A project report

On

Development of Smart Health Monitoring System in IoT Environment

Submitted By

Shobnom Mushtary

ID- BKH1801030F

Session- 2017-18

Under the Supervision

of

Md. Javed Hossain

Professor

Department of CSTE

DEPARTMENT OF COMPUTER SCIENCE & TELECOMMUNICATION ENGINEERRING,

NOAKHALI SCIENCE & TECHNOLOGY UNIVERSITY

Date of Submission: 8th May, 2023

The project entitled “Development of Smart Health Monitoring System in IoT Environment” has been submitted by Shobnom Mushtary, Roll no. BKH1801030F and has been accepted as satisfactory in partial fulfillment of the requirement for the degree of Bachelor of Science in Computer Science & Telecommunication Engineering in Noakhali Science And Technology University.

BOARD OF EXAMINERS

1. _____

Dr. Ashadun Nobi

Professor & Chairman

Department of Computer Science & Telecommunication Engineering

Noakhali Science & Technology University

Chairman

2. _____

Department of Computer Science & Telecommunication Engineering

Noakhali Science & Technology University

1st Examiner

3. _____

Department of Computer Science & Telecommunication Engineering

Noakhali Science & Technology University

2nd Examiner

4. _____

3rd Examiner

STUDENT DECLARATION

I am **Shobnom Mushtary** hereby state that the presented report of my project titled “Development of Smart Health Monitoring System in IoT Environment”, is exclusively prepared and updated by me after completion of my project successfully under the supervision of Prof. Md Javed Hossain.

Shobnom Mushtary

ID - BKH1801030F

Department of CSTE

Noakhali Science & Technology University

STATEMENT OF ORIGINALITY

This is to certify that the project entitled, "Development of Smart Health Monitoring System in IoT Environment" has been submitted by Shobnom Mushtary (Roll No: BKH1801030F) in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science & Telecommunication Engineering (B.Sc. Engg. in CSTE) at the Noakhali Science and Technology University. It is an original work done by her under my supervision and guidance.

Signature of the Supervisor

Date:

Signature of the Candidate

Date:

ACKNOWLEDGEMENT

All praises to THE ALMIGHTY, the supreme Authority of this universe for giving His love, help and strength till I completed my BSc Engineering.

I express my sincere gratitude and deep respect to my supervisor Md. Javed Hossain, Professor, Department of Computer Science & Telecommunication Engineering, Noakhali Science and Technology University, Noakhali for his cordial guidance, insight and generous help during the course of this thesis work. During the successful completion of this project work, my supervisor gives valuable guidance and affection to perform at my best. His sincere sympathies and assistance always encouraged me to carry out the present work firmly. Without his invaluable advices and cooperation, the completion of this thesis would not be possible. I would also like to thank all of my respectful teachers, Department of Computer Science and Telecommunication Engineering, NSTU, for their encouragement and guidance. Their inspiration also helped me to go on my own way to complete the project.

Finally, I also want to give special thanks to my parents and all of my friends who helped and supported me through their appreciation, encouragement and criticism in completing the research work and writing the dissertation.

Shobnom Mushtary

ABSTRACT

The health monitoring system has become popular these days due to uniqueness and diversified usage in the medical field. Everyday many lives are affected because the diseases are not timely and properly diagnosed so we didn't get a chance to provide medical help. Data collected from IoT devices can help physicians identify the best treatment process for patients and reach the expected outcomes. An HMS is a vital tool that provides real-time data using wireless sensor networks. The sensors that attach to a patient's body to monitor the patient health parameters and the software that an HMS includes helps transmit the information to their doctor or a hospital. Health monitoring is very important in terms of prevention, particularly if the early detection of diseases can reduce suffering and medical costs. The diagnosis and prompt treatment of various diseases can radically improve alternatives for the medical treatment of the patient. To deal with these types of situations, this system will help to monitor a patient's certain parameters and predict the patient's condition from time to time. This system is user friendly and reduces human efforts.

Index terms: IoT, NodeMCU, Blood Oxygen and Pulse Sensor, Human Body Temperature Sensor, Blynk IoT.

Table Of Contents

List of Figures

Chapter 1
Introduction

Overview:

- Introduction
  - Concept of Smart Health Monitoring System & IoT
- Proposed System
  - Features of Proposed System
- Motivation
- Objective
- Identification of problems
- Justification of the project

This chapter, I have discussed to give a short introduction about my project and also focus on existing system and proposed system.

1.1 INTRODUCTION

1.1.1 Concept of Smart Health Monitoring System & IoT

Smart Health Monitoring is the term of developing model of a real time monitoring system.

Bangladesh is a developing and over-populated country. Healthcare in Bangladesh is not as sophisticated as in more developed countries. There is a huge shortage of physicians, specialists and clinical equipment in the hospitals. So, it is very difficult to keep each & every patient under observation all the time. As a result, sudden deterioration of a patient causes unexpected death. Specially, in rural hospitals, the facilities for health caring are limited. The poor quality of health management enables issues in health care system. Patients are facing a problematic situation of unforeseen demise due to the specific reason of heart problems and attack which is because of nonexistence of good medical maintenance to patients at the needed time.[1] This is for specially monitoring the old age patients and informing doctors and loved ones.

IoT devices are used in many application fields which make the users' day to day life more comfortable. Internet of Things (IoT) based health monitoring system is the current solution for monitoring the patients all day long. The Internet of Things (IoT) is internetworking of physical devices [2]. This system has ability to transfer data over a network without requiring human-to-human or human-to-computer interaction. IoT sensors are capable of providing information about patients' health, we have proposed an IoT and smart health monitoring system using automation. This IoT based health monitoring system makes use of wireless sensor networks that collects data from different sensors deployed at various nodes and sends it through the wireless protocol. The doctors, nurses and the loved ones of a patient can use this IoT systems to observe the health condition of a patient. In case of any abrupt changes in patient oxygen level, heart rate or body temperature alert is sent about the patient using IoT immediately and the doctors can take the necessary steps as early as possible. This system also shows patients temperature, oxygen level and heartbeat tracked live data with timestamps over the Internetwork. Remote Patient Monitoring arrangement empowers observation of patients outside of customary clinical

settings (e.g., at home), which expands access to human services offices at bring down expenses [3]

1.2 PROPOSED SYSTEM

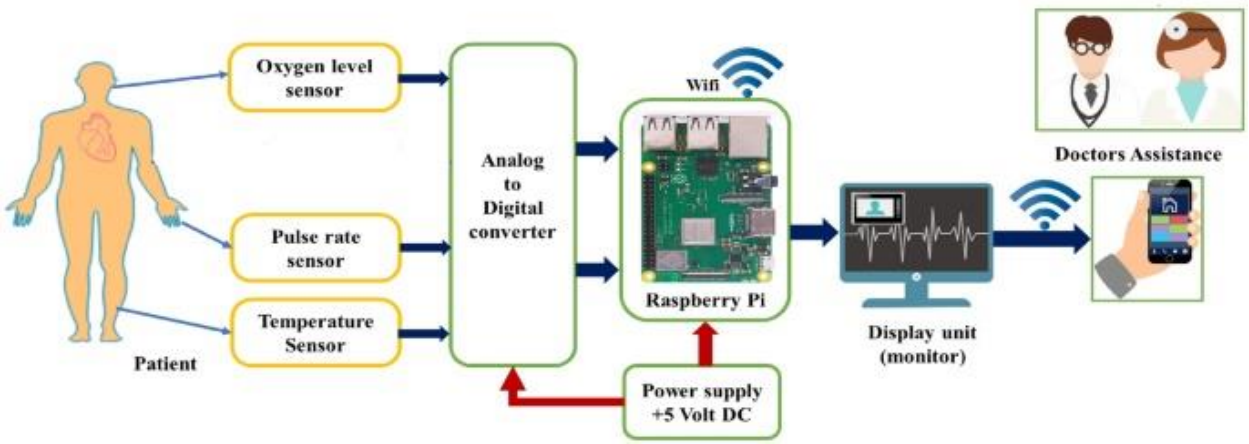

Figure 1.1: Proposed System

The core objective of this project is the design and implementation of a smart patient health tracking system. Fig.1 shows the overview of the proposed system. The sensors are embedded on the patient body to sense the temperature and heartbeat of the patient. Two more sensors are place at home to sense the humidity and the temperature of the room where the patient is staying. These sensors are connected to a control unit, which calculates the values of all the two sensors. These calculated values are then transmitted through a IoT cloud to the base station. From the base station the values are then accessed by the doctor at any other location. Thus based on the temperature and heart beat values and the room sensor values, the doctor can decide the state of the patient and appropriate measures can be taken.

1.2.1 Features of Proposed System

The features included in this Smart health monitoring system are:

1. Measure Human Body Temperature: Measuring body temperature is very important in medicine. Several diseases are characterized by a change in body temperature. LM35 sensor is used for measurement of body temperature.

2. Measure Blood Oxygen level & Pulse Rate: MAX30100 is an integrated pulse oximetry and heart-rate monitor sensor solution.

3. Inform Doctors & Nurses: Doctors & Nurses can monitor patients' health from remote distance by using Internet.

1.3 MOTIVATION

The core to our motivation was to think ahead of our time and to contribute on the sector Internet of Things (IOT) which is undoubtedly the next big thing on the technological market. And IOT has proven itself to be a very handy possession. The idea of this project came to us by observing how easy it is to connect or integrate the normal home appliances with the internet. This led us to the secluded health sector which is still lagging behind from other industries in terms of technology. We observed many case even from our daily lives that in the hectic world we are living right now, it is difficult to keep a watch over everything. Specially, to keep an eye on the elderly people is difficult and costly as well. These real-life situations gave us the ideas and motivation to integrate the two sectors so that it helps in developing such a project which will create an economic and efficient health monitoring system and will pave the way for future work on the field of IOT.

1.4 OBJECTIVE:

- To design and implement an automated system which will help to monitor host remotely is our primary objective.
- To develop health monitoring system that measures body temperature, heart rate, oxygen level in blood.
- To analyze collected data of sensors.
- To track patient health and uses internet to inform doctors, nurses and their loved ones in case of any issues.
- To develop this kind of monitoring systems is to reduce health care costs and also reduce the possibilities of sudden death.

1.5 IDENTIFICATION OF PROBLEM

- Healthcare services and facilities in hospitals and clinics are very poor in Bangladesh.
- Health monitoring systems are not so efficient.
- There is a huge shortage of physicians, specialists, other staffs and clinical equipments in the public and private hospitals. So, keeping each & every patient under observation and monitoring their health all the time is very difficult.
- As a result, suddenly any patient's condition can deteriorate and it may cause unexpected death also.

1.6 JUSTIFICATION OF THE PROJECT:

- Internet of Things (IoT) based health monitoring system is the current solution for monitoring the patients all day long.
- It is a small and affordable system.
- It is an automated system by which doctors, nurses and others related to the patient's healthcare in hospitals can observe the condition of a patient without physically being in front of the patient.

Chapter 2
Literature Review

Overview:

- Related Works
- Research Contribution

In this chapter, I have discussed about related work and its limitations. And also discussed about Research Contribution of my project.

2.1 Related Works:

Smart Health Monitoring system is a real time monitoring system. There have been a number of research and project in Smart Health Monitoring System. An extensive research on the topic related to the system shows a very few of the related works could actually build their own preliminary framework and prototype of the system. Some of the works like the research conducted on the Ambient Assisted Living (AAL) [2] actually did more of a literature survey of the state of its present condition of the monitoring system via IoT. They also tried to identify and highlight the critical issues and the quality of service as well as the user driven experiences in their work. [3] Internet of Medical Things (IoMT): Applications, Benefits and Future Challenges in Healthcare Domain. Here they research about healthcare only. Some, other worked on showing or highlighting the importance of IoT in the health sector and some proposals for the health monitoring architectures. Some related findings used specific models for the health monitoring aspect. Like the abstraction of Model Driven Tree Reference Model (MDTRM), where they explained the necessity of this model in the health field as well as identifying the complexities of the models. They also benched marked the models which came really handy for the initial phase of this research. [5] This project calculates the result of temperature, alcohol and heartbeat show on the LCD screen. By this project it cannot monitor patients from remote place. Some other related model we found are General Domain Model Architecture (GDMA), the health monitoring and sensing with cloud processing was also a helpful source behind the research, as it was useful for generating ideas to get raw data from wearable devices which are compatible and capable of measuring many physical value which we can use to obtain meaningful results. Masimo Radical-7, a health monitor for clinical environment helps to collect data and wirelessly transmits it for ongoing display. This provides high resolution display of information with higher graphical capabilities. It also has a touch-based user interface. But as it can be already assumed how cost effective it is, it can't send an alarm message to notify for any emergencies. Free Scale Home Health Hub reference platform store patient data to cloud via various sensors, where the people related to the patient can have an access. This platform too can't notify for any alarming situation to the people engaged with the patient. [9] Some surveys of ours also lead us to projects which even discussed to monitor the health whole area through wireless network sensors.

2.2 Thesis Contribution

Of all these ideas, models, frameworks and platforms that are surveyed for this research, we differ with all the above with a very basic and fundamental ways. The above researches conducted may use many monitoring variable on the contrary we are using more specific variable. Still, the basic functionality difference is that our motive to develop this project is to generate a functional response and to give feedback to the relatives of the host, so that they can quickly take steps for the wellbeing of the host. The source of this response will be e-mail and social networking site twitter. But the basic idea of the research is totally of a different paradigm. Though, it is true that health monitoring and the prediction from anomalies as well as giving useful feedback to the use is neither a very easy process nor the framework a very easy to set foot to. But this research will take the health monitoring which now one of the most exciting topics (related to IoT) to a very different level. The useful information gathered from the research conducted has provided very handy to carry out and provide a workable model which will be discussed in the next segment of the paper. Workable data is something which we had to toil to obtain but the data were managed properly to continue the research. Thus, most the related works done related to our project was highlighted in this segment and how our research differs from those works were also explained thoroughly. Some of those researches conducted really helped us understanding the importance of this sector and inspired to go the extra-mile on this field. The platform monitoring the daily activities of the host will be useful for the physician attending the host. The daily activities monitoring conducted by the system will help the physician to develop a conclusion and prediction to diagnose the host in a swift manner.

Chapter-3

Methodology

Overview

- Definition of Methodology
  - Methodology of the system
- Tools Requirement
  - Hardware Component
    - Node MCU
    - SMPS
    - Max30100 Pulse Oximeter
    - Temperature Sensor
    - LCD Display
- Feasibility Study

In this chapter, we have tried to discuss about the methodology of my project and also discussed about tools requirement and feasibility studies of this project.

3.1. Definition of Methodology

Methodology is a system of board principles or rules from which specific methods or rules procedures may be derived to interpret or solve different problems within the scope of a particular discipline. Unlike an algorithm, a methodology is not a formula but a set of practices. Methodology is the analysis of the principles or procedures of inquiry in a particular field. It is the procedure where the system of methods and principles used in a particular discipline.

3.1.1 Methodology of the system

There are different methodologies exist to create a smart agriculture.

- Bluetooth based system
- ZigBee based wireless system
- GSM based system
- Internet of Things (IOT) based system
- Respberry pi based system
- ESP8266 NodeMCU based WiFi system

In this project, we choose ESP8266 NodeMCU based WiFi system. The IoT platform will be used in this project is ThingSpeak. This IoT device is an automatic wireless health monitoring system is used to measure a patient's body temperature, heartbeat, oxygen level, room temperature & humidity by using embedded technology. The proposed system uses the sensors like room temperature & humidity sensor, pulse & oxygen level sensor, a human body temperature sensor. These sensors mainly involve monitoring the condition of the patient. It continuously monitors these parameters and updates them to an IoT platform.

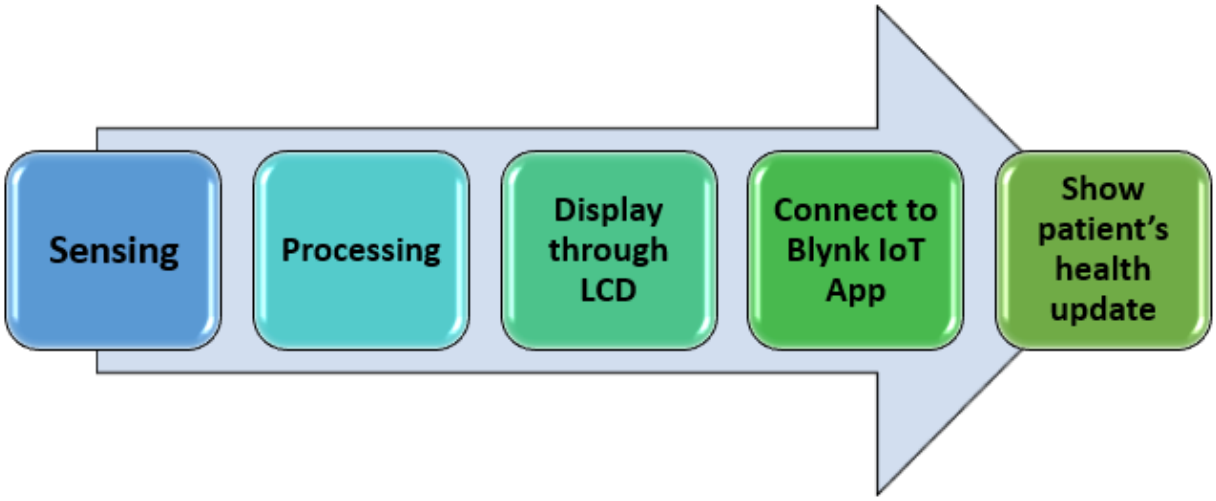

Figure 3.1: Working process of the system.

Smart health monitoring systems using smartphone/laptop/PC/tabloid, NodeMCU and WiFi technology are secured and low cost. The name "NodeMCU" combines "node" and "MCU" (micro-controller unit). The term "NodeMCU" strictly speaking refers to the firmware rather than the associated development kits. The main drawback of the system is sometime the internet connection falls down.

3.2 Tools Requirement

This project consists of NodeMCU as brain. It is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. As well as I am going to utilize below components to successfully execute this project. The components are:

- NodeMCU
- SMPS
- Pulse and Blood oxygen level sensor
- Human body temperature sensor
- LED
- 16*2 LCD display
- Jumper wires
- PC or Smartphone

Hardware Components:

3.2.1 NodeMCU

Here I use two Node MCU or ESP8266 development Board which is transformation of ESP12E wifi module and containing the ESP8266 chip which capacity is 32-bit LX106 RISE. It supports RTOS and it can operate 80MHz to 160MHz with adjustable clock frequency. It contains 1280 KB RAM and also 4 MB of flash memory to store data and programs. In this module we find wi-fi and Bluetooth and deep sleep operating features that make it ideal for IoT Projects. This ESP8266 is a family of Wi-Fi modules developed by the Chinese company Expressive. It is a System on a Chip, abbreviated simply as Sock with integrated TCP/IP protocol stack. It consists of a Ten silica L106 32-bit micro controller and a Wi-Fi transceiver. It has 11 input/output (I/O) pins. And it has a dedicated analog input for the programming purpose. It is programmable and it can be modified in several ways and that is the reason why this chip is the most prevalent IoT device in the market. In my system, I am using this Wi-Fi module because of its higher reliability and easy programming. (Peter, 2007) Wi-Fi module acts as a micro web server and hence eliminates the need for the wired connection between the Arduino board and computer which has many advantages. Main advantages are the reduction in size, cost, and increased convenience.

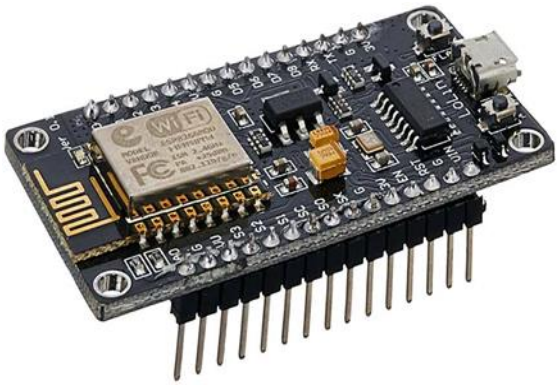

Figure 3.2: Node MCU ESP8266

The Wi-Fi module will require an active internet connection and it will act as the gateway for the Arduino board to communicate with the internet. This proposed system is flexible. I.e. apart from the above-listed components, our system can be interfaced with other sensors according to the user requirement. As Arduino.cc began developing new MCU boards based on non-AVR processors like the ARM/SAM MCU used in the Arduino Due, they needed to modify the Arduino IDE so it would be relatively easy to change the IDE to support alternate tool chains to allow Arduino C/C++ to be compiled for these new processors. They did this with the introduction of the Board Manager and the SAM Core. A "core" is the collection of software components required by the Board Manager and the Arduino IDE to compile an Arduino C/C++ source file for the target MCU's machine language. Some ESP8266 enthusiasts developed an Arduino core for the ESP8266 WiFi SoC, popularly called the "ESP8266 Core for the Arduino IDE".[18] This has become a leading software development platform for the various ESP8266-based modules and development boards, including Node MCUs. The Node MCU (Node Micro Controller Unit) is an open-source software and hardware development environment built around an inexpensive System-on-a-Chip (SoC) called the ESP8266. The ESP8266, designed and manufactured by Espressif Systems, contains the crucial elements of a computer: CPU, RAM, networking (WiFi), and even a modern operating system and SDK. That makes it an excellent choice for Internet of Things (IoT) projects of all kinds.

The NodeMCU (Node Microcontroller Unit) is an open-source software and hardware development environment built around an inexpensive System-on-a-Chip (SoC) called the ESP8266. The ESP8266, designed and manufactured by Espressif Systems, contains the crucial elements of a computer: CPU, RAM, networking (WiFi), and even a modern operating system and SDK. That makes it an excellent choice for Internet of Things (IoT) projects of all kinds. However, as a chip, the ESP8266 is also hard to access and use. [6] You must solder wires, with the appropriate analog voltage, to its pins for the simplest tasks such as powering it on or sending a keystroke to the “computer” on the chip. You also have to program it in low-level machine instructions that can be interpreted by the chip hardware. This level of integration is not a problem using the ESP8266 as an embedded controller chip in mass-produced electronics. It is a

huge burden for hobbyists, hackers, or students who want to experiment with it in their own IoT projects.

But, what about Arduino? The Arduino project created an open-source hardware design and software SDK for their versatile IoT controller. Similar to NodeMCU, the Arduino hardware is a microcontroller board with a USB connector, LED lights, and standard data pins. It also defines standard interfaces to interact with sensors or other boards. But unlike NodeMCU, the Arduino board can have different types of CPU chips (typically an ARM or Intel x86 chip) with memory chips, and a variety of programming environments. There is an Arduino reference design for the ESP8266 chip as well. However, the flexibility of Arduino also means significant variations across different vendors. For example, most Arduino boards do not have WiFi capabilities, and some even have a serial data port instead of a USB port.

However, as a chip, the ESP8266 is also hard to access and use. You must solder wires, with the appropriate analog voltage, to its pins for the simplest tasks such as powering it on or sending a keystroke to the “computer” on the chip. You also have to program it in low-level machine instructions that can be interpreted by the chip hardware. This level of integration is not a problem using the ESP8266 as an embedded controller chip in mass-produced electronics. It is a huge burden for hobbyists, hackers, or students who want to experiment with it in their own IoT projects. But, what about Arduino? The Arduino project created an open-source hardware design and software SDK for their versatile IoT controller. Similar to Node MCU, the Arduino hardware is a microcontroller board with a USB connector, LED lights, and standard data pins. It also defines standard interfaces to interact with sensors or other boards. But unlike Node MCU, the Arduino board can have different types of CPU chips (typically an ARM or Intel x86 chip) with memory chips, and a variety of programming environments. There is an Arduino reference design for the ESP8266 chip as well. However, the flexibility of Arduino also means significant variations across different vendors. For example, most Arduino boards do not have WiFi capabilities, and some even have a serial data port instead of a USB port. The Node MCU Development Board can be easily programmed with Arduino IDE since it is easy to use. Programming Node MCU with the Arduino IDE will hardly take 5-10 minutes. All you need is the Arduino IDE, a USB cable and the

Node MCU board itself. You can check this Getting Started Tutorial for Node MCU to prepare your Arduino IDE for Node MCU. Besides adding WiFi capability, the main claim to fame for the ESP8266 processor over the AVR processor of the standard Arduino is that it has a larger 4 MB of Flash memory and runs at clock speeds of 80 MHz and can sometimes optionally be overclocked to 160 MHz and therefore has a very fast processing speed. The Digital I/O all support PWM and interrupts. In addition they can be configured to have pull-up or pull-down resistors. Though there are 11 digital I/O pins, 2 are typically reserved for use as the TX/RX lines if serial communications are used which leaves 9 digital I/O.

On the down-side, it has only 1 analog input which is probably the most significant limitation for some sensor type applications. That can always be overcome by using an external Analog Mux module like our 16-channel 74HC4067 or our ADS1115 4-channel 16-bit ADC module below if more analog I/O is desired. The module can be powered via the USB port or by using an external 7-12V power supply connected to the VIN pin. The module runs at 3.3V, so keep that in mind when working with I/O. The digital I/O is stated as being 5V tolerant, but the analog input needs to be limited to < 3.3V.The blue on-board LED is connected to D0 (GPIO16) and can be accessed using the LED_BUILTIN constant. The module has a typical 'Reset' button as well as a 'Flash' button. The Flash button is used when programming using the original Node MCU firmware. If the module is being used with the Arduino IDE, the Flash button does not need to be used to program the board and it will program just as any Arduino board would.

The module comes loaded with the Node MCU software that accepts the standard AT command set. The module was initially designed primarily to be programmed using Lua which is an interpreted language, but Lua has been mostly abandoned because it takes up a lot of memory and is slow because it is interpreted and more importantly it is generally buggy. It can also be programmed in C using the Arduino IDE and is how the modules are most often used. An example program is shown down below. If a program is download via the IDE, it will overwrite the Node MCU software or whatever else was loaded before. If that is a problem for what you want to do, the Node MCU software can always be reloaded. here are many instructions for installing and

using ESP8266 based boards with the Arduino IDE, but here is a short-hand version. Note that once the ESP8266 board type is added to the IDE, there will be many more items added to the Tools drop down menu.

3.2.2 SMPS

The full form of SMPS is Switched Mode Power Supply also known as Switching Mode Power Supply. It is an electronic gadget or module that comprises a combination of inductors, capacitors and semiconductor gadgets like diodes and MOSFETs. It is utilized to change over a specific DC voltage to another DC voltage level.

The S.M.P.S operated at input regular voltage AC 100V – 240V.

The S.M.P.S should be capable of a total continuous DC power output of 40 Watts.

The S.M.P.S should be capable of a total peak 50 Watts.

Figure-3.3: SMPS

SENSORS:

3.2.3 Human Body Temperature Sensor:

Measuring body temperature is very important in medicine. Several diseases are characterised by a change in body temperature. With other illnesses, the course of the disease can be followed by measuring body temperature. This allows the doctor to analyse the effectiveness of treatments based on body temperatures. The body temperature on the body surface is about 1-degree centigrade less than the temperature of other parts.[5]

The DS18B20 is a digital thermometer that accurately measures temperature in the range −50°C to 125°C and also includes alarm functions and trigger.

Temperature sensors work by providing readings via electrical signals. Sensors are composed of two metals that generate an electrical voltage or resistance when a temperature change occurs by measuring the voltage across the diode terminals. When the voltage increases, the temperature also increases. A temperature sensor is a key component of any process heating application as it provides temperature feedback about the process, which can be used to monitor or control the process.

Figure 3.4: Temperature Sensor

3.2.4 Blood Oxygen and Pulse Sensor (Pulse Oximeter):

Oxygen enters the lungs and then is passed on into blood. The blood carries oxygen to the various organs in our body. The main way oxygen is carried in our blood is by means of hemoglobin. Body needs a certain amount of oxygen to function properly, and low or high blood oxygen levels can lead to serious complications. Normal oxygen levels in a pulse oximeter usually range from 95 to 100 percent. On the other hand, pulse rate, also known as your heart rate, is the number of times your heart beats per minute (bpm). A normal resting heart rate for adults' ranges from 60 to 100 beats per minute. Abnormal Heart Rates or Heart Beats reflect the cardiac conditions of the body. If unnoticed and untreated, this can sometimes be fatal. Conditions when the heartbeat goes beyond 120-140 beats per minute or falls below 60 beats per minute, can be considered dangerous, and immediate doctor's intervention is a must.[7] If your heart rate is over 100 beats per minute when you are at rest, this is considered fast. A rapid heart rate, also known as tachycardia, can be related to many different health conditions. It's normal for your heart rate to increase when you're exercising or if your body is fighting off an infection.

Figure 3.5: MAX30100 Pulse Oximeter

MAX30100 is an integrated pulse oximetry and heart-rate monitor sensor solution. It integrates two LEDs (IR and Red), a photodetector (Red), optimized optics, and low-noise analog signal processing to detect pulse oximetry and heart-rate signals. It has an I2C digital interface to communicate with a host microcontroller. The pulse oximetry subsystem in MAX30100 consists of ambient light cancellation (ALC), 16-bit sigma delta ADC, and proprietary discrete time filter. It has an ultra-low-power operation which makes it ideal for battery operated

systems.MAX30100 operates on a supply in the range of 1.8 to 3.3V. It can be used in wearable devices, fitness assistant devices, medical monitoring devices, etc. The MAX30100 operates from 1.8V and 3.3V power supplies and can be powered down through software with negligible standby current, permitting the power supply to always remain connected.

Features:

- Integrated LEDs, Photo Sensor, and High-Performance Analog Front-End
- Complete Pulse Oximeter and Heart-Rate Sensor Solution Simplifies Design
    - Integrated LEDs, Photo Sensor, and High-Performance Analog Front-End
    - Tiny 5.6mm x 2.8mm x 1.2mm 14-Pin Optically Enhanced System-in-Package
- Ultra-Low-Power Operation Increases Battery Life for Wearable Devices
    - Programmable Sample Rate and LED Current for Power Savings
    - Ultra-Low Shutdown Current
- Advanced Functionality Improves Measurement Performance
    - High SNR Provides Robust Motion Artifact Resilience
    - Integrated Ambient Light Cancellation
    - High Sample Rate Capability
    - Fast Data Output Capability

Application:

- Fitness Assistant Devices
- Medical Monitoring Devices
- Wearable Devices
- **Package includes:**

  1 x Max30100 Pulse Oximeter Heart-Rate Sensor Module

3.2.5 LCD Display

LCD20*04, or 20*04 character-type liquid crystal display, is a kind of dot matrix module to show letters, numbers, and characters and so on. It's composed of 5x7 or 5x11 dot matrix positions; each position can display one character. There's a dot pitch between two characters and a space between lines, thus separating characters and lines. The model 20*04 means it displays 4 lines of 20characters.

Generally, LCD20*04 has parallel ports, that is, it would control several pins at the same time. LCD20*04 can be categorized into eight-port and four-port connections. If the eight- port connection is used, then all the digital ports of the Sun Founder Uno board are almost completely occupied. If you want to connect more sensors, there will be no ports available. Therefore, the four-port connection is used here for better application.

Pin	Function
VSS	connected to ground
VDD	connected to a +5V power supply
VO	to adjust the contrast
RS	A register select pin that controls where in the LCD's memory you are writing data to. You can select either the data register, which holds what goes on the screen, or an instruction register, which is where the LCD's controller looks for instructions on what to do next.
R/W	A Read/Write pin to select between reading and writing mode
E	An enabling pin that reads the information when High level (1) is received. The instructions are run when the signal changes from High level to Low level.
D0-D7	to read and write data
A	Pins that control the LCD backlight. Connect A to 3.3v.
K	Pins that control the LCD backlight. Connect K to GND.

Tabel.01: Lcd display outline

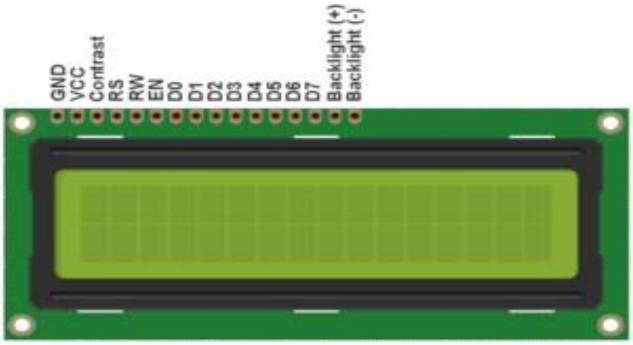

Figure- 3.6: LCD Display

3.2.6 IoT Server:

At whatever point the patient goes to the healing center premises, sensors sense the physiological signs and these signs are changed over to electrical signs [8]. Then simple electrical flag is changed over to advanced flag (computerized information) which is put away in RFID. From nearby server the information is exchanged to the therapeutic server through WLAN. At the point when the information is exchanged to the therapeutic server, it checks whether the patient has any past medicinal record then the server adds the new information to that record and exchanges to the specialist. In the event that patient does not have any past therapeutic record then the server makes new ID and stores the information in its database [9]. This information is exchanged to the specialist for diagnosis. The complete Data transmission using IoT is given in Fig. 2.

Figure 3.7: Data Transmission using IoT

Sensor data	Datatype
Patient ID	Int
Body Temperature	Float
Pulse rate	Float
Blood Oxygen level	Float

Table 02: Database Structure

3.3. Feasibility Study

Feasibility study includes consideration of all the possible ways to provide a solution to the given problem. The proposed solution should satisfy all the user requirements and should be flexible enough so that future changes can be easily done based on the future upcoming requirements.

Software feasibility:

We can make this system easily by using Arduino IDE.

Chapter-4
System Design & Implementation

Overview

- Block Diagram
  - Concept of Block Diagram
  - Different component of Block Diagram
- Circuit Diagram
- Final Illustration of the Project

In this chapter, I have discussed about Block Diagram and Circuit Diagram of my project. And also include some information about different equipments which are related to the diagram.

4.1. BLOCK DIAGRAM

4.1.1. Concept of Block Diagram

A Block Diagram is a diagram of a system in which the principal parts or functions are represented by blocks connected by lines that show the relationships of the blocks. They are heavily used in engineering hardware design, electronic design, software design, and process flow diagram.

4.1.2 Block Diagram:

The health monitoring sensors are used to collect health-related data i.e. for data acquisition. Communication can be done by controller for sending data on internet wirelessly.

These data are then received with IoT platform. With the values received the doctor then diagnose the disease and the state of health of the patient.[10]

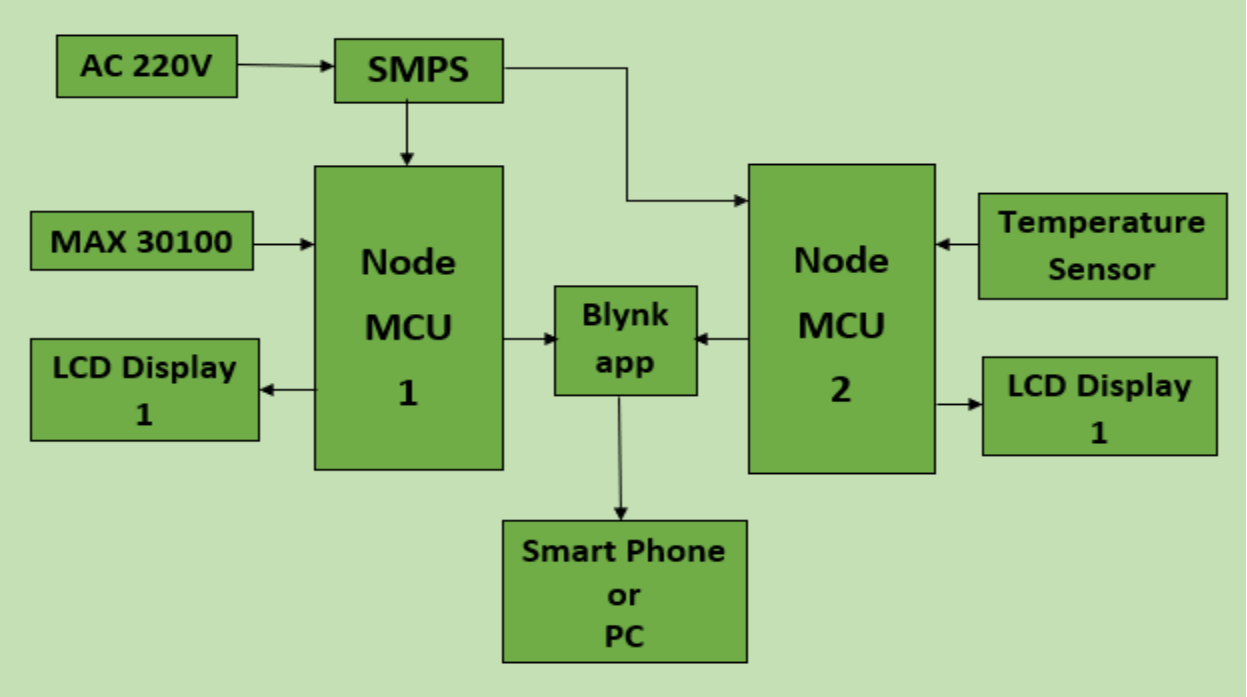

Figure 4.1: Block diagram of the whole system

4.2 CIRCUIT DIAGRAM

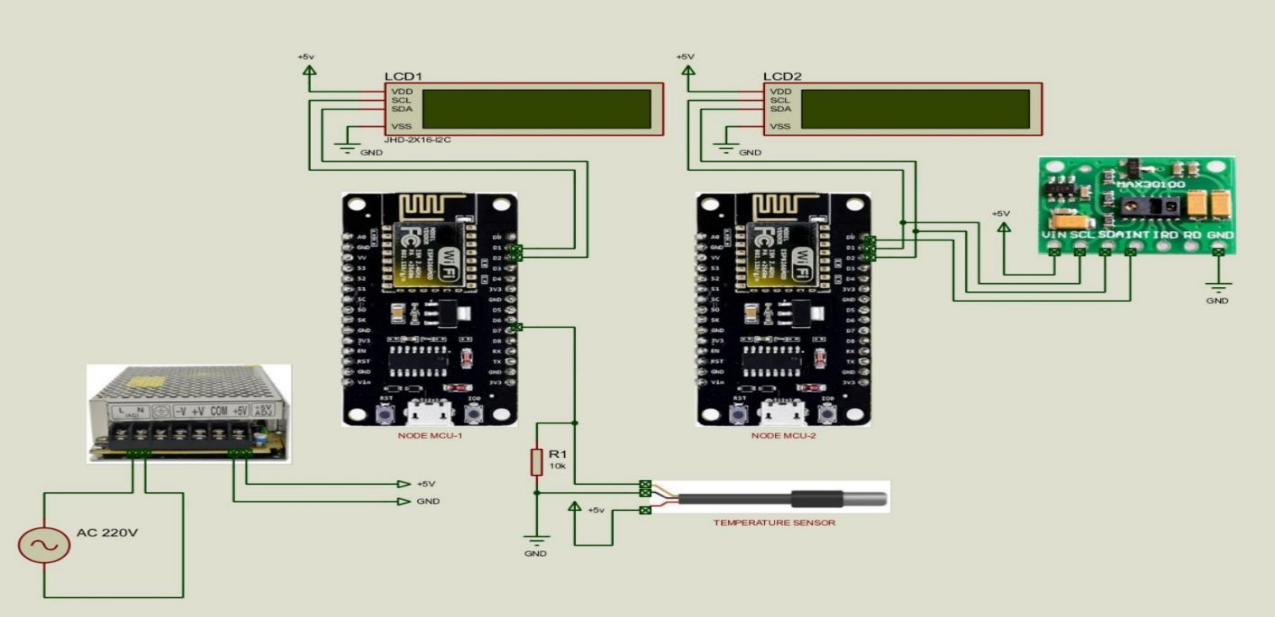

Figure 4.2: Circuit Diagram of Smart Health Monitoring System

4.3 Final Illustration of the Project

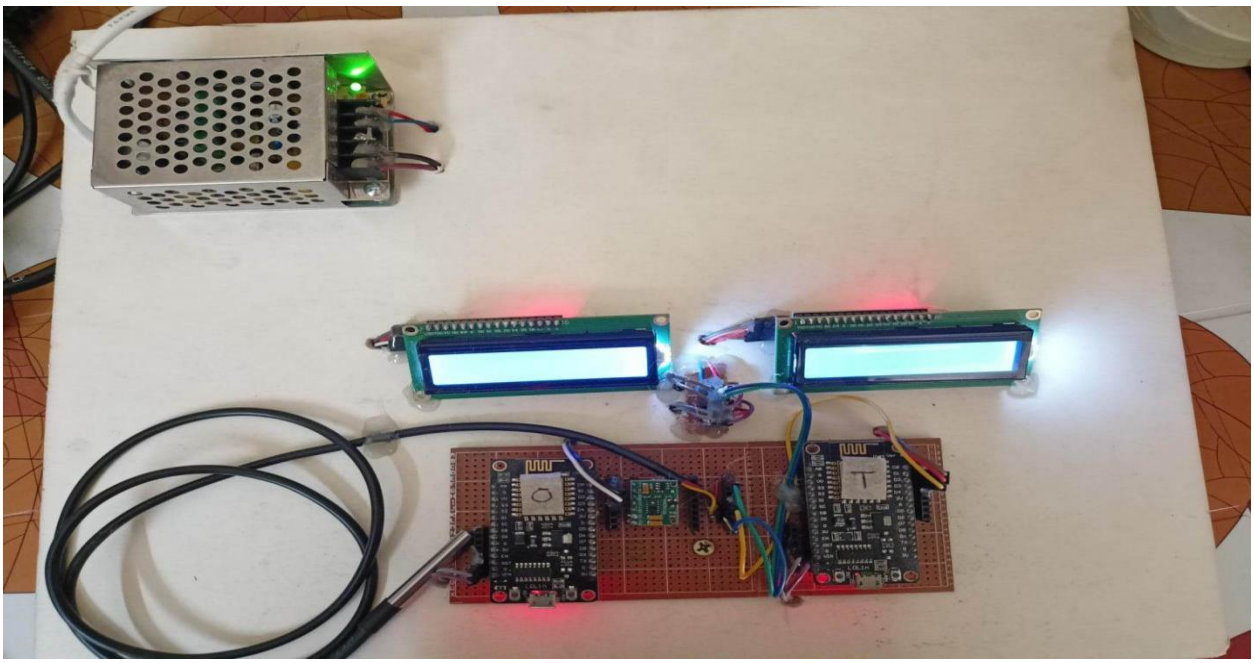

Figure 4.3: Final Illustration of my project

Chapter 5
Software and Simulation

Overview:

- Arduino IDE
- Blynk App

In this chapter, I have discussed about the software and applications and also their simulation I have used in this project.

5.1 Arduino IDE

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. It connects to the Arduino hardware to upload programs and communicate with them. The Arduino IDE is an open-source software, which is used to write and upload code to the Arduino boards. The IDE application is suitable for different operating systems such as Windows, Mac OS X, and Linux. It supports the programming languages C and C++. Here, IDE stands for Integrated Development Environment.

The program or code written in the Arduino IDE is often called as sketching. We need to connect the Genuino and Arduino board with the IDE to upload the sketch written in the Arduino IDE software. The sketch is saved with the extension '.ino.'

The Arduino IDE will appear as:

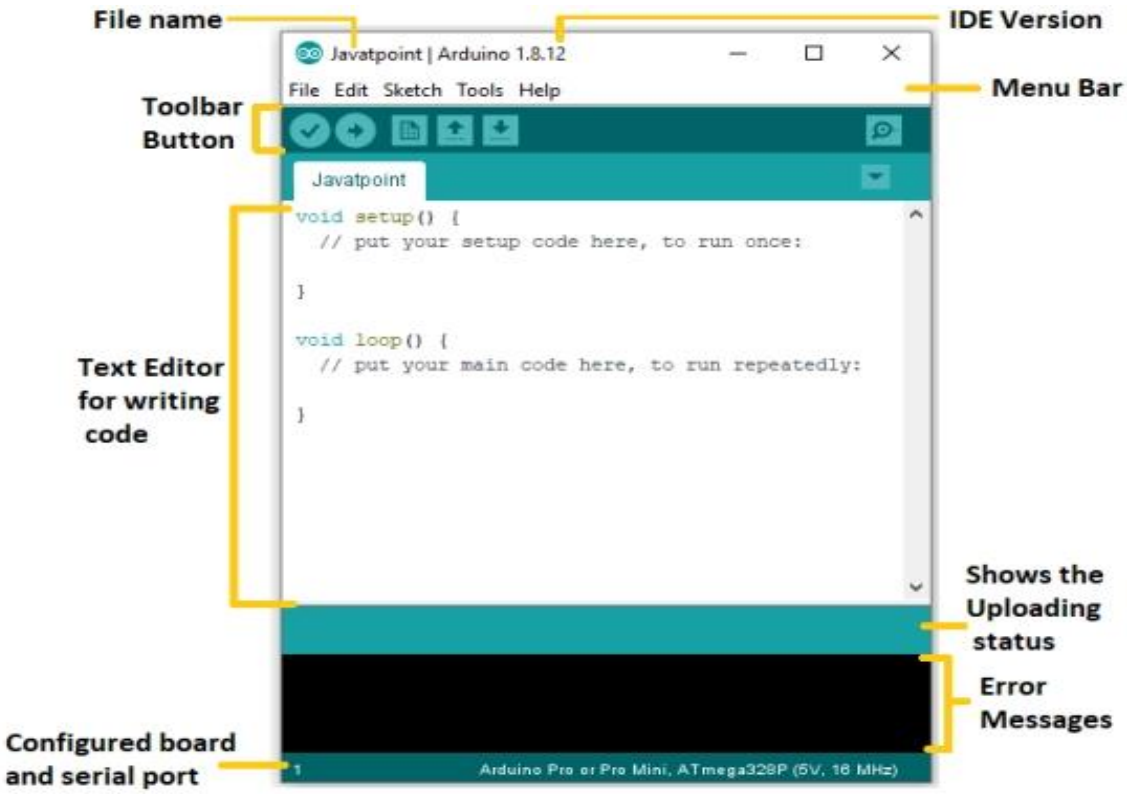

Figure 5.1 Arduino IDE

The editor contains the four main areas:

1. A Toolbar with buttons for common functions and a series of menus. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.

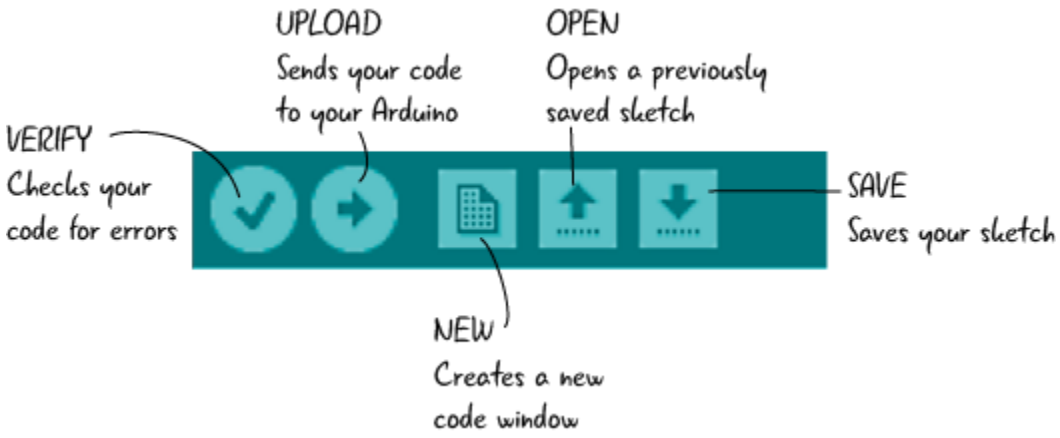

Figure 5.2 Arduino IDE Toolbar

2. The message area, gives feedback while saving and exporting and also displays errors.

3. The text editor for writing your code.

4. The text console displays text output by the Arduino Software (IDE), including complete error messages and other information.

The bottom right-hand corner of the window displays the configured board and serial port.

5. Connect your Arduino or Genuino board to your computer.

6. Now, you need to select the right core & board. This is done by navigating to **Tools > Board > Arduino AVR Boards > Board**. Make sure you select the board that you are using. If you cannot find your board, you can add it from **Tools > Board > Boards Manager**.

7. Now, let's make sure that your board is found by the computer, by selecting the port. This is simply done by navigating to **Tools > Port**, where you select your board from the list.

5.2 Blynk App

With Blynk, you can create smartphone applications that allow you to easily interact with microcontrollers or even full computers such as the Raspberry Pi. The main focus of the Blynk platform is to make it super-easy to develop the mobile phone application. As you will see in this course, developing a mobile app that can talk to your Arduino is as easy as dragging a widget and configuring a pin. With Blynk, you can control an LED or a motor from your mobile phone with literally zero programming. This is actually the first experiment that I will demonstrate in this course. But don't let this simplicity make you think that Blynk is only useful for trivial applications. Blynk is a robust and scalable tool that is used by hobbyists and the industry alike.

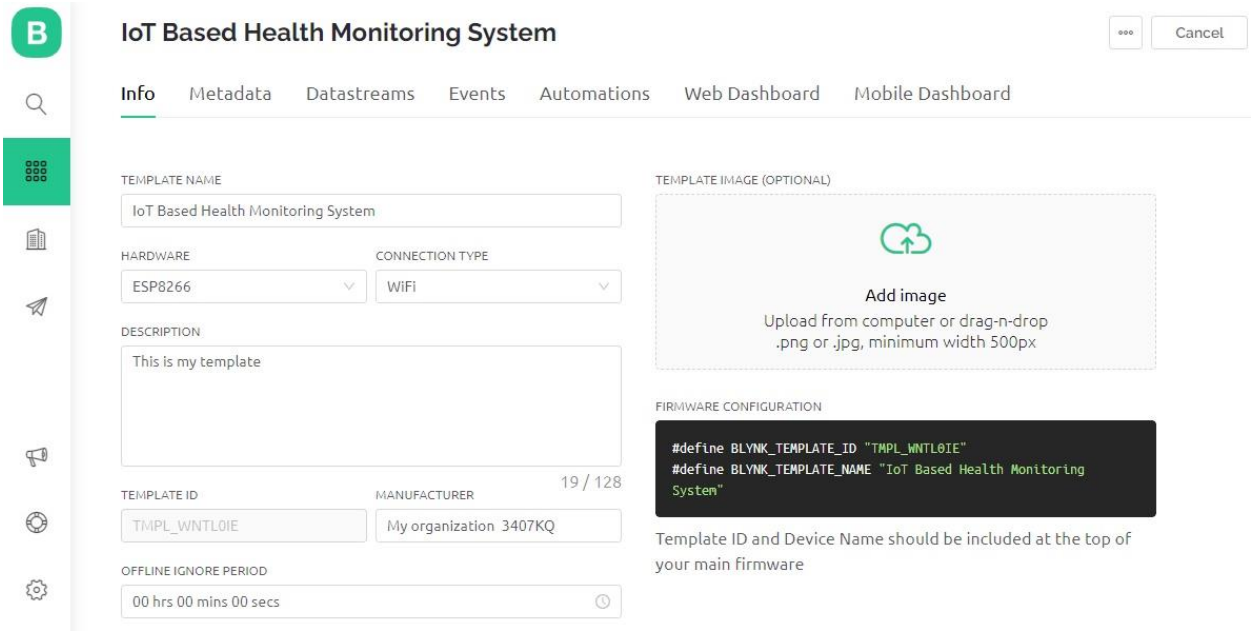

Figure 5.3: Blynk app Info

You can use it to monitor the soil humidity of your vegetable garden and turn on the water, or open up your garage door, with your phone. You can also use it to control smart furniture that can learn from your routines, or embed IoT and AI to traditional industrial products such as a boiler, or forimproving the integrity and safety of oilfields. Blynk is free to use for personal use and prototyping. Their business model generates profits by selling subscriptions to businesses that want to publish Blynk-powered apps for their hardware products or services.

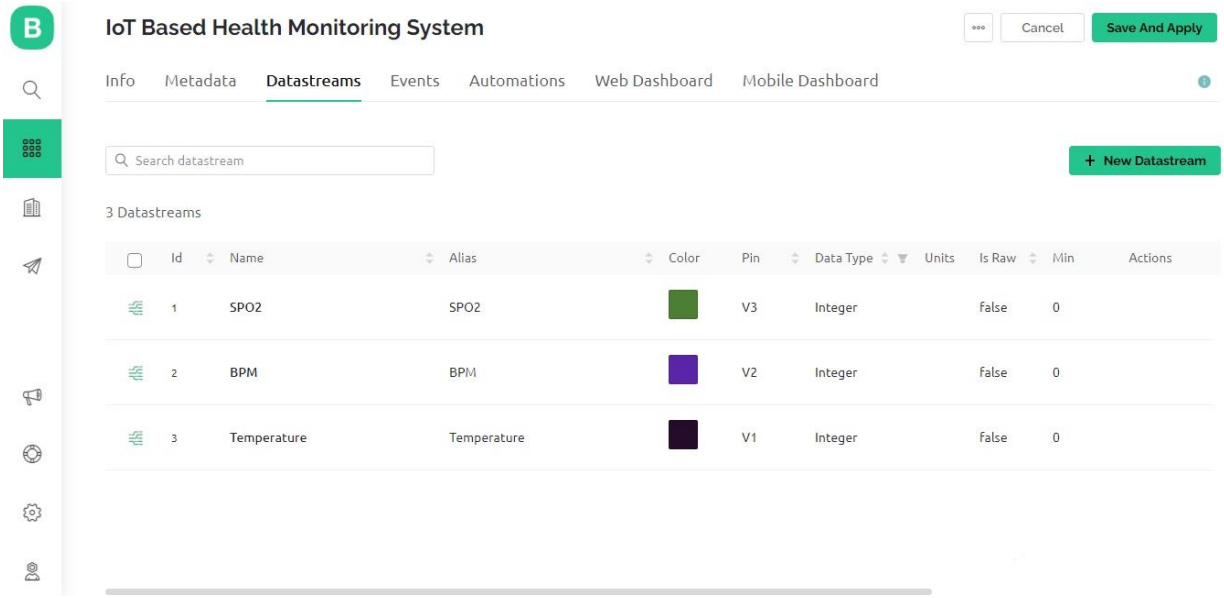

Figure 5.4: DataStream's

The Blynk innovation is the smartphone app. The Blynk app is is really an app editor.

It allows you to create one or more projects. Each project can contain graphical widgets, like virtual LEDs, buttons, value displays and even a text terminal, and can interact with one or more devices. With the help of the Blynk library, it is possible to control Arduino or ESP32 pins directly from your phone, without having to write any code at all. It is also possible to share a project with friends and even customers so that they can access the connected devices but not be able to modify the project. Imagine a scenario where you build a smartphone application where you can control lights, window blinds and room temperature from your phone. You can share the project with other family members so that they can also access the functionality.

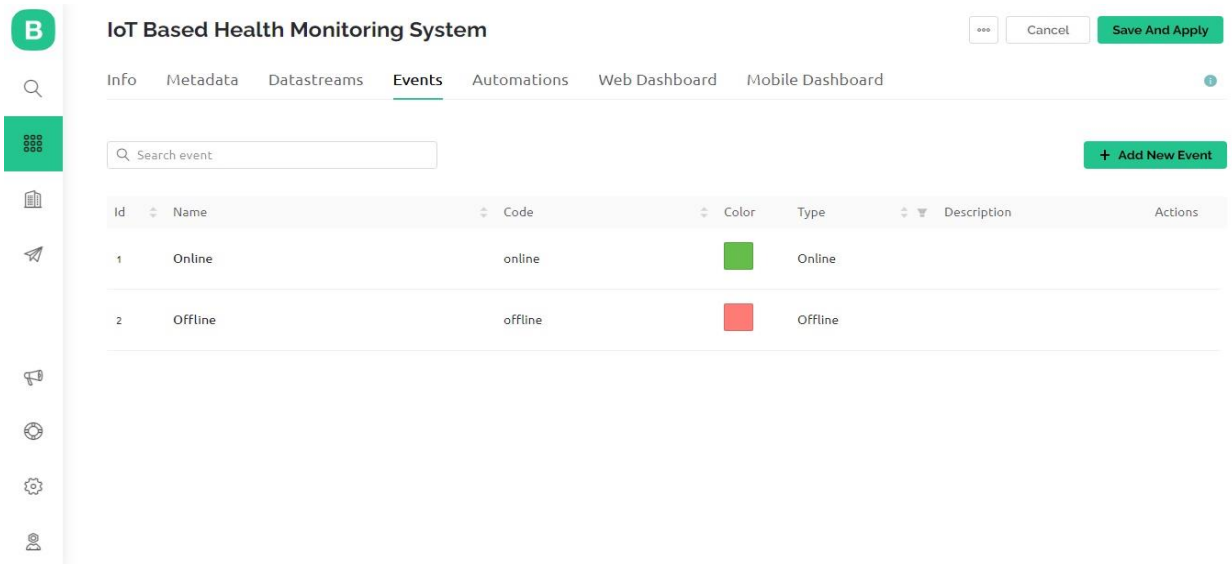

Figure 5.5: Events

I am amazed by the range of devices and connectivity types that the Blynk platform supports. The support is implemented by means of a Blynk library that targets a device and connectivity type combination. For example, if you want to use your Arduino Uno with an Ethernet shield, you would use the library "BlynkSimpleEthernet" which contains the Blynk firmware plus the required connectivity support.

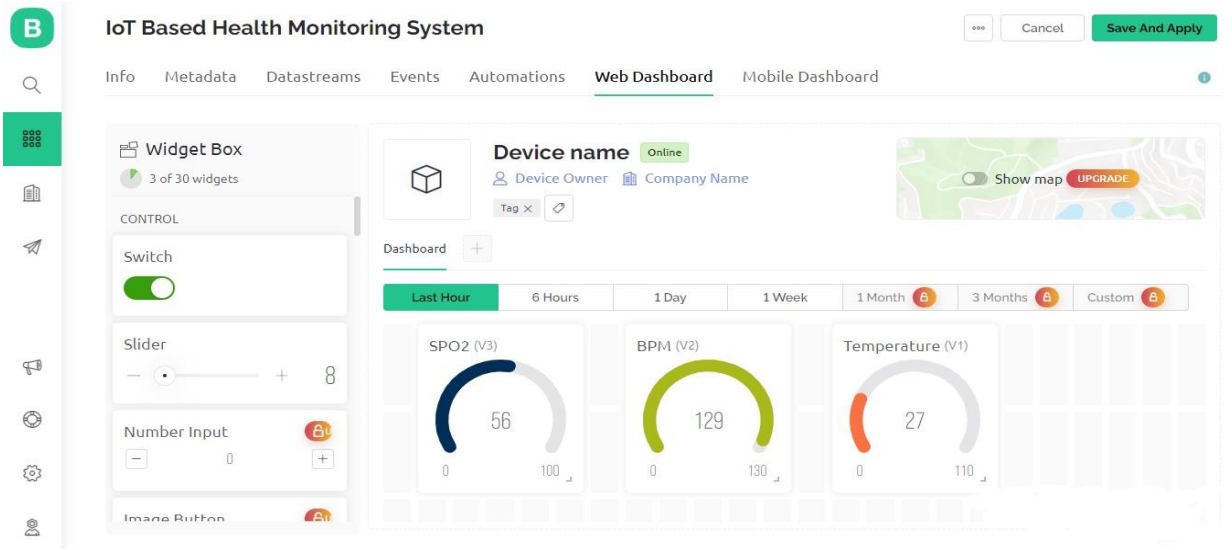

Figure 5.6: Web Dashboard

Both sketches would use the same Blynk infrastructure, such as the physical pins that are part of the Arduino's hardware, as well as the virtual pins which are implemented in software by the Blynk Platform. This means that with a bit of planning, you can write sketches that can be easily shared among different target devices. You can potentially write a sketch for an Arduino Uno and and with minimal modification run it on an Arduino MKR1010. Blynk also supports clients that are not a microcontroller. You can write client code is JavaScript, Python or Lua thanks to the available Blynk libraries for these languages. This means that you can have a Blynk project that interacts with an Arduino and with Python code running on a Linux virtual machine somewhere on the Cloud. The possibilities are really very exciting.

Unlike IoT platform such as IFTTT, Twillio, and even Adafruit IO, you can host a private instance of the full Blynk server and connect your smartphone Blynk app to it.

The Blynk Cloud server is an excellent choice for most projects, as it is always there, ready to use. We will use the Cloud server in the first few experiments in this course to help you get started with minimal effort. However, as you will see, the Cloud Blynk server has imposed limitations. Some limitations are due to the topology of the server: depending on your geographical location, the server may be in a different continent, which makes communications between the app, the devices and the server slow due to the amount of time it takes packets to travel across the Internet.

Another imposed limitation is that in the Cloud server, you can only use a small number of widgets. Blynk is using the concept of "energy" to implement a pricing system for its widgets. In the Cloud server you may start a new project with 1000 energy units. An LED widget may cost you 200 units, leaving 800 units for other widgets. On a private server, you can set your own energy limits. You can configure your server to allocate 100,000 energy units to new users. It's totally up to you.

However, a private Blynk server gives you additional benefits:

- Essentially unlimited energy units, so that you can build any Blynk application you can think of.

- Minimal latency, which is useful when your application is used in a limited geographic area and responsiveness is important.
- Total control of your data. You can keep your own backups of your Private server, migrate your server to a new host, implement whichever security mechanisms you wish, and finely control your users.

In this course, I will show you how to install and configure a private Blynk server so you can see the related benefits and costs. You can run an instance of the Blynk server on any host that has a Java runtime environment. In this course, I use a $15 Raspberry Pi Zero W, running the Raspberry Pi operating system. This is a computer with just half a gigabyte of RAM, and still runs the Blynk server without any issues at all. You can use any Raspberry Pi with WiFi, it does not need to be a Zero W. In fact, you can use any Linux or Windows host you want, including virtual machines.

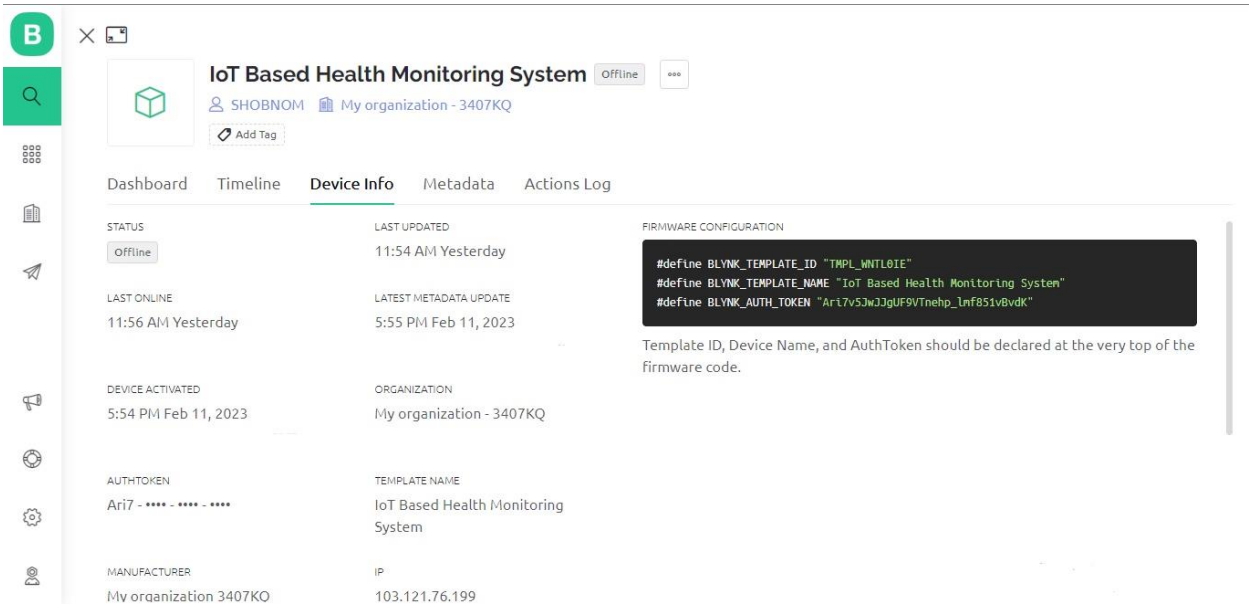

Figure 5.7: Device Info

Chapter 6
Result and Discussions

Overview

- Result of the Project
- Application of the Project
- Feature of the System
- Benefit of the System

In this chapter, I have tried to discuss about the result of the project. And also have discussed about the features of the system and benefits of the system.

6.1 Result of the Project

I have physically worked on this project and I gained a success in implementing the task. In this report, I projected the role of smart health monitoring system as an efficient way to cut consumption of energy and other resources. To simplify the working of the project we had split the task into multiple phases. Each phase has its own importance, which we had used a WiFi module. In this phase the WiFi hotspot of the phone can also be used to control the entire electronic appliance which is connected through the NodeMCU just with a touch we can control the appliances.

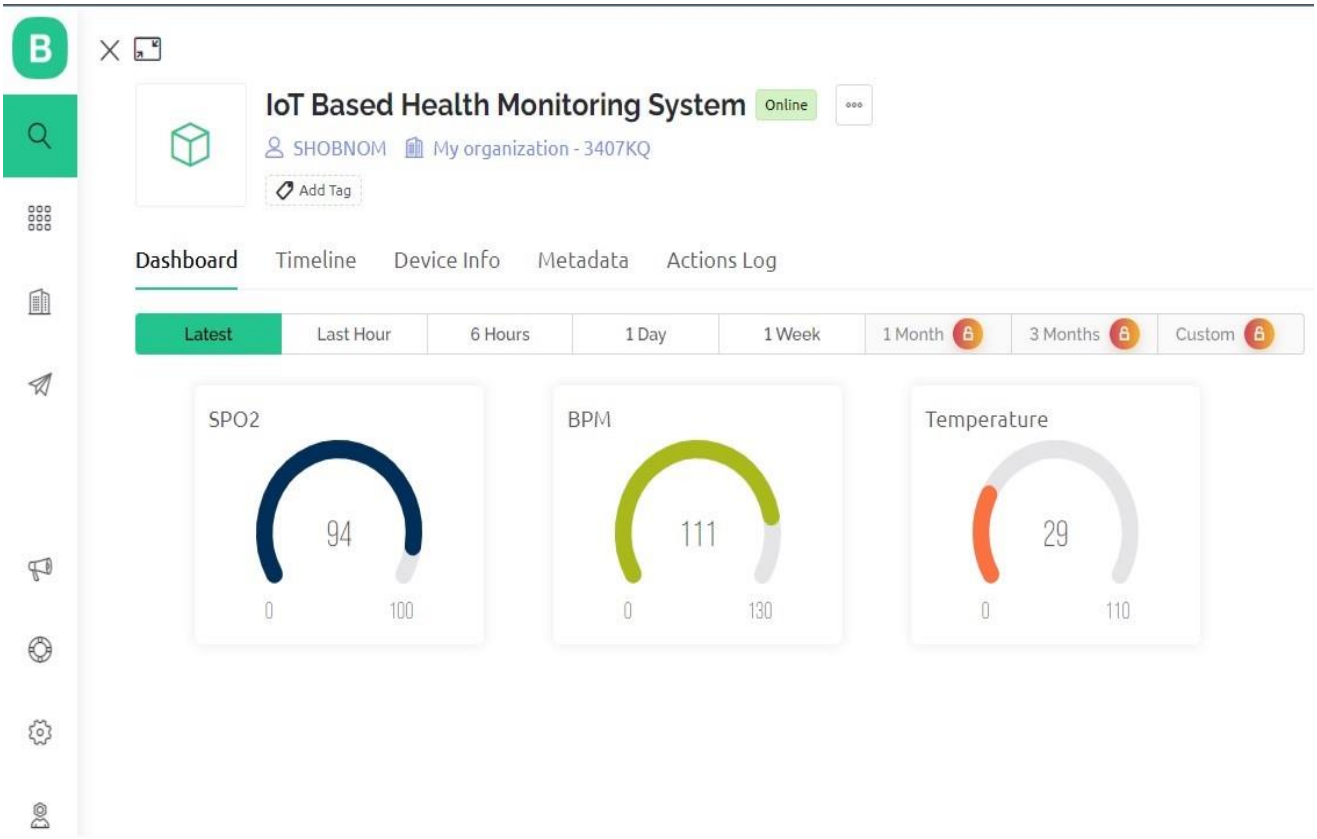

Figure 6.1: Output from Blynk IoT App

6.2 Application of the Project

Enhanced Comfort: Smart Health Monitoring System let you easily adjust the comfortable situation. We don't need to recheck the patient's situation.

Energy Saving: This system is energy saving. We can use battery also instead of adapter or USB cable. Which is safer to licking electricity.

Increased Convenience: This system gives you entire control of the health monitoring. Doctors don't have to be physically present in front of the patients. They can monitor them from their cabin or home.

6.3 Features of the System

- Measure Blood Oxygen level
- Measure Heart rate
- Measure Body Temperature
- Inform doctors & nurses from remote distance.

6.4 Benefits of the System

- Patients' health condition can be monitored.
- Decreased the rate of sudden death.
- Early detection of diseases.
- Time saving.
- Cost effective.

Chapter-7
Conclusion

Overview

- Conclusion
- Limitations
- Future Work

In this chapter, we have tried to draw a conclusion of my project. For this we have focused on Limitations and future work of this project.

7.1 CONCLUSION:

The Internet of Things is considered now as one of the feasible solutions for any remote value tracking especially in the field of health monitoring. It facilitates that the individual prosperity parameter data is secured inside the cloud, stays in the hospital are reduced for conventional routine examinations and most important that the health can be monitored and disease diagnosed by any doctor at any distance.[10] The early identification of any health problem can help the patient to take necessary emergency measures, whichcan potentially save the patient's life. IoT can help in this regard. IoT based health monitoring systems can monitor the patients in real-time and warn the patient of any abnormalities. However, the IoT architecture must have the facilities to ensure the proper security of sensitive data. Also, the used sensors must be small in size so that they can be easily incorporated into various systems. In this paper, an IoT based health monitoring system was developed. The system monitored body temperature, pulse rate and room humidity and temperature using sensors, which are also displayed on a LCD. These sensor values are then sent to a medical server using wireless communication. These data are then received in an authorized personals smart phone with IoT platform. With the values received the doctor then diagnose the disease and the state of health of the patient.

The system is very useful in the case of infectious disease like a novel coronavirus (COVID-19) treatment. The developed system will improve the current healthcare system that may protect lots of lives from death.[11]

7.2 LIMITATIONS

- ESP8266 or NodeMCU which are IOT enabled boards that connect our projects to internet via Wi-Fi. That is why if Internet connection dropped, we couldn't get the result.
- The data cannot be stored in any database.

7.3 FUTURE WORKS

- Integrating more sensors for more specific data acquisition and analysis.
- Will be applicable in army services in active situation.
- Will be used to provide health service to rural areas in affordable price.
- Huge database will be built for doctors to diagnose people from different areas and cultures.

Multiple parameters like Blood pressure, room temperature & humidity, retinal size, age and weight can be included as controlling parameters in the future.

This system also developed by using advanced GSM and GPRS technology in future.

To combat such situations, medical IoT devices and applications can gather vital data and transfer it to doctors and health personnel for real-time tracking. Also, these mobile applications and IoT devices can also send notifications regarding a patient's critical conditions irrespective of place, and time. Additionally, this knowledge helps EMTs choose the most appropriate institution for a patient. Similar technology in the form of an IoT-enabled badge that tracks blood pressure, pulse, body temperature, and breathing rate may be used in emergency medical treatment in the future of IoT in healthcare.

Chapter-8
Appendix

Overview

- Implementation of program Code
- References

In this Section, we have included some of References from where we have gathered all of this data and information of this project work and also have included program code which used in NodeMCU board.

8.1 Implementation of Program Code

Pulse-Oximeter Sensor

```
#include <Wire.h>

#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);


#define BLYNK_TEMPLATE_ID "TMPLdUvkAe0H"

#define BLYNK_DEVICE_NAME "IoT Based Pulse Oximeter"

#define BLYNK_AUTH_TOKEN "gcRNdH0tFj_SMHKDkzTyZPOsmIAIwMo3"

#define BLYNK_PRINT Serial


#include <Wire.h>

#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>

#include "MAX30100_PulseOximeter.h"

#include <SimpleTimer.h>

#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>

// BPM GAUGE CONNECT V2

// SP02 GAUGE CONNECT V3
```

```
char auth[] = BLYNK_AUTH_TOKEN;

// Your WiFi credentials.

// Set password to "" for open networks.

char ssid[] = "Engineers";

char pass[] = "25802580";


#define REPORTING_PERIOD_MS   3000

PulseOximeter pox;

uint32_t tsLastReport = 0;

void onBeatDetected()

{

;

}

BlynkTimer timer;


void setup()

{

  lcd.begin(16,2);

  lcd.init();   // initializing the LCD

  lcd.backlight(); // Enable or Turn On the backlight

  lcd.clear();
```

```
Serial.begin(115200);

Blynk.begin(auth, ssid, pass);

Serial.print("Initializing pulse oximeter..");

if (!pox.begin()) {

Serial.println("FAILED");

for (;;);

} else {

Serial.println("SUCCESS");

digitalWrite(1, HIGH);

}

pox.setIRLedCurrent(MAX30100_LED_CURR_24MA);

// Register a callback for the beat detection

pox.setOnBeatDetectedCallback(onBeatDetected);

timer.setInterval(3000L, getSendData);

}


void loop()

{

timer.run(); // Initiates SimpleTimer

Blynk.run();
```

```
lcd.setCursor(0,0);
lcd.print("BPM:");
lcd.print(pox.getHeartRate());
lcd.setCursor(0,1);
lcd.print("SpO2:");
lcd.print(pox.getSpO2());
// Make sure to call update as fast as possible
pox.update();
if (millis() - tsLastReport > REPORTING_PERIOD_MS) {
// to android cell phone application
Serial.print("BPM:");
Serial.print(pox.getHeartRate());
//blue.println("\n");
Serial.print("  SpO2:");
Serial.print(pox.getSpO2());
Serial.print("%");
Serial.println("\n");
Blynk.virtualWrite(V2, pox.getHeartRate() );
Blynk.virtualWrite(V3, pox.getSpO2());
tsLastReport = millis();
}
}
```

```
void getSendData()

{

}
```

Temperature Sensor

```
#include <OneWire.h>

#include <DallasTemperature.h>

#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27, 16, 2);


#define BLYNK_TEMPLATE_ID "TMPL_WNTL0IE"

#define BLYNK_TEMPLATE_NAME "IoT Based Health Monitoring System"

#define BLYNK_AUTH_TOKEN "Ari7v5JwJJgUF9VTnehp_lmf851vBvdK"

#define BLYNK_PRINT Serial


#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>


char auth[] = BLYNK_AUTH_TOKEN;


// Your WiFi credentials.

// Set password to "" for open networks.

char ssid[] = "Engineers";

char pass[] = "25802580";

int depth =20;
```

```
BlynkTimer timer;

#define ONE_WIRE_BUS D3
OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature DS18B20(&oneWire);
float temp;

void setup()
{
  lcd.begin(16,2);
  lcd.init();   // initializing the LCD
  lcd.backlight(); // Enable or Turn On the backlight
  lcd.clear();
  Serial.begin(115200);
  Blynk.begin(auth, ssid, pass);
  DS18B20.begin();
  timer.setInterval(1000L, getSendData);
}

void loop()
{
  timer.run(); // Initiates SimpleTimer
  Blynk.run();
}

/**************************************************
 * Send Sensor data to Blynk
```

```
 **********************************************/
void getSendData()
{
  DS18B20.requestTemperatures();
  temp = DS18B20.getTempCByIndex(0); // Celcius
  Serial.println(temp);
  Blynk.virtualWrite(V1, temp); //virtual pin V3

    ////////////////// lcd//////////
    lcd.clear();
    lcd.setCursor(1,0);
    lcd.print("Temp = ");
    lcd.print(temp);
    lcd.print("C");
    delay(1000);
    lcd.clear();
}
```

8.2 REFERENCES:

[1] DSR Krishnan, SC Gupt, Tanupriya Choudhury - International Conference on Advances in Computing and Communication Engineering (ICACCE), 23 August 2018.

[2] Steve Ranger, “Internet of Things: The Basics Explained”, February 3, 2020.

[3] Gulraiz J. Joyia, Rao M. Liaqat, Aftab Farooq, and Saad Rehman, “Internet of Medical Things (IOMT): Applications, Benefits and Future Challenges in Healthcare Domain”, Journal of Communications Vol. 12, No. 4, April 2017.

[4] S Patil, S Pardeshi – “Health Monitoring System” Int. Res. J. Eng. Technol. (IRJET) Journal, Volume: 05 Issue: 04 | Apr-2018.

[5] Komal Suresh.1, Shaikh Hina Ilyas.2, Prof. Tekale N. S., “Patient Health Monitoring System Using GSM Module”, International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering, Vol. 8, Issue 5, May 2019.

[6] Hamid Al-Hamadi and Ing-Ray Chen, “Trust-Based Decision Making for health IoT Systems” IEEE Internet of Things Journal, 10.1109/JIOT.2017.2736446.

[7] Ayush Bansal, Sunil Kumar, Anurag Bajpai, Vijay N. Tiwari, Mithun Nayak, Shankar Venkatesan, Rangavittal Narayanan, “Remote health monitoring system for detecting cardiac disorders”, ET Syst.Biol., 2015, Vol. 9, Iss. 6, pp. 309–314.

[8] P. Chavan, P. More, N. Thorat, S. Yewale, and P. Dhade, “ECG- Remote patient monitoring using cloud computing,” Imperial Journal of Interdisciplinary Research, vol. 2, no. 2, 2016.

[9] Ruhani Ab. Rahman, NurShima Abdul Aziz, MurizahKassim, Mat IkramYusof, “IoT-based Personal Health Care Monitoring Device for Diabetic Patients”, IEEE Journal, 978-1-5090-4752-9/17/2017

[10] Rajiv .K. Bhatia and Varsha Bodade,"Smart Grid Security and Privacy: Challenges, Literature Survey and Issues", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, Issue 1, January 2014.

[11] Md. Milon Islam, Ashikur Rahaman, Md. Rashedul Islam, “Healthcare Monitoring System with IoT”, SN Computer Science, 14 May 2020.