Fig.4: ESP8266-01 Wi-Fi module

Pin details:

- **VCC:** It is the power pin through which 3.3V is supplied.
- **GND:** It is the ground pin.
- **TX:** This pin is used to transmit serial data to other devices.
- **RX:** The RX pin is used to receive serial data from other devices.
- **RST:** It is the Reset Pin and it is an active LOW Pin. (ESP8266 will reset if the RST pin receives a LOW signal).
- **CH_PD(EN):** This is the chip enable pin and it is an active HIGH Pin. It is usually connected to 3.3V.
- **GPIO0:** The GPIO0 (General Purpose I/O) Pin has dual functions – one for normal GPIO Operation and other for enabling the Programming Mode of ESP8266.
- **GPIO2:** This is a GPIO Pin.

Features:

- It has 32- bit microcontroller
- CPU: 80 MHz (default) or 160 MHz
- Memory: 32 KiB instruction RAM, 80 KiB user data RAM, 32 KiB instruction cache RAM, 16 KiB ETS system – data RAM.
- Input: 16 GPIO pins

- Processor: L106 32-bit RISC microprocessor core based on the Tensilica Xtensa Diamond standards 106 Micro running at 80 MHz.
- External QSPI flash: up to 16 MiB is supported (512 KiB to 4 MiB typically included)
- The new version of the ESP8266 Wi-Fi module has increased the flash disk size from 512KB to 1MB.
- IC (software implementation)
- IS interfaces with DMA (sharing pins with GPIO)
- 10-bit ADC (successive approximation ADC).
- Supports serial communication hence compatible with many development platform like Arduino
- Can be programmed using Arduino IDE or AT-commands or Lua Script.

3.3.3 Pulse Sensor:

Fig.5 shows the structure of pulse sensor. A pulse wave is the change in the volume of a blood vessel that occurs when the heart pumps blood, and a detector that monitors this volume change is called a pulse sensor. The Pulse Sensor (or Heart Rate or Heart Beat sensor) is a plug-and-play heart-rate sensor for Arduino. The sensor has two sides, on one side the LED is placed along with an ambient light sensor and on the other side it has some circuitry. This circuitry is responsible for the amplification and noise cancellation work. The LED on the front side of the sensor is placed over a vein in our human body. This can either be the Fingertip or ear tips, but it should be placed directly on top of a vein. Now the LED emits light which will fall on the vein directly. The veins will have blood flow inside them only when the heart is pumping, so it can monitor the flow of blood, and monitor the heart beats as well. If the flow of blood is detected then the ambient light sensor will pick up lighter since they will be reflected by the blood, this minor change in received light is analysed over time to determine our heart beats.

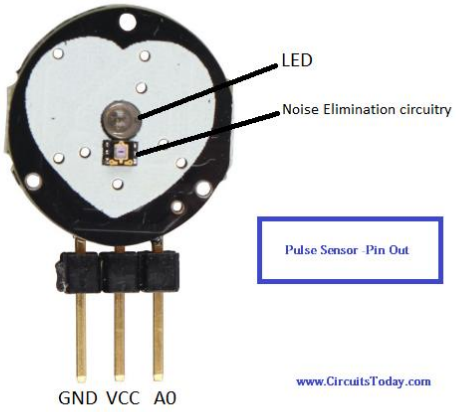


Fig.5: Pulse sensor

Pin details:

- Pin-1 (GND): Black Colour Wire – It is connected to the GND terminal of the system.
- Pin-2 (VCC): Red Colour Wire – It is connected to the supply voltage (+5V otherwise +3.3V) of the system.
- Pin-3 (Signal): Purple Colour Wire – It is connected to the pulsating o/p signal.

Features:

- This is a heartbeat detecting and biometric pulse rate sensor
- Its diameter is 0.625
- Its thickness is 0.125
- The operating voltage is ranges +5V otherwise +3.3V
- This is a plug and play type sensor
- The current utilization is 4mA
- Includes the circuits like Amplification & Noise cancellation

- This pulse sensor is not approved by the FDA or medical. So, it is used in student-level projects, not for the commercial purpose in health issues applications.

3.3.4 LM35 Temperature Sensor:

Fig.6 shows the LM35 sensor which are precision integrated-circuit temperature devices with an output voltage linearly-proportional to the Centigrade temperature. The LM35 device is rated to operate over a −55°C to 150°C temperature range, while the LM35C device is rated for a −40°C to 110°C range (−10° with improved accuracy).

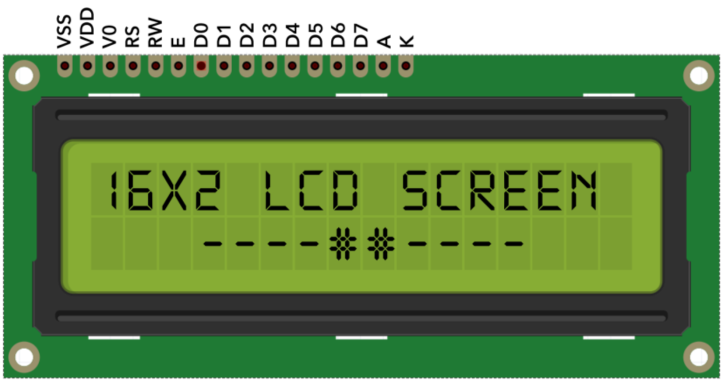
Fig.6: LM35 Temperature sensor

Features:

- Calibrated Directly in Celsius (Centigrade)
- Linear + 10-mV/°C Scale Factor.
- 0.5°C Ensured Accuracy (at 25°C)
- Rated for Full −55°C to 150°C Range.
- Suitable for Remote Applications.
- Low-Cost Due to Wafer-Level Trimming.
- Operates From 4 V to 30 V.
- Less Than 60-µA Current Drain.

3.3.5 LCD Display:

Fig.7 shows the Liquid Crystal Display or LCD uses liquid crystal for the production of visible images.16×2 LCD is named so because; it has 16 Columns and 2 Rows. There are a lot of combinations available like, 8×1, 8×2, 10×2, 16×1, etc. but the most used one is the 16×2 LCD. So, it will have (16×2=32) 32 characters in total and each character will be made of 5×8 Pixel Dots.



No	Symbol	Function
1	VSS	Ground
2	VDD	5V +
3	V0	Contrast
4	RS	Register
5	RW	Read/Write
6	E	Enable
7	D0	Data bus
8	D1	Data bus
9	D2	Data bus
10	D3	Data bus
11	D4	Data bus
12	D5	Data bus
13	D6	Data bus
14	D7	Data bus
15	A	Anode (5V+)
16	K	Cathode (GND)

Fig.7:16*2 LCD Screen

Features:

- Operating Voltage is 4.7V to 5.3V
- Current consumption is 1mA without backlight
- Alphanumeric LCD display module, meaning can display alphabets and numbers
- Consists of two rows and each row can print 16 characters.
- Each character is built by a 5×8-pixel box
- Can work on both 8-bit and 4-bit mode
- It can also display any custom generated characters

A 16×2 LCD has two registers - data register and command register. The RS (register select) is mainly used to change from one register to another. When the register set is '0', then it is known as a command register. Similarly, when the register set is '1', then it is known as a data register.

The main function of the Command register is to store the instructions of command which are given to the display. So that predefined task can be performed such as clearing the display, initializing, setting the cursor place, and display control. Here commands processing can occur within the register.

The main function of the Data register is to store the information which is to be exhibited on the LCD screen. Here, the ASCII value of the character is the information which is to be exhibited on the LCD screen. Whenever the information is sent to LCD, it transmits to the data register, and then the process will be starting there. When register set =1, then the data register will be selected.

3.3.6 Potentiometer:

Fig.8 shows a potentiometer (also known as a pot or pot meter) which is defined as a 3 terminal variable resistor in which the resistance is manually varied to control the flow of electric current. A potentiometer acts as an adjustable voltage divider.

A potentiometer is a passive electronic component. Potentiometers work by varying the position of a sliding contact across a uniform resistance. In a potentiometer, the entire input voltage is applied across the whole length of the resistor, and the output voltage is the voltage drop between the fixed and sliding contact.

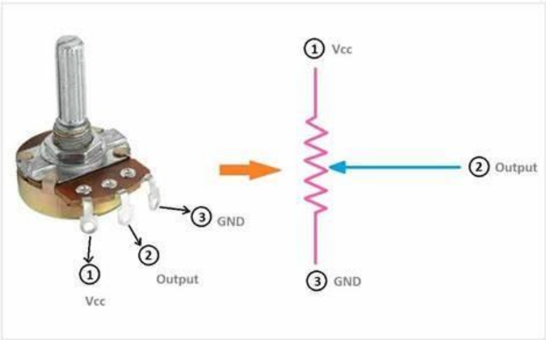


Fig.8: Potentiometer

This kind of potentiometer includes two terminal contacts where a consistent resistance can be located in a semi-circular model. And also, it includes a terminal in the middle that is allied to the resistance using a sliding contact that is connected through a rotating knob. Rotary potentiometer has a wiper that rotates across two terminals for varying the resistance of the potentiometer.

3.3.7 Jumper wires:

As shown in Fig.9, Jumper wires are simple wires that have connector pins at each end, allowing them to be used to connect two points to each other without soldering. Jumper wires are typically used with breadboards and other prototyping tools in order to make it easy to change a circuit as needed. There are two types of jumpers - male and female.

Fig.9: Jumper Wires

3.4 Design aspects

In this design, the Arduino Uno microcontroller which acts as the brain of our project is interfaced with the temperature sensor, pulse sensor, ESP8266 WiFi module and LCD display respectively.

The Analog pins A0 and A1 of the Arduino are connected to the output pins (Vout) of the pulse sensor and temperature sensor respectively. The LCD display is interfaced with the Arduino for the display of read sensor data, i.e., the data pins D4, D5, D6, D7 of LCD display are connected to the digital pins 5,4,3,2 of the Arduino respectively. The enable and register pin of the LCD display which determines the state and mode of the LCD are connected to pin 9 and pin 12 of the Arduino respectively. The VEE pin of the LCD display is connected to the potentiometer, to adjust the contrast of the LCD display.

The transmitting (TX) and receiving (RX) channel pins of the ESP8266 WiFi module is connected to the pin 10 and pin 11 of the Arduino i.e., pins used for SPI communication (for communication over short distance). This interface will help ESP8266 WiFi module to fetch sensor data that is to be transmitted to the cloud platform. The enable and Vcc pins of the ESP8266 W-Fi module are connected to the power supply.

3.5 Working principle

The Arduino Uno, an open-source microcontroller board, is connected to two sensors - pulse sensor and temperature sensor (LM35). The pulse sensor is used to easily incorporate live heart rate data (BPM) using an open-source monitoring platform that graphs pulse in real time. The LM35 temperature sensor is a precision integrated-circuit temperature device used to reflect the change in the body temperature. The readings of the parameters from the sensors will be displayed on the LCD display interfaced with the microcontroller.

The ESP8266 WIFI module interfaced with the Arduino device provides internet connectivity to the setup by connecting to a known access point/WiFi. The module works as a station in our setup. Hence it can easily fetch the sensor data and upload it to the Cloud platform. Here, the ThingSpeak platform provides a very good tool for IoT based implementation. By using the ThingSpeak site, data can be monitored and can be controlled over the Internet, using the channels and web pages provided by ThingSpeak to notify the patients' health to their family members also allowing doctors to easily access the patients' health condition through their smartphones from any part of the world for assistance during any medical emergency.

4. Results

Fig.10 and Fig.11 shows the output from the sensor and amplifier circuit which was connected to the Arduino. The observed output signal was periodic ac signal with amplitude varying from peak to peak according to person. A model sinusoidal signal and the output from the sensor were fed to the Arduino and the counted pulse rate was successfully sent via Wi-Fi module. The counted signal from the sensor to measure the heartbeat was relatively a weak signal which needed to be amplified and filtered before it was sent to the Arduino. So, the signal was amplified using an operational amplifier. The amplified signal was then filtered to get the desired output of heartbeat which was then sent to the microcontroller for further processing. The microcontroller then sends the received data of both heartbeat and temperature of a patient to a remote end via Wi-Fi module. The output consists of the data from sensors. The measured heartbeat and temperature vary for different individuals depending upon their age-group.

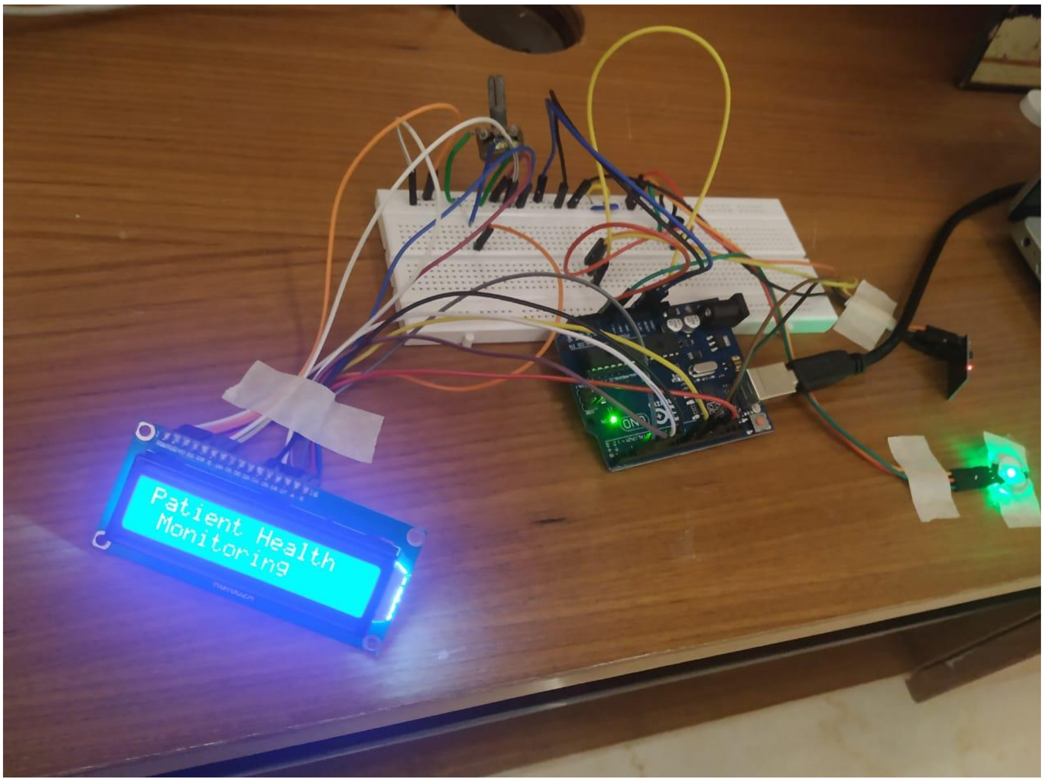


Fig.10: Hardware implementation of IoT Based Patient Health Monitoring System

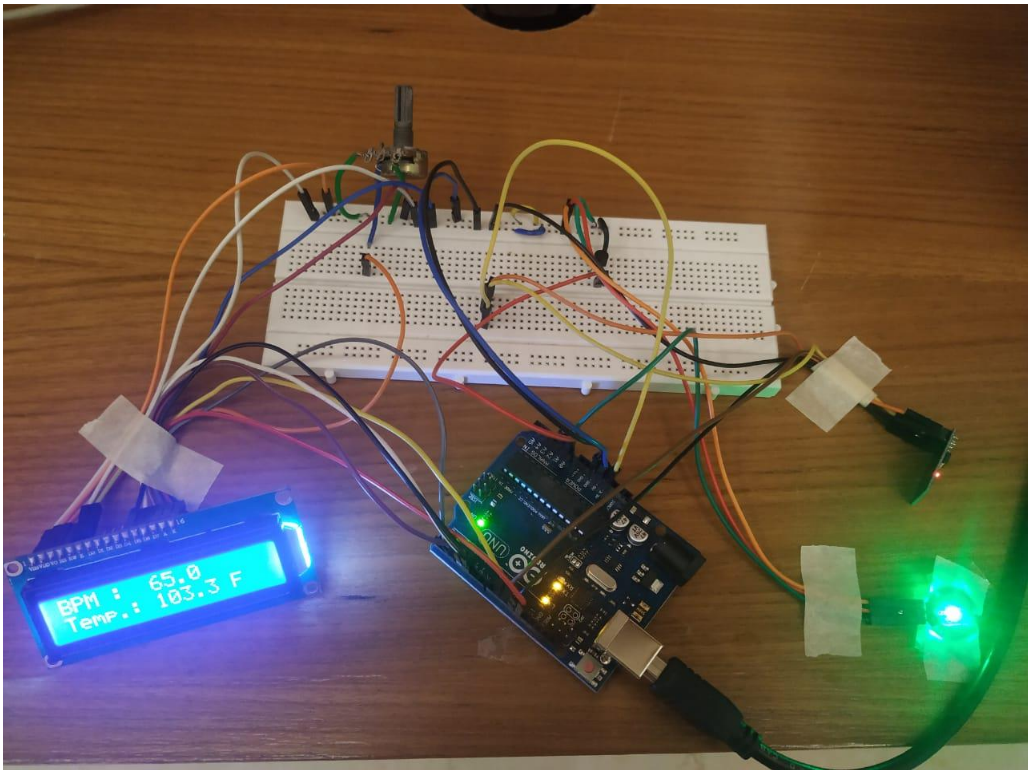


Fig.11: Hardware implementation of IoT Based Patient Health Monitoring System

ThingSpeak output for person 1:

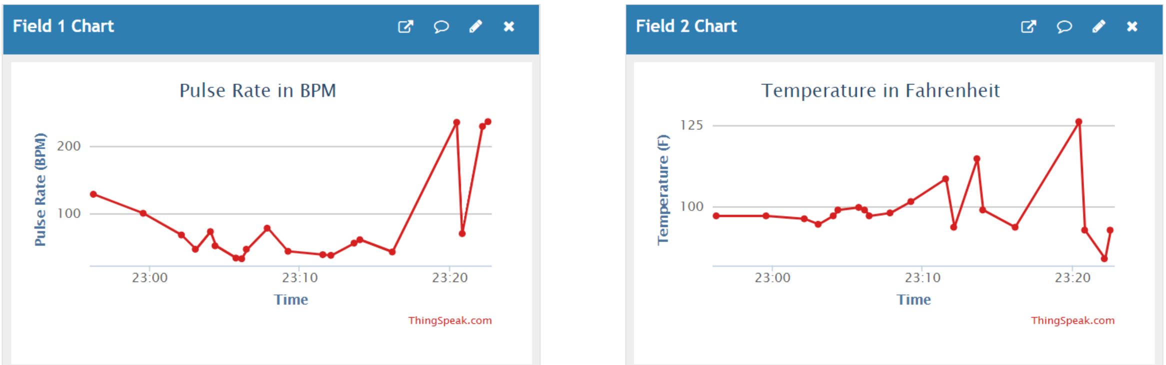
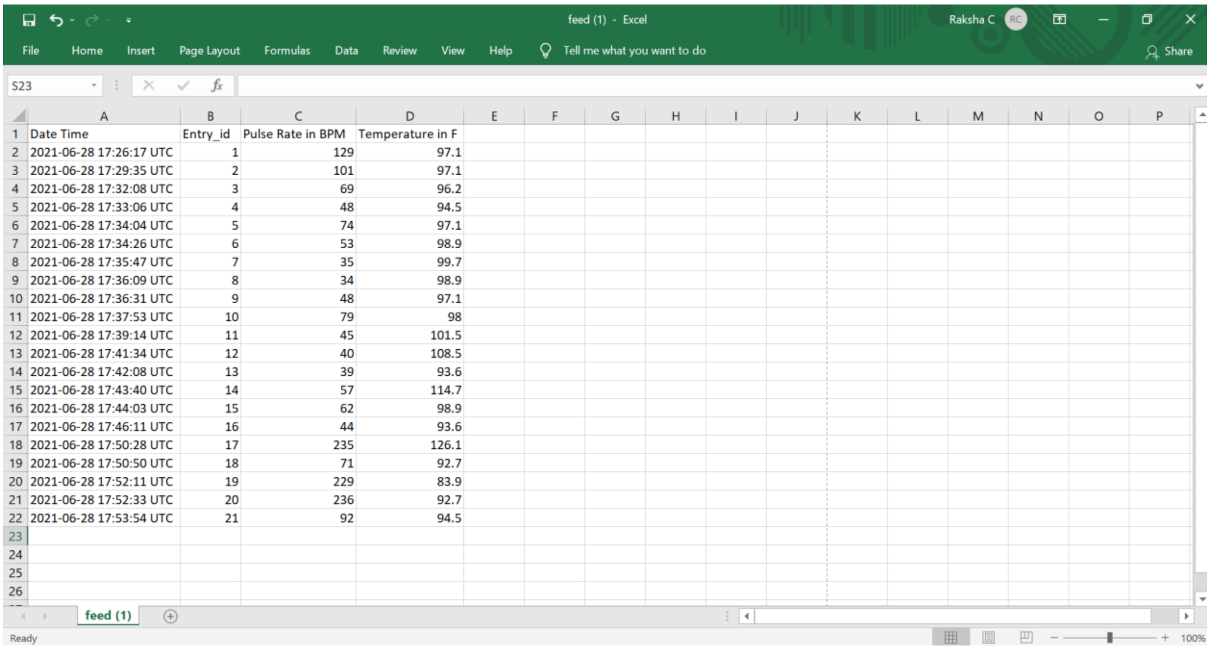


Fig.12: ThingSpeak output for person 1 (Age between 20-25 years)

Person 1 data record:



Date Time	Entry_id	Pulse Rate in BPM	Temperature in F
2021-06-28 17:26:17 UTC	1	129	97.1
2021-06-28 17:29:35 UTC	2	101	97.1
2021-06-28 17:32:08 UTC	3	69	96.2
2021-06-28 17:33:06 UTC	4	48	94.5
2021-06-28 17:34:04 UTC	5	74	97.1
2021-06-28 17:34:26 UTC	6	53	98.9
2021-06-28 17:35:47 UTC	7	35	99.7
2021-06-28 17:36:09 UTC	8	34	98.9
2021-06-28 17:36:31 UTC	9	48	97.1
2021-06-28 17:37:53 UTC	10	79	98
2021-06-28 17:39:14 UTC	11	45	101.5
2021-06-28 17:41:34 UTC	12	40	108.5
2021-06-28 17:42:08 UTC	13	39	93.6
2021-06-28 17:43:40 UTC	14	57	114.7
2021-06-28 17:44:03 UTC	15	62	98.9
2021-06-28 17:46:11 UTC	16	44	93.6
2021-06-28 17:50:28 UTC	17	235	126.1
2021-06-28 17:50:50 UTC	18	71	92.7
2021-06-28 17:52:11 UTC	19	229	83.9
2021-06-28 17:52:33 UTC	20	236	92.7
2021-06-28 17:53:54 UTC	21	92	94.5

Fig.13: First person's recorded data list

ThingSpeak output for person 2:

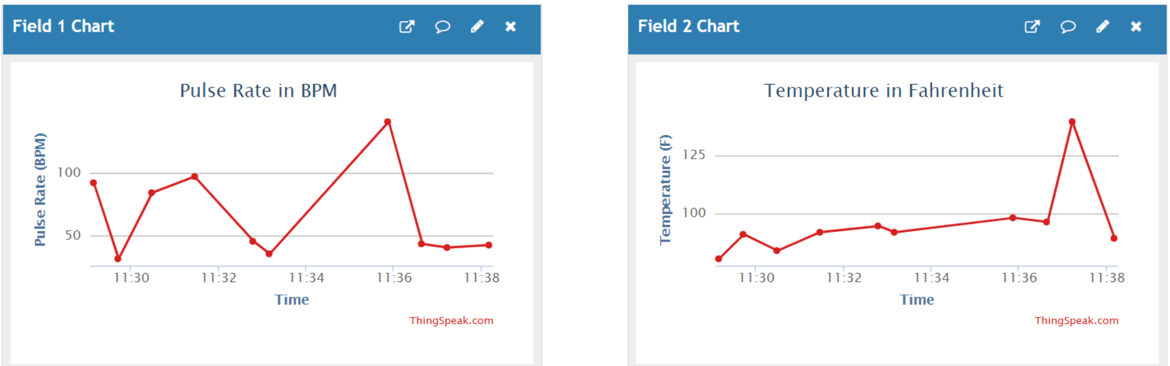
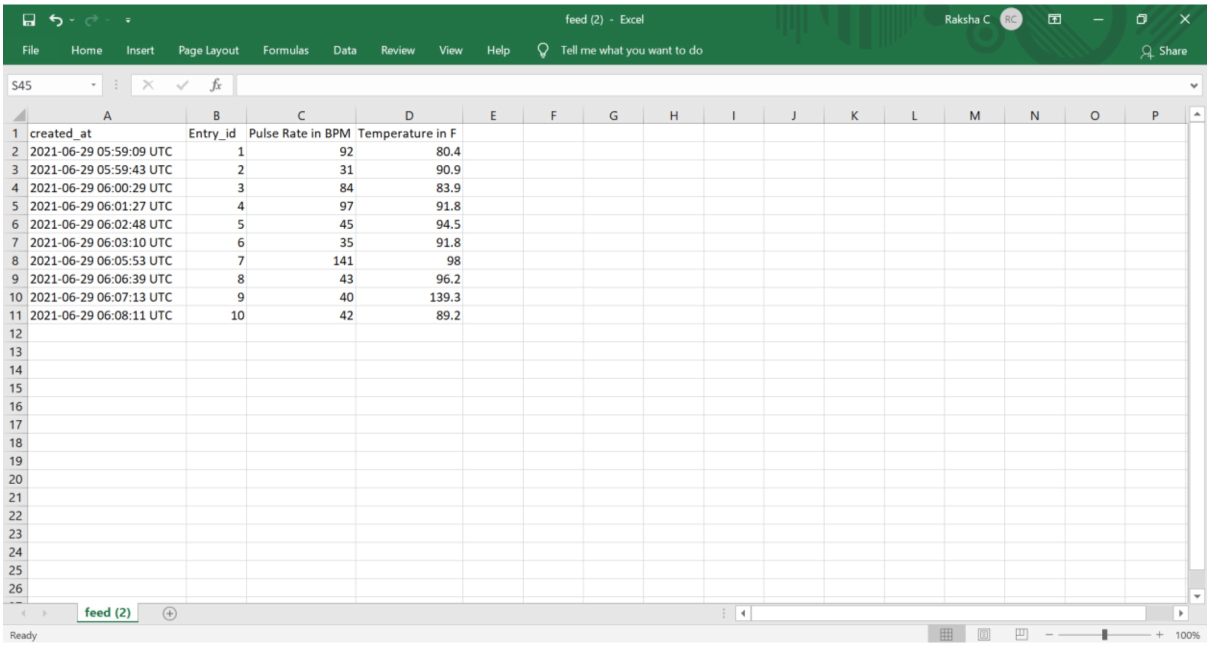


Fig.14: ThingSpeak output for person 2 (Age between 40-45 years)

Person 2 data record:



created_at	Entry_id	Pulse Rate in BPM	Temperature in F
2021-06-29 05:59:09 UTC	1	92	80.4
2021-06-29 05:59:43 UTC	2	31	90.9
2021-06-29 06:00:29 UTC	3	84	83.9
2021-06-29 06:01:27 UTC	4	97	91.8
2021-06-29 06:02:48 UTC	5	45	94.5
2021-06-29 06:03:10 UTC	6	35	91.8
2021-06-29 06:05:53 UTC	7	141	98
2021-06-29 06:06:39 UTC	8	43	96.2
2021-06-29 06:07:13 UTC	9	40	139.3
2021-06-29 06:08:11 UTC	10	42	89.2

Fig.15: Second person’s recorded data list

5. Conclusion and Future Enhancements

With the extensive use of the internet, this project concentrated on executing internet technology which communicated with the internet to monitor a patient's health. The main objective of the design, of obtaining the patient's pulse rate and temperature on cloud, was successfully achieved. The design thus put forward is focused mainly on making life more convenient for people with health challenges and who need to visit the hospital regularly. This device reduces the number of hospital visits, queues in the hospital and reduction in the cost of taking care of the sick and can thus be used to implement social distancing in times of a medical pandemic. All the individual modules, pulse sensor, temperature sensor, ESP8266 and Arduino gave out the intended results.

The design covers concepts of computer science and embedded systems period being applied practically. Circuit analysis knowledge was used during design and fabrication of the individual modules. Electromagnetic field analysis used in the wireless transmission between microcontrollers and software programming used during programming of the microcontrollers to come up with a final finished circuit system.

Future enhancements

Development is a forever on-going process. With advancements in science and technology, the existing systems can be enhanced to perform better and efficiently.

The whole health monitoring system which has been framed can be integrated into a small compact unit as small as a cell phone or a wrist watch. This will help the patients to easily carry this device with them wherever they go.

In the proposed system, an enhancement would be to connect more sensors to the internet which measures various other health parameters like ECG, SpO2, RESP, CO, NIBP, IBP, IPLETH, etcetera.

Remote viewing of data: problems associated with having data online. Tackle Distributed denial of service. DDOS, and Data privacy/security especially of medical systems.

IoT based Remote Patient Monitoring System can be enhanced to detect and collect data of several anomalies for monitoring purposes such as home ultrasound, Brain signal monitoring, tumor detection etc.

More research on problems associated with having data online, data privacy as IoT is managed and run by multiple technologies and multiple vendors are involved in it. Security algorithms and certain precautions by the users will help avoid any security related threats in the IoT network.

The interface can be designed to control which sensors can be used by consumers according to their needs.

Web UI can be enhanced to perform several activities which include controlling the hardware, real-time graphs, history and analysis graphs to observe anomalies etc.

6. References

1. https://www.circuito.io/blog/arduino-uno-pinout/
2. https://www.etechnophiles.com/arduino-uno-pinout-pin-diagram-specifications/
3. https://bc-robotics.com/tutorials/using-a-tmp36-temperature-sensor-with-arduino/#:~:text=The%20TMP36%20temperature%20sensor%20is%20an%20easy%20way,very%20low%20cost%2C%20making%20it%20a%20popular%20choice.
4. https://microcontrollerslab.com/tmp36-temperature-sensor/
5. https://bc-robotics.com/tutorials/using-a-tmp36-temperature-sensor-with-arduino/
6. https://learn.adafruit.com/tmp36-temperature-sensor/overview
7. https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Datasheet1.pdf
8. https://theiotprojects.com/iot-based-fall-detection-using-nodemcu-and-mpu6050-sensor/
9. https://dzone.com/articles/an-overview-of-iot-monitoring-systems
10. https://microcontrollerslab.com/esp32-mpu6050-accelerometer-gyroscope-temperature-arduino/#:~:text=ESP32%20with%20MPU6050%20Accelerometer%2C%20Gyroscope%2C%20and%20Temperature%20Sensor,Display%20MPU-6050%20Readings%20on%20Serial%20Plotter.%20More%20items
11. https://how2electronics.com/measure-tilt-angle-mpu6050-arduino/
12. https://www.woolseyworkshop.com/2018/08/17/interfacing-an-mpu6050-gyroscope-accelerometer-sensor-module-to-an-arduino-uno/
13. https://www.electronicshub.org/getting-started-arduino-mpu6050/
14. https://theiotprojects.com/monitor-mpu6050-tilt-angle-on-blynk-using-nodemcu/
15. https://randomnerdtutorials.com/arduino-mpu-6050-accelerometer-gyroscope/
16. https://www.electrical4u.com/potentiometer/
17. https://www.elprocus.com/potentiometer-construction-working-and-applications/
18. https://electricalshouters.com/potentiometer-working-types-applications/

19. https://www.tutorialspoint.com/basic_electronics/basic_electronics_resistors.htm
20. https://byjus.com/physics/resistor/
21. https://www.theengineerspost.com/types-of-resistor/
22. https://byjus.com/physics/resistor-colour-codes/
23. https://www.elprocus.com/what-is-a-wire-wound-resistor-types-and-applications/#:~:text=Advantages%20of%20Wire%20Wound%20Resistor%201%20This%20resistor,by%20the%20noise%203%20It%20is%20thermally%20constant.
24. https://www.elprocus.com/light-emitting-diode-led-working-application/
25. https://www.javatpoint.com/led
26. Software Testing: https://en.wikipedia.org/wiki/Software_testing