WEATHER MONITORING

A project report submitted in partial fulfilment for the

award of degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING
(ECS1002)

Submitted by

Alapati Sai Harsha Vardhan -20BCE7401

Penumarthi Sai Karthik-20BCE7434

Bodinapalli Venkata Siva Sai Revanth-20BCE7403

Yarlagadda Lakshmi Akshay Kumar-20BCE7408

Ramini Siva Aakash -20BCD7240

Malla Sriraj - 20BCN7117

under the guidance of

Prof. D.Venkata Lakshmi, SCOPE

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING
VIT-AP UNIVERSITY

AMARAVATI

ANDHRA PRADESH, INDIA

APRIL 2023

Abstract

Now-a-days many weather reporting applications are available which gives us information about climatic changes that are going to take place by which man can be aware of present and 4 future climatic changes. Most of the weather reporting applications extracts the data from accurate weather system. Here we are building our own weather reporting system which would give us information about present temperature, humidity etc. We can setup this in our home and get time to time changes in climate which would help us in planning our daily work easily. Like It would be helpful for a farmer in this agricultural activity by which he can protect his crops climatic changes. It would help in transportation giving information of weather conditions etc.

INDEX

List of Figures

1.INTRODUCTION:

Weather condition plays an important role in our daily life as weather and climate are the most ubiquitous factors for home and environment planning. Moreover, the tremendous development of Internet nowadays made possible to monitor weather conditions and collect the respective data in-situ. All the objects, sensors and devices can be linked through Internet to share and analyze the data collected at various locations. The Internet of Things (IoT) can be much more extensive in predicting and knowing the weather conditions in particular place by connecting this weather station to the Internet. The climate in general is capricious that is hard to predict nowadays . With advanced technology to help humanity and bring convenience to the society, it is now the time for the weather broadcasting to be implemented into mobile phone instead of keep checking through only television or radio. However, with the mobile weather checking system we have in this era still often we see people rushing for schedule under the rain without umbrella; laundries are still showering by rain; home planted plants are wilted due to the hot and dry weather. Therefore, the objective of this project is to create an online weather system which enables user to check real time weather parameters of a place anytime and anywhere with just a few buttons click. On top of that, people will receive real time notification or reminder. Weather prediction will be done which allows users to get themselves prepared for their plans in the current weather. Google Cloud platform is chosen because it is easy to set up, easy to run, and with security built in features. It will be used to do data analysis and building a mobile weather checking applications. Google Cloud is suitable to be used in this project that needs several functions provided in this platform in order to obtain desired results. References are some of the weather stations that have already been developed by other researchers.

2.BACKGROUND:

However, majority of the technologies were employed to support teaching process, such as presentation, organization and integration of learning content, evaluation of students' learning performance, mutual interaction between the student and the teacher, etc. The physical factors, such as temperature, humidity, noise, thermal, air pressure, ventilation, air quality, acoustic, dust, vibration, lighting, radiation were taken insufficient account. Additional, considerable evidence shows that there is an explicit relationship between the physical characteristics of school buildings, and the spaces within them, and educational outcomes. Recently, more and more researchers have seen the importance of physical environment in the TRC, and providing the students with comfortable physical environment has cause concern of the academic community.

Based on the demands of new generation of students for the reform of learning environment and the analyzing of challenges for both the online learning environments and classroom former environments, Huang et al proposed the concept of "smart" learning environment which is the high level of digital learning environment with the aim at facilitating "easy, engaged and effective" learning for learners. This concept of "smart" learning environment covers not only the devices and instructional software, but also the physical aspect of learning environment. To build an easy, healthy and comfortable environment is one objectives of the "smart" learning environment. As one of the most important component of smart learning environment, campus weather station is a kind of school or college facilities with the function of automated or manual measurements of temperature, humidity, wind speed and direction, pressure, solar radiation and soil temperatures updated every 10 minutes.

This paper is the further study of smart learning environment, which focuses on physical environment. The aim is to develop the WMS based on Raspberry Pi to collect weather data of the campus, and transmit to TRC which equipped with indoor environmental systems.

3.PROBLEM DEFINITION:

The problem statement for weather monitoring is the need for accurate and reliable weather data for various applications. Traditional weather monitoring systems can be expensive and difficult to install, and may not provide accurate data for specific locations. Furthermore, these systems often require specialized knowledge to operate and maintain, making them inaccessible to many individuals and organizations.

Weather monitoring using a Raspberry Pi addresses these issues by providing a cost-effective and accessible solution for collecting and analyzing weather data. By utilizing sensors for measuring various weather parameters, the Raspberry Pi can retrieve data and store it in a database or send it to a cloud server for further analysis and processing. This allows for more widespread data collection and analysis, which can be used by farmers to plan their irrigation and cropping schedules, by meteorologists to predict weather patterns and make forecasts, by environmentalists to monitor climate change, and by individuals to plan outdoor activities or dress appropriately for the weather.

Therefore, the problem statement for weather monitoring is to develop an accessible and cost-effective system for collecting and analyzing weather data that provides accurate and reliable information for various applications.

Weather monitoring system being very hand for better performance of the solar plants has the issue of higher cost. The hard drive based data logging facility requires a separate computer setup for its operation and many a times, the data stored cannot be manipulated in a useful mean.

4.OBJECTIVES:

- Collect and store accurate and reliable weather data: The primary objective of weather monitoring is to collect and store accurate and reliable weather data in real-time. This data can include temperature, humidity, pressure, rainfall, wind speed, and direction.

- Provide accessible and cost-effective weather monitoring: Weather monitoring using a Raspberry Pi should be accessible and cost-effective, so that it can be used by individuals and organizations with limited resources. This can be achieved by using low-cost sensors and a Raspberry Pi microcontroller.

- Analyze and process weather data: The collected weather data should be analyzed and processed to provide useful information for various applications. This can include generating weather forecasts, predicting weather patterns, and identifying trends in climate change.

- Enable remote access to weather data: Weather data should be accessible remotely, either through a web application or a mobile application. This allows users to access weather data from anywhere and at any time.

- Provide customizable weather alerts: Weather monitoring should provide customizable alerts based on specific weather conditions. This can include alerts for high winds, heavy rainfall, or extreme temperatures.

- Foster collaboration and data sharing: Weather monitoring should encourage collaboration and data sharing among individuals and organizations. This can be achieved by using open-source software and making weather data publicly available.

- The aim of weather monitoring system is to detect, record and display various weather parameters such as temperature ,humidity ,Light intensity ,Rain fall, Smoke and Gas Values. This system makes use of sensors for detecting and monitoring weather parameters and then this collected information is sent to the cloud which can be accessed using the internet.

5.APPARATUS:

- Raspberry Pi Pico W
- 16×2 I2C LCD Display
- Connecting Wires
- Breadboard
- Micro-USB Cable
- Gas Sensors
- Smoke Sensor
- DHT11(Temperature & Humidity sensors)
- LDR

Required Software Applications:

- Thonny IDE
- Thing view app

Raspberry Pi Pico:

Raspberry Pi Pico is a low-cost, high-performance microcontroller board developed by Raspberry Pi Foundation. It is designed to be used for a wide range of projects, from simple embedded systems to advanced robotics and IoT applications. Raspberry Pi Pico is based on the RP2040 microcontroller, which is designed and developed by Raspberry Pi Foundation. It features a dual-core ARM Cortex-M0+ processor running at 133MHz, 264KB of on-chip RAM, and up to 2MB of external flash memory. It also includes a variety of hardware peripherals, including SPI, I2C, UART, and PWM interfaces, as well as a USB 1.1 interface for programming and debugging.

Fig-1: Raspberry pi pico

Gas Sensors:

Gas sensors are electronic devices that are used to detect the presence of various gases in the air. They are widely used in industrial and residential applications to detect and measure the concentration of gases such as carbon monoxide, methane, propane, hydrogen, and many others.

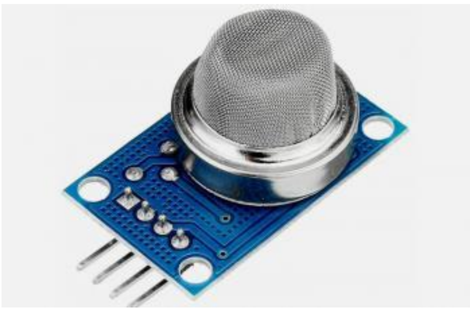


Fig-2: Gas Sensor

Temperature Sensor:

Temperature sensors are electronic devices that are used to measure the temperature of a system or environment. They are widely used in a variety of applications, from monitoring the temperature of food and beverages to monitoring the temperature of industrial equipment and machinery.

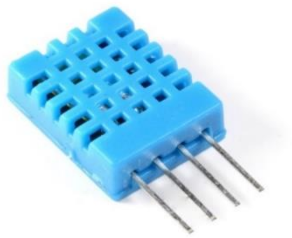


Fig-3: Temperature Sensor

Humidity Sensors:

Humidity sensors are electronic devices that are used to measure the amount of moisture or humidity in the air. They are widely used in a variety of applications, including HVAC systems, industrial processes, and weather monitoring.

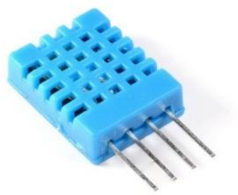


Fig-4: Humidity Sensor

Lcd Display:

A liquid crystal display (LCD) has liquid crystal material sandwiched between two sheets of glass. Without any voltage applied between transparent electrodes, liquid crystal molecules are aligned in parallel with the glass surface.

Fig-5: Lcd Display

LDR (Light depended resister):

LDRs or Light Dependent Resistors are very useful especially in light/dark sensor circuits. Normally the resistance of an LDR is very high, sometimes as high as 1000 000 ohms, but when they are illuminated with light resistance drops dramatically.

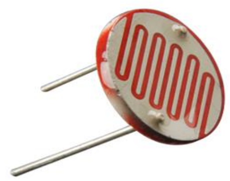


Fig-6: LDR

Connecting Wires:

Connecting wires allows an electrical current to travel from one point on a circuit to another because electricity needs a medium through which it can move. Most of the connecting wires are made up of copper or aluminum.

Fig-7: Connecting Wires

Smoke Sensor:

A smoke detector is an electronic fire-protection device that automatically senses the presence of smoke, as a key indication of fire, and sounds a warning to building occupants.

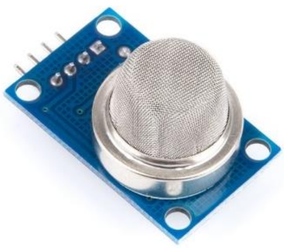
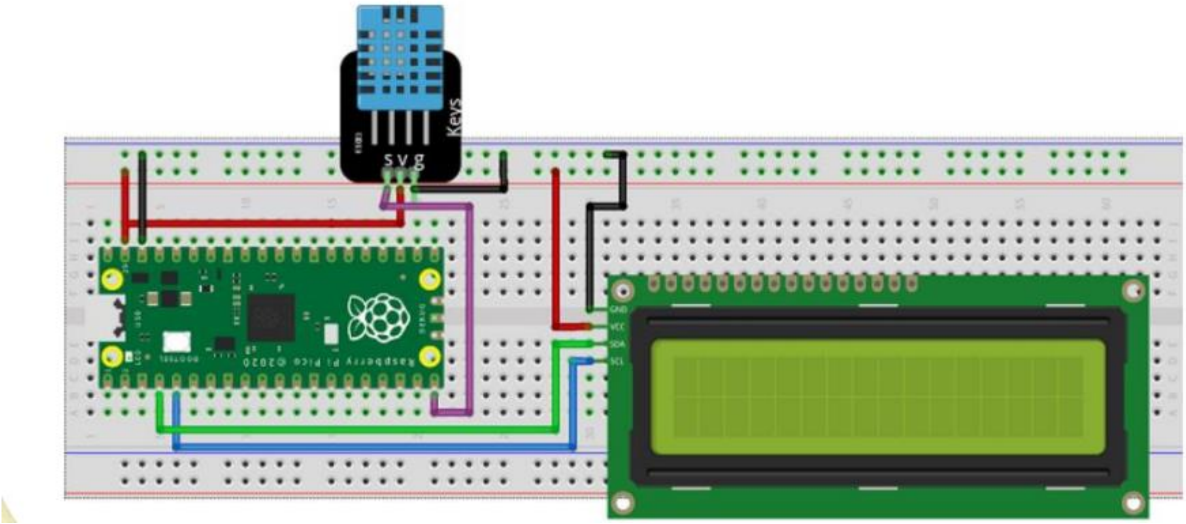


Fig-8: Smoke Sensor

Circuit Design:



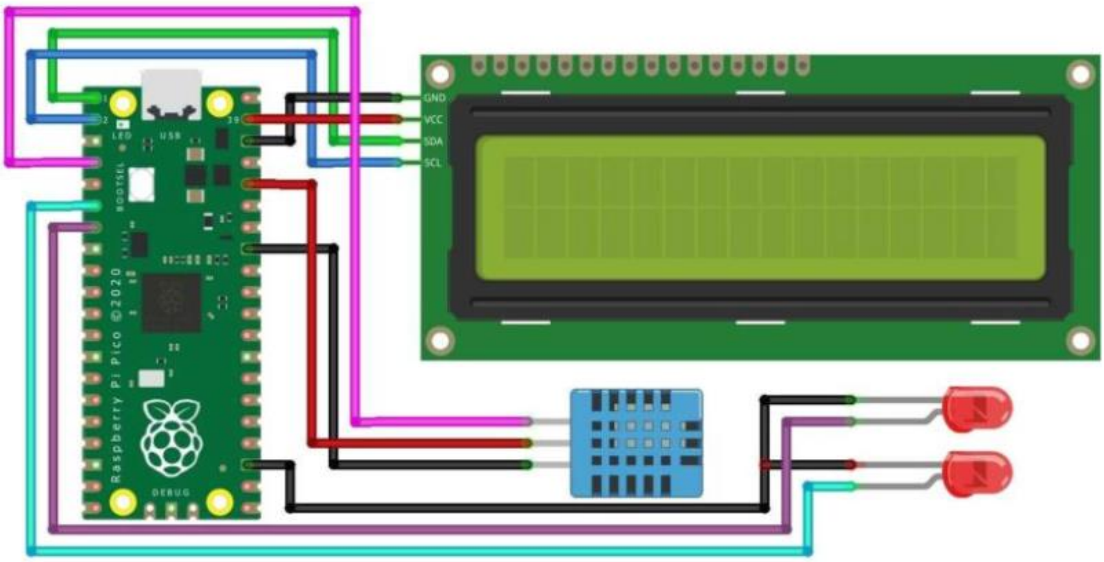


Fig-9:Circuit Design

Block Diagram:

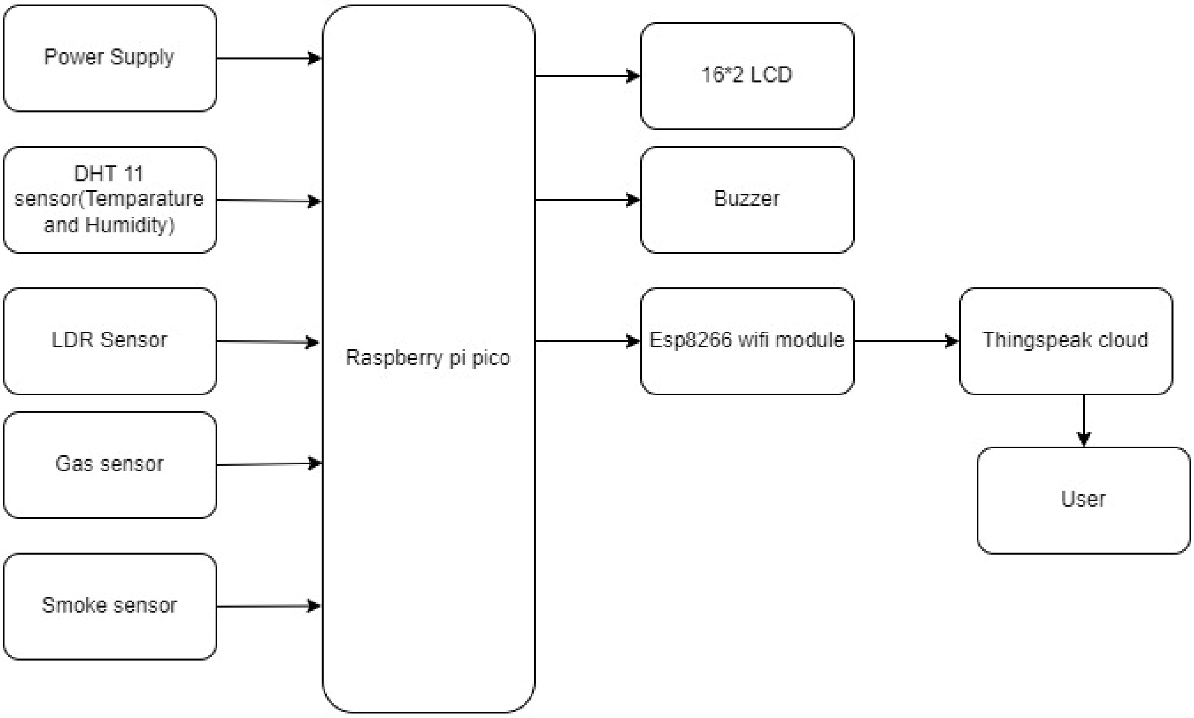


Fig-10: Block Diagram

6.PROCEDURE:

1.Place the breadboard on a flat surface.

2.Connect the DHT11 temperature sensor to the breadboard using jumper wires. Connect the VCC pin to a 3.3V pin on the Raspberry Pi, the GND pin to a GND pin on the Raspberry Pi, and the data pin to GPIO pin 17.

3.Connect the DHT11 humidity sensor to the breadboard using jumper wires. Connect the VCC pin to a 3.3V pin on the Raspberry Pi, the GND pin to a GND pin on the Raspberry Pi, and the data pin to GPIO pin 27.

4.Connect the first MQ-2 gas sensor to the breadboard using jumper wires. Connect the VCC pin to a 5V pin on the Raspberry Pi, the GND pin to a GND pin on the Raspberry Pi, and the data pin to GPIO pin 18.

5.Connect the second MQ-2 gas sensor to the breadboard using jumper wires. Connect the VCC pin to a 5V pin on the Raspberry Pi, the GND pin to a GND pin on the Raspberry Pi, and the data pin to GPIO pin 22.

6.If you are using an Ethernet cable, plug one end into the Ethernet port on the Raspberry Pi and the other end into your router or modem.

7.If you are using a Wi-Fi dongle, insert it into one of the USB ports on the Raspberry Pi.

8.Plug the power supply into a power outlet and the other end into the Raspberry Pi.

9.Wait for the Raspberry Pi to boot up.

10. First you need to create account on ThingSpeak website and create a ‘New channel’ in it. In new channel you have to define some fields for the data you want to monitor, like in this project we will create three fields for Humidity, Temperature and Pressure data.

11.Now click on ‘API keys’ tab and save the Write and Read API keys, here we are only using Write key. You need to Copy this key in ‘key’ variable in the Code.

12.After it, click on ‘Data Import/Export’ and copy the Update Channel Feed GET Request URL.

13.Now we need this ‘Feed Get Request URL’ in our Python code to open “api.thingspeak.com” and then send data using this Feed Request as query string. And Before sending data user needs to enter the temperature, humidity and pressure data in this query String using variables in program.

14.after creating channel successfully we will get a channel id. we can share this channel id and use this id to search in Think View app to monitor the weather.

7.RESULTS AND DISCUSSION:

On the system side Raspberry Pi board operates as a data acquisition mode and as a web server mode. It collects data from Temperature and Humidity sensor, Pressure and Altitude sensor, Light intensity sensor and rain water Level sensor. This data is then sent to the client side using HTTP protocol. On client side real-time data can be seen from anywhere in the world on gecko.com. Internet connection to the board is given by using LAN through Ethernet port or by using USB dongle through USB port. On this website one channel is created and all six fields are placed in this channel. Field 6 shows temperature, Field 7 shows humidity, Field 8 shows rain water level.

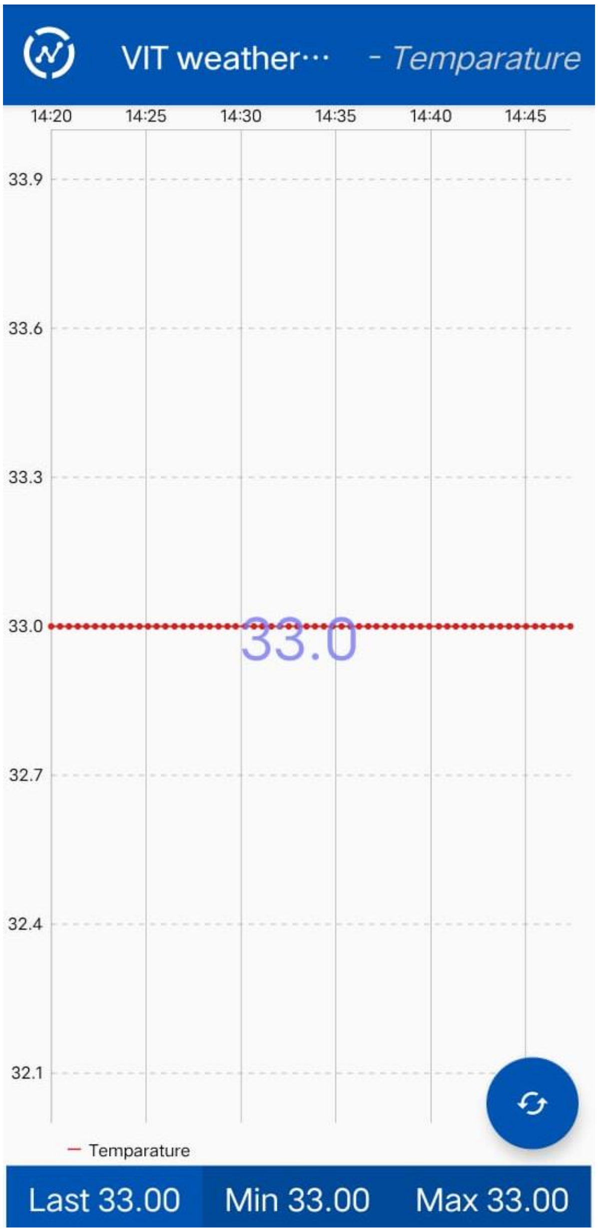


Fig-11:Temperature Graph

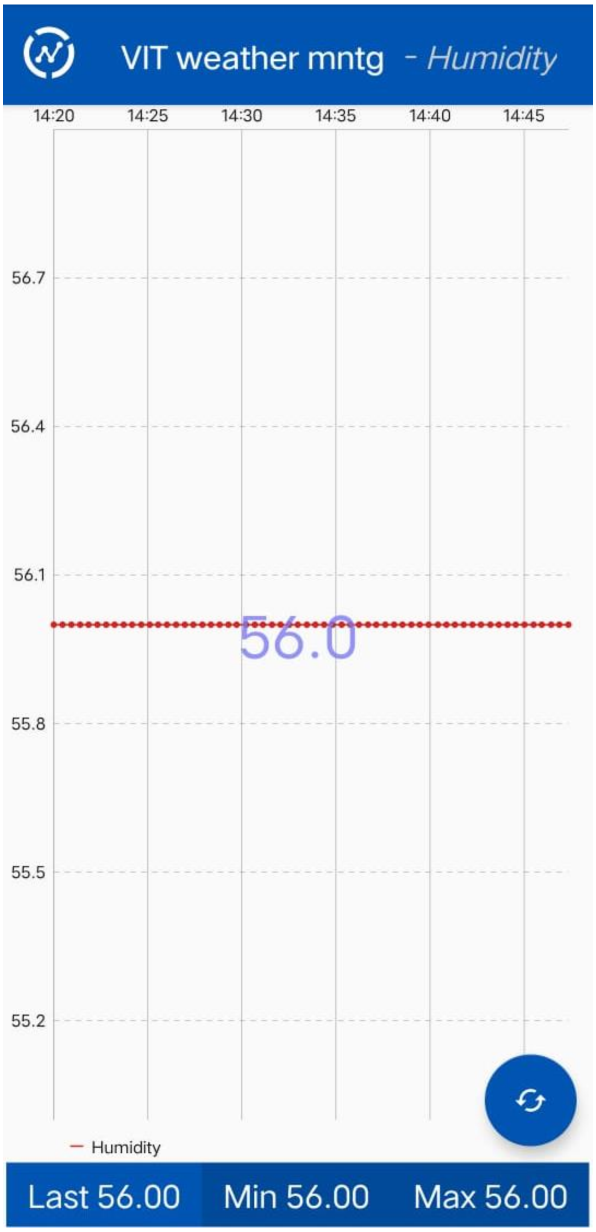


Fig-12: Humidity Graph

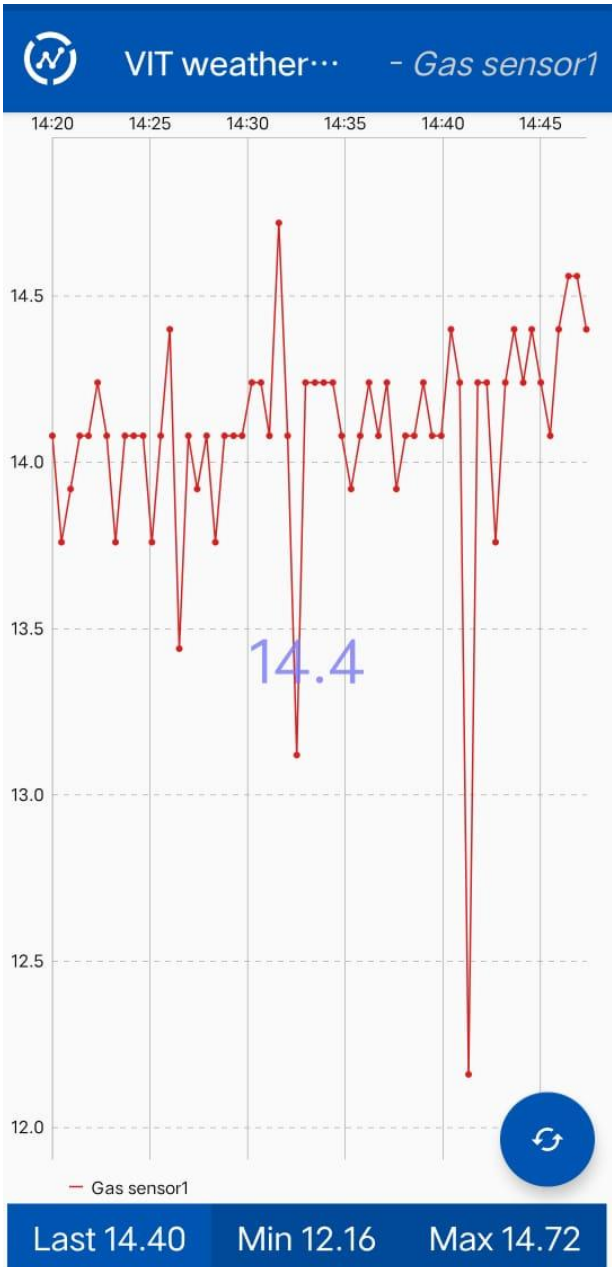


Fig-13:Gas Sensor Graph

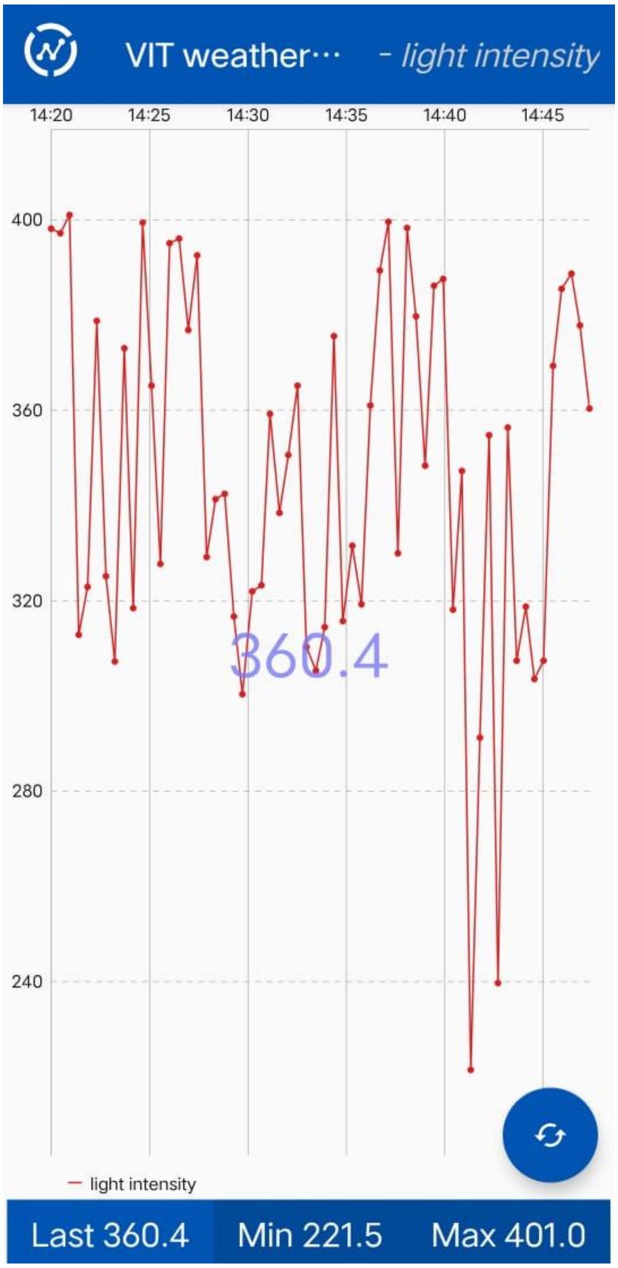
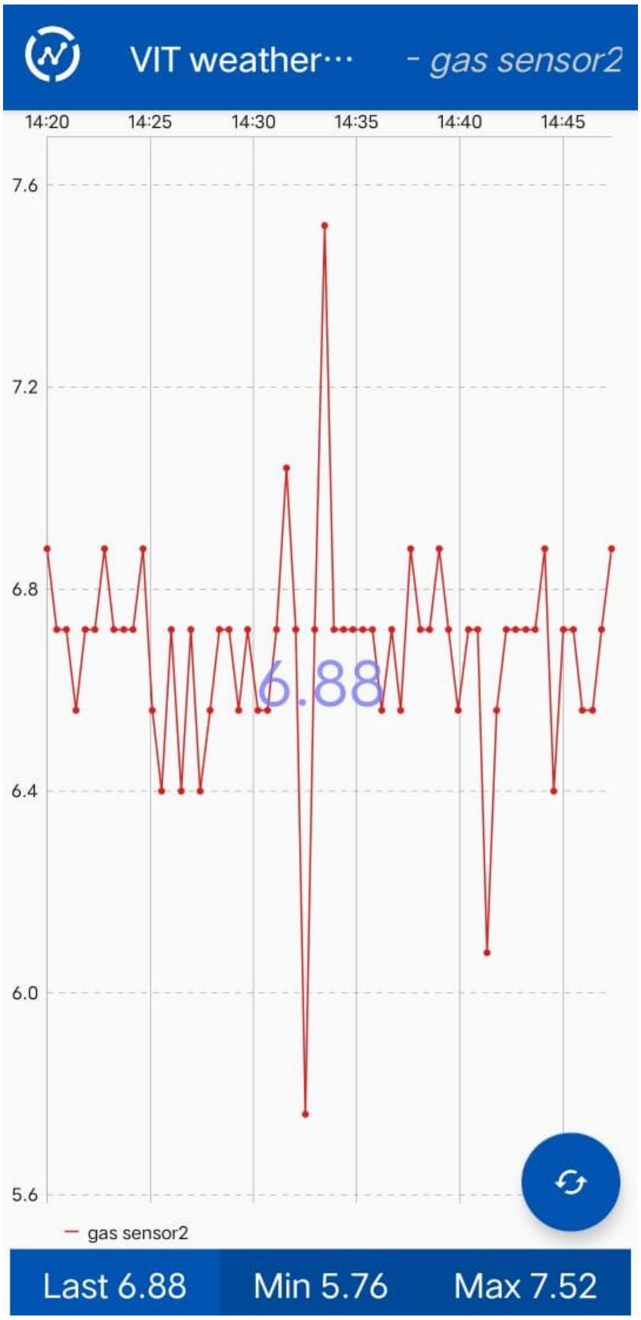


Fig-14: Light Intensity Graph



VIT weather···
- gas sensor2
14:20
14:25
14:30
14:35
14:40
14:45
7.6
7.2
6.8
6.4
6.0
5.6
6.88
gas sensor2
Last 6.88
Min 5.76
Max 7.52

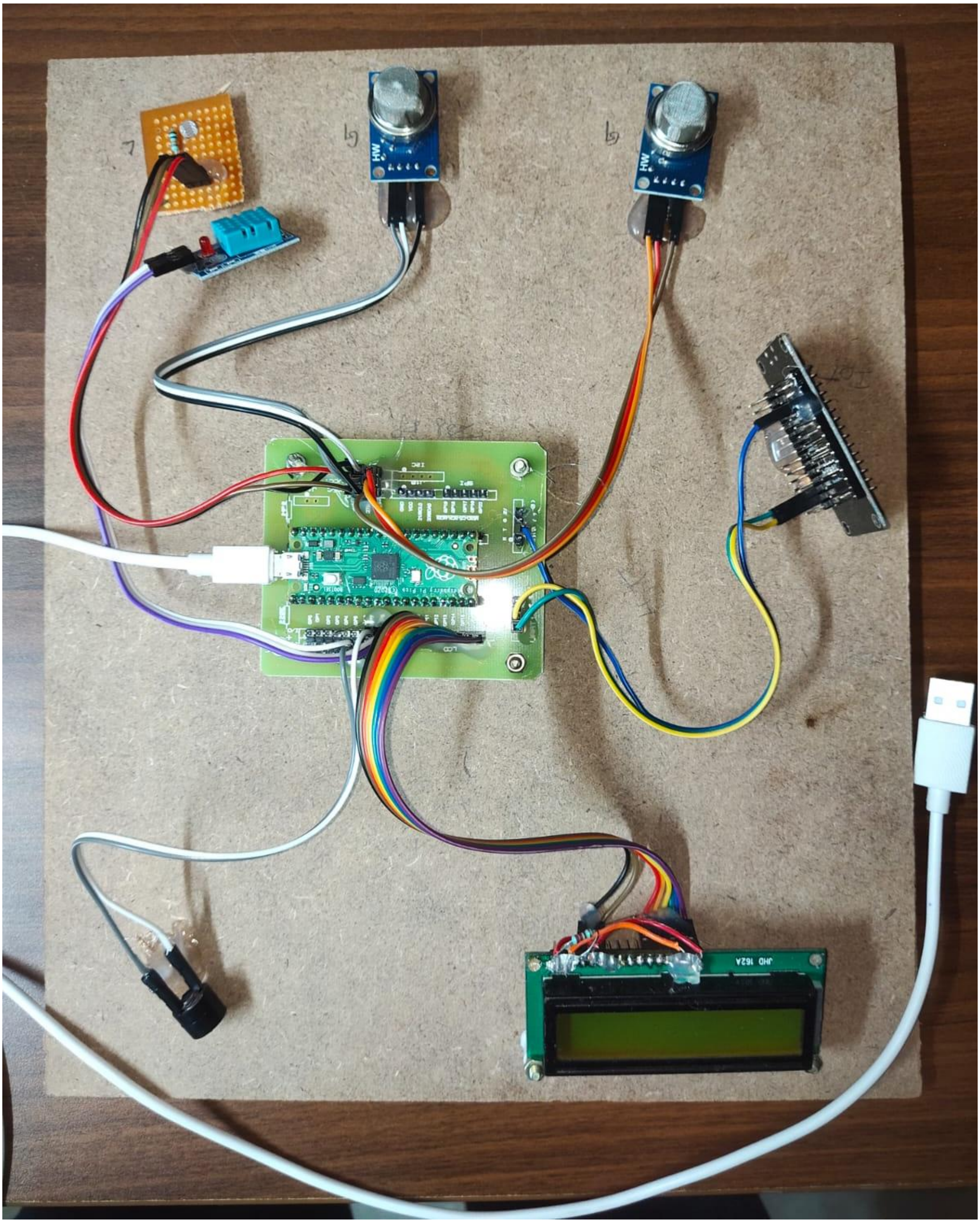


Fig-15: Weather Monitoring

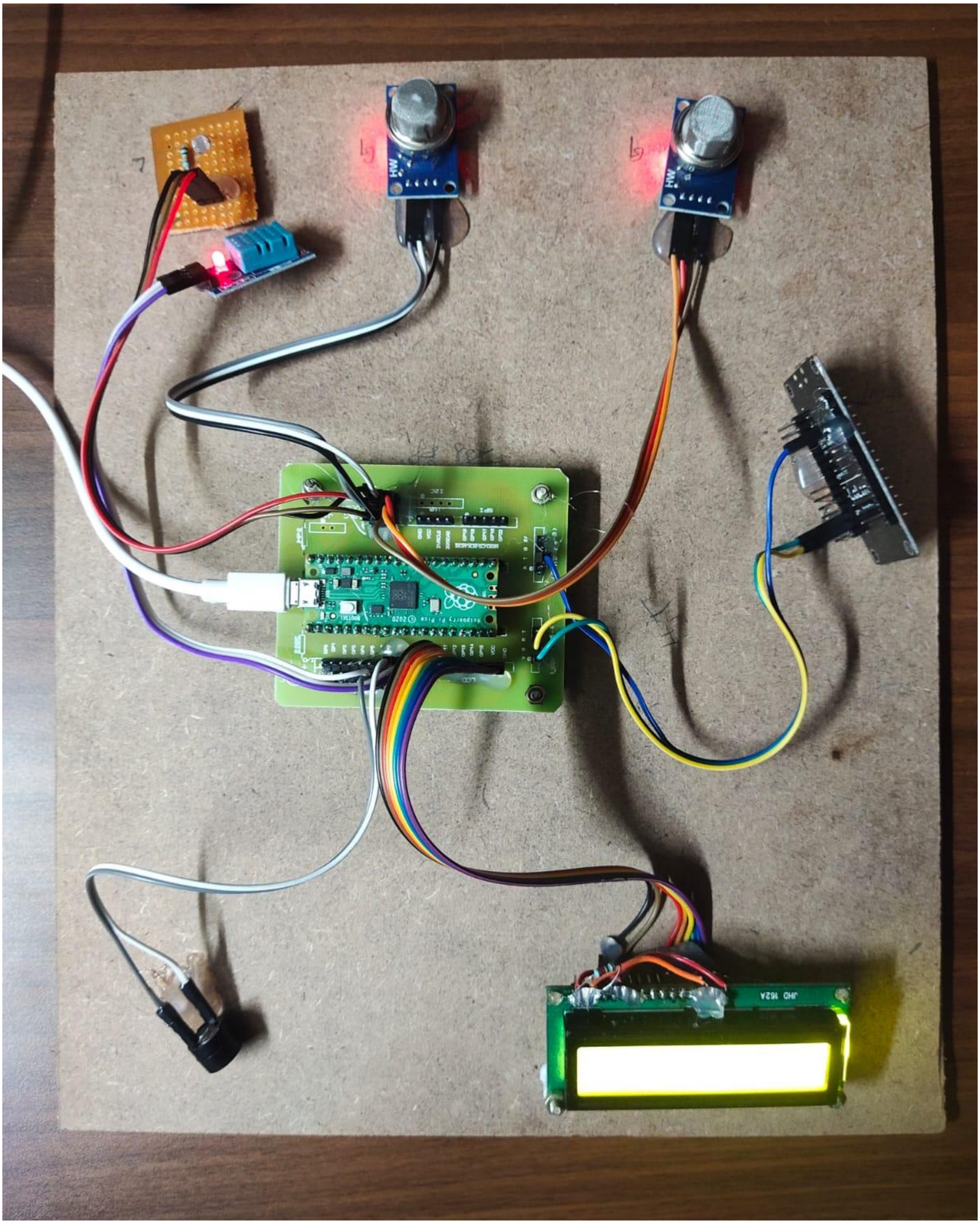


Fig-16:Weather Monitoring

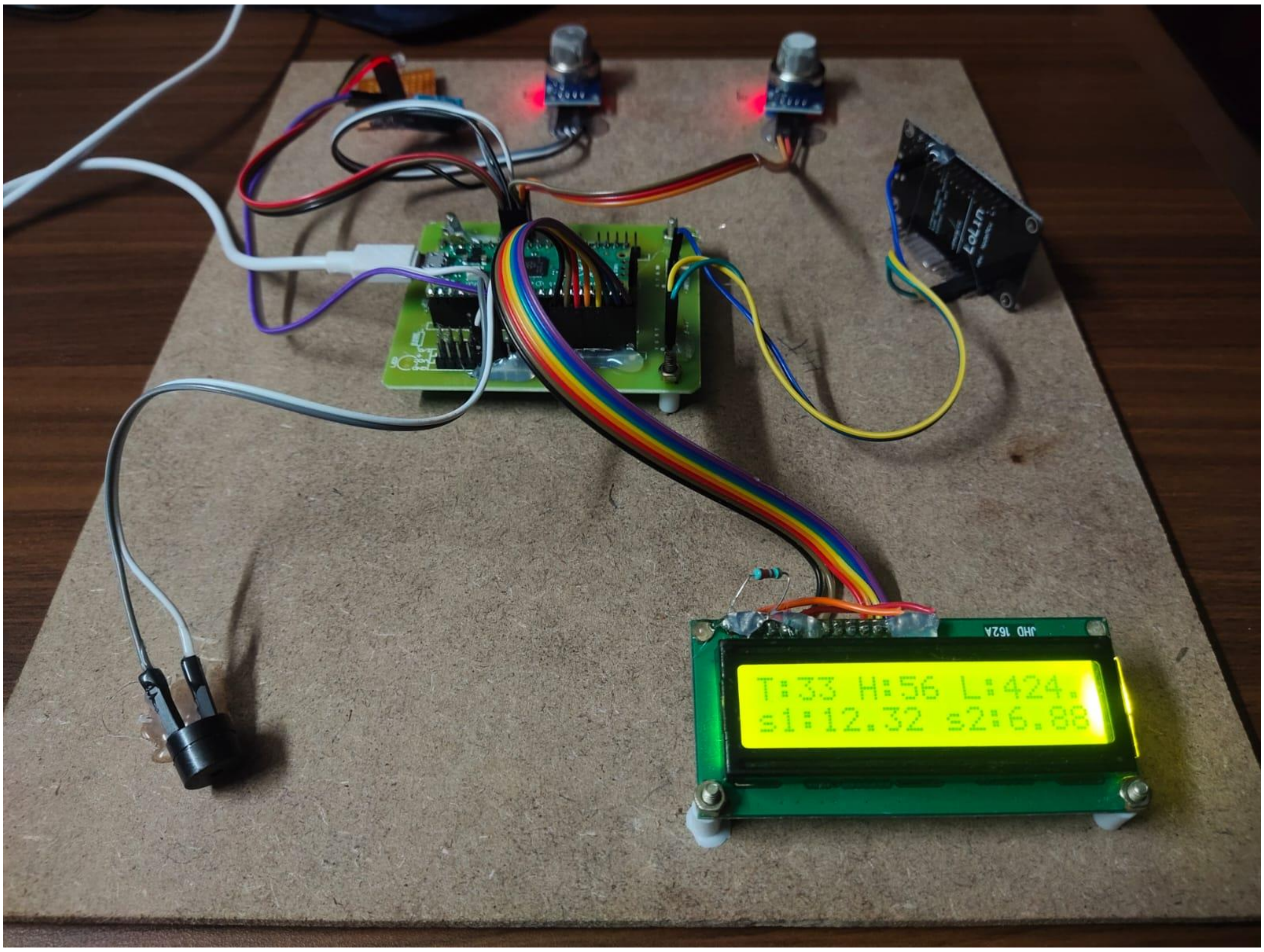


Fig-17: Weather Monitoring

8.CONCLUSION & FUTURE SCOPE:

CONCLUSION:

Weather prediction is a very important factor, which forecasts the climate in a region based upon the values of weather parameters. So the calculated results from this system can be made use in forecasting the weather of that locality for a period of time. As we made use of Raspberry pi in this model, immediate alert message or e-mail can be sent to the mobile phone, when the parameters changes are drastic. The technology changes day by day. Using the sensors for air temperature, air humidity, light, soil moisture, and rain detection in combination with Raspberry PI a prototype had been developed. Data from the sensors is transmitted to sever where it can be viewed globally which will be easily accessible to everyone. This IoT based system gives real-time monitoring of environmental parameters.

FUTURE SCOPE:

- The project can be enhanced by using a sensor to note the soil moisture value such that usage of unnecessary fertilizers can be reduced. A water meter can be added to estimate the amount of water used water irrigation and thus giving cost estimation. Further, it also reduces the investment of farmers.
- Advanced analytics and machine learning: As the amount of weather data collected continues to grow, there will be an increasing need for advanced analytics and machine learning algorithms to process and analyze this data. These algorithms can be used to identify patterns, predict weather events, and generate more accurate weather forecasts.
- Integration with smart home systems: Weather monitoring systems can be integrated with smart home systems to automate tasks based on weather conditions. For example, a smart irrigation system can be programmed to water plants only when there is no rainfall.
- Integration with autonomous vehicles: Weather monitoring systems can be integrated with autonomous vehicles to improve safety and efficiency. For example, autonomous cars can adjust their speed and route based on weather conditions.
- Integration with renewable energy systems: Weather monitoring systems can be integrated with renewable energy systems to optimize energy production. For example, solar panels can be angled to maximize energy production based on the position of the sun and cloud cover.
- Improved data visualization: Weather data can be visualized using advanced data visualization techniques to make it more understandable and actionable. This can include interactive maps, graphs, and charts.
- Citizen science: Weather monitoring systems can be used to engage citizens in data collection and analysis. This can help to improve the quality and coverage of weather data, and also foster greater public engagement with science and technology.

9.REFERENCES:

1. The International Journal of Engineering and Science (IJES) Volume 4 2015 ISSN (e): 2319 – 1813 'Zigbee based weather monitoring system'

2. Documentation for the Python programming language :http://www.python.org.

3. DeHennis, A. D., and K. D. Wise.A wirelessmicrosystem for the sensing of temperature, and relative humidity. Journal Micro electro mechanical Systems 14(1):ISO7730. 1984, Geneva, Switzerland.

4. Ong, K. G., , C. L. Robbins, and R. S. Singh. 2001.Design of a wireless, passive, resonant-circuit. environmental monitoring sensors. Sensors and Actuators A 93: 33–43.

10.APPENDIX:

CODES IN APPENDIX:

```
from machine import Pin,ADC
import time,utime
from dht import DHT11
sensor = DHT11(Pin(9,Pin.OUT, Pin.PULL_DOWN))
buz = Pin(15, Pin.OUT)
ss = ADC(27)
ms = ADC(28)
ldr= ADC(26)
trigger = Pin(3, Pin.OUT)
echo = Pin(2, Pin.IN)
pump1=Pin(18,Pin.OUT)
pump2=Pin(19,Pin.OUT)
rs = machine.Pin(10,machine.Pin.OUT)
e = machine.Pin(11,machine.Pin.OUT)
d4 = machine.Pin(12,machine.Pin.OUT)
d5 = machine.Pin(13,machine.Pin.OUT)
d6 = machine.Pin(14,machine.Pin.OUT)
d7 = machine.Pin(15,machine.Pin.OUT)
def get_distance():
   trigger.low()
   utime.sleep_us(2)
   trigger.high()
```

```
    utime.sleep_us(5)
    trigger.low()
    while echo.value() == 0:
        signaloff = utime.ticks_us()
    while echo.value() == 1:
        signalon = utime.ticks_us()
    timepassed = signalon - signaloff
    dist = (timepassed * 0.0343) / 2
    return dist
def pulseE():
    e.value(1)
    utime.sleep_us(40)
    e.value(0)
    utime.sleep_us(40)
def send2LCD4(BinNum):
    d4.value((BinNum & 0b00000001) >>0)
    d5.value((BinNum & 0b00000010) >>1)
    d6.value((BinNum & 0b00000100) >>2)
    d7.value((BinNum & 0b00001000) >>3)
    pulseE()
def send2LCD8(BinNum):
    d4.value((BinNum & 0b00010000) >>4)
    d5.value((BinNum & 0b00100000) >>5)
    d6.value((BinNum & 0b01000000) >>6)
    d7.value((BinNum & 0b10000000) >>7)
    pulseE()
    d4.value((BinNum & 0b00000001) >>0)
    d5.value((BinNum & 0b00000010) >>1)
    d6.value((BinNum & 0b00000100) >>2)
    d7.value((BinNum & 0b00001000) >>3)
    pulseE()
def setUpLCD():
    rs.value(0)
    send2LCD4(0b0011)#8 bit
    send2LCD4(0b0011)#8 bit
    send2LCD4(0b0011)#8 bit
    send2LCD4(0b0010)#4 bit
    send2LCD8(0b00101000)#4 bit,2 lines?,5*8 bots
```

```
    send2LCD8(0b00001100)#lcd on, blink off, cursor off.
    send2LCD8(0b00000110)#increment cursor, no display shift
    send2LCD8(0b00000001)#clear screen
    utime.sleep_ms(2)#clear screen needs a long delay

uart0 = machine.UART(0,baudrate=9600,tx=Pin(0),rx=Pin(1))
print(uart0)
second=0
while True:
    #temp = sensor.temperature
    #humidity = sensor.humidity
    time.sleep(1)
    sval = ss.read_u16()/100
    mval = ms.read_u16()/100
    lval = ldr.read_u16()/100
    temp= 33#int(sensor.temperature)
    humidity= 56#int(sensor.humidity)
    #fval=1-fs.value()
    distance=get_distance() #Getting distance in cm
    print(distance)
    print("Temperature: {}°C   Humidity: {:.0f}% ".format(temp, humidity))
    print("SMOKE:"+ str(sval) + "   GAS:" + str(sval1))
    print("s1:"+ str(sval) + " M:" + str(mval))
    print("L:"+ str(lval))

    line1="T:"+ str(temp) + " H:" + str(humidity) + " L:" + str(lval)
    line2="s1:"+ str(sval) + " s2:" + str(mval) + " L:" + str(distance)

    if(temp>40 or humidity>85 or sval>50 or  sval1>50 or lval<100):
        buz.value(1)
        time.sleep(1)
        buz.value(0)
        time.sleep(1)

    setUpLCD()
    rs.value(1)
    for x in line1:
        send2LCD8(ord(x))
```

```
time.sleep(0.02)
rs.value(0)
time.sleep(0.02)
send2LCD8(0b11000000)#clear screen
time.sleep(0.02)

rs.value(1)
for x in line2:
send2LCD8(ord(x))
time.sleep(0.02)
second=second+1
if(second==16):
 uart0.write(str(temp)+","+str(humidity) + ","+str(sval)+ ","+str(sval1)+
","+str(lval)+",0\n")
  second=0
print("uploading")
```