Lebanese University 

Faculty of Engineering III

Electrical and Electronic Department

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Iot Health Monitoring System

Communication and Mini-Project

by

**Dana Dagher 6436**

**Marwa Chehab 6385**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Spring 2023-2024**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Supervisor: Dr Rida El-Chall, Dr Abdullatif Samhat.**

# DEDICATION

We dedicate this project to our families and friends who supported us throughout all the phases.

Dana & Marwa

# Acknowledgment

First of all, we want to express our gratitude both Dr.Abdullatif Samhat and Dr.Rida El-Chall, professors of Electrical Engineering, for their support and encouragement throughout this project.

In addition, we want to thank our families for standing by our side all this time by encouraging us through the ups and downs as we worked to complete this project.

# Abstract

Nowadays, IoT has played an essential role in many fields such as smart home, smart cars and transportation, education, and healthcare. In this project, we demonstrated IoT’s efficiency in healthcare field by implementing one of the health monitoring systems that enables real-time remote patient monitoring to improve the quality of people’s lives. The current system detects heart pulse and body temperature of any person and send data via an ESP8266 NodeMCU microcontroller which acts like a bridge to connect and transfer data from those sensors to the network, specifically to the ThingSpeak channel every 15 seconds showing a fluctuation graph.

In this way, a real live monitoring of the patient’s healthcare was provided and the doctor will always be aware of any abnormal rates anytime and anywhere.

In addition, a lcd will continuously show the pulse and temperature rate taken.

In the other part of the project, in order to improve it, a lcd with an RTC were added to show real date and time as well as two LEDs that refers to an ambulance siren. This project is part of a university lab initiative aimed at enhancing our skills and knowledge in the hardware field.

**Keywords: IoT, healthcare, microcontroller, network, monitoring, real time, data, pulse rate, body temperature.**

**Table of Contents**

[DEDICATION 1](#_Toc169504934)

[Acknowledgment 2](#_Toc169504935)

[Abstract 3](#_Toc169504936)

[List of Figures 6](#_Toc169504937)

[List of Tables 7](#_Toc169504938)

[List of Abbreviations 8](#_Toc169504939)

[Chapter 1 Introduction 9](#_Toc169504940)

[1.1 Background Information 9](#_Toc169504941)

[1.2 Problem Statement 10](#_Toc169504942)

[1.3 Objective and Motivation 11](#_Toc169504943)

[Chapter 2 Literature Review 12](#_Toc169504944)

[2.1 Introduction 12](#_Toc169504945)

[Chapter 3 System Design 14](#_Toc169504946)

[3.1 Introduction 14](#_Toc169504947)

[3.2 System Requirements 14](#_Toc169504948)

[3.2.1 Use Case Diagram 15](#_Toc169504949)

[3.3 System Architecture 16](#_Toc169504950)

[3.4 Non-Functional Requirements 16](#_Toc169504951)

[3.5 Conclusion 16](#_Toc169504952)

[Chapter 4 Implementation and Results 17](#_Toc169504953)

[4.1 Introduction 17](#_Toc169504954)

[4.2 Implementation Tools 18](#_Toc169504955)

[4.2.1 Software Tools 18](#_Toc169504956)

[4.2.2 Hardware Tools 23](#_Toc169504957)

[4.3 Implementation Steps 24](#_Toc169504958)

[4.4 Results 26](#_Toc169504959)

[4.5 Conclusion 26](#_Toc169504960)

[Chapter 5 Conclusion 27](#_Toc169504961)

[5.1 Conclusion 27](#_Toc169504962)

[5.2 Future work 27](#_Toc169504963)

[References 28](#_Toc169504964)

# List of Figures

[**Figure 1 IoT challenges** 10](#_Toc169504965)

[**Figure 2 Use case diagram** 15](#_Toc169504966)

[**Figure 3 System architecture** 16](#_Toc169504967)

[**Figure 4 ThingSpeak Channel** 23](#_Toc169504968)

[**Figure 5 First circuit diagram** 24](#_Toc169504969)

[**Figure 6 Second system diagram** 25](#_Toc169504970)

# List of Tables

[**Table 1 IoT inventions and research** 13](#_Toc169437812)

[**Table 2 First System libraries** 18](#_Toc169437813)

[**Table 3 Second System libraries** 20](#_Toc169437814)

[**Table 4 Summary of the implementation Tools** 22](#_Toc169437815)

[**Table 5 LCD pins** 23](#_Toc169437816)

# List of Abbreviations

|  |  |
| --- | --- |
| IC  IoT | Integrated Circuit  Internet of things |
| RTC | Real time clock |
| 5G  RFID | fifth-generation wireless technology  radio frequency identification |
| BT | body temperature |
| ECG | electrocardiograph |
| HR | heart rate |
| PAT | Pulse Arrival Time |
| BP | blood pressure |
| PPG | Photoplethysmography |
| GND | Ground |
| MCU | Microcontroller unit |
| GPIO | General purpose input output |

# Introduction

## Background Information

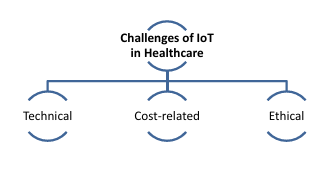
The term Internet of Things (IoT) was invented by Kevin Ashton in 1999 and refers to data on the Internet that are connected to evolving global service architecture.

Entering the 2020 decade, more devices are connected to the internet than ever before. Therefore, the growth of IoT technology has driven interest in a wide range of health practices to improve population health more specifically. Recent reviews have overviewed the various services and applications of IoT in health care. These services have been detailed extensively and can have many applications across single condition and cluster condition management, including, for example, the ability to track and monitor health progress remotely by health care professionals, improve self-management of chronic conditions, assist in the early detection of abnormalities, fast-track symptom identification and clinical diagnoses, deliver early intervention, and improve adherence to prescriptions. These applications can make better use of health care resources and provide quality and low-cost medical care.

Additionally, several academic papers have previously discussed the potential of IoT in healthcare. Traditional literature reviews and systematic literature reviews were used by researchers to summarize IoT research in healthcare. Several aspects of IoT adoption in healthcare have been examined, ranging from exploring the privacy and security issues in IoT-enabling healthcare systems and exploring IoT for elderly care, to examining the potential of the technology for combatting the current COVID-19 pandemic. For example, Dimitrov in 2017 examines IoT and big data applications in healthcare and finds that health education, fitness, symptom monitoring, care coordination, and collaborative disease management are just a few of the areas where IoT wearables are making a positive impact.

## Problem Statement

The challenges of IoT in health care can be classified as technical, cost-related, ethical.



**Figure 1 IoT challenges**

* **Technical considerations**: arise from the fact that IoT is still not a part of everyday life. In most countries, the fifth-generation wireless technology(5G) and subsequently IoT services are not available. Not only patients but also the vast majority of healthcare practitioners and researchers are not well acquainted with what IoT comprises and what can IoT do in everyday life let away from health care.

The next technical challenge lies with the integration of data. Multiple sources of data mean multiple devices. In the healthcare sector, there is a variety of wearables and data collecting devices that cannot be easily modified to a unified pattern of data collection for technical and financial reasons.

* **Financial Challenges:** From a financial point of view, IoT belongs to the sphere of remote health applications. According to the International Data Corpo ration (IDC), the current budget for remote health monitoring in Europe is e10.41 billion, and it grew to over e12.4 billion. So, the high cost of monitoring equipment in the IoT healthcare system is a serious issue that should be addressed.
* **Ethical Challenges:** There are ethical challenges to privacy and security. Hackers can easily have access to all medical records that are stored in the cloud or a web server. That’s why security and privacy should be addressed in IoT system design in order to maintain user authentication, data ownership and data-protection policies.
* In addition, there is the sensor accuracy challenge/problem where the values obtained by the IoT based sensors are not 100% accurate and of course this fact should be taken into consideration.

Finally, many other challenges could be confronted such as delays, connectivity, wiring, memory consumption, ect…

## Objective and Motivation

In our perspective, this project idea has become more popular after the global COVID-19 pandemic, where hospitals weren’t able to provide healthcare to all patients and track essential vitals such as heart pulse, body temperature and oxygen rate.

By implementing an Iot health monitoring system, the health staff will be able to monitor all covid patient’s vitals using the network while they are at home and enable the staff to provide alerts and advices without the need to be at the hospital.

The goal of this project is to demonstrate the efficiency of such technology in monitoring real-time important vital signs, exporting and visualization of data.

# Literature Review

## Introduction

The proliferation of high-tech hardware and software platforms, the expansion of communication channels, and the development of cutting-edge data analysis tools have all contributed to the rapid rise of the Internet of Things (IoT), which is defined as a network of interconnected devices that monitor environmental variables.

One of the main ideas behind IoT is to enable objects that generate and/or collect data to interconnect via technologies such as radio frequency identification (RFID), actuators, sensors, and mobile phones.

IoT health monitoring played a major role in developing medical field. Many studies and academic papers as well as many monitoring systems were implemented through the past years.

The wearable device developed by Wu et al. [16] monitors various physiological parameters, including body temperature (BT), electrocardiograph (ECG), and heart rate (HR). Using Pulse Arrival Time (PAT) to measure ECG and PPG, it is possible to estimate blood pressure (BP). The interaction between humans and remote monitoring programs is straightforward because all the components are designed within a rigid framework. In addition, the power consumption of the devices is low, and they can communicate wirelessly to make tailored measurements of a specific physiological signal.

The table below summarize some of the recent studies and implementations:

|  |  |  |  |
| --- | --- | --- | --- |
| Study | Aims | Advantages | Disadvantages |
| Islam et al. | To collect data on patient’s BT, HR and other vital signs as well as environmental factors in the hospital room, such as CO, CO2, and humidity. Then medical staff can view data in real-time and remotely. | This system could save more lives and could be helpful in managing infectious-disease outbreaks, such as COVID-19. | The system still lacks some epidemic-related sensors that need to be evaluated once implemented. |
| Al-Sheik and Ameen | Remotely monitors patient’s vital signs, including BT, ECG, and blood-oxygen saturation (SpO2) and uses Wi-Fi to send the data to a cloud service on the IoT platform called Blynk. | For security and privacy reasons, the results are sent to a specific smartphone that the doctor can monitor. | Two microcontrollers, Arduino and NodeMCU are used, which still need to be improved.  For long-distance transmission, Wi-Fi technology is not the ideal option. |
| Ahmadi et al. | review 60 studies to examine several facets of healthcare IoT architecture, the barriers to IoT adoption, and the ways different parts of IoT architecture interoperate | - | - |
| Mieronkoski et al. | summarize 62 studies to examine how IoT can support basic nursing practices. | - | - |
| Qadri et al. | To explore the ways in which the Internet of Nano Things and Tactile Internet are reshaping healthcare IoT systems  To discuss the future path for enhancing the Quality of Service using these novel technologies. | - | - |

**Table 1 IoT inventions and research**

This project examines about the open API administration called IoT Wi-Fi with NodeMCU ESP8266 gadget. It plays as a host for various quantities of sensors detecting and observing patients’ pulse and temperature information at ThingSpeak remotely and on LCD simultaneously. Furthermore, a buzzer will be either on or off depending on heart pulse rate.

# System Design

## Introduction

The healthcare system is divided into two sub-systems:

1. The first sub-system is composed of an Uno Board, a DS3231 RTC module to keep track of the time, an LCD, and red and blue LEDs. All components are connected using 20 cm male to female jumpers.

2. The second sub-system, which is the main objective of this project, involves using a NodeMCU\_ESP8266, two wearable devices: a pulse sensor to detect heart rate and an LM35 temperature sensor to detect body temperature, an LCD screen, a buzzer. All components are connected using 20 cm male to female jumpers as well as 30 cm female to female jumpers and a mini breadboard.

The entire system is housed in a custom-designed 40\*34\*10 plastic box.

## System Requirements

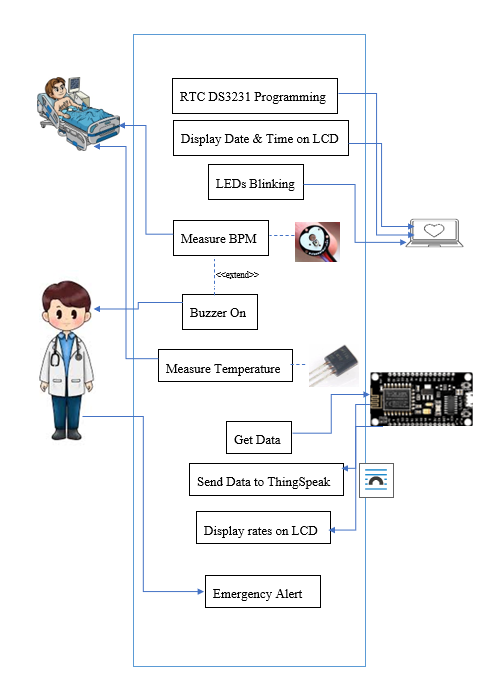
1. The first sub-system uses:

* Uno Board that interface with and control electronic components such as sensors.
* The DS3231 RTC module used to keep track of the time even if the power is off. It takes its main power from a 3V battery and it has a low power consumption with accurate time tracking.
* An LCD is implemented so that the patient (user) could see the current date and time
* Red and blue LEDs are used to represent ambulance siren colors.

1. The second sub-system involves:

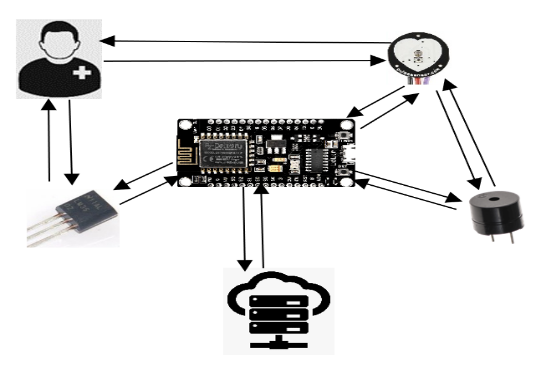
* A NodeMCU\_ESP8266, which is essentially a WIFI module integrated with Microcontroller. It allows the health monitoring system to connect to the internet wirelessly and transmitting real-time sensors data to remote servers or cloud platforms which is ThingSpeak in our case for storage, analysis and processing
* Two wearable devices:
* A pulse sensor that continuously measures the patient’s heart rate. It works based on the principles of PPG.
* A LM35 temperature sensor to detect body temperature.
* An LCD screen that displays heart and temperature rates simultaneously and are updated every 15 seconds.
* A buzzer to alert caregivers if any heart rate value is out of the normal range, which is under 50 or above 120.

### Use Case Diagram



**Figure 2 Use case diagram**

## System Architecture



**Figure 3 System architecture**

## Non-Functional Requirements

The health monitoring system is addressed to every individual patient or not that are interested in remotely tracking their important sign vitals such as heart and body temperature rate. Additionally, it is addressed to all caregivers such as nurses and doctors that will be able to track their patient anytime and anywhere.

On the other hand, this project can be supported with high security and encryption to prevent any attack of patient privacy.

Furthermore, the heart pulse and temperature sensors used maintain high availability to ensure continuous monitoring of health parameters in real time and with high accuracy.

## Conclusion

In conclusion, our implemented IoT Health Monitoring system is able to detect and transmit pulse and temperature rate of a given patient with high accuracy, low cost and low power consumption. Then, it can send those data into any cloud or server which will then facilitate monitoring patient health by caregivers who will be able to visualize all real time vital signs rates.

# Implementation and Results

## Introduction

Regarding hardware part, two separated systems where implemented:

The first one used an Arduin UNO Board where we have added an LCD, a red LED, a blue LED and a DS3231 RTC Module using a 20 cm male to female jumpers. Connected to a laptop as a power supply for UNO Board it will display time and date on the LCD by the help of DS3231, as well as blinking two LEDs simultaneously.

The second system used a NodeMCU ESP8266 Wi-Fi module where we connected an active buzzer, a heart pulse sensor, an LM35 temperature sensor and an LCD. Connected to a laptop as a power supply for the microcontroller, we can detect BPM and body temperature using those sensors and send them via Thingspeak server and also displaying it on an LCD. In addition, the buzzer will start functioning if any out-of-range BPM rate is detected.

For the software coding, two separated codes, one for each system was written in C++ language and uploaded using Arduino IDE. Additionally, ThingSpeak platform was used to send and collect real-time data.

The testing procedure, started with ensuring that every component is working and turned on. Then test every sensor alone to test its accuracy by tracking the serial monitor. After that, a condition was added on the code to let the buzzer ON if the BPM is under 40 or above 120 and to be OFF otherwise. After everthing was completed, we have started by trying to send data to ThingSpeak server and track all data to ensure that it is compatible with what the serial monitor is showing.

## Implementation Tools

In this paragraph, we will list and talk about all used software Tools such as libraries, as well as all hardware Tools used with some diagrams and explanations.

### Software Tools

* **Arduino IDE:**

1. Time and Date system:

|  |  |
| --- | --- |
| Libraries | Aims to |
| <Wire.h> | Facilitate communication with I2C devices. |
| <LiquidCrystal\_I2C.h> | support LCD displays that use I2C communication. |
| <DS3231.h> | designed to interface with the DS3231 Real-Time Clock (RTC) module |

**Table 2 First System libraries**

Now let us explain some parts of the code:



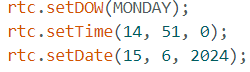
Define pin for each used LED.



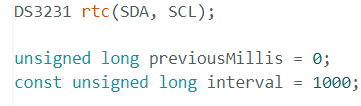
0x27=I2C address.

16= number of columns.

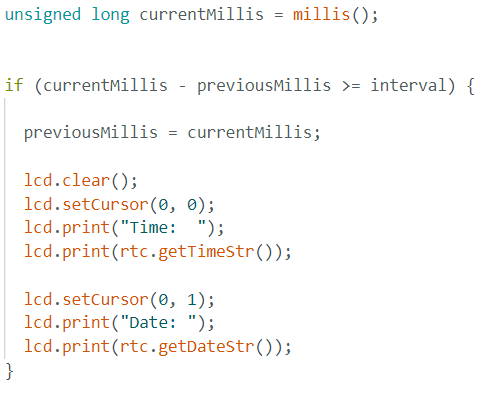
2=number of rows.



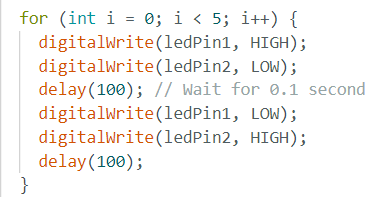
Aims to setup the DS3231 time module with the current date and time and then it will be able to continue tracking real date and time even if the power if off.



* Initialization of RTC DS3231 where SDA & SCL represent the data and clock lines for I2C communication.
* previousMillis to store the previous time.
* Interval define the interval between two actions which is 1 second.



If Loop in order to update the time every 1 second obtained from the RTC and print it in the first row of LCD, as well as, printing the date in the second row.



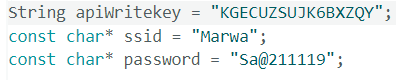
This for loop is used to obtain LEDs blinking where each LED will be on and off every 0.1 s.

1. Health monitoring system:

|  |  |
| --- | --- |
| Libraries | Aims to |
| <Wire.h> | Facilitate communication with I2C devices. |
| <LiquidCrystal\_I2C.h> | support LCD displays that use I2C communication. |
| <ESP8266WiFi.h> | Designed for ESP8266 WIFI Module. |
| <PulseSensorPlayground.h> | designed for interfacing with Pulse Sensor devices. |

**Table 3 Second System libraries**

Now let us explain some parts of the code:

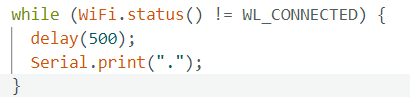


Define API Write Key for our ThingSpeak channel where the values will be sent and collected.

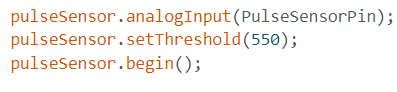
SSID & password are the two initial credentials to connect ESP8266 with WIFI.

* In the setup loop:

In this loop, is used for initialization of our esp8266 Arduino project such as LCD, Heart pulse sensor, WIFI connection, buzzer ect…



Check WIFI status every 500 ms until the ESP8266 is connected.



Initialize pulse sensor and adjust the value obtained using the threshold value to be as accurate as possible.



Ensure that the buzzer is low at startup.

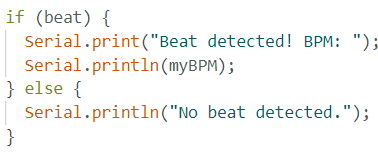
* In void loop ():



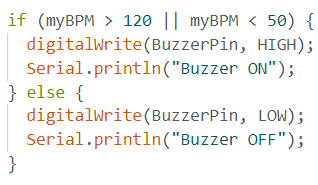
Read the temperature using serial port of LM35 and convert the value from Fahrenheit to Celsius.



Read from pulse sensor analog pin the BPM and check if the beat is detected or not.



Serial monitor debugging messages indicating whether a beat is detected or not.



This loop defines the range of BPM that control when the buzzer should be on or off meaning HIGH or LOW.



Connectivity with the ThingSpeak server in order to send BPM and temperature rates, and at the same time, those values will be printed on LCD display.

Note that, the values will be updated every 15 seconds on both ThingSpeak and LCD.

* **Thignspeak platform:**

ThignSpeak platform is widely used in IoT project, in order to Sens, collect and visualize charts of real time data. Those data will be sent using WIFI with the help of a NodeMCU ESP8266 to a channel having the API writing key mentioned within the Arduino IDE code.



**Figure 4 ThingSpeak Channel**

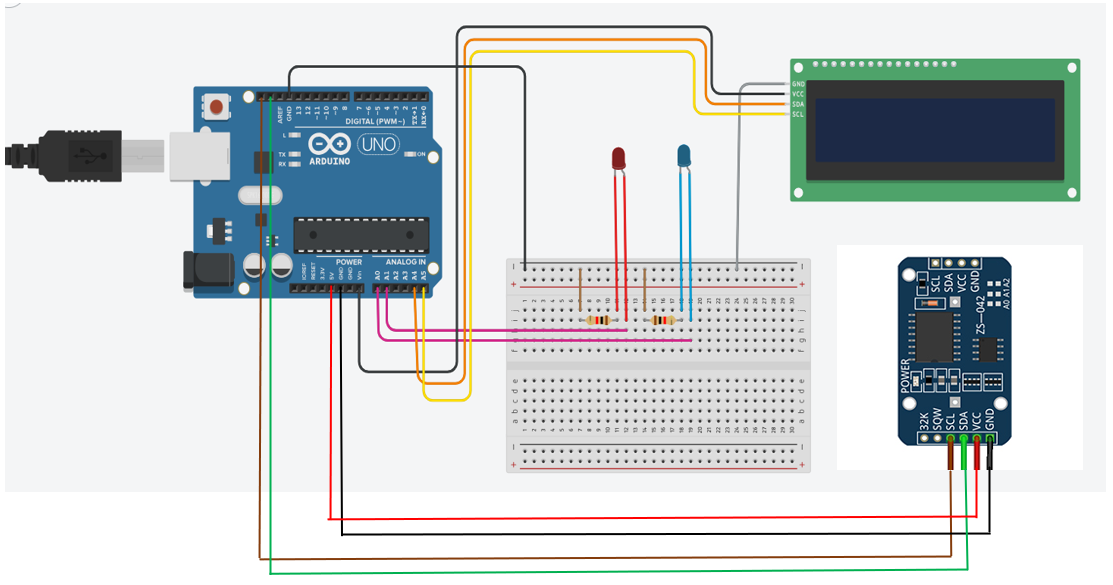
### Hardware Tools

**Table 4 Summary of the implementation Tools**

|  |  |
| --- | --- |
| Components | Usage |
| Male to Female Jumpers | Components connectivity |
| Female to Female Jumpers | Components connectivity |
| Uno Board | Microcontroller |
| ESP8266 NodeMCU | Microcontroller with WIFI |
| LCD with I2C | Characters display |
| LEDs | ON/OFF |
| DS3231 RTC Module | Real time tracking |
| Active Buzzer | As an alert |
| LM35 | Temperature Sensor |
| Pulse sensor | Detect BPM |

## Implementation Steps

1. **First circuit:**



**Figure 5 First circuit diagram**

The main board used in this circuit is an Arduino UNO Board connected to our PC for power supply.

The breadboard is used to facilitate connections between the microcontroller and the components. A common GND is implemented connecting the GND of UNO Borad to the (-) of breadboard.

The LCD is composed of 4 pins:

|  |  |  |
| --- | --- | --- |
| Pin (from up to down) | Connection | Description |
| GND | (-) of the breadboard | - |
| Vcc | Vin which is 5 volts | Power supply for LCD |
| SDA | To A4 | Data Line for I2C communication |
| SCL | To A5 | Serial Clock Line for I2C communication |

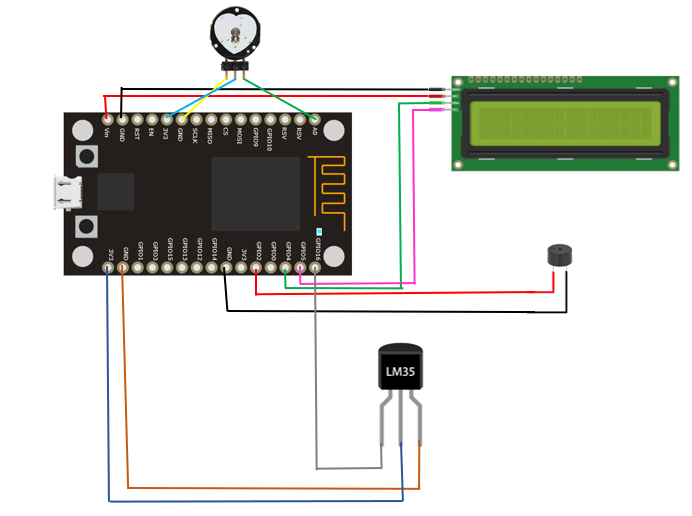
**Table 5 LCD pins**

In general, each led is composed of two legs, the long leg which is the anode is connected to an UNO analogue pin. The short leg which is the cathode is connected to one side of the resistor and the second side is connected to the ground.

In our case, the anode of red LED is connected to A1 and the anode of blue LED connected to A1. A 220-ohm resistor is used.

Regarding DS3231 RTC Module, it is composed of 6 pins where the GND pin is connected to ground, Vcc to 5V, SDA and SCL to SDA and SCL of the UNO Board respectively. In addition, it has a built-in 3V backup battery to keep time tracking even when the Vcc is disconnected.

1. **Second circuit:**



**Figure 6 Second system diagram**

In this circuit, an ESP8266 NodeMCU board is used which combine between a microcontroller unit (MCU) and WIFI connectivity capability.

The LCD composed of 4 pins where the first top pin is connected to the ground. The second one which is Vcc is connected to Vin referring to 5 volts power supply. The following pin which is SDA is connected to GPIO4 and the last one which is SCL is connected to GPIO5.

The buzzer presents only to pin where the longest one is connected to GPIO2 and the shortest to GND.

Regarding LM35 temperature sensor, it has 3 pins where the left pin is connected to GPIO16, the middle on to 3V3 and the last one to the ground.

For pulse sensor, it also has three pins, where the left pin connected to ground, the middle one to 3V3 and the last pin to A0.

## Results

By running the two codes simultaneously, the first system will display the actual date and time and update it every 1 second approximatively. Additionally, the two LEDs will blink one after another fast with a delay of 0.1 seconds.

Working with the second system, we will be able to remotely take our pulse rate and body temperature that then will be sent, collected and monitored, in real-time, in a ThingSpeak server, and also will be displayed on LCD screen.

The data is updated every 15 seconds.

Finally, as we desired, the buzzer will be turned on every time the pulse rate goes below 50 or above 120.

## Conclusion

In conclusion, the NodeMCU ESP8266 is a low cost yet very efficient chip to be used for data transmission via WIFI, ensuring real time monitoring of all vital signs in an IoT health system.

The two used sensor are accurate most of the time and don’t need high power supply to complete desired tasks.

# Conclusion

## Conclusion

After completing this project, we can conclude the following:

The used sensors for tracking human vitals signs are most of the time accurate, but, definitely not all the time.

To make sensor data more reasonable, we should test many times the system, then, using the code, we should find the best adjustment according to real life vitals.

The ESP8266 NodeMCU is a really efficient bridge between IoT systems and any cloud/server but demands a really good connectivity to send data remotely.

In contrast, ESP8266 doesn’t support a high number of sensors.

Implementing an advanced IoT health system can definitely lower the cost of healthcare, and it also plays a major role in pandemics such as COVID-19 where the patient vitals can be monitored by the caregivers without the need to be in a room hospital.

A high security should be available to keep patient record private.

## Future work

* Add more health-related sensors such as ECG, oximeter, muscle sensor.
* Collaborate with cybersecurity to elevate data encryption, secure communication protocols, in order to increase patient’s data privacy.
* Create a mobile application that will interface with IoT health system. It will simplify caregivers’ tasks in monitoring patient’s vital signs.
* Replace ESP8266 NodeMCU by ESP32 for higher performance and features such as Bluetooth connectivity, higher throughput WIFI, and handling more complex systems with higher number of sensors.

# References

* [9 Biometric Sensors Arduino Compatible | Random Nerd Tutorials](https://randomnerdtutorials.com/9-biometric-sensors-arduino-compatible/)
* [NodeMCU ESP8266 Pinout, Specifications, Features & Datasheet (components101.com)](https://components101.com/development-boards/nodemcu-esp8266-pinout-features-and-datasheet)
* [Introduction to NodeMCU ESP8266 - IoTEDU (iot4beginners.com)](https://iot4beginners.com/introduction-to-nodemcu_esp8266/#google_vignette)
* <https://howtomechatronics.com/tutorials/arduino/arduino-ds3231-real-time-clock-tutorial/>
* [IoT-Based Healthcare-Monitoring System towards Improving Quality of Life: A Review - PMC (nih.gov)](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9601552/)
* [The Internet of Things (IoT) in healthcare: Taking stock and moving forward - ScienceDirect](https://www.sciencedirect.com/science/article/pii/S2542660523000446#fig0001)
* <https://iotdesignpro.com/projects/iot-based-heartbeat-monitoring-system-using-nodemcu>
* [IoT Based Patient Health Monitoring System using ESP8266 & Arduino (youtube.com)](https://www.youtube.com/watch?v=Utlnd_FDB9A)
* <https://how2electronics.com/iot-bpm-monitoring-on-thingspeak-using-esp8266-pulse-sensor/>
* [IoT Digital Thermometer using NodeMCU ESP-12 and LM35 Temperature Sensor (circuitdigest.com)](https://circuitdigest.com/microcontroller-projects/iot-digital-thermometer-using-nodemcu-esp12-and-lm35)
* <https://circuitdigest.com/microcontroller-projects/interfacing-ds3231-rtc-with-arduino-and-diy-digital-clock>
* [A systematic review of IoT in healthcare: Applications, techniques, and trends - ScienceDirect](https://www.sciencedirect.com/science/article/abs/pii/S1084804521001764)