IOT BASED CONTACTLESS TEMPERATURE SCANNING SYSTEM FOR SMART AND SAFE CAMPUS

B. SAI SHANKAR (19JJ1A0407)

A. CHANDRAVADAN (19JJ1A0401)

MD. ARBAZ (19JJ1A0429)

Department of Electronics and Communication Engineering

Jawaharlal Nehru Technological University Hyderabad
University College of Engineering Jagtial

(Accredited by NAAC with 'A+' Grade)

Nachupally (Kondagattu), Jagtial Dist - 505501, Telangana

2023

IOT BASED CONTACTLESS TEMPERATURE SCANNING SYSTEM FOR SMART AND SAFE CAMPUS

MINI PROJECT REPORT

SUBMITTED IN PARTIAL FULFILMENT

OF THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

BY

B. SAI SHANKAR (19JJ1A0407)

A. CHANDRAVADAN (19JJ1A0401)

MD. ARBAZ (19JJ1A0429)

Department of Electronics and Communication Engineering

Jawaharlal Nehru Technological University Hyderabad
University College of Engineering Jagtial

(Accredited by NAAC with 'A+' Grade)

Nachupally (Kondagattu), Jagtial Dist - 505501, Telangana

2023

Jawaharlal Nehru Technological University Hyderabad
University College of Engineering Jagtial

(Accredited by NAAC with 'A+' Grade)

Nachupally (Kondagattu), Jagtial Dist - 505 501, Telangana

Department of Electronics and Communication Engineering

CERTIFICATE

Date: /12/2022

This is to certify that the project work entitled ***IoT-BASED CONTACTLESS BODY TEMPERATURE SCANNING SYSTEM FOR SMART AND SAFE CAMPUS*** *is a bonafide work carried out by* ***B. SAI SHANKAR, A. CHANDRAVADAN and MD. ARBAZ*** *bearing Roll Nos.* ***19JJ1A0407, 19JJ1A0401, and 19JJ1A0429*** *in partial fulfillment of the requirements for the degree of* ***BACHELOR OF TECHNOLOGY*** *in* ***ELECTRONICS & COMMUNICATION ENGINEERING*** *by the Jawaharlal Nehru Technological University Hyderabad during the academic year 2022-23.*

The results embodied in this report have not been submitted to any other University or Institution for the award of any degree or diploma.

----------	----------
Dr. Dhiraj Sunehra	**Dr. Dhiraj Sunehra**
Professor	**Professor**
Project Guide	**Head of the Department**

Acknowledgements

The success in this project would not have been possible without timely help and guidance by many people. We wish to express our sincere gratitude to all those who have helped and guided us for the completion of the project.

It is our pleasure to thank to our Project Guide and Project Coordinator, **Dr. Dhiraj Sunehra, Professor & Head, Department of ECE,** for the guidance and suggestions throughout the project, without which we would not have been able to complete this project successfully.

We express our sincere gratitude to **Prof. D. Ramesh, Principal and Prof. T. Venugopal, Vice-Principal, JNTUH UCEJ,** for their encouragement and providing facilities to accomplish our project successfully.

We would like to thank our classmates, all faculty members and non-teaching staff of ECE Department, JNTUH UCEJ for their direct and indirect help during the project work.

Finally, we wish to thank our family members and our friends for their interest and assistance that has enabled to complete the project work successfully.

Abstract

In modern world, the IoT is at its peak. Nowadays, everyone is required to check the body temperature and record personal details whenever he arrives at a different place. However, there are some problems when the actions areperformed. Therefore, this project proposed an Internet of Things (IoT)-based contactless body temperature scanning system. This project proposes an IoT-based contactless body temperature scanning system for a smart and safe Campus.

The proposed system helps users to measure their body temperature and recordtheir details without interaction with other people. Besides human resources can be reduced as no worker is needed to check people's body temperature. Furthermore, users can upload their details without using a mobile phone as only an identity card is required.

Keywords: IoT, Node MCU, RFID, Contactless body temperature sensor, Blynk app

List of Figures

List of Tables

CONTENTS

Chapter 1
Introduction

1.1 Introduction

We know that how covid pandemic influenced our lives. In order to face any type of pandemic in near future we need to adapt to the latest technologies. During a pandemic situation, everyone is required to check the body temperature and record personal details whenever they enter the campus. When the security officer checks the body temperature of students using a conventional thermometer, there is a possibility of spread of infection due to physical contact and it takes more time for the measurement. So, in order to avoid physical contact, we can use an IoT-Based Contactless Body Temperature Scanning System for recording attendance with measuring temperature, creating alerts if he/she has abnormal temperature with Buzzer and LED system and upload data to Blynk app.

1.2 Aim of the Project

The aim of this project is to implement a Contactless body temperature scanning system that records the attendance of students with the temperature and alerts us if he has abnormal temperature via buzzer and LED system and uploads data to Blynk app.
The objectives of the project are as follows:

1. To understand the architecture of the Node MCU Board and salient features of Arduino IDE and Blynk IoT App.
2. To interface the various components Contactless Temperature sensor, EM18 Reader module, Liquid Crystal Display [LCD], LED, Buzzer to the Node MCU board.
3. To develop the necessary code using Embedded C and the Arduino IDE software tool.
4. To test the system for satisfactory operation.

1.3 Methodology

The project is aimed at implementation of an IoT based contactless body temperature scanning system for a smart and safe campus. This device helps to automate the

attendance registry and temperature check at entrance of campus. The architecture of the Node MCU is understood including the ports and pins. The written code is uploaded to the Node MCU board. The contactless temperature sensor, EM18 Reader module, Liquid Crystal Display, buzzer and LED are interfaced with the Node MCU Board. The system is tested for satisfactory results.

1.4 Significance of the work

In this project we attempt to develop a device which can be fixed at the main entrance of campus to ensure their safety of every student and staff from ill students who are main cause of virus transmissions . It has the following advantages:

- Easy to operate.
- Economical
- Less power consumption.
- Efficient
- Provides the real-time temperature of the student as well as details of the student via RFID card. The details will only be visible to those who have Blynk app credentials.
- Can also help in alerting the abnormal temperature through the buzzer and LED system.

1.5 Organization of the thesis

This thesis is divided into six chapters including introduction and conclusions. The block diagram, features, pin diagram and other functional units of Arduino Board are explained in Chapter 2. The description of various hardware components and the software used in the project is explained in chapter 3 and chapter 4. The schematic diagram, flowchart, experimental setup and results are discussed in chapter 5.

Chapter 2
Description of Arduino Uno

2.1 Introduction

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. The Arduino generally takes the input from various sensors. The sensors presently available in the market may be analog or digital in nature. It processes the input and gives output as programmed by the user. The Arduino is the heart for processing data in a large number of applications designed to meet basic needs. Hence, it is the CPU for various applications. An Arduino board is preferred as it is an open-source. The main motive of the designers of this board was that they want to provide the beginners with a simplified programming environment that even a student with least programming skills should easily deal with the Arduino. The basic microcontroller of the AVR family is selected for design. The controller is of 8-bit to which they have provided various connections and added some components and integrated into a board called Arduino. The arduino boards are designed from smaller boards with 8-bit controllers to a board which can support the Internet of things. Every Arduino board comes with a preloaded boot loader.

2.2 History of Arduino

The Arduino project started at the “Interaction design institute Ivrea” in Ivrea, Italy. At that time, the students used a “Basic stamp” microcontroller at a cost of $100, a considerable expense for many students. In 2003, Hernando Barragan created the development platform “Wiring” as a Master’s thesis project at IDII, under the supervision of Massimo Banzi and Casey Reas, who are known for their work on the “Processing” language. The project goal was to create simple, low-cost tools for creating digital projects by non-engineers. The Wiring platform consisted of a printed circuit board (PCB) with an ATmega168 microcontroller, an IDE based on processing

and library functions to easily program the microcontroller. In 2003, Massimo Banzi, with David Mellis, another IDII student, and David Cuartielles, added support for the cheaper ATmega8 microcontroller to Wiring. But instead of continuing the work on Wiring, they forked the project and renamed it ARDUINO.

2.3 Types of Arduino

Different Arduino boards available in the market, with wide specifications. Each board uses different kinds of microcontroller with different operating voltages, CPU speed, RAM size and different number of analog and digital inputs. Some of the Arduino boards are given below:

2.3.1 Arduino Uno

Arduino Uno is the most preferred board among all the Arduino boards. The processor used is ATmega328P. Its operating and input voltage are 5V and 7 to 12 V respectively, 16MHz clock frequency, 6 analog and 14 digital IO, 1 kB EEPROM, 32 kB Flash, a regular USB and 1 UART.

2.3.2 Arduino Due

First ARM based microcontroller Arduino board is the Arduino Due. It is a 32-bit microcontroller based Cortex-M3 Central Processing Unit. It is a very powerful microcontroller with 54 digital I/O, 12 analog inputs, 2 analog outputs and 4 UART serial interfaces. If we give more than 5V power supply to the Arduino, it will damage. It is 32-bit ARM core, it has 512KB of flash memory, 96KB SRAM in which two banks are there and they are 64KB and 32KB.

2.3.3 Arduino Mega 2560 Model

The Arduino Mega 2560 model is an advanced version of the previous existing model Arduino Due. It has 256 KB of RAM memory and 54 input output pins. In that 16 pins are used as analog pins and 14 pins are used for Pulse Wave Modulation (PWM).

2.3.4 Arduino Uno (R3)

The Uno is a huge option for your initial Arduino. This Arduino board depends on an ATmega328P based microcontroller. It consists of 14-digital I/O pins, where 6-

pins can be used as PWM output pins, 6-analog inputs, a reset button, a power jack, a USB connection, an In-Circuit Serial Programming header (ICSP), etc.

2.3.5 Arduino Leonardo Board

The first development board of an Arduino is the Leonardo board. This board uses one microcontroller along with the USB. That means, it can be very simple and cheap also. Because this board handles USB directly, program libraries are obtainable which let the Arduino board follow a keyboard of the computer, mouse, etc.

2.4 Arduino Uno

Arduino Uno, which is a small board but useful for many applications.

2.4.1 Features of Uno

Table 2.1 provides the salient features of Arduino Uno board.

Table 2.1 Features of Arduino UNO

S.No	Specifications	Value
1	Microcontroller	ATmega328P
2	Operating Voltage	5V
3	Input Voltage	7V to 12V
4	Digital Pins	14 (6 PWM)
5	PWM Digital I/O Pins	6
6	Analog Input Pins	6
7	DC Current (3.3V Pin)	50 mA
8	Flash Memory	32KB
9	SRAM	2KB
10	EEPROM	1KB
11	Clock Speed	16MHz
12	LED BUILTIN	Pin 13
13	Length X Width	68.6 mm X 53.4mm
14	Weight	25g

2.4.2 Description of Arduino UNO

Figure 2.1 shows the layout diagram of the Arduino Uno board showing various components.

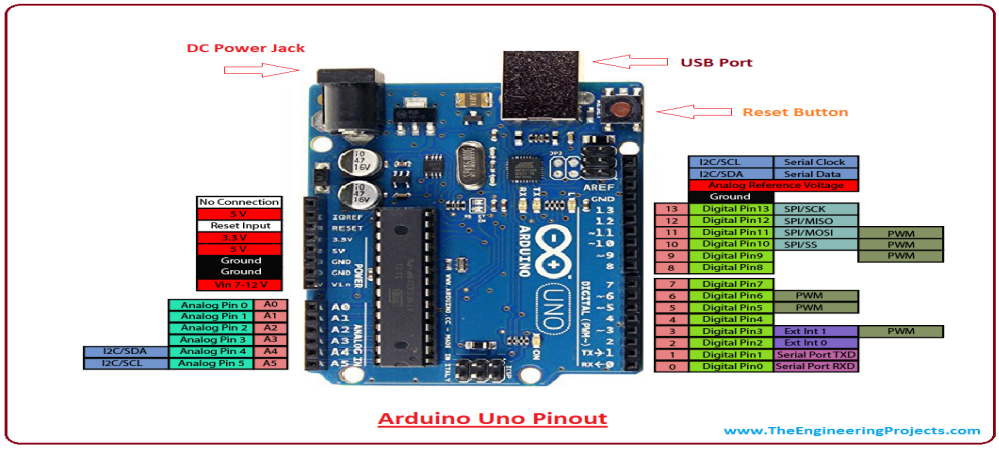


Fig. 2.1 Arduino Board Top view

Arduino Board consists of the following**:**

Power:

Arduino is powered by using a USB port of computer or by using a DC adapter or by a barrel jack fixed to the board.

Pins:

The connections to the Arduino can be made through the pins. The black plastic 'headers' are used to plug and connect a wire into the board. The functions vary from pin to pin.

GND: GND means 'Ground'. There are sufficient GND pins to connect more modules to the Arduino board.

5V & 3.3V: The power supply to the modules is the 5V and 3.3V pins. 5V pin supplies power at 5 volts and the 3.3V pin supplies power at 3.3 volts. In general, most of the components work at the specified voltages given above.

Analog: A0 to A5 pins are analog. The input to these pins is analog in nature. These pins receive the data from an analog sensor. The smoke sensor and temperature sensor gives analog values. There is an ADC to convert it into a digital value that will be processed by the controller.

Digital: Pins 0 to 13 are digital. These pins take the digital input from the sensor and give digital output to the load.

PWM:

The tilde (~) is shown adjacent to some of the digital pins on the Arduino board. These pins can be used as Pulse-Width Modulation pins and also as digital pins.

AREF (Analog Reference):

It acts as an external reference voltage.

Reset Button:

When Reset is pressed, that pin is grounded and restarts the code loaded to Arduino.

Power LED Indicator:

When a power source is connected, the LED turns ON. If the LED doesn't turn on, the connections need to be checked. If the connections are good then the board might have been damaged.

Tx Rx LEDs:

The Tx (transmitter) and Rx (receiver) LEDs will glow whenever there is data transfer between Arduino and PC. It also glows for other serial communication data transfer.

Voltage Regulator:

The voltage regulator controls the fluctuations in the voltage that is supplied to the Arduino board.

Main IC:

The main IC on the Arduino is from the ATMEL Company, usually ATmega series.

- 512 KB SRAM.
- Software Security.

2.4.3 Pin Description

The pin diagram or pin layout of Arduino board is shown in Fig. 2.3.

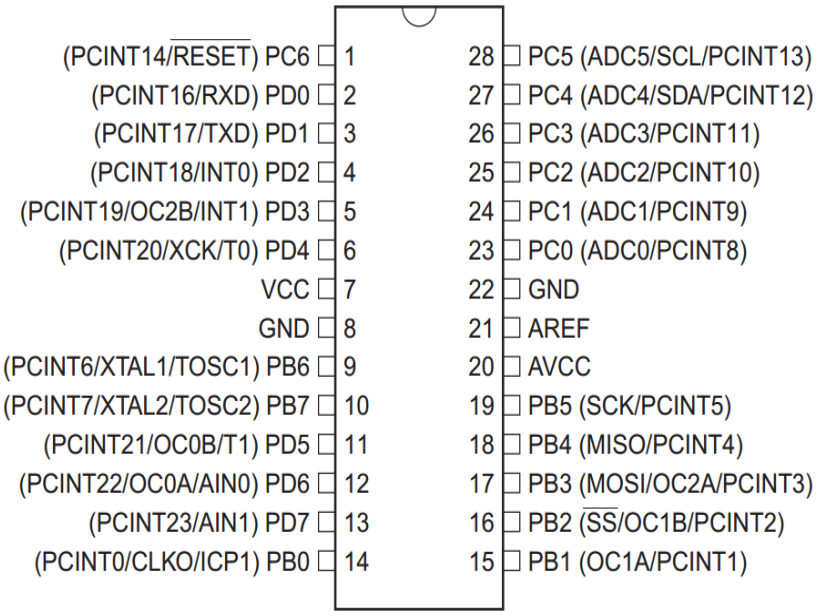


Fig.2.2 Pin diagram of Arduino board

V_{cc}: Power source.

GND: The pin GND is used to connect the Ground.

Port B (PB-7, PB-6):

PB7 and PB6 are used as TOSC2 and TOSC1 input for Asynchronous Timer. The Port B is used as a system clock.

Port C (PC-5:0):

Port C has 7-bit bidirectional input and output port. Pull up registers also consisted in the port C. The PC5 and PC0 outputs have some similar characteristics. The sink capability of the source is more.

RESET: To reset the system PC-6 is used as a reset button or pin.

Port D (PD-7: PD-0):

The Port D consisted pull up registers for output and pulled low for inputs. It has an 8-bit bi directional input output port. It consists of pull-up resistors. Port D pins are externally pulled low for inputs. Then pull up resistors were active. These pins are in tristate even though the clock is not running mode.

AV$_{CC}$: For Analog to Digital converter, we are using the AVcc Converter. For the external connection to V_{cc}, PC-3 pins are used to connect. As well as PC-0, ADC-7 and ADC-6 are also used to connect to the V_{cc}.

A$_{REF}$: A_{REF} means Analog Reference.

ADC-7, ADC-6:

ADC7, ADC6 (For external supply) pins are used as input pins while the ADC process is going on.

2.5 Conclusions

In this chapter the history of arduino and salient features of different types of Arduino boards are explained. The architecture of Arduino Uno board is discussed along with its pin configuration.

Chapter 3
Hardware Description

3.1 Introduction

The microcontroller (Node MCU) is programmed in a way that it can interface with the modules like contactless temperature sensor, EM18 Reader, LED, buzzer and LCD. Upon receiving the valid details from RFID tag, the sensors read the values and give it to the Node MCU which in turn transmits to the Wi-fi module so that it can send the details to Blynk app via Wi-fi protocol. Also, the RED LED emits the light and buzzer sounds only on receiving the abnormal temperatures. The temperature data and personal details of user will be received by sensor and RFID scanner and transmitted to respective components via Node MCU.

3.2 Block Diagram

Figure 3.1 shows the block diagram of the project.

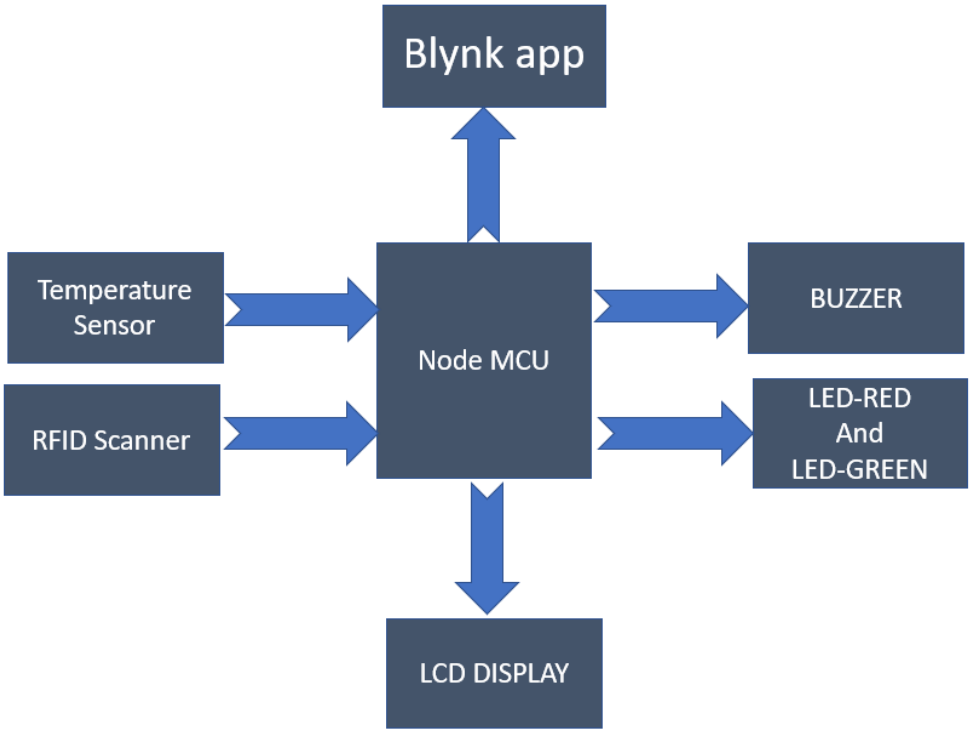


Fig.3.1 Block Diagram of the Project

Figure 3.1 represents the block diagram which consists of a contactless temperature sensor , EM18 Reader module, ie RFID Scanner, LCD Display, LED, buzzer, power

supply, and a NODE MCU board. The EM18 Reader module both have a one-way communication i.e., transmitting data and NODE MCU board has a one way communication established with the other components to send the commands or to receive the inputs from sensors. The NODE MCU board gets power from external power supply where as the components get power supply from board.

3.3 Node MCU

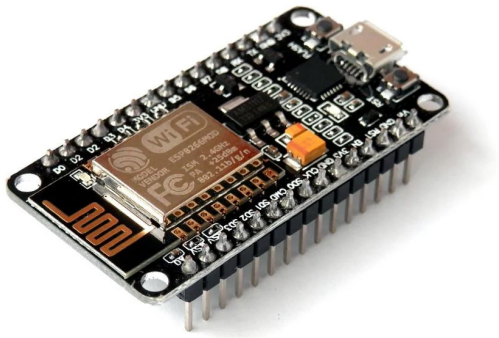


Fig 3.2: Node MCU

3.3.1 Features

- Open Source & Arduino like hardware.
- Micro USB port & Reset/Flash buttons.
- Interactive, Programmable & Low cost.
- ESP8266 with inbuilt Wi-Fi.
- USB to UART converter and Status LED.
- Voltage: 3.3V
- Current consumption: 10uA~170mA
- Flash memory attachable: 16MB max (512K normal)
- Integrated TCP/IP protocol stack.
- Processor: Tensilica L106 32-bit.
- Processor speed: 80~160MHz
- RAM: 32K+ 80K
- Maximum concurrent TCP connections: 5
- GPIOS: 17 (multiplexed with other functions). Analog to Digital: 1 input with 1024 step resolution.

3.3.2 Description

Node MCU is an open-source firmware and development kit that plays a vital role in designing an IoT product using a few script lines. Multiple GPIO pins on the board allow us to connect the board with other peripherals and are capable of generating PWM, 12C, SPI, and UART serial communications.

The interface of the module is mainly divided into two parts:

- Firmware – Runs on the ESP8266 Wi-Fi SoC.
- Hardware – Based on the ESP-12 Module

Open-source firmware allows us to edit, modify and rebuild the existing module and keep changing the entire interface until you succeed in optimizing the module as per your requirements.

- USB to UART converter is added on the module that helps in converting USB data to UART data which mainly understands the language of serial communication.
- Instead of the regular USB port, Micro USB port is included in the module that connects it with the computer for dual purposes: programming and powering up the board.
- The board incorporates status LED that blinks and turns off immediately, giving you the current status of the module if it is running properly when connected with the computer.
- The ability of module to establish a flawless Wi-Fi connection between two channels makes it an ideal choice for incorporating it with other embedded devices like Raspberry Pi.

3.3.3 Pin Configuration

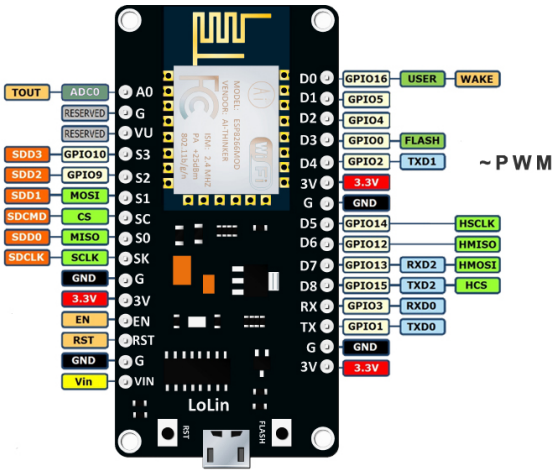


Fig 3.3: Pin Configuration of Node MCU

The Pin configuration of Node MCU is as follows:

1) Power Pins: There are four power pins namely - V_{in} pin and three 3.3V pins.

- The V_{in} pin can be used to directly supply ESP8266 and its components if you have a controlled 5V voltage source.
- The 3.3V pins are the output of the voltage board controller. These pins can be used to supply power to external parts.

2) GND (Ground): It is the ground pin of the ESP8266 Node MCU development board.

3) I2C Pins: These are used to integrate all types of I2C sensors and parameters in the project. Both I2C Master and I2C Slave are supported. The clock frequency is 100 kHz at maximum speed. It should be noted that the frequency of the I2C clock should be greater than the frequency of the slowest clock of the slave device.

4) GPIO Pins: The ESP8266 Node MCU has 17 GPIO pins that can be assigned to various functions such as I2C, I2S, UART, PWM, IR Remote Control, LED Light, and Button respectively. Each GPIO digitally-enabled can be adjusted to internal drag or drop or set to high intensity. When set as input, it can also be set to Edge-trigger or level-trigger to produce CPU interference.

5) ADC channel: NodeMCU is embedded 10-bit with SAR ADC accuracy. These two functions can be performed using the ADC viz. V_{DD} 3P3 pin power supply and T_{out} pin power input. However, they cannot be used simultaneously.

6) UART: The ESP8266 NodeMCU has 2 UART domains, namely UART0 and UART1, which offer different connections (RS232 and RS485), and can communicate

up to 4.5 Mbps. UART0 pins (TXD0, RXD0, RST0 & CTS0) can be used for communication. However, the UART1 (TXD1 pin) includes a data transfer signal only, so it is used to print the log.

7) SPI Pins: ESP8266 incorporates two SPIs (SPI and HSPI) into slave and master modes. These SPIs also support the following SPI features:

- 4-time modes for SPI format transfer
- Up to 80 MHz with split clocks of 80 MHz
- Up to 64-Byte FIFO

8) SDIO Pins: ESP8266 incorporates the Secure Digital Input / Output Interface (SDIO) which is used to connect directly to SD cards. 4-bit 25 MHz SDIO v1.1 and 4-bit 50 MHz SDIO v2.0 are supported.

9) PWM Pins: The board has 4 Pulse Width Modulation (PWM) channels. PWM output can be programmed and used to drive digital engines and LEDs. The frequency range of PWM ranges from 1000 μs to 10000 μs, eg between 100 Hz and 1 kHz.

10) Control Pins: These are used to control ESP8266. These anchors include Chip Enable pin (EN), Reset pin (RST), and WAKE pin.

- **EN Pin (Enable):** The ESP8266 chip is enabled when the EN pin is pulled INSIDE. When pulled LOW the chip works at low power.
- **RST Pin (Reset):** The RST pin is used to reset the ESP8266 chip.
- **Wake Pin:** The use of Wake pin is used to wake up the chip from a deep sleep.

3.4 EM18 Reader Module

Figure 3.5 shows the EM 18 Reader Module (Top View).

Fig 3.4: EM 18 Reader Module

3.4.1 Specifications and Features of EM-18 RFID Reader Module :-

- Operating Voltage: 5V DC Supply
- Reading Distance: 6-10 cm.
- Read frequency: 125 kHz.
- EM4001 64 – bit RFID tag compatible.
- 9600bps ASCII output.
- Current : <50 mA
- Operating Frequency : 125 Khz
- Read Distance : 5 cm
- Compatible Tags : 125KHz EM4100 Tags
- Size : 32mm(length) * 32mm(width) * 8mm(height)

3.4.2 Description

- Radio frequency Identification (RFID) is a wireless identification technology that uses radio waves to identify the presence of RFID tags.
- Just like Bar code reader, RFID technology is used for identification of people, object etc. presence.
- In barcode technology, we need to optically scan the barcode by keeping it in front of reader, whereas in RFID technology we just need to bring RFID tags in range of readers. Also, barcodes can get damaged or unreadable, which is not in the case for most of the RFID.
- RFID is used in many applications like attendance system in which every person will have their separate RFID tag which will help identify person and their attendance.
- RFID is used in many companies to provide access to their authorized employees.
- It is also helpful to keep track of goods and in automated toll collection system on highway by embedding Tag (having unique ID) on them.

RFID based system has two basic elements

1. **RFID Tag:**

RFID tag includes microchip with radio antenna mounted on substrate which carries 12 Byte unique Identification number.

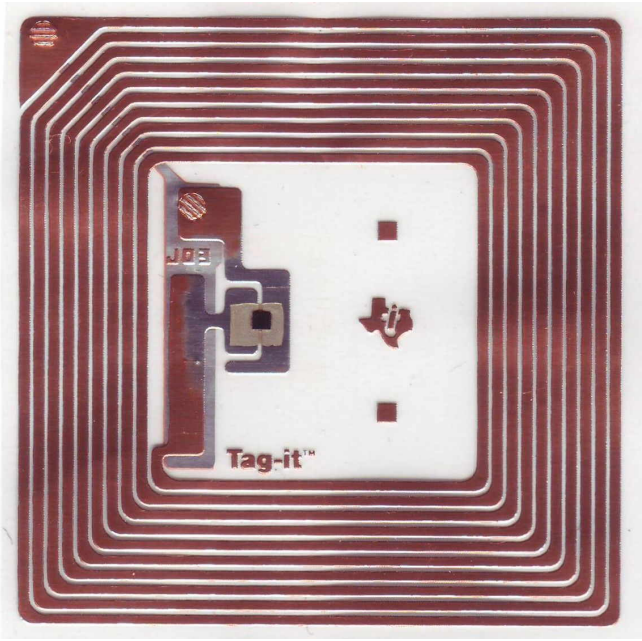


Fig 3.5:RFID Tag Inside

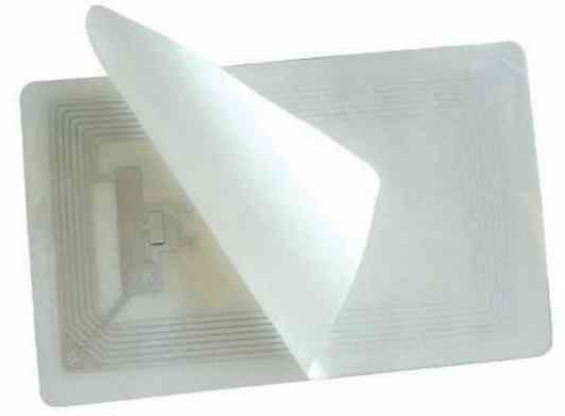


Fig 3.6:RFID Tag

2. RFID Reader:

It is used to read unique ID from RFID tags. Whenever RFID tags comes in range, RFID reader reads its unique ID and transmits it serially to the microcontroller or PC. RFID reader has transceiver and an antenna mounted on it. It is mostly fixed in stationary position.

Fig 3.7: EM18 RFID Reader Module

How RFID system works?

RFID Reader has transceiver which generates a radio signal and transmits it through antenna. This signal itself is in the form of energy which is used to activate and power the tag.

When RFID tag comes in range of signal transmitted by the reader, transponder in the tag is hit by this signal. A tag draws power from the electromagnetic field created by reader. Then, the transponder converts that radio signal into the usable power. After getting power, transponder sends all the information it has stored in it, such as unique ID to the RFID reader in the form of RF signal. Then, RFID reader puts this unique ID data in the form of byte on serial Tx (transmit) pin. This data can be used or accessed by PC or microcontroller serially using UART communication.

When we bring RFID tag near to the reader, it reads the data from the tag and immediately transmits 12-byte unique ID on the serial port.

12 Byte Unique Identification Number

RFID Tags contain 12 Byte unique ID.

1st 10-Byte Data	**Last 2-Byte**
Tag no.	Checksum

These checksum bytes will be the XOR result of 1st 10 bytes, i.e. Tag no.

3.4.3 Pin Configuration

Figure 3.6 shows the pin details of the EM 18 Reader Module.

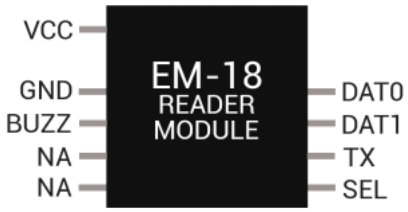


Fig.3.8 Pin Configuration of EM 18 Reader Module

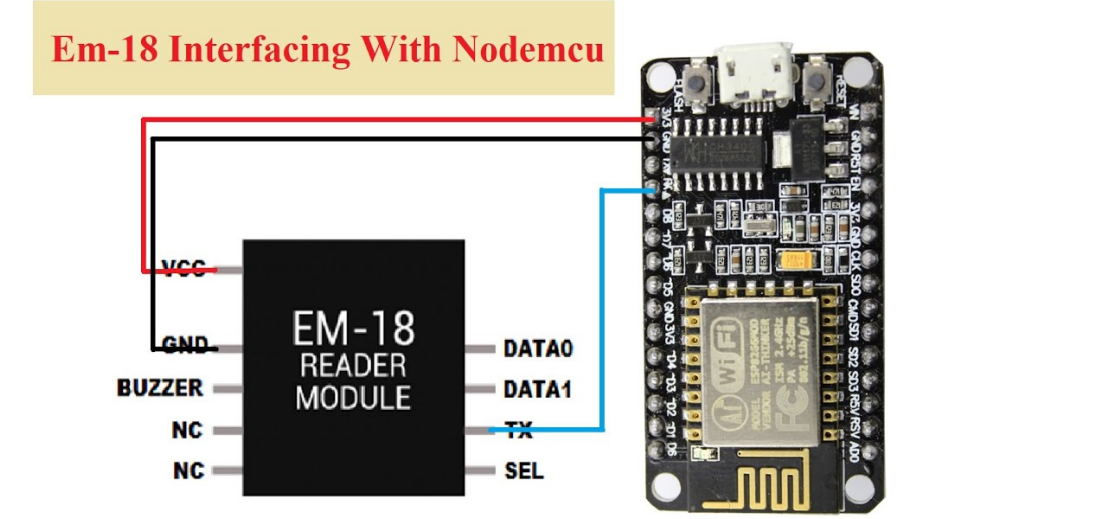
PIN NAME	DESCRIPTION
VCC	Should be connected to positive of power source
GND	Should be connected to ground
BUZZ	Should be connected to buzzer
NC	No Connection
SEL	SEL=1 then o/p = RS232 SEL=0 them o/p=WEIGAND
TX	DATA is given out through TX of RS232
DATA1	WEIGAND interface DATA HIGH pin
DATA0	WEIGAND interface DATA LOW pin

Table 3.1:Pin Configuration of EM18 Reader Module

3.4.4 Interfacing

Figure 3.7 shows the interfacing of Node MCU with EM 18 Reader module.

Fig.3.9 Interfacing with EM18 Reader Module



The connection for interfacing of a EM 18 Reader Module with NODE MCU is easy

- Connect the ground pin on the Node MCU board to the ground pin of the EM Reader module.
- Connect the receiver pin on the Node MCU board to the transmitter pin of the EM 18 Reader module.

3.5 Liquid Crystal Display (LCD) 2x16:

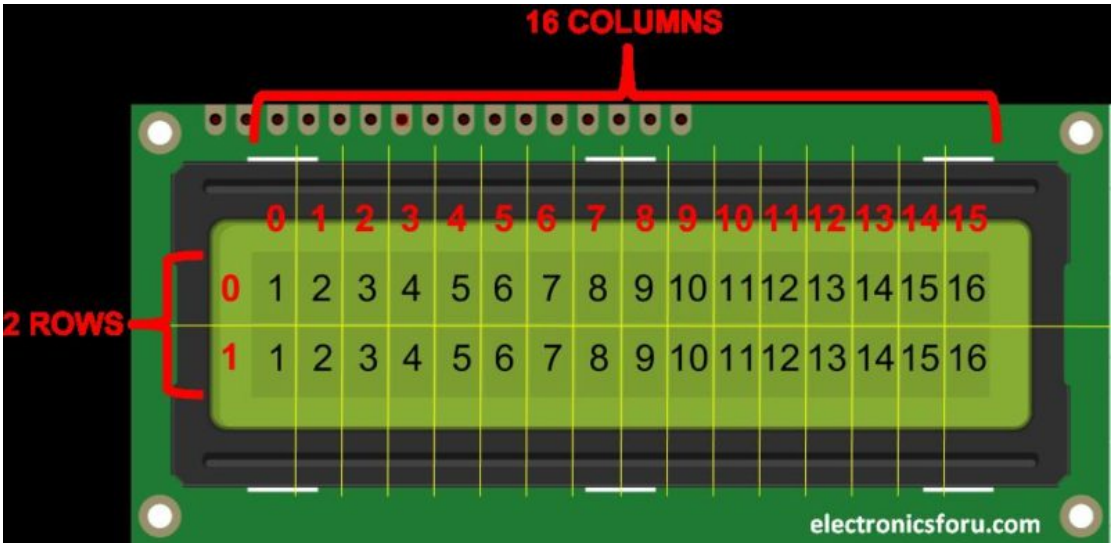


Fig.3.10 Liquid Crystal Display [2x16]

Features of 16×2 LCD module

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

3.5.2 Description

Alphanumeric displays are used in a wide range of applications, including palmtop computers, word processors, photocopiers, point of sale terminals, medical instruments, cellular phones, etc. The 16 x 2 intelligent alphanumeric dot matrix display is capable of displaying 224 different characters and symbols. A full list of the characters and symbols is printed on pages 7/8 (note these symbols can vary between brand of LCD used). This booklet provides all the technical specifications for connecting the unit, which requires a single power supply (+5V).

3.5.3 Pin Configuration

Figure 3.9 shows the pin description of LCD [2x16] Display.

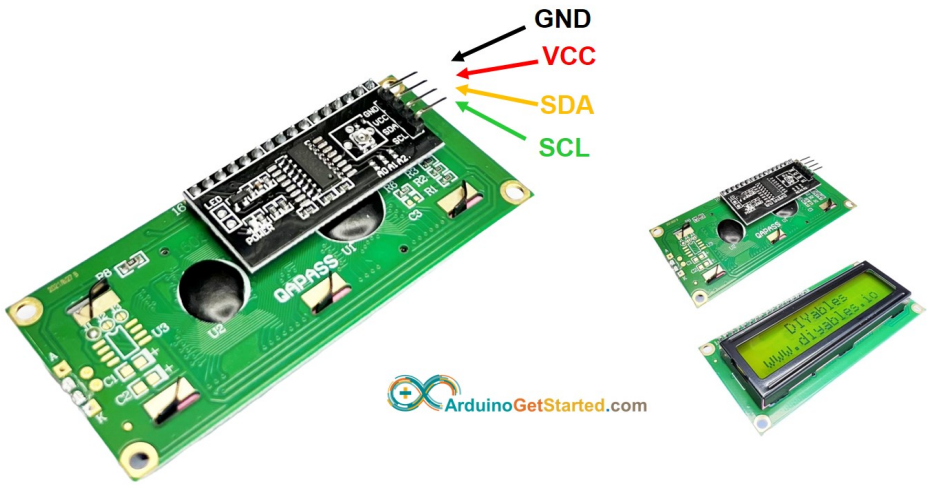


Fig.3.11 Pin Configuration of LCD Display

Pin Name	Description
GND	Needs to be connected to ground 0v
VCC	The power supply for LCD needs to be connected to 5v
SDA	I2C Data Signal
SCL	I2C Clock Signal

Table 3.2 : Pin Configuration of LCD Display

3.5.4 Interfacing

Figure 3.12 shows the interfacing of LCD Display with Node MCU.

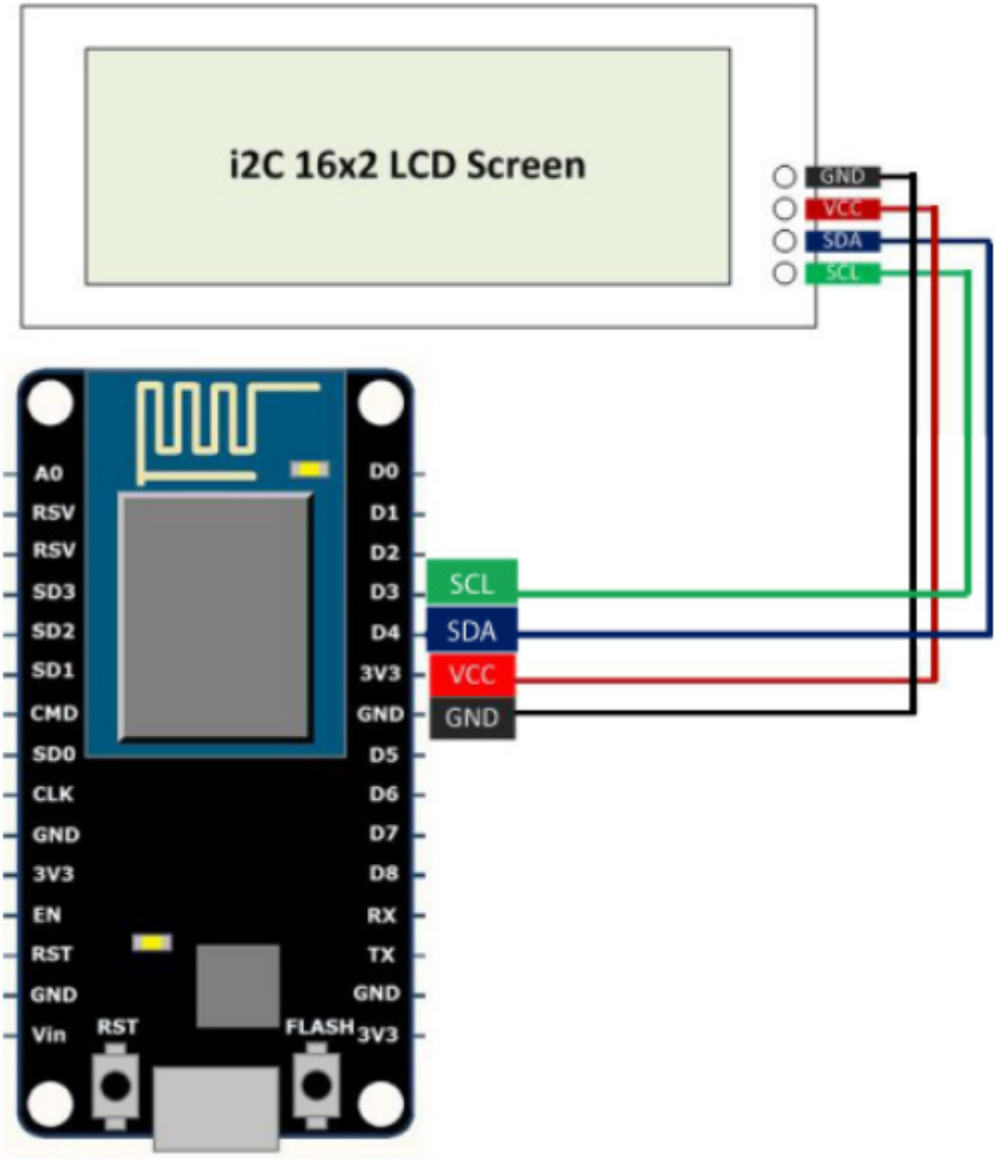


Fig.3.12 Interfacing with LCD Display

The connection for interfacing of the LCD with Node MCU board is very easy.

1)Connect the ground pin of the Node MCU with GND pin of I2C LCD.

2)Connect the Vcc pin of I2C LCD with Node MCU 3.3v pin.

3)Connect Serial Clock (SCL) pin of LCD with Node MCU D3 pin.

4)Connect Serial Data (SDA) pin of LCD with Node MCU D4 pin

3.6 Contactless Temperature Sensor [MLX90614]

Figure 3.11 shows the Contactless temperature Sensor used in the project.

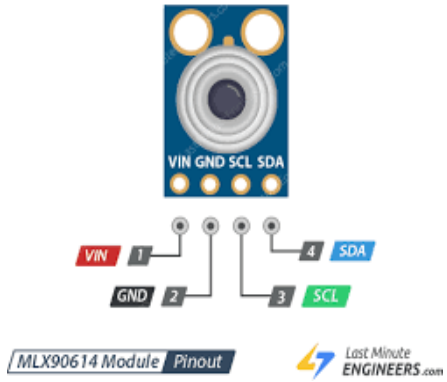


Fig.3.13 Contactless Temperature Sensor [4486]

3.6.1 Features

- Operating Voltage: 3.6V to 5V (available in 3V and 5V version)
- Supply Current: 1.5mA
- Object Temperature Range: -70° C to 382.2°C
- Ambient Temperature Range: -40° C to 125°C
- Accuracy: 0.02°C
- Field of View: 80°
- Distance between object and sensor: 2cm-5cm (approx.)

3.4.2 Description

The MLX90614 is a **Contactless Infrared (IR) Digital Temperature Sensor** that can be used to measure the temperature of a particular object ranging from -70° C to 382.2°C. The sensor uses IR rays to measure the temperature of the object without any physical contact and communicates to the microcontroller using the I2C protocol. The MLX90614 sensor can measure the temperature of an object without any physical contact with it. This is made possible with a law called Stefan-Boltzmann Law, which states that all objects and living beings emit IR Energy and the intensity of this emitted IR energy will be directly proportional to the temperature of that object or living being. The key feature of MLX90614 is that it is a contactless IR temperature sensor with high accuracy. So it can be used in industries to measure the temperature

of moving objects like a rotating motor shaft. Due to its high accuracy and precision, it is also used in a wide range of commercial, health care, and household applications like room temperature monitoring, body temperature measurement, etc.

3.6.3 Pin Configuration

Figure 3.14 shows the pin description of contactless temperature sensor.

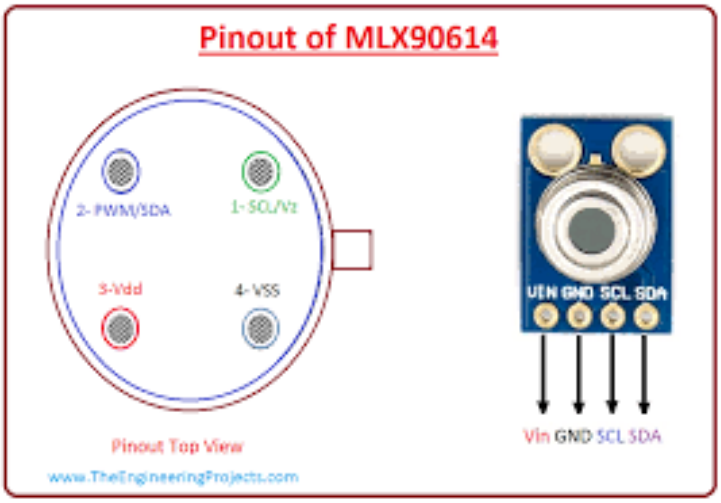


Fig.3.14 Pin description of contactless temperature Sensor

Pin No.	Pin Name	Description
1	Vin	Vin can be used to power the sensor, typically 5v
2	GND	The metal can also ground
3	SDA-Serial Data	Serial data pin used I2C communication
4	SCL-Serial Clock	Serial Clock Pin used for I2C Communication

Table 3.3:Pin Configuration of MLX90614

3.6.4 Interfacing

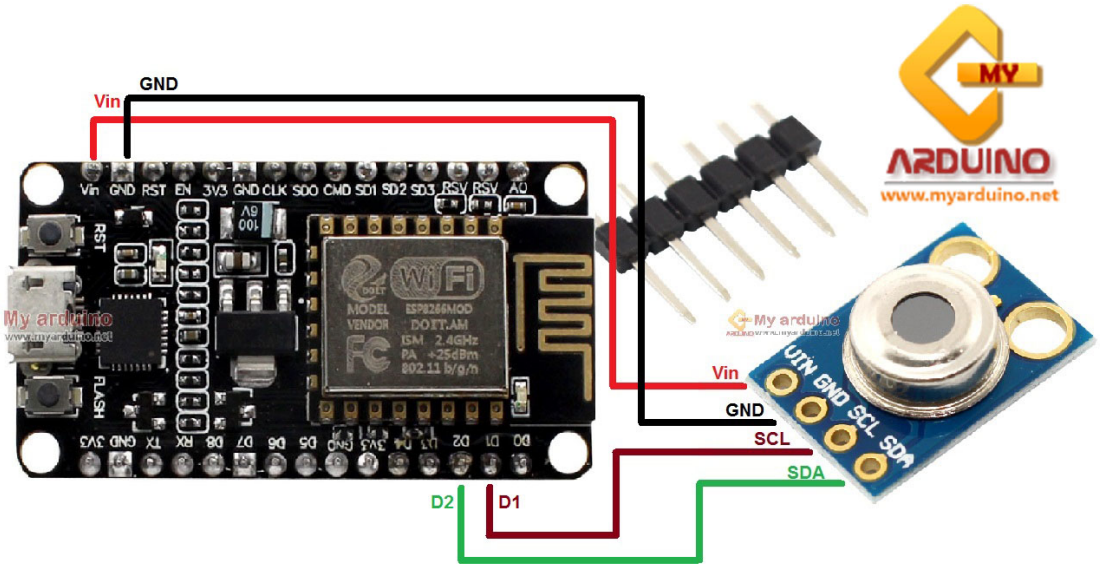


Fig.3.15 Interfacing with Contactless temperature Sensor

The connection for interfacing of the MLX90614 with Node MCU board is very easy.

1)Connect the ground pin of the Node MCU with GND pin of MLX90614.

2)Connect the Vcc pin of sensor with Node MCU 3.3v pin.

3)Connect Serial Clock (SCL) pin of sensor with Node MCU D1 pin.

4)Connect Serial Data (SDA) pin of sensor with Node MCU D2 pin.

3.7 Light Emitting Diode

An LED is a semiconductor Light source that emits light when current flows through it. The different semiconductor materials and different impurities result in different colors of light from the LED.

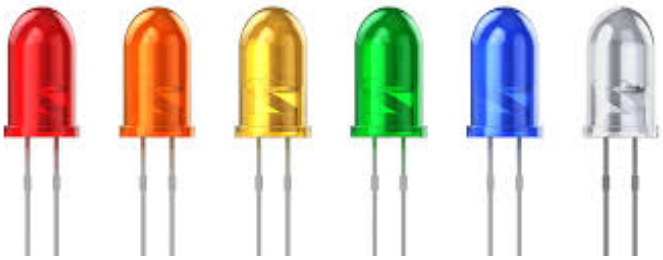


Fig.3.16 Light Emitting Diodes

Table 3.4 Typical LED Characteristics

Typical LED Characteristics			
Semiconductor Material	Wavelength	Colour	V_F @ 20mA
GaAs	850-940nm	Infra-Red	1.2v
GaAsP	630-660nm	Red	1.8v
GaAsP	605-620nm	Amber	2.0v
GaAsP:N	585-595nm	Yellow	2.2v
AlGaP	550-570nm	Green	3.5v
SiC	430-505nm	Blue	3.6v
GaInN	450nm	White	4.0v

3.8 Buzzer

Figure 3.15 shows the Buzzer.

Fig.3.17 Buzzer

A buzzer is a device that provides an audio signal in a circuit when a voltage is applied to it. It comes in many different forms which include Mechanical, Electromechanical, and Piezoelectric. Having the ability to produce sound in

electronics has many different applications. The buzzer is an electronic component that has the ability to produce sound when a voltage is applied to it. This can be utilised in many applications. One of the biggest applications where a buzzer plays a crucial role is a smoke alarm. When you are sleeping your sense of smell and sight are impaired. If a fire starts in your home while you are sleeping, you will have no way of knowing. A buzzer provides a way of letting you know a fire has started through sound. Other applications include alarm clocks, doorbells, electronic musical instruments, greeting cards, security systems and many more applications. So, a buzzer's main function in a circuit is to provide an audio signal. Buzzers come in a variety of construction, size, and specifications. Different types and sizes of buzzers are used for different applications. Based on construction, there are the following kinds of buzzers:

1. Piezoelectric buzzers
2. Magnetic buzzers
3. Electromagnetic buzzers
4. Mechanical buzzers
5. Electromechanical buzzers

3.9 Power supply

The power requirements of various components involved in this project are with the help of the following three components

1. Lithium-ion battery charger.
2. Bridge Rectifier
3. Voltage Regulator
4. LED
5. Capacitor

Lithium-ion battery charger

A Lithium-ion or Li-ion battery is a type of rechargeable charge which uses the reversible reduction of lithium ion to store energy. It is the predominant battery type used in portable consumer electronics and electric vehicles. It also sees significant use for grid-scale energy storage and military and aerospace applications. Compared to other rechargeable battery technologies, Li-ion batteries have high energy densities, low self-discharge, and no memory effect (although a small memory effect reported in LFP cells has been traced to poorly made cells).

Fig 3.18 Lithium-ion Battery

Bridge Rectifier

The bridge rectifier circuit is made of four diodes D_1, D_2, D_3, D_4, and a load resistor R_L. The four diodes are connected in a closed-loop configuration to efficiently convert the alternating current (AC) into Direct Current (DC). The main advantage of this configuration is the absence of the expensive center-tapped transformer. Therefore, the size and cost are reduced. ridge rectifier is a type of full-wave rectifier that uses four or more diodes in a bridge circuit configuration to efficiently convert alternating (AC) current to a direct (DC) current.

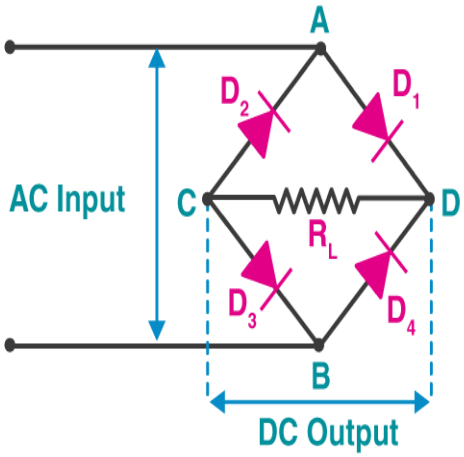


Fig 3.19 Bridge Rectifier

Voltage Regulators LM7805

The MC78XX/LM78XX/MC78XXA are a series of three terminal positive voltage regulators. This series of fixed-voltage integrated-circuit voltage regulators are designed for a wide range of applications such as on-card regulation for elimination of noise and distribution problems associated with single-point regulation. Each of these regulators can deliver up to 1.5 A of output current. The internal current-limiting and thermal-shutdown features of these regulators essentially make them immune to overload. In addition to use as fixed-voltage regulators, these devices can be used with external components to obtain adjustable output voltages and currents, and also can be used as the power-pass element in precision regulators.

Features of these voltage regulators:

- Output Current up to 1A
- Output Voltages of 5, 6, 8, 9, 10, 12, 15, 18, 24V
- Thermal Overload Protection
- Short Circuit Protection
- Output Transistor Safe Operating Area Protection

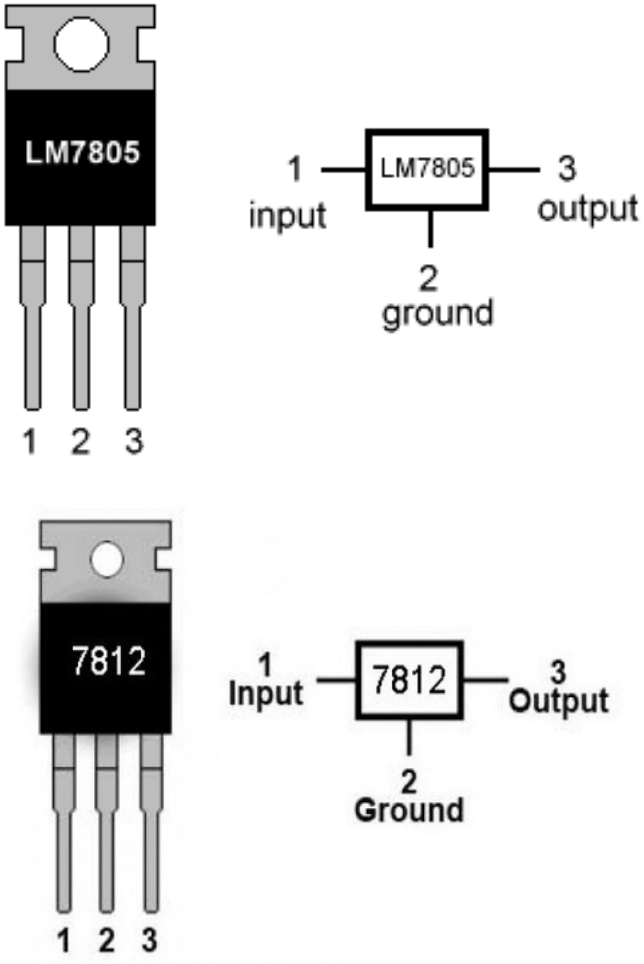


Fig 3.20 Voltage Regulators

Capacitor

The capacitor is an electric component that has the ability to store energy in the form of electrical charges that creates a potential difference, which is a static voltage, much like a small rechargeable battery.

The most basic design of a capacitor consists of two parallel conductors (Metallic plate), separated with a dielectric material. When a voltage source is attached across the capacitor, the capacitor plate gets charged up. The metallic plate attached to the positive terminal will be positively charged, and the plate attached to the negative terminal will be negatively charged.

Fig 3.21 Capacitor

3.10 Conclusion

In this chapter different hardware modules are discussed and their interfacing is explained.

Chapter 4

Software Tools

4.1 Introduction

The software tool which is used in this project is the Arduino IDE. In Arduino IDE, project code is written to respond to the abnormal temperatures and to identify identity using RFID cards.

4.2 Arduino IDE

The Node MCU connects to the computer via USB cable. The Arduino IDE is user friendly where code can be written and various toolbars are available for easy programming. Code can be dumped into Node MCU using this software. Arduino IDE (Integrated Development Environment) is used to write code to interface different modules with an Arduino board i.e., Node MCU. Arduino IDE tool sketch is shown in Fig. 4.1.

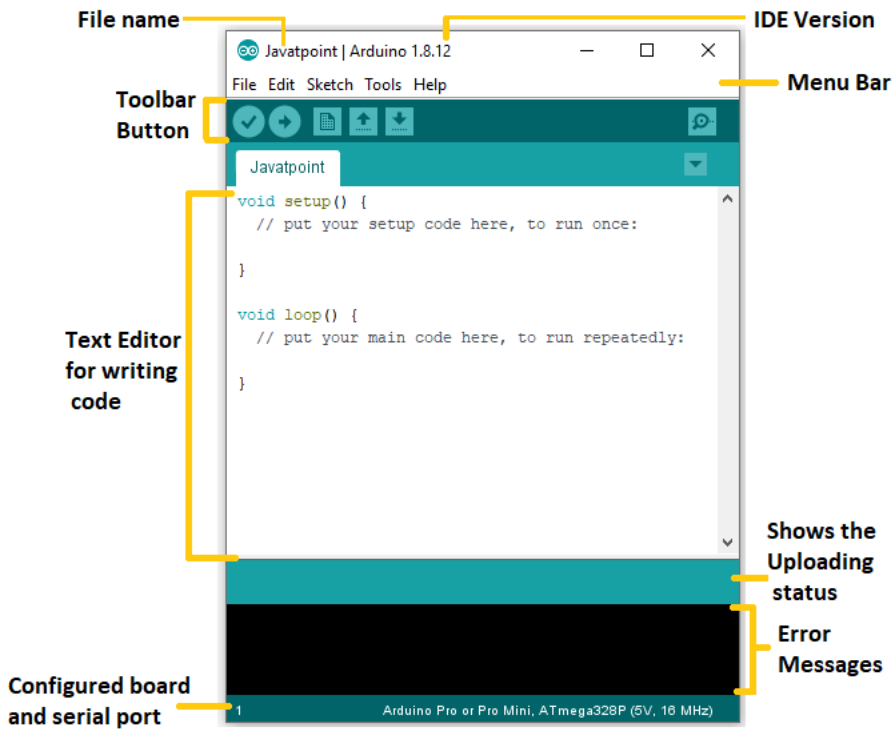


Fig. 4.1 Arduino IDE tool sketch

Steps to follow for Arduino programming IDE:

Step 1: Choose a suitable Arduino board and connect it to pc using USB cable.

Step 2: Arduino IDE Software should be downloaded from the website and install it.

Step 3: Provide power supply.

Most Arduino boards have USB as a power source. Adaptor can also be a choice. The power LED named PWR glows on the power supply.

Step 4: Start Arduino IDE.

Open the installed software by double click.

Step 5: Initiate a project.

It can be done in two ways:

- A new project can be designed.
- An existing project can be viewed.

For a new project, we need to select File, select New from the options and write a program.

Step 6: Type of Arduino board needs to be selected.

Any discrepancy can be eliminated while uploading code if a compatible board is chosen. Board can be chosen from the toolbar.

Step 7: Configuration of Serial port.

See the port to which Arduino is connected and select that port like COM3.

Step 8: The program can now be uploaded to the board.

The code is finally dumped and the result can be verified.

4.3 Embedded C

Embedded C is a set of language extensions for the C programming language by the C Standards Committee to address commonality issues that exist between C extensions for different embedded systems. In 2008, the C Standards Committee extended the C language to address these issues by providing a common standard for all implementations to adhere to. It includes a number of features not available in normal C, such as, fixed-point arithmetic, named address spaces, and basic I/O hardware addressing.

4.4 Blynk IoT App

Introduction:

1. Blynk is a full suite of software required to prototype, deploy, and remotely manage connected electronic devices at any scale: from personal IoT projects to millions of commercial connected products.
2. With Blynk anyone can connect their hardware to the cloud and build a no-code iOS, Android, and web applications to analyze real-time and historical data coming from devices, control them remotely from anywhere in the world, receive important notifications, and much more…

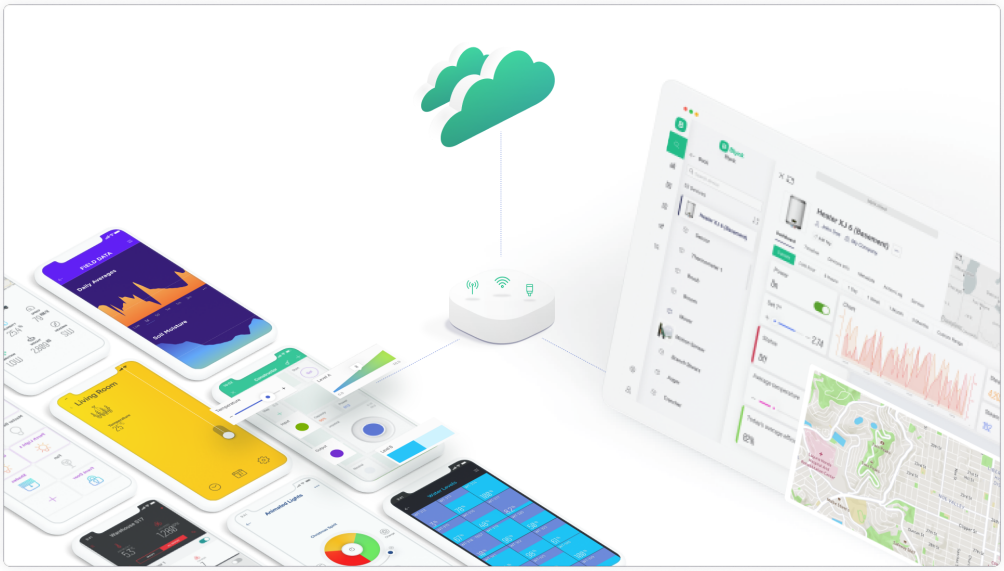


Fig 4.2 Overview of Blynk App

Send Data From Hardware To Blynk:

1. With Blynk you can send raw or processed data from any sensor or actuator connected to the MCU board
2. When you send data to Blynk it flows through a Datastream using Blynk protocol. Then every value is automatically timstamped and stored in the Blynk.Cloud database (you can also send batches of timestamped data if needed).
3. Datastream is a channel that tells Blynk what type of data is flowing through it.

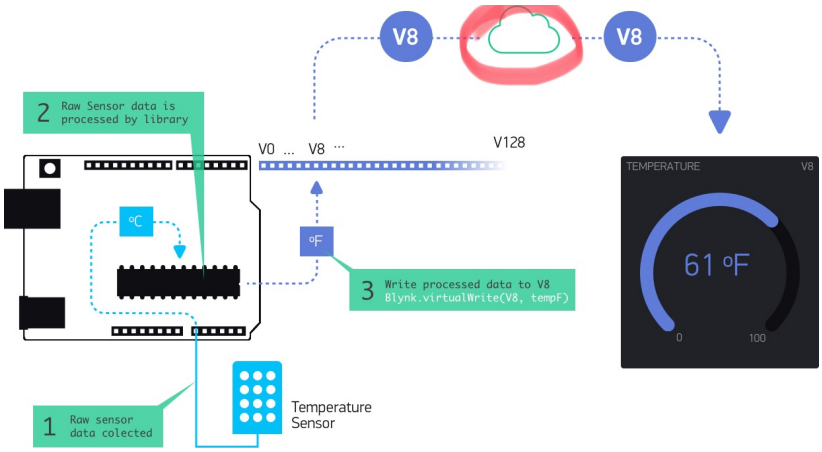


Fig 4.3 Display of sensor data on Blynk App

Automations

1. Automations are designed to work across all properly configured devices. This means that one condition can be used to perform actions on multiple devices.
2. Note that the value of a Data stream may be used by a condition, and action can be configured to set a Data stream to a new value. Using widgets on the Blynk.Console and/or Blynk .App, you can visualize and change the values of Data streams. You may also access Data streams within the firmware of an IoT device, and by using the Blynk. Cloud HTTPS API.
3. One or more actions may be executed sequentially or simultaneously when a condition's requirement has been met. The types of actions include:
   - Sending a mobile app notification
   - Forwarding device data to some other device data stream
   - Sending an email
   - Setting a delay
   - Setting a Data stream value

4.5 Proteus Software

The Proteus Design Suite is a proprietary software tool suite used primarily for electronic design automation. The software is used mainly by electronic design engineers and technicians to create schematics and electronic prints for manufacturing printed circuit boards.

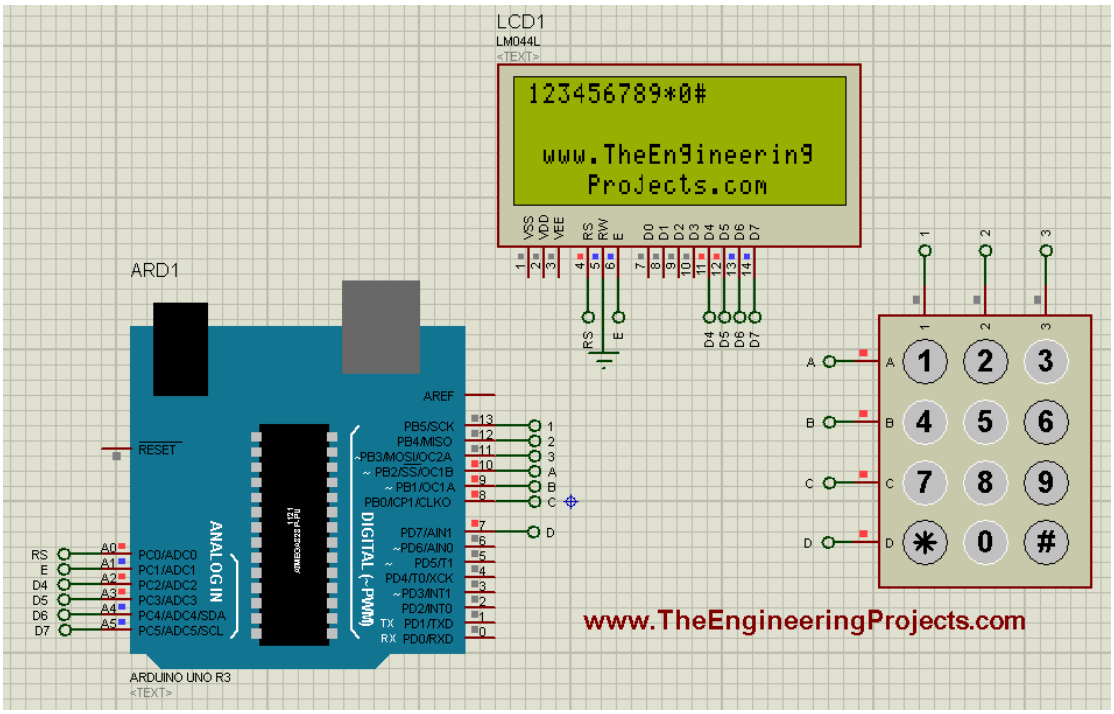


Fig 4.4 Schematic capture using proteus software

Features of Proteus

- There are 2 main parts of proteus first is used to design and draw different circuits and the second is for designing of PCB layout.
- First is ISIS that used to design and simulate circuits. And second is ARES that used for designing of a printed circuit board.
- It also provides features related to the three-dimensional view of design in PCB.

4.6 Conclusions

In this chapter software tools like Arduino IDE, Embedded C, Blynk App and proteus software are studied.

Chapter 5
Results and Discussion

5.1 Introduction

In this chapter we discuss the projects outputs and the sequential process of the project, i.e. flowchart of the project, and schematic diagram of setup (by fritzing software), explanation of experimental setup, and results of the project.

5.2 Schematic diagram

Figure 5.1 shows the Schematic diagram of the complete project.

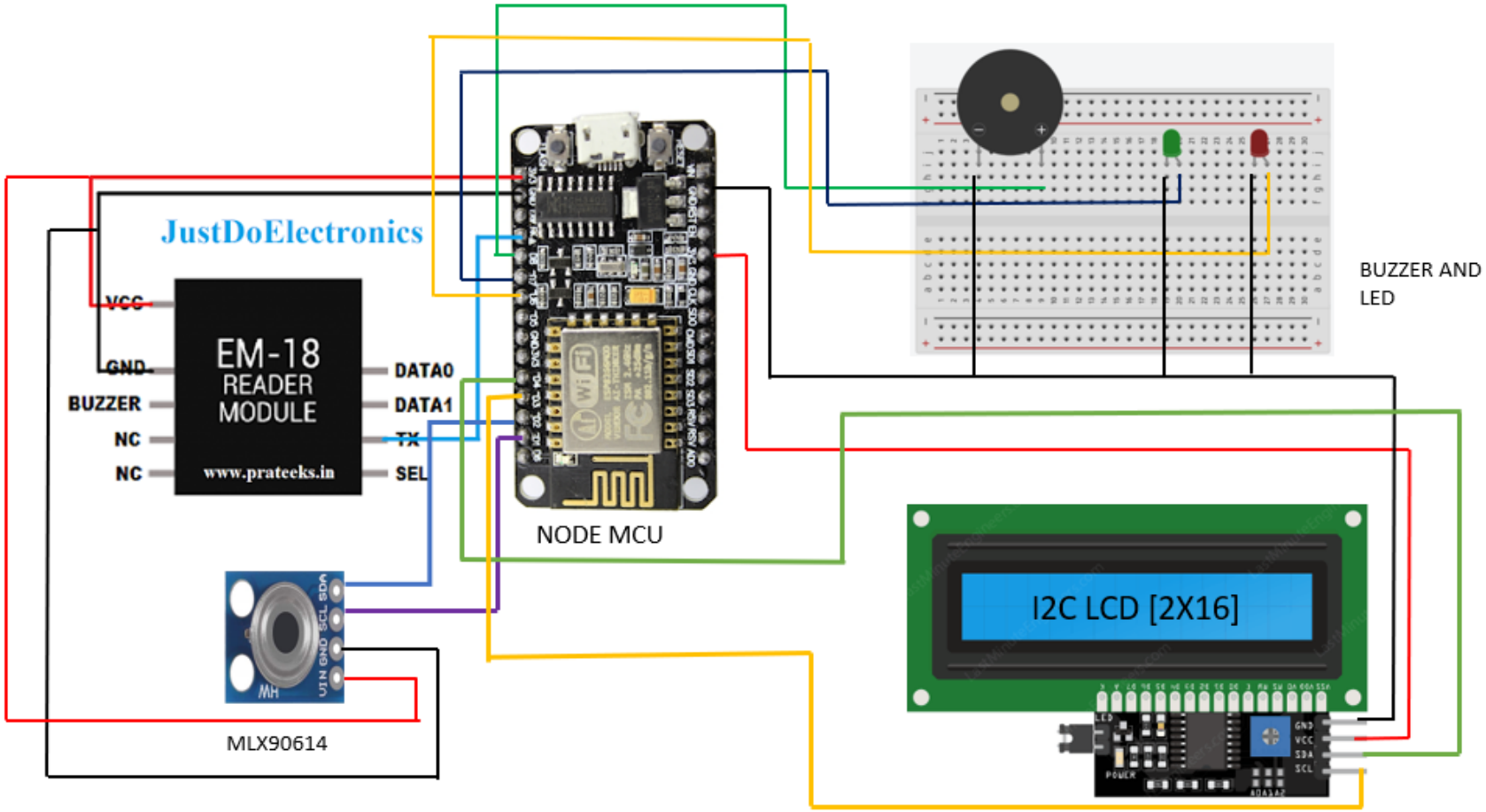


Fig 5.1 Schematic diagram

5.3 Connections

Table 5.1 shows the pin connections of the components used in the project.

Table 5.1: Description of connections between various components

S.No	Component	Selected Pin	Pin to be connected to
1.	**EM18 Reader Module**	GND	GND
		TX	NODE MCU RX
		Vcc	3.3 volts
2.	**I2C LCD Display**	GND	GND
		Vcc	3.3 volts
		SCL	NODE MCU D1
		SDA	NODE MCU D2
3.	**Contactless temperature Sensor**	GND	GND
		Vcc	3.3 volts
		SCL	NODE MCU D3
		SDA	NODE MCU D4
4.	**Buzzer**	-ve	GND
		+ve	NODE MCU D8
5.	**LCD Green**	-ve	GND
		+ve	NODE MCU D7
6.	**LED RED**	-ve	GND
		+ve	NODE MCU D6

5.4 Flowchart

Figure 5.2 shows the flowchart of the Contactless Body temperature Scanning System

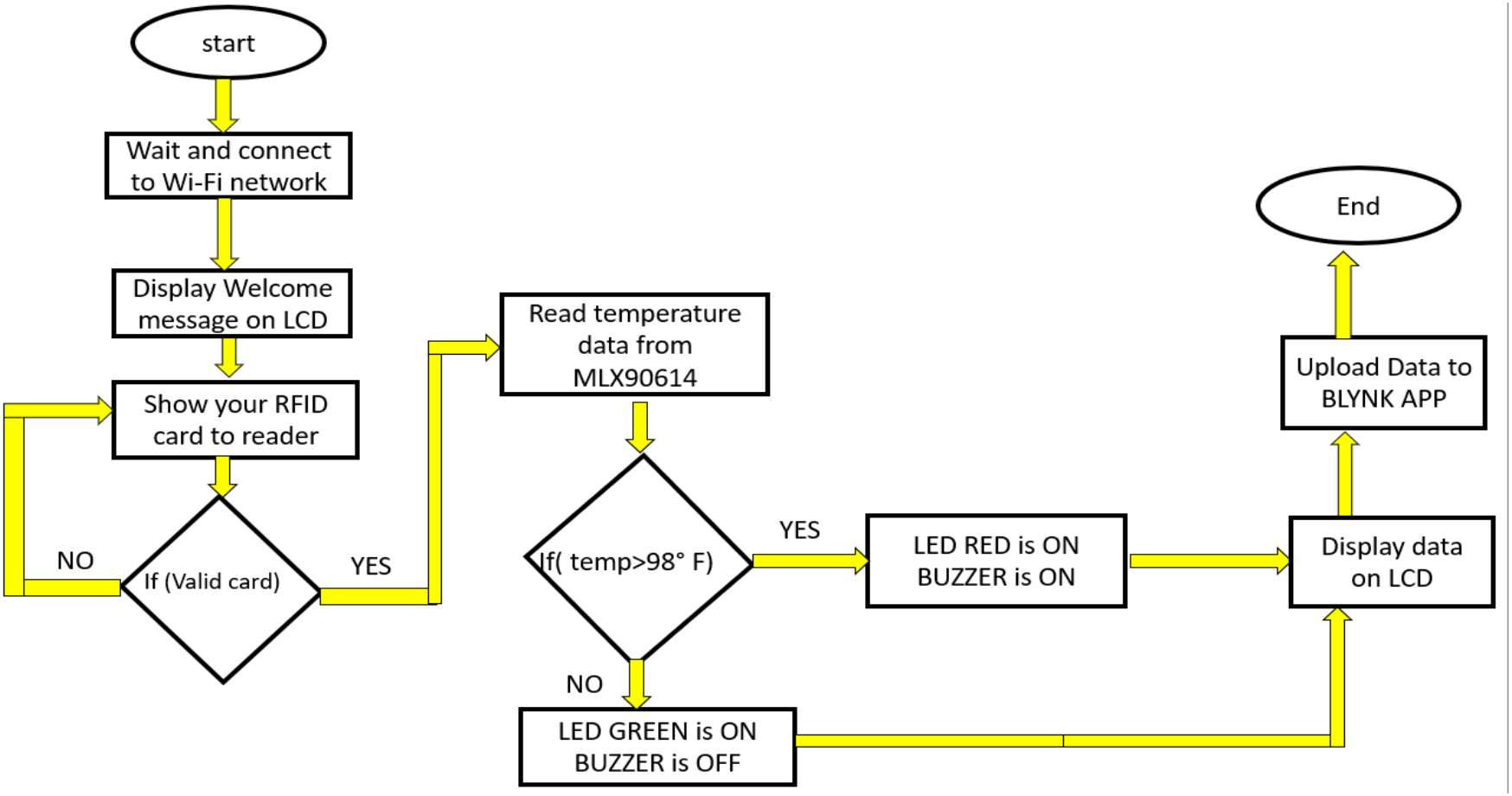


Fig. 5.2 Flowchart

Description of Flowchart:

After the device is on the response of device can be tracked as follows

Step 1:

The device detects the Wi-Fi networks in the given and connects to the selected network whose credentials are uploaded in code. If Wi-Fi gets connected it will move to step 2 or else wait until it gets connected to given network.

Step 2:

In response the LCD shows a message about status of Wi-Fi connection and displays a Welcome message.

Step 3:

Now the user/Student needs to show his RFID card to the EM18 Reader Module.

If the shown card is valid :

In response the temperature value is recorded.

Else:

Device traces back its step and asks user to show card again until its valid.

Step 4:

In response the temperature is recorded from contactless temperature sensor and passes through a condition

If the given temperature is greater than 98.6 Fahrenheit:

Buzzer will be ON for 10 seconds and LED RED will be ON for alerts.

Else if temperature less than 98.6 Fahrenheit:

Buzzer will be OFF and LED GREEN will be ON for safe entry signal.

Else:

User needs to show his card again and follow up from step3.

Step 5:

The data will be displayed along with his name on LCD Display and also uploaded to BLYNK IOT App simultaneously.

5.5 Experimental Setup

Figure 5.3 shows the Experimental setup of the project.

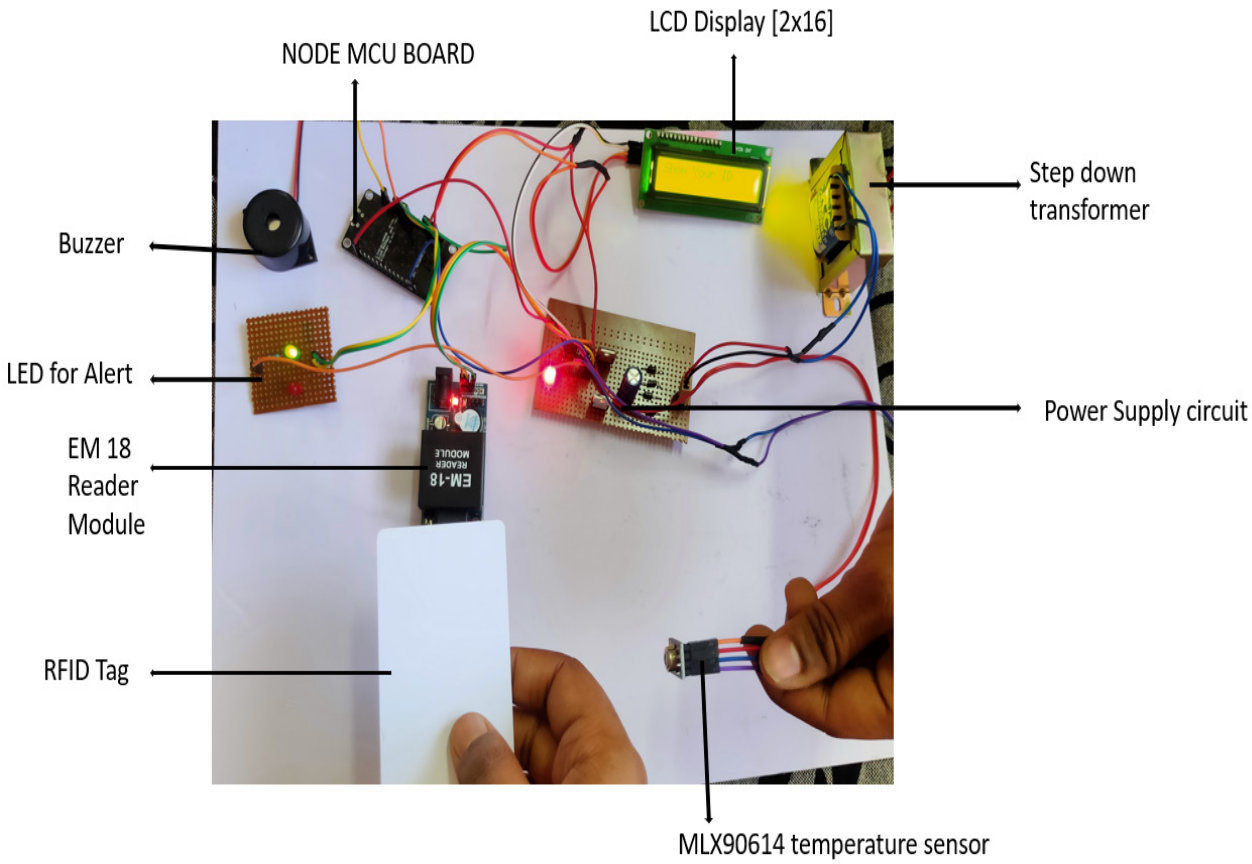


Fig. 5.3 Experimental Setup

5.6 Results

1. Figures 5.4 and 5.5 show the results obtained during this project.
2. Figure 5.6 show the results of three students/users who accessed the device using their cards in BLYNK IOT App Dashboard

.

This following figure conveys output when body temperature is normal.

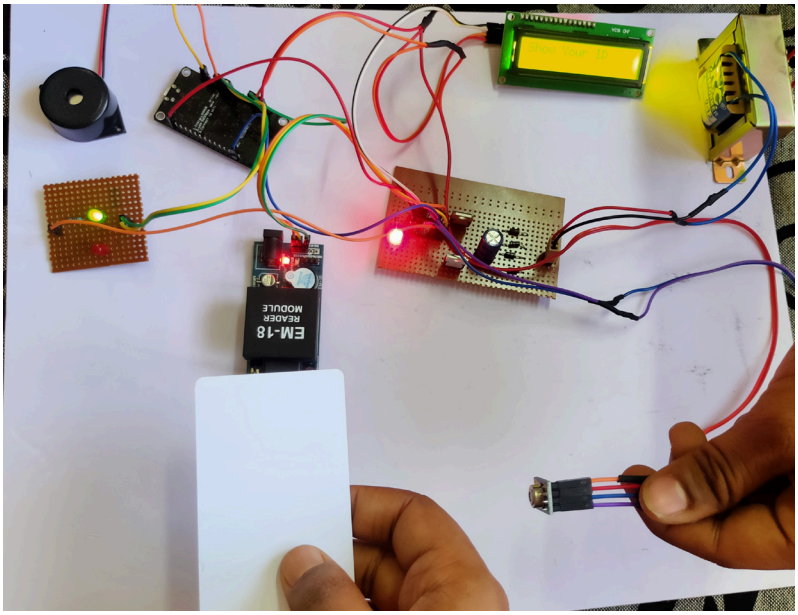


Fig 5.4 LED GREEN is ON and LED RED is OFF

This following figure conveys output when body temperature is abnormal.

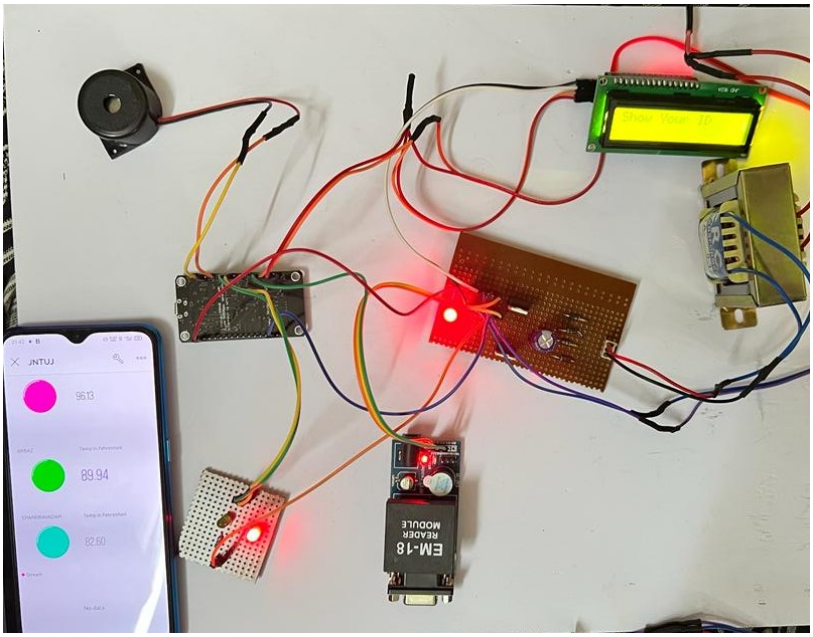


Fig 5.5 LED RED is ON and LED GREEN is OFF

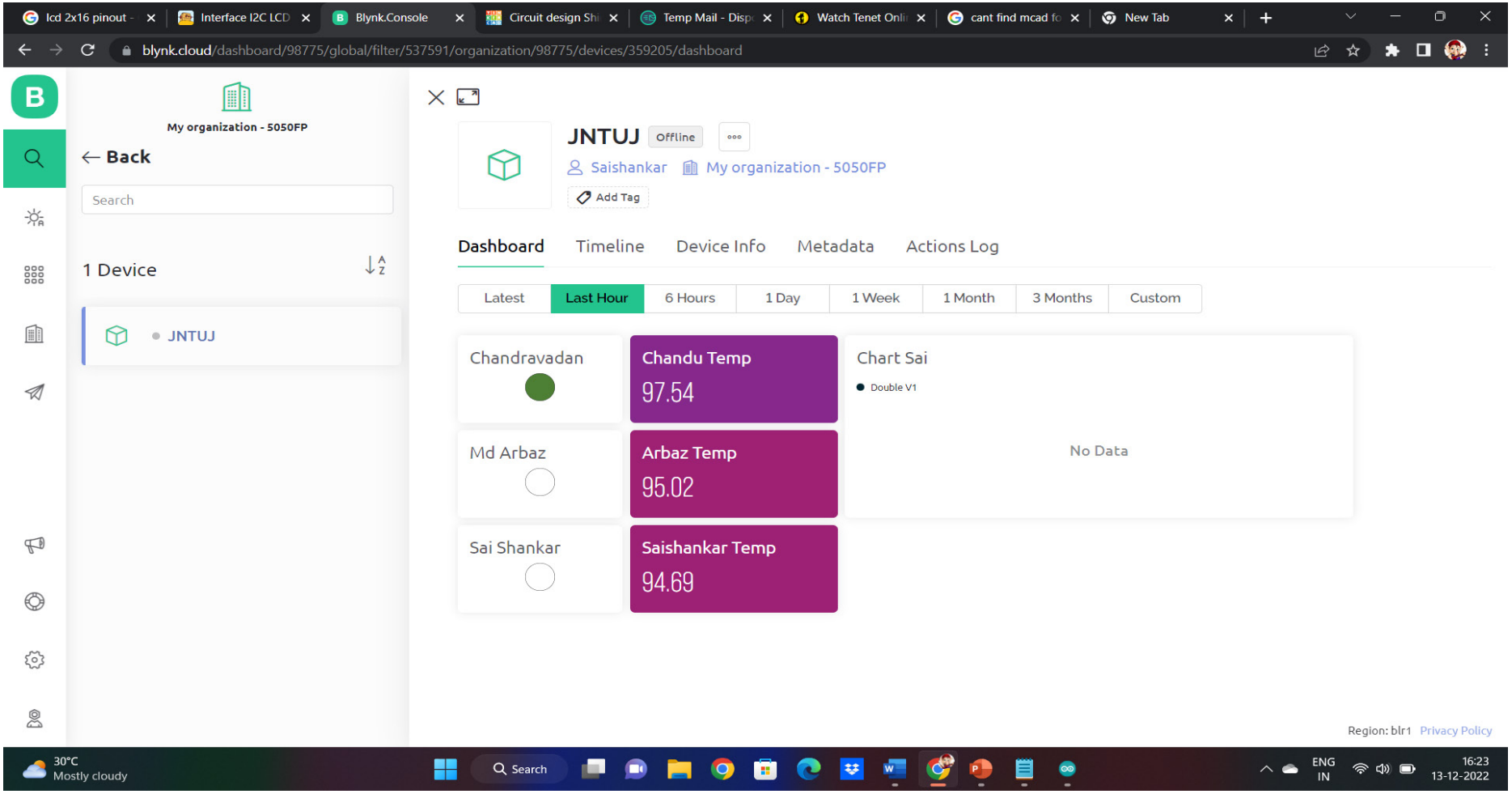


Fig 5.6 shows the body temperature data that was uploaded and displayed on Blynk APP Dashboard

5.7 Conclusions

In this chapter we have seen the schematic diagram, flowchart, experimental setup of the project and their results during various operations.

Chapter 6
Conclusions

The contactless temperature scanning device is capable of acting as a smart Internet of Things device. It provides administration with the attendance record with body temperature along with buzzer and LED system to find abnormal temperature and isolate ill candidates at the entrance right away which ensures a safe campus environment. The smart contactless body temperature scanning system can be enhanced much more in the future by using faster 5G connections which can be used for faster data transmissions. Also a more power efficient model will have to be created which will be capable of holding the battery for a longer time.

References

1. https://www.arduino.cc/
2. https://www.electronicshub.org/different-types-sensors/
3. https://www.electronicshub.org/arduino-introduction/
4. https://ieeexplore.ieee.org/document/9567862