DESIGN AND IMPLEMENTATION OF AN IOT-BASED HEALTHCARE MONITORING SYSTEM WITH AUTOMATIC HAND SANITIZER

by

MOHAMMED ZAKI

MOHAMMED MOINUL ISLAM

BACHELOR OF SCIENCE IN ELECTRICAL AND ELECTRONIC ENGINEERING

Department of Electrical and Electronic Engineering
INTERNATIONAL ISLAMIC UNIVERSITY CHITTAGONG

DECEMBER 2021

DESIGN AND IMPLEMENTATION OF AN IOT-BASED HEALTHCARE MONITORING SYSTEM WITH AUTOMATIC HAND SANITIZER

By

MOHAMMED ZAKI

MOHAMMED MOINUL ISLAM

A project
submitted as partial fulfillment of the requirement for the degree of

BACHELOR OF SCIENCE IN ELECTRICAL AND ELECTRONIC ENGINEERING

Department of Electrical and Electronic Engineering
INTERNATIONAL ISLAMIC UNIVERSITY CHITTAGONG

DECEMBER 2021

CERTIFICATE OF APPROVAL

The project entitled as "**Design and Implementation of an IoT-based Healthcare Monitoring System with Automatic Hand Sanitizer**" submitted by **Mohammed Zaki**, bearing Matric ID. **ET151056R** and **Mohammed Moinul Islam**, bearing Matric ID. **ET151065** of session **Spring 2015**, to the Department of Electrical and Electronic Engineering, International Islamic University Chittagong, has been accepted as satisfactory in partial fulfillment of the requirements for the degree of Bachelor of Science in Engineering and approved for the examination held on **24 December, 2021**.

Supervisor
Engr. Md. Sk. Golam Mostafa
Associate Professor,
Department of Electrical and Electronic Engineering,
International Islamic University Chittagong.

DECLARATION

It is hereby declared that this work has been done by us and no portion of the work contained in this project has been submitted elsewhere for the award of any degree or diploma, except where due acknowledgment is made in the thesis.

____

Mohammed Zaki

____

Mohammed Moinul Islam

ACKNOWLEDGMENT

All praises are due to Allah, the most Beneficent and the most Merciful, for helping us to accomplish this work.

Our heartily thanks, gratitude and appreciation to our supervisor Engr. Sk. Md. Golam Mostafa for his encouragement, help and supervision. All his technical and editorial advice, suggestions and guidance were a real support to carry out this project.

Moreover, we would like to express our gratitude for all the honourable teachers of our department and all the office and lab staffs for their invaluable efforts throughout our university life.

Last but not the least, special thanks to our family, especially our parents, for their understanding and encouragement during the different phases of our study life.

Mohammad Zaki

Mohammed Moinul Islam

ABSTRACT

According to ways-to-die website, over 150,000 people die every day. And the most common cause of death, i.e., about 20% of all deaths, is heart diseases. So, the most crucial contribution from our side to lower this percentage can be to monitor the cardiac values as much as possible. There are conventional methods to measure patients' health and condition, but they are laborious; have possibilities of errors; and nocturnal monitoring has as well been very difficult. Moreover, since 2019, COVID19 has caused more than five million deaths all around the world, as stated by WHO. And it made the physical presence of doctors and caretakers almost impossible. So, we have designed an up-to-date IoT-based project that continuously monitors the patient's body temperature, heart-rate and oxygen saturation level; keep the data readings in display before the patient and on the screen of the doctor's mobile; and it also provides a non-touching hand-sanitizing system. The proposed design integrates NodeMCU, DS18B20 Temperature sensor, Max30100pulse-oximeter, and other required materials in a small box. The readings are as accurate as the conventional medical equipments while it just takes less than a minute of time to perform the whole procedure. The developed project has outperformed the conventional method by providing a safer, less complex, cost effective and faster service.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

LCD	Liquid Crystal Display
SPO2	Peripheral capillary oxygen saturation
BPM	Blood Pressure Measurement
COVID19	Corona Virus Disease 2019
URL	Uniform Resource Locator
COPD	Chronic Obstructive Pulmonary Disease
IoT	Internet of Things
WBASN	Wireless Body Area Sensor Network
GSM	Global System for Mobile
IR	Infrared
LED	Light Emitting Diode
RFID	Radio Frequency Identification
UMTS	Universal Mobile Telecommunications System
MCU	Microcontroller Unit
RAM	Random Access Memory
MRAM	Magneto-resistive Random Access Memory
AVR	Automatic Voltage Regulation
PIC	Peripheral Interface Controller
ARM	Atmospheric Radiation Monitoring
SAM	Software Asset Management
IDE	Integrated Development Environment
VDD	Voltage Drain-to-Drain
GND	Voltage Drain-to-Drain
COM	Common
NO	Normally Open
NC	Normally Closed
I/O	Input / Output
DC	Direct Current
AC	Alternating Current
SMPS	Switched-Mode Power Supply
USB	Universal Serial Bus
DRAM	Dynamic Random Access Memory
CPU	Central Processing Unit

PCB	Power Control Board
ROM	Read Only Memory
R	Resistance (Ohm)
V	Voltage (Volt)
mAh	Mili-ampere
SCL	Serial Clock
SDA	Serial Data
EEG	Electro Encephalogram
ECG	Electro Cardiogram
ICU	Intensive Care Unit
WiFi	Wireless Fidelity
App	Application

CHAPTER 01
INTRODUCTION

1.1 Introduction

People are advancing in technology, but we can't ignore health concerns. Health concerns affect a country or even the whole world economically and socially. According to Ways-to-die, over 150,000 people die every day. The most common cause of death, i.e., about 20% of everyday deaths, is mainly heart diseases [1]. Moreover, CoVid19 recently has affected the whole world, especially medical system and facilities of the countries all over the world. This type of situation demands remote monitoring technologies for everything. IoT can be considered the best solution for such technologies [2]. This can help the doctors to observe patients from a far-away room, and the patient can also consult from his own home which will save both his time and money [3]. Different kinds of sensors are used to track people's health; and using internet helps transmit their conditions to their concerned ones. Other aims behind remote healthcare are cost reduction and size deduction by IoT-based monitoring platforms. The data can be seen by several people including family members and caretakers by means of web-server [4]. All the above mentioned facts are performed through a URL or a site passed. There are GSM-based monitoring systems as well. Here, SMS strategy is used to transmit the data. Some of the countryside regions also don't have immediate physical doctoring facilities and it takes a long time for the patients to reach hospitals and undergo medical procedures [5].

So, we have designed an IoT based system which uses NodeMCU as a microcontroller unit, and it receives readings of temperature, heartbeat and SpO2 values by using DS18B20 temperature sensor and Max30100 pulse-oximeter. The readings are displayed in an LCD beside the patient and the physician's mobile screen using Blynk App. It also has an IR sensor which detects objects placed in front and it turns the relay on to pump out the sanitizer without even being touched.

1.2 Objectives

1. To ensure continuous monitoring of body temperature, heart-rate, and SpO2 values by using Max30100 and DS18B20 sensors.
2. To add an automatic non-touching hand sanitizing system.

The implemented project work will:

i) Display real time data with LCD.

ii) Display the data on Android app via Wi-Fi communication system.

iii) Replace the direct presence of a physician.

1.3 Motivation

Healthcare is a basic human right. Over 20 percent of everyday deaths are caused by heart-related diseases. Heart problems and oxygen level in blood aren't only related to Corona Virus, they are factors for a lot of other diseases such as COPD (Chronic Obstructive Pulmonary Disease), asthma and many other lung infection diseases. Minor symptoms of heart diseases and low oxygen level in blood are generally ignored by many people as they don't recognize the serious effects primarily. And eventually they develop diseases like hypoxia. In most cases, even if people consider the symptoms, it takes them a long time undergoing clinical tests and waiting for advanced symptoms like shortness of breath or pneumonia. Moreover, CoVid19 made our lives so difficult that even doctors attending patients physically has become almost impossible. These all facts lead the patient to critical conditions and eventually death in some cases [6].

To overcome such situations, regular monitoring of the patient's heart-rate and blood saturation level will help medical staffs to take actions as soon as possible. Doctors, patients and the patients' families won't have to face sudden stress of being transmitted by any disease. And IoT-based platform helps the doctor know patients' conditions through web-server and prescribe them the necessary steps.

1.4 Outline of the Report

This whole report book is organized in five chapters.

The first chapter is started with the background introduction and the primary objectives of our project along with its motivation.

The second chapter discusses some literature reviews on healthcare monitoring system and the details of all our project components.

The third chapter is on our experimental setup including circuit diagram, block diagram and flowchart.

The measured results of our project and its related discussions are written in the fourth chapter.

And the final chapter concludes the projects and describes some possible future works.

CHAPTER 02
LITERATURE REVIEW

2.1 Literature Review of Healthcare System

There are a lot of research works done on the usage of wireless sensors in the past. We are going to review a few of those below:

The first research paper proposed a system on healthcare monitoring. It is based on IoT with Raspberry Pi. The components used are: LM35 temperature sensor, blood pressure sensor, heartbeat sensor, ECG sensor and GSM module. The paper has made the system into software and hardware sections. The software section worked on interfacing. While the hardware section focused on transmitting and receiving. The transmitter section is connected to human body. The master device, Raspberry Pi, connects the different sensors used with IoT server. The data from the sensors are taken, processed and stored in the memory of the master device. The data is then sent to the server. This helps all the readings from temperature sensor and heartbeat sensor appear on the LCD at the receiver end. So, any abnormal situation of the patient is known by the doctor or the concerned person directly through GSM. The system overall is simple, power efficient and easily understandable [7]. But, major drawback of this research is that the processor used in the system is too costly.

In the second one, it also is a similar approach. The research is an IoT-based intelligent home-centric healthcare platform, i.e., iHome system. The system mainly worked on three sections: (1) Smart medical service, (2) Medication management, and (3) Cloud integration. Smart sensors are connected to Rasberry Pi. The sensors take readings from the patient’s body. The daily medications are managed by monitoring the data conditions of the patient. This approach provides opportunities that help various e-health services with a good efficiency [8]. But this system too is very costly to install and doesn’t work on patient sanitizing section.

The third system uses a NodeMCU, two Arduino Uno boards, an acceleration sensor, a heartbeat sensor, an IR sensor, a temperature sensor (LM35) and an LCD display. The programming is done in Embedded C and Arduino Software. The parameters taken from sensors are monitored, recorded and stored as a database. The parameters include body temperature, acceleration, heart rate and saline level. This system also performs for patients in remote areas when he is out of reach. In case of abnormal condition, the system is expected to inform the doctor through cloud computing [9]. Major drawback of this approach is that it used a total of 3 MCU boards – a NodeMCU and two Arduino Unos, which ultimately makes the project more complex and costlier.

All these reviewed systems used processors that are costly to install. And these systems don't provide sanitizing facility as well, which is one of today's major demands.

2.2 Node MCU

Node MCU is an IoT based open source platform. Its firmware works on ESP8266 Wi-Fi SoC from Espressif Systems. And the hardware is based on the ESP-12E module. Some open source projects are used there such as lua-cjson and spiffs. A picture of NodeMCU is shown in **Fig. 2.1**.

When Arduino.cc developed new MCU boards on non-AVR processors such as ARM/SAM MCU, they were used in Arduino. The Arduino IDE is easier to use as it supports tool chains. It also compiles Arduino C and C++ to new processors. These processes were performed with the development of Board Manager and SAM Core.

Fig.2.1 Node MCU [10].

The core collects software components for the Board Manager and the Arduino IDE. It helps to run Arduino C and C++ source files down to MCU's machine language. The ESP8266 Core for the Arduino IDE is now among the leading software developing platforms that produces many kinds of ESP8266 based modules and development boards, for example, Node MCU [10].

2.3 Max30100 Module

MAX30100, shown in **Fig. 2.2**, is a monitoring sensor which measures both heart rate and oxygen saturation level. It consists of two LEDs, a photodetector, optimized optics, and low-noise analog signal processing. The MAX30100 works on 1.8V to 3.3V power supplies as its operating voltage. A negligible standby current powers the oximeter. So the power supply remains connected at all times [11].

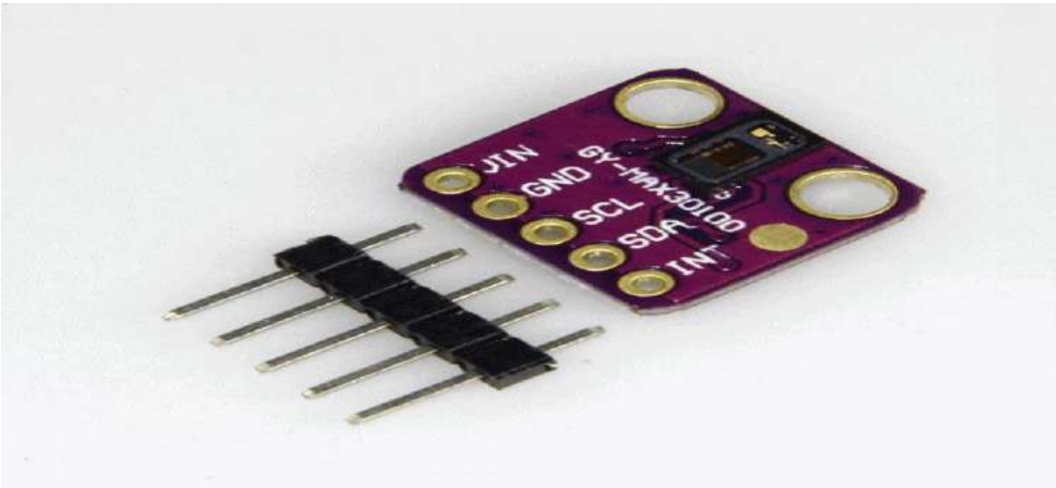


Fig.2.2 Max30100 module [11].

2.3.1 Applications:

i) Wearable Devices

ii) Fitness Assistant Devices

iii) Medical Monitoring Devices [11]

2.3.2 Benefits and Features:

- The pulse oximeter which works on both heart rate and SpO2 values has integrated LEDs, photo sensor, and optically enhanced system with low analog signal processing.
- Its operating low power supports battery life of the wearable devices. It saves the LED and programmable sample rate in a very low shutdown current of 0.7 µA.

- Its high SNR gives robust motion artifact resilience and Integrated ambient light cancellation.
- It has high sample rate capability.
- It has fast data output capability.

The MAX30100 IC operates at 1.8V to 3.3V and its working principle is described in **Fig. 2.3**. It uses two regulators to achieve this voltage. However, A 4.7k pull-up resistor is used with the SCL and SDA pins to the V_{CC}. As a result, it doesn't work properly with high logic level microcontrollers [12].

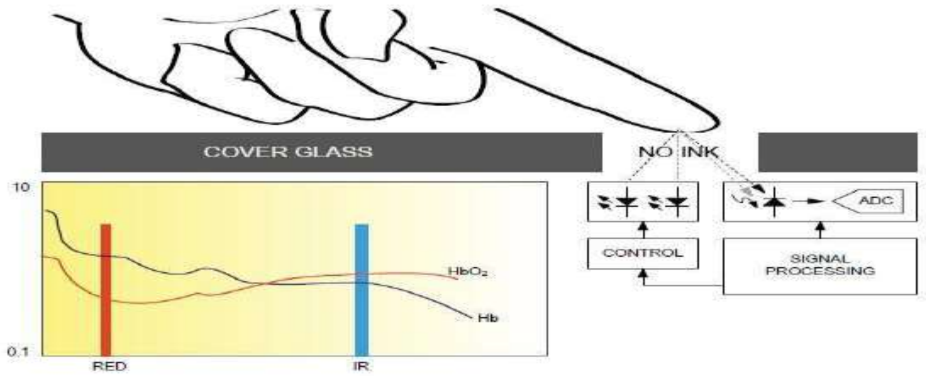


Fig.2.3 Max30100 working principle [12].

2.4 16*2 LCD

LCD stands for Liquid Crystal Display. It works in two modes: 4-bit mode and 8-bit mode. The first 4-bit nibble, D0 to D3 is sent and it is followed by next 4 bits, D4 to D7, which is the higher nibble. In this way, we can send 8 bits of data in one stroke as we are using all the 8 data lines. So, it works faster than the 4-bit mode.

Fig.2.4 16*2 LCD [13].

But there is a problem with the 8-bit line when it is connected to a microcontroller. It needs to connect all the 8 lines and as a result, all the input and output pins of the MCU runs out. So, 4-bit mode is widely used. With the help of programming, we can change its modes without controlling the pins [13]. **Fig. 2.4** picturizes a 16/2 LCD.

2.5 I^{2}C Module

I^{2}C Module as shown in **Fig. 2.5** is a great IIC/I^{2}C/TWI/SPI Serial Interface. The number of pins in microcontroller units like Arduino and NodeMCU are limited. When we connect a couple of sensors or SD card, the pins run out.
But, I2C Module is the solution to this problem. It only needs two wires to display data in the LCD and monitor [14].

Fig.2.5 I^{2}C module [14].

2.6 Buck Converter

A buck converter is a DC to DC voltage regulator and is shown in **Fig. 2.7**. It steps down the input voltage (ranging from 3.2V to 40V) to an output voltage ranging from 1.25V to 35V. However, buck converter steps up the current from supply to load.
It is a class of switched-mode power supply (SMPS). It has two semiconductor devices, an energy storage element, a capacitor or an inductor. The two semiconductor devices typically are a diode and a transistor. Sometimes, both the devices are transistors for a synchronous rectification. A capacitor is used at the output to filter voltage ripples. And sometimes, the capacitor is used with an inductor in a converter’s input (supply side filter) and output (load side filter) [15].

In **Fig. 2.6**, the circuit operates in two modes. When the switch is on, input current rises and flows through inductor and load. When the switch is off, the freewheeling diode conducts due to energy stored in the inductor. The inductor current then flows through diode, inductor and load. The inductor current releases until the switch is open for next cycle.

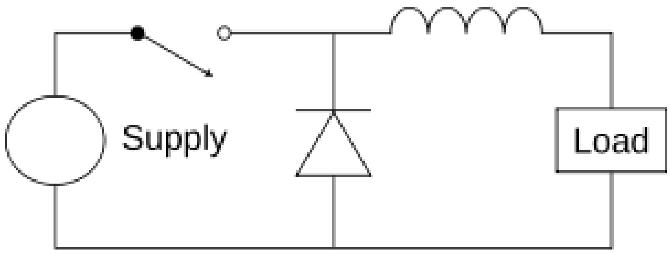


Fig.2.6 Buck converter circuit diagram [15].

Fig.2.7 Buck converter [15].

Switching converters like buck converters have greater power efficiency than linear regulators. The linear regulators are simpler with lowering voltages and they dissipate power as heat. And they don’t step us the output current [15].

Buck converters can be high efficiency (often higher than 90%). Hence, they are used to convert a computer's main supply voltage (often 12V) down to lower voltages needed by USB, DRAM and the CPU (1.8 V or less) [16].

2.7 Veroboard

Veroboard, shown in **Fig. 2.8**, is a multipurpose printed circuit board for electronic projects. It has rows of copper tracks with holes drilled in. Electronic components are soldered to construct circuits. Like the pins of ICs, veroboard has its holes and tracks at a distance of one by tenth inch. The holes in a row are electrically connected by a copper strip. Sometimes, it needs to break the circuit to cut the tracks. Because we don’t want continuous strip while using an IC for example [17].

Veroboard is easily available and cheap. It takes a relatively short time to connect the circuit. It is more reliable and permanent than breadboard.

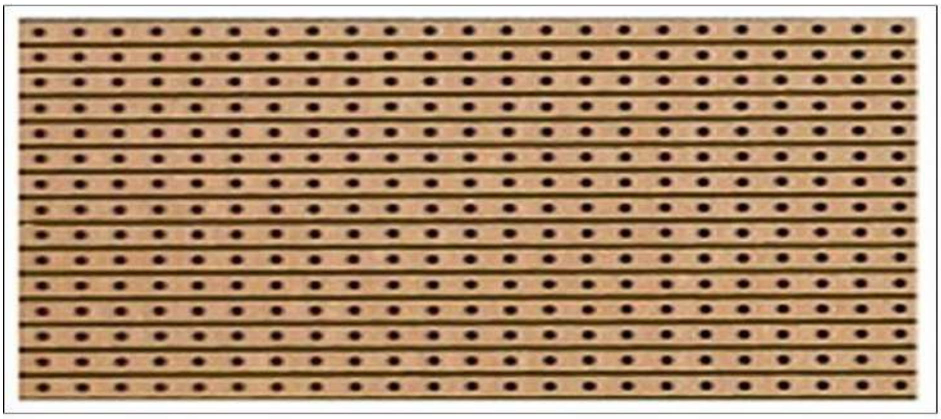


Fig.2.8 Veroboard [18].

Fig.2.9 1961 Veroboard unit – components [18].

Fig.2.10 1961 Veroboard unit – copper [18].

This sort of boards might be used to wire initial electronic circuit development. It may build prototypes for bench testing. And it may produce complete electronic units in small quantities.

Firstly in 1961, Vero Electronics Department contructed prototypes using veroboard as shown in **Fig. 2.9**. The assembled components as well as the copper conductors with needed discontinuities are discovered in some photos of a decade counter sub-unit as shown in **Fig. 2.10**. Connectors installed on a motherboard were used to connect some of the sub-units. Each sub-unit had a digital capacity of half a byte of data storage, requiring 2,000,000 sub-units to store 1 megabyte.

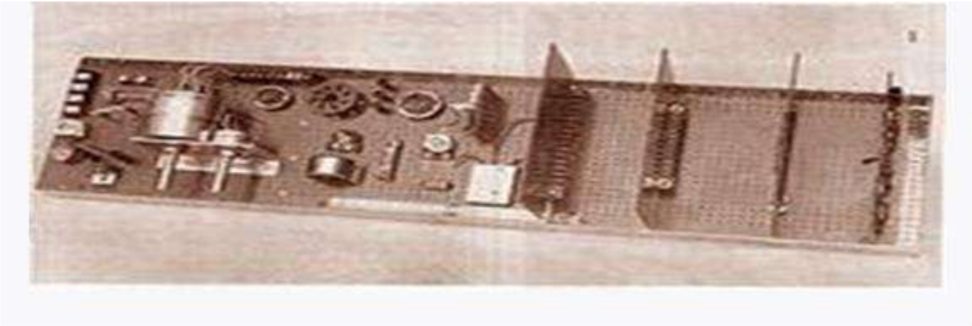


Fig.2.11 1961 Veroboard Display Board [18].

There are two types of Veroboard made. One having 2.54 mm or 0.1 inch distant hole pitch; and another one with 3.5 mm 0.15 inch. The one with a wide pitch is simpler to put together. Especially by the builders were more experienced using valves and tag strips. Dual in-line integrated circuits, which are being more familiar, would fit on the 0.1 boards. As a result, the 0.1 pitch became the most popular. Integrated circuits protruded out of the sides of IC packages, as did the prevalent architecture of short parallel strips. And it motivated to develop special type boards like Verostrip.

Veroboard has a different length than normal. The board is long and thin. And the copper strips are stacked transversely. A ready-cut central gap isolated the sides of the IC. **Fig. 2.11** describes the 1961 model of Veroboard Display Board.

A drawing from 1979 Vero Electronics Ltd production portrays a special Veroboard product. It was developed for RS Components Ltd [17]. Various type of design implementations of Veroboard or Stripboards were developed at that time. It can be found on the Internet.

2.8 Temperature Sensor (DS18B20)

The DS18B20 in **Fig. 2.12** is a single-wire temperature sensor. It is a naturally programmable and digital sensor. This sensor requires only one pin from our NodeMCU for transferring data. We can measure multiple temperatures by connecting it to MCU. While it takes the least number of number of pins on our development board to do so.

It is widely used for measuring temperatures in hard environments like mines. It measures temperatures of chemical solutions, soil, thermostat control systems, high voltage AC supply systems, etc. [19].

Fig.2.12 DS18B20 temperature sensor [19].

2.8.1 How to use the DS18B20 Sensor

The sensor measures data ranging from -55˚C to +125˚C with an accuracy of +/-0.5˚C. the data reading it performs is 9 to 12 bits. It has 3 pins: V_{CC}, GND and Data. There is a built-in pull-up resistor of 4.7 kilo-ohm with its Data pin.

The Data pin is connected to the NodeMCU with the pull-up resistor. V_{CC} and GND pins are used for power as in fig. 2.13.

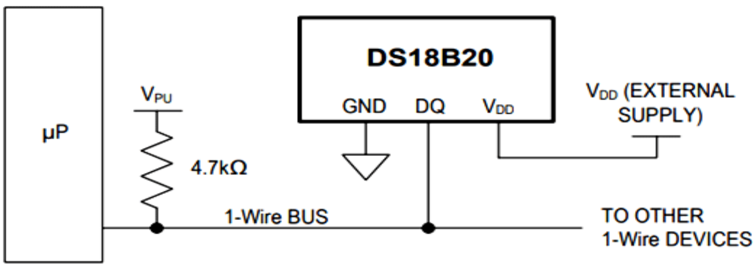


Fig.2.13 Working principle of DS18B20 sensor [19].

As illustrated in **Fig. 2.13**, a pull-up resistor is used with the Data pin that keeps the line high in case the bus suddenly misconduct. The temperature value is stored in a 2-byte register inside the sensor. The data is then sent in a sequence. It is read by using the single-wire method. Two types of command are used to read the values: ROM command and function command. The ROM command is used to select which DS18B20 that the master wants to communicate with. And the function command allows the master to read and to write from the DS18B20's scratchpad, etc. [20].

2.9 Resistor

Resistors are passive electrical components. They have two terminals: positive and negative. They reduce current flow and lower voltages at certain points in electrical circuits. They are also used to adjust signal levels, bias active elements and terminate transmission lines. There are high power resistors as well which dissipate big watts of power as heat. They are effective in motor controlling, power distribution system and testing loads for generation. **Fig. 2.14** describes the picture of two color coded resistors.

Fig 2.14 Resistor [21].

There are two basic types of resistors: linear resistor and non-linear resistor. Again, linear resistors are divided into fixed and variable resistors.

Fixed resistors don't vary or slightly vary their resistance with temperature, time or operating voltage.

Variable resistors adjust circuit elements such as volume control or a lamp dimmer. Variable resistors are also used as sensing devices for heat, light, force, humidity or chemical activity.

The resistance value of a resistor defines its function. Commercial resistors come in a wide variety of sizes, spanning more than nine orders of magnitude. The nominal resistance of a component is found within the manufacturing tolerance [21].

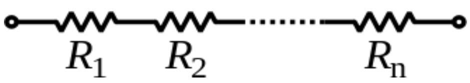
2.9.1 Theory of Operation

The resistance of a resistor is mainly applied in Ohm's law. In Ohm's law, resistance (R) is the proportionality constant while voltage across resistor (V) is proportional to current (I). Some practical resistors hold a bit of inductance and capacitance. It affects the relation between voltage and current in AC circuits.

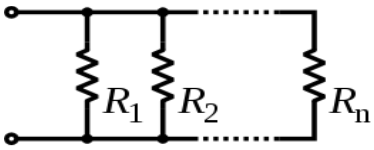
The SI unit of Resistance is ohm, Ω. It was named after Georg Simon Ohm. One ohm is equivalent to one volt per ampere. Resistors are manufactured for different range of values. Some commonly used derived units are milliohm (1 mΩ = 10^{-3} Ω), kilo-ohm (1 kΩ = 10^3 Ω), and mega-ohm (1 MΩ = 10^6 Ω) [22].

2.9.2 Series and Parallel Resistors

The sum of individual resistance values of resistors is defined as the total resistance when connected in series.



The reciprocal of the sum of the reciprocals of the individual resistors is the total resistance when connected in parallel.



For example, a 30-ohm resistor is connected in parallel with a 10-ohm resistor and a 20-ohm resistor. The total resistance will be 1 / (1/30 + 1/10 + 1/20) ohms, or 60/11 = 5.454 ohms.

2.10 3.7V Rechargeable battery

The battery used in the system is an 18650 battery and is shown in **Fig. 2.15**. It is a lithium-ion rechargeable battery. The first four digits of the battery model 18650 refer to its physical dimension. And the fifth one refers to the cylinder cell. The size of this kind of a standard battery has 18mm in length and 65 mm around. They are often used in devices that need a portable power, such as in laptop battery packs, electric vehicles, cordless tools, flashlights, etc [23].

2.10.1 Style

- Flat top: here, the positive terminal of the battery has a flat surface.
- Button top: here, the positive terminal of the battery is protruding.

Fig 2.15 3.7V Rechargeable Battery [24].

2.10.2 Nominal Capacity

- The minimum capacity that the battery holds.
- It is measured in mAh (miliamp hours).
- The unit ‘milliamp hours’ measures electrical power over the time.
- A higher mAh rating indicates a fully-charged battery. It can power a device which requires power.
- And it provides charging for a long time until it becomes depleted. Then it needs to be recharged.

2.10.3 Continuous Discharge Rating

The maximum current at which battery can be discharged continuously without being damaged or reducing its capacity is defined as Continuous Discharge Rating.

2.10.4 Pulse Discharge Rating

The maximum current at which battery can be discharged for a short period without being damaged or reducing its capacity is defined as Pulse Discharge Rating.

2.10.5 Nominal Voltage

Nominal voltage is an assigned value for the purpose of classification. The mid point of the range within which operation takes place is called the Nominal Voltage.

2.10.6 Protected

There is a small electronic circuit in protected batteries integrated into the cell packaging. The circuit prevents the battery from common dangers. The dangers can be overcharging, over discharging, short circuit, over current, temperature, etc. Due to its integrated circuit, protected batteries are usually slightly longer [24].

2.11 Connecting Wire

A wire is a metal rod which carries mechanical load, electricity and telecommunication signals. It is mostly cylindrical in shape and has flexible strand. Jumper wires are wires with two connector pins to connect two points in a circuit without soldering. The jumper wires have different colors on it. But it doesn't mean anything different. The user can use the color to recognize different polarities.

There are three types of jumper wire: male to male, male to female, and female to female. The male end means a pin is protruded to be pushed into a hole. And female ends don't have protruded pins. They have holes to receive male pins. A bundle of male to male wires is shown in **Fig. 2.16**.

Wires are commonly made by stretching a metal through a hole in a die or draw plate. Wire gauges are found in various standard sizes, described by means of a gauge number.

Wire rope in mechanics or a cable in electricity is a multi-stranded wire with a bundle of such strands. Wires are commonly in solid core, stranded, or braided forms. Though wires are mostly circular in cross- section, wires can be made in square, hexagonal, flattened rectangular or other cross- sections. The shapes are decided for decorative purposes or for technical purposes. Some of the technical purposes are high- efficiency voice coils in loudspeakers, edge-wound coil springs, such as the Slinky toy [25].

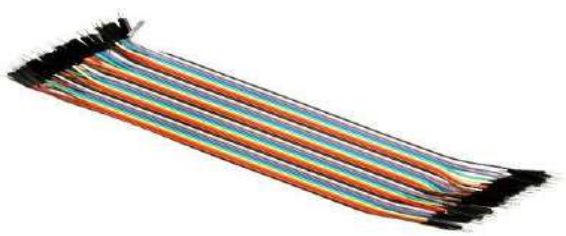


Fig 2.16 Connecting Wire [25].

2.12 IR Sensor

An Infrared Sensor is an electronic device that detects heat and motion of objects from surroundings. To do so, it emits light radiations, whereas passive IR only measures infrared radiations. Objects radiate heat in the infrared spectrum. These radiations are invisible to our eyes. They are only detected by IR sensors. **Fig. 2.17** shows the external appearance of an IR sensor.

IR sensor has an IR LED as its emitter and an IR photodiode as its detector. The photodiode senses IR light of same radiation as emitted by LED. When photodiode receives an IR light, the resistance and output voltage change with respect to the magnitude of IR light received as illustrated in **Fig. 2.18**.

IR sensor radiations are of 0.75 to 1000µm wavelength. The radiation falls between visible and microwave regions of electromagnetic spectrum. The IR spectrum can be divided into three regions based on its wavelength: Near Infrared, Mid Infrared, and Far Infrared.

Fig 2.17 IR Sensor [26].

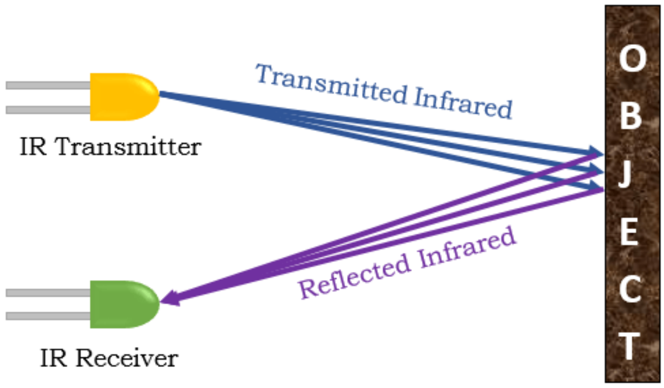


Fig 2.18 IR Sensor Working Principle [27].

The wave lengths of these spectrums are:

Near IR is 0.75µm to 3 µm; Mid IR is 3 µm to 6 µm, and Far IR is > 6 µm [27].

2.13 Relay Module

A relay is an electromechanically controlled switch. It can be controlled by a low power signal. Also, several circuits can altogether be controlled by a single signal. The relay module shown in **Fig. 2.19** works for a wide range of microcontroller units such as Arduino, AVR, PIC, and ARM. It works with digital outputs.

Fig 2.19 1-channel 5V Relay Module [29].

The relay used in this project is a one-channel 5V DC relay. The maximum load it can take is 250V/10A for AC and 30V/10A for DC. The triggering current is 5 miliampere and working voltage is 5V. It has VCC, GND and IN pins in one side; and NC, NO and COM pins in the other side.

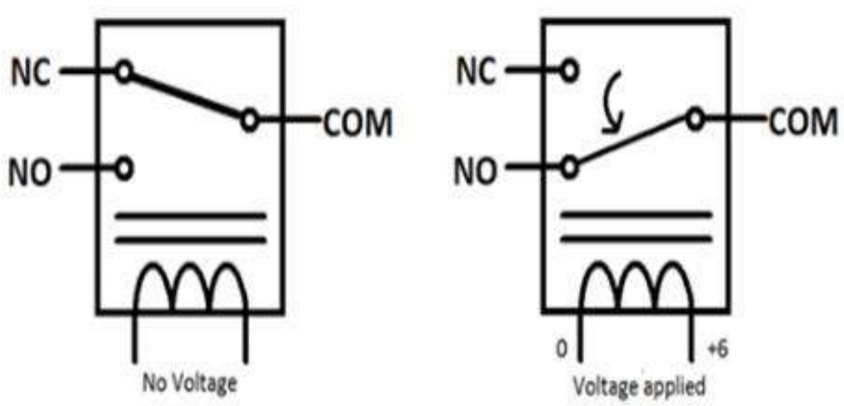


Fig 2.20 Relay Working Principle [28].

Fig. 2.20 describes the working principle of a relay. When no voltage is applied to the input pin, the armature stays connected to the NC pin (Normally Closed). When the input pin gets voltage, the armature moves to the NO (Normally Open) and the relay triggers. The COMMON pin is always connected to the positive of the load [28].

2.14 Water Pump

A pump is a device that convert electrical energy into mechanical energy to move fluids. There are three basic types of pumps: lift pump, displacement pump and gravity pump. They can be operated by electricity from engine or wind power. They can be found in various sizes [30].

The pump used in this project is a micropump as shown in **Fig. 2.21**. It lifts the sanitizer fluid in our system and sprays it out to peoples. Its rating is 130 to 220 miliampere and it works on 2.5V to 6V supply. The rate at which the pump moves sanitizer is 1.25 litre per minute. And it can lift from 40 to 110 centimeters [31].

Fig 2.21 Water Pump [30].

2.15 Buzzer

A buzzer or beeper is an audio-signaling device. It can be mechanical, electromechanical, or piezoelectric. Buzzers are mainly used in alarm devices, timers, and when confirming user inputs such as a mouse click or a keystroke.

A buzzer is shown in **Fig. 2. 22**. It has two pins: positive and negative. The short one is connected to ground and the long one to supply. The buzzer operates on 4V to 8V DC and its current rating is less than 30 miliampere. The resonant frequency is approximately 2300 Hz. It is Breadboard and Perf board friendly.

Fig 2.22 Buzzer [32].

2.15.1 Applications of Buzzer

- Alarming circuitries.
- Communication devices
- Automobile electronics.
- Portable devices, due to its compact size [32].

CHAPTER 03

METHODOLOGY

3.1 Block Diagram and its Analysis

The block diagram of the proposed system is shown in **fig. 3.1**. In the block diagram, the NodeMCU which has a built in WiFi module is powered by a DC supply. Then it takes data from pulse oximeter sensor (Max30100) and the temperature sensor (DS18B20).

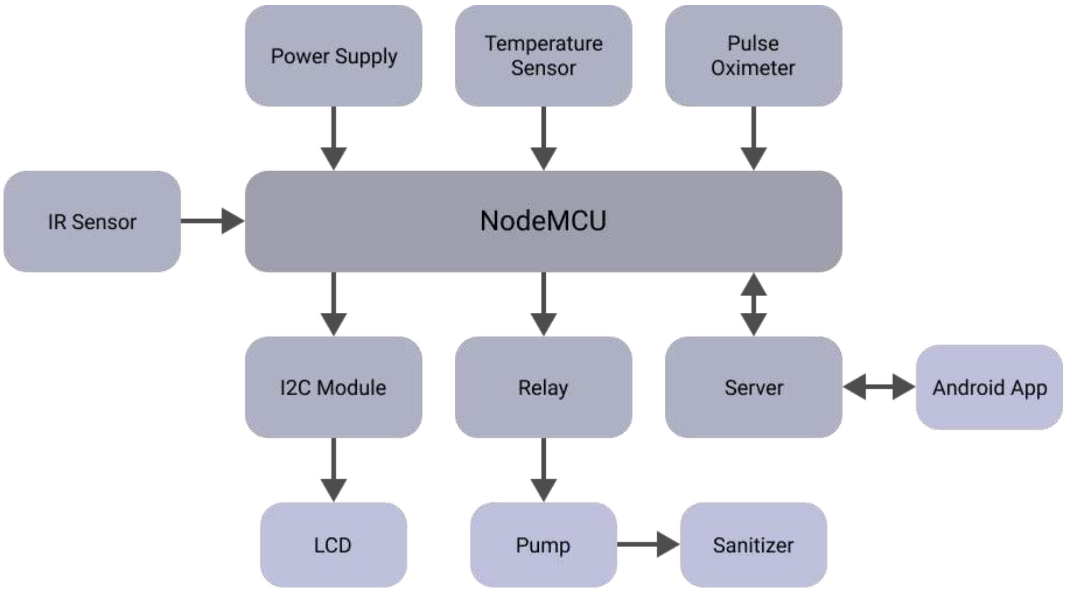


Fig. 3.1 Block diagram of the system.

By using logic and coding, the NodeMCU processes stores all the data and shows it on the LCD by means of I2C Module. It also sends the data to an android app (Blynk IoT) using the WiFi communication. There, an IR sensor is also connected to the NodeMCU which detects objects placed in front of it and as a result, it turns on the relay to pump the sanitizer out to the object.

3.2 Flow Chart and its Description

As **fig. 3.2** describes, firstly, the circuit is initialized. Then it checks whether NodeMCU is connected to Wi-Fi or not. If NodeMCU is connected to Wi-Fi then it checks Max30100 sensor. If the finger is placed on the sensor, it shows the pulse rate and oxygen saturation level on the (LCD) display and android App together using Blynk. If the finger is not placed, it keeps checking for the finger.

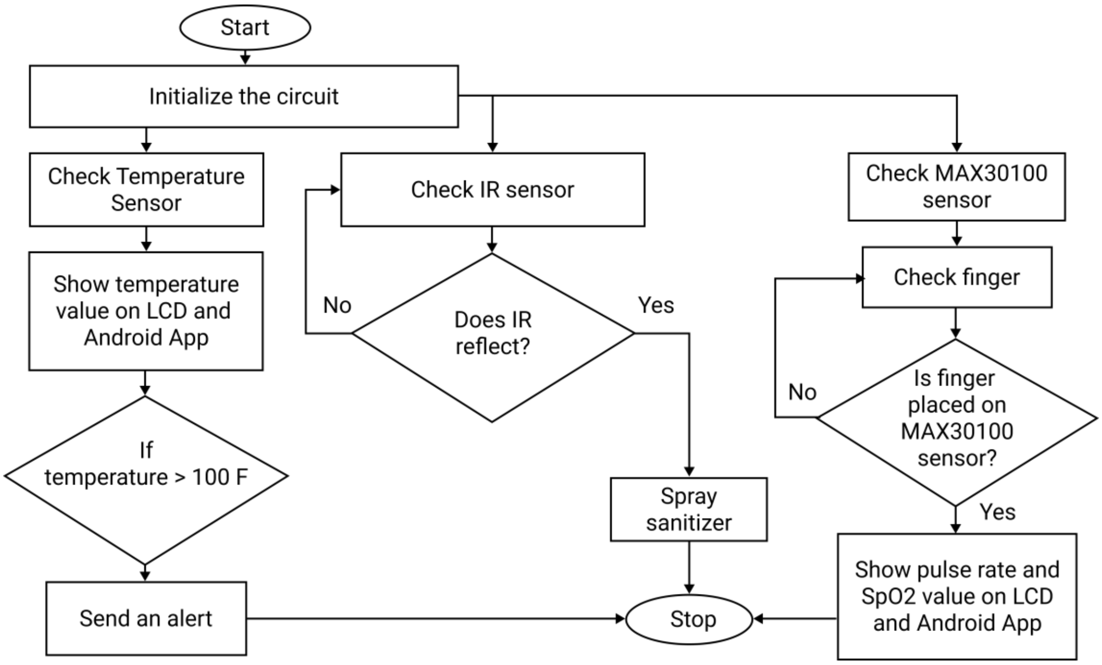


Fig. 3.2 Flowchart of the system.

Then, the NodeMCU checks the temperature sensor and shows the reading on the LCD and the Android App. In case the temperature exceeds 100 ˚F, it sends an alert through buzzer. Then, it stops.

It then checks the IR sensor. If IR sensor doesn't reflect any object, it keeps checking. If IR sensor detects the object, it sprays sanitizer to it and then stops.

This system continuously works in a loop with this programming logic.

3.3 Circuit Diagram

In **fig. 3.3**, first of all, we have used four series-connected batteries of 3.7V each, which gives us an approximately 14.8V DC source. As our microcontroller unit works with 5V DC source, we have used a buck converter which steps down the battery voltage to 5V. The input pin of the NodeMCU, V_{IN}, receives the output of the buck converter and starts to operate. It then takes the temperature reading from DS18B20 sensor whose Data pin along a 4.7 kΩ pull-up resistor is connected to the input pin D5 of the MCU. It also takes the data of heart rate and blood saturation level from the Max30100 sensor whose SCL and SDA pin are connected to D1 and D2 of NodeMCU through I2C Module respectively.

Since the Max30100 works on about 3V supply, its VCC and GND pin are connected to the 3.3V pin and GND pin of the NodeMCU. And the I2C Module is again connected to the 16*2 LCD through SCL and SDA pin where we can read the data of the sensors. The VCC and GND pins are connected accordingly.

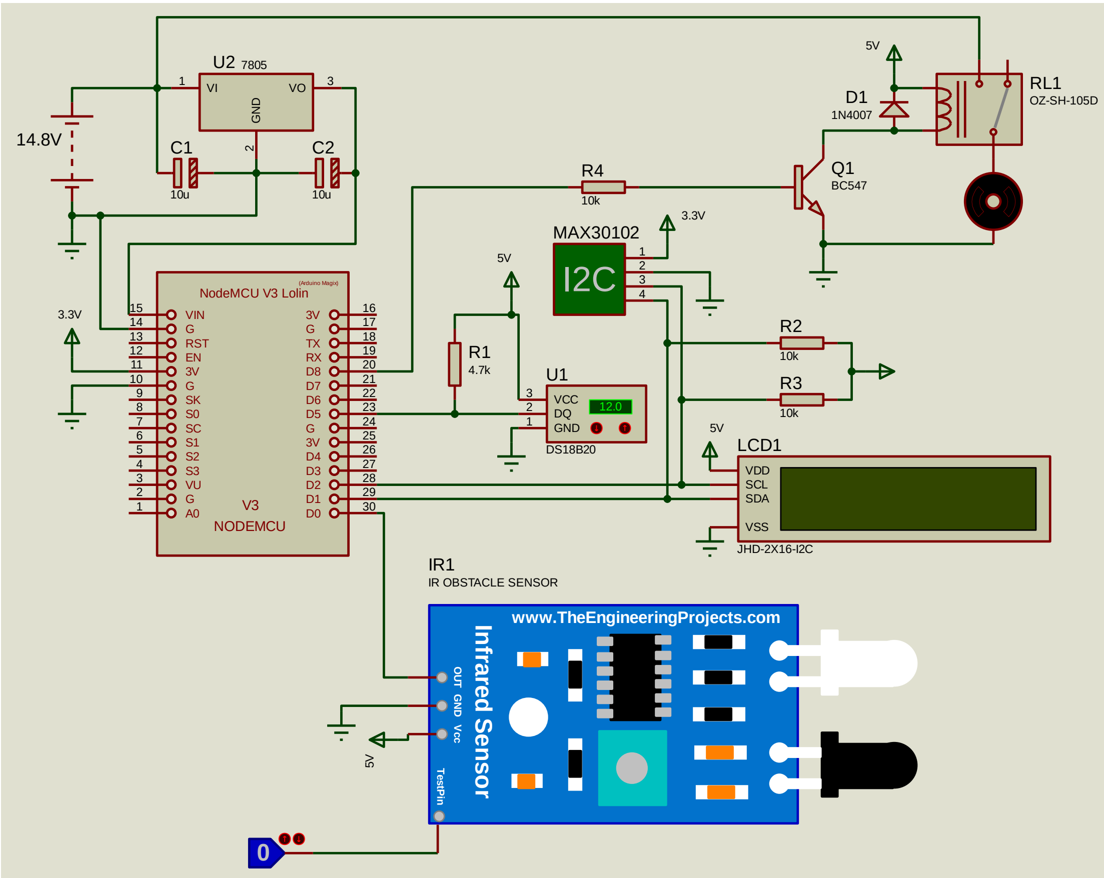
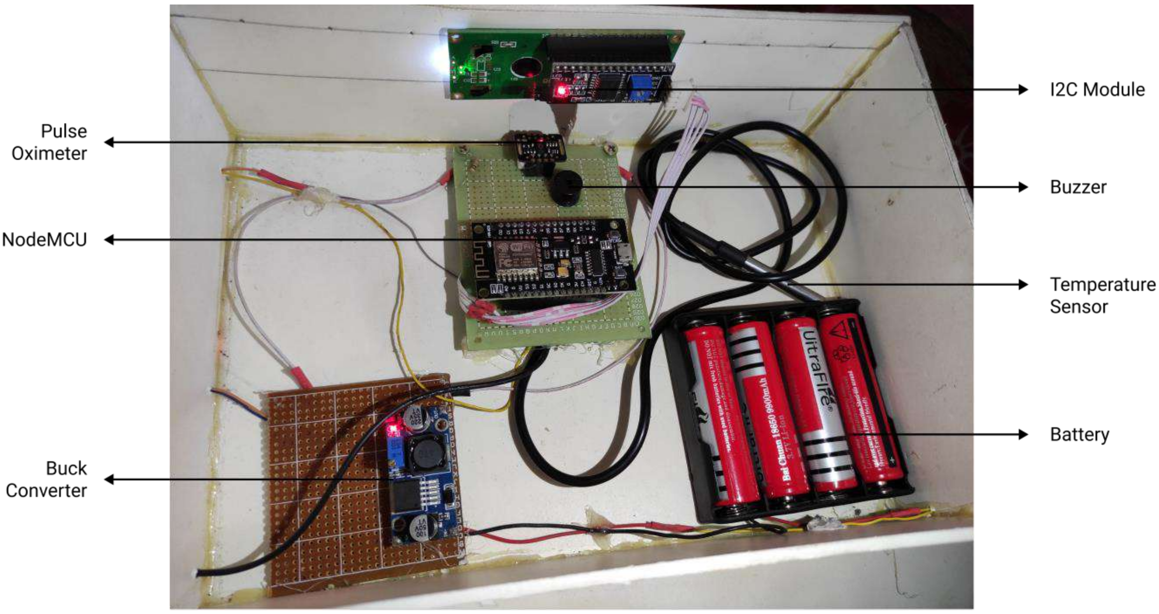


Fig.3.3 Circuit diagram of the system.

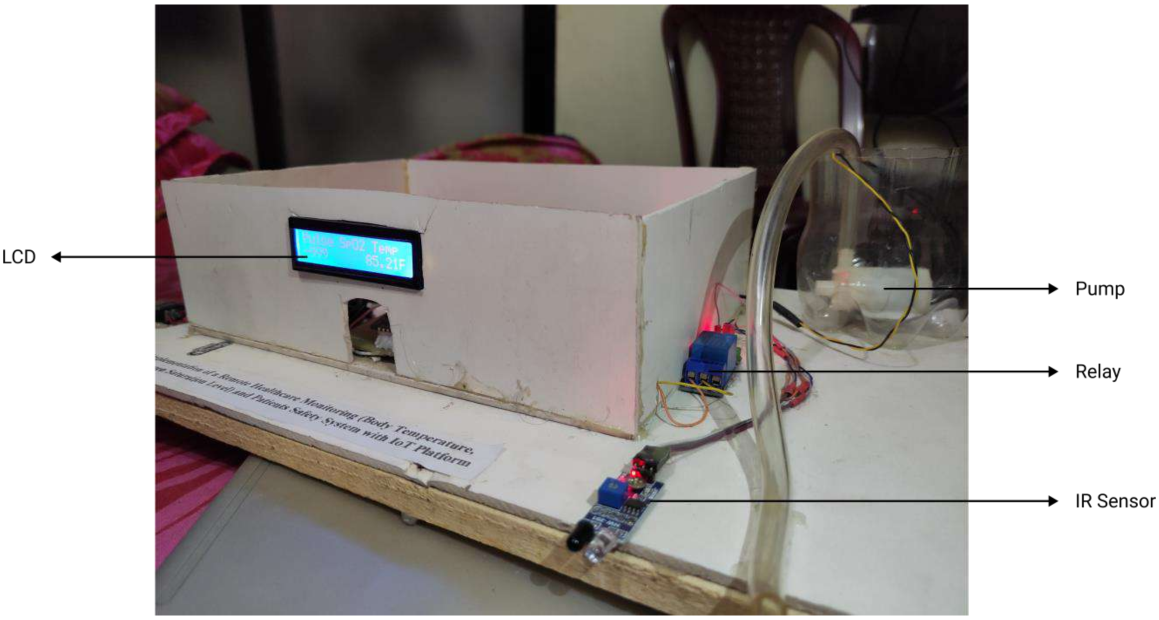
An IR sensor is then connected to detect any object placed in front of it. Its Output pin is connected to the signal pin of the relay. The NO (Normally Open) pin of the relay is connected to the VCC of the NodeMCU. And the Common pin (COM) is connected to the positive terminal of the water pump. The water pump then pumps sanitizer to the patient as soon as the IR sensor detects any hand or other objects in front of it. Two pull-up resistors are used with SCL and SDA pins of the I2C module for a continuous high line without any voltage drop.

3.4 Hardware Implementation

Fig. 3.4 (a) shows the components inside the box while they were in operating mode. And **fig. 3.4 (b)** shows the components attached outside the box. All the components are captioned accordingly by means of arrows as shown below.



(a)



(b)

Fig.3.4 Components view within the box.

CHAPTER 04
RESULTS AND DISCUSSION

4.1 Measured Data from People

With the approached project, we are able to monitor the patient's body temperature, pulse rate and oxygen saturation level without physical meeting of the patient and the doctor. The output results are shown below:

Table 4.1 Comparison of data between medical device and proposed device

Patient Name	Gender	Age	Actual Temperature	Sensor temperature	Actual Pulse (BPM)	Sensor Pulse(BPM)	Actual SpO2	Sensor SpO2
Sample 1	Male	22	96.3 ˚F	96.13 ˚F	107	107	98%	99%
Sample 2	Male	24	97.2 ˚F	97 ˚F	102	105	97%	99%
Sample 3	Male	23	98.6 ˚F	98.26 ˚F	110	107	98%	99%
Sample 4	Male	21	96.8 ˚F	95.9 ˚F	82	83	96%	99%
Sample 5	Male	25	94.2 ˚F	94 ˚F	96	99	97%	99%
Sample 6	Male	23	97.8 ˚F	96.9 ˚F	69	67	97%	97%
Sample 7	Male	25	97.6 ˚F	97.1 ˚F	81	79	99%	98%
Sample 8	Male	23	97.7 ˚F	98 ˚F	102	100	99%	99%
Sample 9	Male	27	96.7 ˚F	96.9 ˚F	85	85	98%	98%
Sample 10	Male	17	95.5 ˚F	95.2 ˚F	69	67	97%	97%

In **Table 4.1**, the measured parameters by using the proposed project are compared to the values measured by conventional medical devices. Here, firstly, a thermometer modeled as XHF2001 was used to compare with the project's DS18B20. Both the heads of thermometer and DS18B20 were placed under the armpit of 10 different persons of different ages. After keeping them there for one minute, their readings were noted down.

Then a Jumper Oximeter modeled JPD-500D was used to measure heart rate and SpO2 values of those 10 persons and they were noted down. Similarly, the readings from the Max30100 sensor was noted down to compare it to the values measured by the Jumper.

4.1.1 Comparison of temperature between thermometer and proposed device

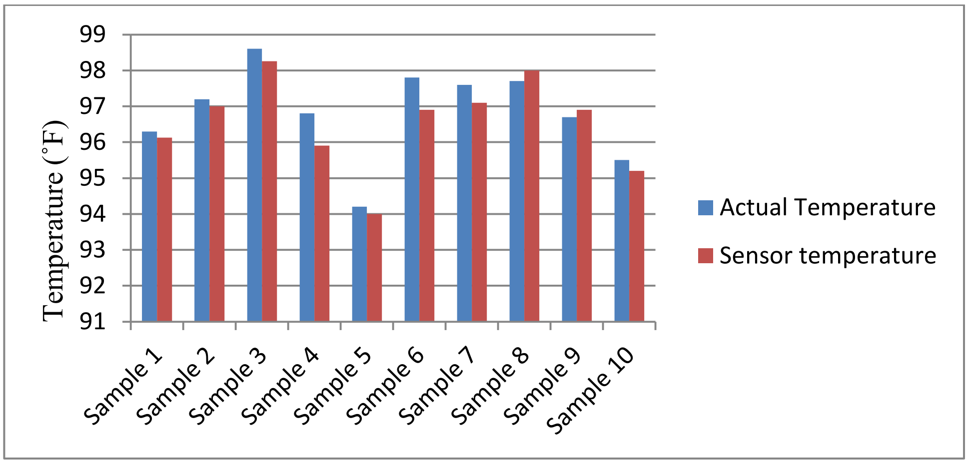


Fig. 4.2 Comparison of temperature between thermometer and proposed device.

The different temperature values of 10 people are compared between readings in the thermometer and the DS18B20 sensor in **fig. 4.2**, 8 out of 10 people had shown a bit lower temperature in the proposed project than in the medical thermometer; and two of them showed a bit higher value than the thermometer's one; the difference is ranging from 0.0 to 0.5˚C which is considered accurate according to DS18B20 sensor datasheet [18].

4.1.2 Comparison of pulse data between JPD-500D and Max30100 sensor

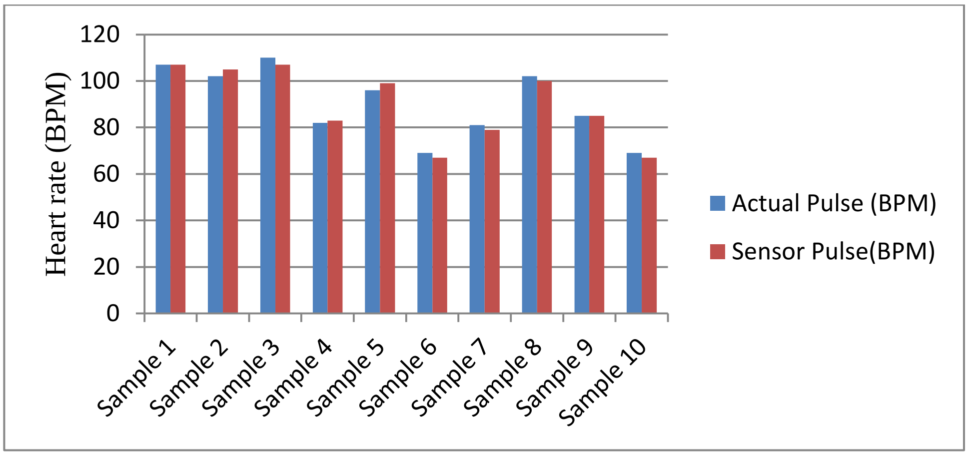


Fig.4.3 Comparison of pulse data between JPD-500D and Max30100.

In **fig. 4.3**, heart rate values are compared between the proposed approach and the Jumper device found in the market. In the project, heart rate was measured by placing the fingers of the 10 different personalities on the Max30100 sensor. After waiting for four seconds according to the sensor demand, the readings were recorded. While the pulse rate reading from the Jumper JPD-500D was recorded after placing the same fingers inside the jumper mouth. It also took 3 to 4 seconds to show the value. As fig. 4.3 describes, some of the data between both sensors are exactly same, while a few of them are differing by one or two BPMs. So it can also be said to be well accurate.

4.1.3 Comparison of SpO2 Values between Jumper JPD-500D and Max30100 sensor

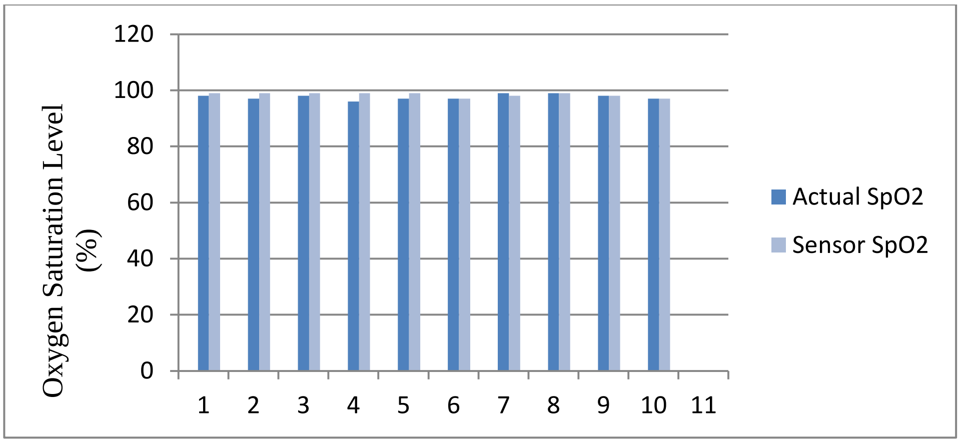


Fig.4.4 Comparison of SpO2 values between JPD-500D and Max30100.

Fig. 4.4 compares the oxygen saturation level between the proposed device and Jumper JPD-500D. Both the devices have shown almost same values between 97 and 100 percentages. This value, SpO2 is processed and shown on the basis of oxygenated hemoglobin and deoxygenated hemoglobin. There are some manufacturers who have their own look up system on SpO2 value. Typically, this project work calculates the SpO2 value on the ratio of red light absorbed and infrared light absorbed. If the R/IR ratio is 0.5, then SpO2 equals approximately 100%. And if the ratio stands at 1.0, the SpO2 value equals about 82%, and for the ration of 2, the SpO2 value becomes almost zero [12].

4.2 Displaying Measured Parameters

Fig.4.5 shows the upper front side of the enclosed project. This view is showing only the switch, LCD sanitizing pump and IR sensor. While the components inside the box are battery, buck converter, NodeMCU, DS18B20 temperature sensor, Max30100 pulse oximeter, and I2C module. In this picture, a message can be seen on the LCD which reads as “Initializing Sensor”. After connecting the system to WiFi, this message appears on the LCD for two seconds.

Fig.4.5 Upper front side of the enclosed project.

Fig.4.6 Output data shown on LCD.

Then, when a finger is placed on the Max30100 sensor, pulse rate and SpO2 values are shown on the LCD and Android App. **Fig.4.6** represents the project's LCD while **Fig. 4.7** describes the mobile screen through Blynk Applicaion. In **Fig. 4.6** and **Fig. 4.7**, it can clearly be seen that the pulse is 93, SpO2 value is 99%, and temperature is 81.95˚F. The value of temperature came from the DS18B20 temperature sensor

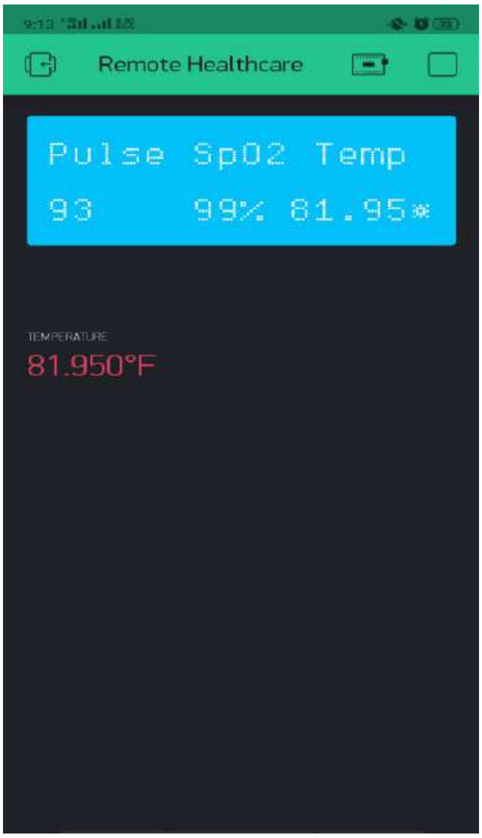


Fig.4.7 Output data shown on Android (Blynk App).

CHAPTER 5

CONCLUSION AND FUTUREWORK

5.1 Conclusion

People are advancing in technology, but we can’t ignore health concerns. Health concerns affect a country or even the whole world economically and socially. 20% of everyday deaths are mainly due to heart diseases. Recently, COVID19 for example, has affected the whole world, especially medical system and facilities of the countries all over the world. This type of situation demands remote monitoring technologies for everything. IoT can be considered the best solution for such technologies. This can help the doctors by observing the patient from a far-away room, and the patient can also consult from his own home which saves both his time and money. Different kinds of sensors are used to track people’s health, and using internet helps transmit their conditions to their concerned ones. Other aims behind remote healthcare are cost effectiveness and size reduction by IoT-based monitoring platforms. The readings can be seen by several people including family members and caretakers by means of web-server. All the discussed functionalities are processed by means of a URL or a site. Some of the countryside regions also don’t have immediate physical doctoring facilities and it takes a lot of time for the patients to reach hospitals and undergo medical procedures.

So, the proposed design, an IoT platform which uses NodeMCU as a microcontroller unit that receives readings of temperature, heartbeat and SpO2 values using DS18B20 temperature sensor and Max30100 pulse oximeter. The readings are displayed on an LCD in front of the patient and on the mobile screen of the physician and concerned ones using Blynk app. This project also used an IR sensor which detects objects placed in front and it turns the relay on to pump the sanitizer without even being touched. The project is working properly compared to the conventional method and it only takes a minute to operate. This project has outperformed the conventional method by giving us a safer, less complex, cost effective and faster service.

5.2 Future Works

The system proposed in this project mainly aims heart patients. There are a lot of scopes for further development both in hardware and software. It can be designed to monitor the whole ward room and detect the physical situation of patients through WiFi Module. In case of software implementation, the webserver can be developed with more advanced features and apps. The screen can be used of a bigger size where all the details related to the patient including his name, time, and his family's contact will be present The developed softwares can be uploaded to Google Play Store so that people can easily access it and use it. Internet of Things is always there to let us build comprehensive services.

REFERENCES

[1] "Most Common Causes of Death," Accessed: Oct 21, 2021. [Online]. Available: Most Common Causes of Death Around the World (US, UK and More) (ways-to-die.com)

[2] I. Azimi, A. Anzanpour, A. M. Rahmani, P. Liljeberg, T. Salakoski, "Medical Warning System Based on Internet of Things Using Fog Computing," in *2016 Int. Workshop Big Data and Inf. Security*, Dept. of Information Technology, University of Turku, Finland, 2016, pp. 2-4.

[3] V. Pardeshi, S. Sagar, S. Murmurwar, P. Hage, "Health Monitoring Systems using IoT and Raspberry Pi – A Review", in *Int. Conf. Innovative Mechanisms for Industry Applications*, 2017, pp. 2-3.

[4] R.N. Kirtana, Y.V. Lokeswari, "An IoT Based Remote HRV Monitoring System for Hypertensive_Patients," in *IEEE Int. Conf. Computer, Communication and Signal Processing - ICCCSP*, 2017, pp. 3-4.

[5]. R. A. Rahman, N. S. A. Aziz, M. Kassim, M. I. Yusof, "IoT based Personal Health Care Monitoring Device for Diabetic Patients," in *2017 IEEE Symp. Computer Applications and Industrial Electronics (ISCAIE)*, 1 April 2017, pp. 2-5.

[6]. C.A. Boano, M. Lasagni, K. Römer, T. Lange, "Accurate Monitoring Of Cardiac Rhythms Using Wearable Body Sensor Networks," in *10th ACM/IEEE Int. Conf. Information Processing and Sensor Networks*, 2011, pp. 169– 170.

[7] V. Pardeshi, S. Sagar, S. Murmurwar, P. Hage, "Health Monitoring Systems using IoT and Raspberry Pi – A Review," in *Int. Conf. Innovative Mechanisms and Industry Applications – ICIMIA*, 2017.

[8] S. Lavanya, G. Lavanya, J. Divyabharathi, "Remote Prescription and I-home Healthcare IoT," in *IEEE Int. Conf. Innovations Green Energy and Healthcare Tech.*, 2017.

[9] M. S. Chandana, Dr. M.R. Arun, "An IoT based Patient Health Monitoring System using Node MCU," in *Int. Research Journal of Engineering and Technology (IRJET)*, Oct 2020, vol. 7, Iss. 10.

[10] "Getting Started w/ NodeMCU ESP8266 on Arduino IDE," Accessed: July 12, 2021. [Online]. Available: Getting Started w/ NodeMCU ESP8266 on Arduino IDE - Arduino Project Hub

[11] "Interfacing MAX30100 Pulse Oximeter Sensor with Arduino," Accessed: July 15, 2021. [Online]. Available: Interfacing MAX30100 Pulse Oximeter Sensor with Arduino (how2electronics.com)

[12] "MAX30100 Heart rate Sensor IC: Specification, Pinout and Datasheet," Accessed: July 15, 2021. [Online]. Available: MAX30100 Heart-Rate Sensor IC : Specification, Pinout and Datasheet (utmel.com)

[13] "LCD 16×2 Pin Configuration and Its Working," Accessed: July 21, 2021. [Online]. Available: LCD 16x2: Pin Configuration, Features and Its Working (elprocus.com)

[14] "Introduction to I2C LCD Adapter Module," Accessed: July 21, 2021. [Online]. Available: Introduction to I2C LCD Adapter Module - Homemade Circuit Projects (homemade-circuits.com)

[15] “Buck Converter – Circuit, Design, Operation and Examples,” Accessed: July 30, 2021. [Online]. Available: Buck Converter - Circuit, Design, Operation and Examples (electricaltechnology.org)

[16] S. Palanidoss, V. Theja.V S, “Experimental Analysis of Conventional Buck and Boost Converter with Integrated Dual Output Converter,” in *Int.Conf. Electrical, Electronics, Communication, Computer and Optimization Techniques (ICEECCOT)*, 2017.

[17] “How to use Veroboard,” ACCESSED: Aug. 5, 2021. [Online]. Available: How to use veroboard - Hobby electronic soldering and construction (bestsoldering.com)

[18] “Veroboard,” Accessed: Aug. 5, 2021. [Online]. Available: Veroboard - Wikipedia

[19] “DS18B20 Temperature Sensor,” Accessed: Aug. 6, 2021. [Online]. Available: DS18B20 Temperature Sensor Pinout, Specifications, Equivalents & Datasheet (components101.com)

[20] “DS18B20,” Accessed: Nov. 6, 2021. Available: pic_examples.byethost3.com/DS18B20.html

[21] “Resistors,” Accessed: Aug. 8, 2021. [Online]. Available: Resistors - learn.sparkfun.com

[22] “Resistors and its Operation,” Accessed: Aug. 5, 2021. [Online]. Available: Resistor: What is it & What Does it Do? (Examples Included) | Electrical4U

[23] N. Xu, L. Kong, G. Wen, M. G. Pecht, “Protection Devices in Commercial 18650 Lithium-Ion Batteries,” in *Digital Object Identifier 10.1109/ACCESS.2021.3075972*, April 27, 2021.

[24] “18650 Battery in Review,” Accessed: Aug. 28, 2021. [Online]. Available: Best 18650 Battery in Review - Updated August 28, 2021 (powergenixsystems.com)

[25] “Jumper Wire,” Accessed: Aug. 8, 2021. [Online]. Available: What is a Jumper Wire? (sparkfuneducation.com)

[26] “IR Sensor Interfacing with Arduino,” Accessed: Aug. 15, 2021. [Online]. Available: Arduino ir proximity sensor interfacing | circuits4you.com

[27] “Infrared Sensor,” Accessed: Aug. 15, 2021. [Online]. Available: Infrared Sensor - How it Works, Types, Applications, Advantage & Disadvantage (electricalfundablog.com)

[28] “Relay Module Interfacing,” Accessed: Aug. 15, 2021. [Online]. Available: Relay Module interfacing with pic16f877a microcontroller - Ac load control (microcontrollerslab.com)

[29] “Relay,” Accessed: Aug. 15, 2021. [Online]. Available: relay picture - Bing images

[30] “Micropump,” Accessed: Aug. 30, 2021. [Online]. Available: Micropump - Wikipedia

[31] “Micropump: An Overview,” Accessed: Aug. 30, 2021. [Online]. Available: Micropump - an overview | ScienceDirect Topics

[32] “Active Passive Buzzer,” Accessed: Sept. 2, 2021. [Online]. Available: Buzzer Pinout, Working, Specifications & Datasheet (components101.com)

APPENDIX

```
#define BLYNK_PRINT Serial
#include <LiquidCrystal_I2C.h>
#include <Wire.h>
#include <OneWire.h>
#include <DallasTemperature.h>
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <SimpleTimer.h>
#include <DFRobot_MAX30102.h>

DFRobot_MAX30102 particleSensor;
const int DS18B20 = D5;
const int buzzer = D8;
////int SPO2=0;
//int heartRate=0;
float bodytemperature;
LiquidCrystal_I2C lcd(0x27, 16, 2);
OneWire oneWire(DS18B20);
DallasTemperature sensors(&oneWire);
char auth[] = "2MGDIR7nLbYHk5uSLZjeA7sf2Nt-Xapt";
char ssid[] = "remotehealthcare1";
char pass[] = "Remote.healthcare1";
WidgetLCD lcd1(V2);
SimpleTimer timer;

int32_t SPO2 = 0; //SPO2
int8_t SPO2Valid; //Flag to display if SPO2 calculation is valid
int32_t heartRate = 0; //Heart-rate
int8_t heartRateValid; //Flag to display if heart-rate calculation is valid

void setup()
{

Blynk.begin(auth, ssid, pass);
pinMode(buzzer, OUTPUT);
lcd.init();
lcd.backlight();
lcd.begin(16, 2);
Serial.begin(115200);
Serial.println("Initializing...");
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Initializing");
lcd.setCursor(0, 1);
lcd.print("Sensor");
delay(2000);


Serial.begin(115200);
```

```
while (!particleSensor.begin()) {
Serial.println("MAX30102 was not found");
delay(1000);
}

particleSensor.sensorConfiguration(/*ledBrightness=*/50,
/*sampleAverage=*/SAMPLEAVG_4, \
/*ledMode=*/MODE_MULTILED, /*sampleRate=*/SAMPLERATE_100, \
/*pulseWidth=*/PULSEWIDTH_411, /*adcRange=*/ADCRANGE_16384);
 lcd.clear();
}

void loop()
{

heartRate = 0;
Serial.println(F("Wait about four seconds"));
particleSensor.heartrateAndOxygenSaturation(/**SPO2=*/&SPO2,
/**SPO2Valid=*/&SPO2Valid,    /**heartRate=*/&heartRate,
/**heartRateValid=*/&heartRateValid);
//Print result
Serial.print(F("heartRate="));
Serial.print(heartRate, DEC);
Serial.print(F(", heartRateValid="));
Serial.print(heartRateValid, DEC);
Serial.print(F("; SPO2="));
Serial.print(SPO2, DEC);
Serial.print(F(", SPO2Valid="));
Serial.println(SPO2Valid, DEC);
// heart
lcd.clear();

lcd.setCursor(0, 0);
lcd.print("Pulse");
lcd.setCursor(6, 0);
lcd.print("SpO2");
lcd.setCursor(0, 1);
 lcd.print(heartRate);
lcd1.clear();
lcd1.print(0, 0, "Pulse");
lcd1.print(0, 1, heartRate);
lcd1.print(6, 0, "SpO2");
//// Blynk.virtualWrite(V7, heartRate);
// Blynk.virtualWrite(V8, SPO2); body_temp();
Blynk.run();

// if (SPO2 > 99)
// {
// // beatAvg = 0;
// SPO2 = 0;
// delay(50);
// }
```

```
if (heartRate > 50)
{
lcd.setCursor(6, 1);
lcd.print("97%");
lcd1.print(6, 1, "97%");

}
if (heartRate > 70)
{
lcd.setCursor(6, 1); lcd.print("98%");
lcd1.print(6, 1, "98%");

}
if (heartRate > 80)
{
lcd.setCursor(6, 1); lcd.print("99%");
lcd1.print(6, 1, "99%");

}
if (heartRate >= 150)
{
heartRate = 0;
digitalWrite(buzzer, HIGH);
delay(500);
digitalWrite(buzzer, LOW);
delay(500);
}
//heartRate = 0;
delay(100);

}

void body_temp()
{
//lcd.clear(); s
ensors.requestTemperatures();
bodytemperature = sensors.getTempFByIndex(0);
Serial.print("Body Temperature: ");
Serial.print(bodytemperature);
Serial.println("°F");
lcd.setCursor(11, 0);
lcd.print("Temp");
lcd1.print(11, 0, "Temp");
lcd.setCursor(10, 1);
lcd.print(bodytemperature);
lcd1.print(10, 1, bodytemperature);
lcd.print("F");
lcd1.print(15, 1, "F");
delay (3000);
Blynk.virtualWrite(V9, bodytemperature);
}
```