A MINI- PROJECT REPORT ON

"Weather Forecasting and Pollution Monitoring System"

SUBMITTED BY

Mr.NIKHILINGALE (17IT3001)

Mr.VAIBHAV KAMBLE(17IT1016)

Mr.VIJAY DUBHALKAR(17IT1002)

Supervisor

Dr. NIKITA KULKARNI

Department of Information Technology

Dr. D.Y.Patil Group's

Ramrao Adik Institute of Technology

Nerul, Navi Mumbai
(Affiliated to University of Mumbai)

(2019)

CERTIFICATE

This is to certify that the project entitled **" Weather Reporting and Pollution Monitoring System "** being submitted by **Nikhil Ingale (17IT3001) Vaibhav Kamble (17IT1016) and Vijay Dubhalkar (17IT1002)** to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of **'TE / IT'** in **"Internet of Things"**.

Project Guide	External Examiner	Head of Department
(Dr. Nikita Kulkarni)	( )	(Dr. Ashish Jadhav)

DECLARATION

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Name and Roll No. of Student:	**Signature:**
1. NIKHIL INGALE (17IT3001)	( )
2. VAIBHAV KAMBLE (17IT1016)	( )
3. VIJAY DUBHALKAR (17IT1002)	( )

DATE:

PLACE:

ACKNOWLEDGEMENT

The project **“ Weather Forecasting and Pollution Monitoring System ”** is creative work of many minds. A proper synchronization between individual it must for any project to be completed successfully. One cannot imagine the power of the force that guides us all and neither can we succeed without acknowledging it.

We would like to express our gratitude to principal **Dr. Mukesh D Patil** and **Dr.Ashish Jadhav,** our Head of the department, Information Technology Engineering for encouraging and inspiring us to carry out the project in the department lab.

We would also like to thank our Guide **Dr. Nikita Kulkarni**, Department of the Information Technology Engineering for her expert guidance, encouragement and valuable suggestions at every step.

We also would like to thank all the staff members Department of the Information Technology Engineering for providing us with the required facilities and support towards the completion of the project.

Last but not the least we are thankful to our parents and friends for their constant Inspiration, encouragement and well wishes by which we have made a challenging project.

PREFACE

We take great opportunity to present this mini project report on "**Weather Forecasting and Pollution Monitoring System"** and put before readers some useful information regarding our project. We have made sincere attempts and taken every care to present this matter in precise and compact form, the language being as simple as possible. We are sure that the information contained in this volume certainly prove useful for better insight in the scope and dimension of this project in it true perspective. The task of the completion of the project though being difficult was made quite simple, interesting and successful due to deep involvement and complete dedication of our group members.

TABLE OF CONTENTS

TABLE OF FIGURES

ABSTRACT

IoT is one of the dominant position all over the world in technological development. It is another information industry following computer, Internet and mobile communication In IOT enabled weather monitoring system, Raspberry Pi measures 2 weather parameters using DHT11 sensor. This sensor measures Humidity and Temperature. This sensor is directly connected to Raspberry Pi since it has inbuilt Analog to digital converter. In the system, IOT is coupled with Machine Learning. Using Machine Learning the outputs of the sensor are compared with the readings of some weather stations established in India and the reporting is done accordingly. It also uses Rain-drop sensor for the presence of rain.

This system also includes pollution monitoring which uses 2 sensors respectively.

One is MQ2-Gas sensor and other is Noise sensor. Both of these sensors are to be placed in certain locations where pollution is severe and needs to be monitored frequently. These sensors will detect any major change and will alert the officials about the condition of that particular region.

1. Introduction to IOT :

The IOT concept was coined by a member of the Radio Frequency Identification (RFID) development community in 1999, and it has recently become more relevant to the practical world largely because of the growth of mobile devices, embedded and ubiquitous communication, cloud computing and data analytics. Internet of Things (IoT) is an ecosystem of connected physical objects that are accessible through the internet. The 'thing' in IoT could be a person with a heart monitor or an automobile with built-in-sensors, i.e. objects that have been assigned an IP address and have the ability to collect and transfer data over a network without manual assistance or intervention. The embedded technology in the objects helps them to interact with internal states or the external environment, which in turn affects the decisions taken.

Internet of Things can connect devices embedded in various systems to the internet, When devices/objects can represent themselves digitally, they can be controlled from anywhere.

The connectivity then helps us capture more data from more places, ensuring more ways of increasing efficiency and improving safety and IoT security.

IoT is a transformational force that can help companies improve performance through IoT analytics and IoT Security to deliver better results. Businesses in the utilities, oil &amp gas, insurance manufacturing, transportation, infrastructure and retail sectors can reap the benefits of IoT by making more informed decisions, aided by the torrent of interactional and transactional data at their disposal.

1.1 Characteristics of IoT:

Intelligence :

Together algorithms and compute (i e. software & amp: hardware) provide the "intelligent spark" that makes a product experience smart Consider Misfit Shine, a fitness tracker, compared to Nest's intelligent thermostat. The Shine experience distributes compute tasks between a smartphone and the cloud. The Nest thermostat has more compute horsepower for the Al that make them smart.

Connectivity:

Connectivity in the IoT is more than slapping on a Wi-Fi module and calling it a day, Connectivity enables network accessibility and compatibility. Accessibility is getting on a network while compatibility provides the common ability to consume and produce data. If this sounds familiar, that's because it is Metcalfe's Law and it rings true for IoT.

Sensing:

We tend to take for granted our senses and ability to understand the physical world and people around us. Sensing technologies provide us with the means to create experiences that reflect a true awareness of the physical world and the people in it.

Expressing:

Expressing enables interactivity with people and the physical world. Whether it is a smart home or a farm with smart agriculture technology, expressing provides us with a means to create products that interact intelligently with the real world. This means more than just rendering beautiful UIs to a screen Expressing allows us to output into the real world and directly interact with people and environment.

Energy:

Without energy we can't bring our creations to life. The problem is we can't create billions of things that all run on batteries. Energy harvesting, power efficiency, and charging infrastructure are necessary parts a power intelligent ecosystem that we must design. Today, it is woefully inadequate and lacks the focus of many product teams.

Safety:

As we gain efficiencies, novel experiences, and other benefits from the IoT, we must not forget about safety. As both the creators and recipients of the IoT, we must design for safety. This includes safety of our personal data and the safety of our physical well-being. Securing the endpoints, the networks and the data moving across all of it means creating a security paradigm that will scale.

1.2 ARCHITECTURE:

The Internet of Things is considered as the third wave of the World Wide Web (WWW) after static web pages and social networking based web. The IOT is a worldwide network that connects different type of objects at anytime and anywhere via a popular internet protocol named Internet Protocol (IP). According to most of the researcher's opinions about conventional IoT architecture, it is considered as three layers: - 1) Perception Layer 2) Network Layer 3) Application Layer In other aspects, some researchers analyzed one more layer which is also included in IoT's latest architecturethat is a support layer that lies between the application layer and network layer. The support layer consists of fog computing and cloud computing. The cloud computing is also the hottest topic today in research. The perception layer is also called the recognition layer. The perception layer is the lowest layer of the conventional architecture of IoT. This layer's main responsibility is to collect useful information/data from things or the environment (such as WSN, heterogeneous devices, sensors type real world objects, humidity, and temperature etc.) and transform them in a digital setup. The main purpose of objects is unique address identification and communication between short-range technologies such as RFID, Bluetooth. Near-FieldCommunication (NFC), 6LOWPAN (Low Power Personal Area Network This layer is the brain of conventional IoT architecture This layer's main responsibility is to help and secure data transmission between the application and perception layer of IoT architecture. This layer mainly collects information and delivers to the perception layer toward several applications and servers. Basically, this layer is a convergence of internet and communication- based networks. According to a current study performed on several communication - based technologies researchers concluded that the network layer is the most developed layer of conventional IOT architecture.

1.3 Problem Statement :

Typical weather stations have the following instruments:

- Thermometer for measuring air temperature
- Barometer for measuring atmospheric pressure
- Hygrometer for measuring humidity
- Anemometer for measuring wind speed
- Ultra Sound Sensor for sensing Excessive Noise
- Advance Gas Sensor for detecting pollution in Air

These meteorological instruments are very expensive in India, resulting in low resolution of stations.

There is a need of low cost and high accuracy systems which can be used to improve the monitoring over a wide area with high resolution.

Today in hospitals, agriculture and for learnings purposes, the accurate local environmental parameters are needed. Hence a low cost high precision model can solve this problem.

1.4 Objective:

- The aim of this project is to achieve a completely automatic low cost weather station with IOT and Raspberry Pi.
- To achieve a good accuracy of results we will be comparing the outputs of the sensors with the readings of some weather stations established in India.
- Weather monitoring, pollution monitoring and warning the Officials about the pollution.

2 Literature Survey :

1. FireWxNet: A Multi-Tiered Portable Wireless System for Monitoring Weather Conditions in Wildland Fire Environments∗[1]
   .Authors: Carl Hartung, Richard Han
   In this paper FireWxNet, a multi-tiered portable wireless system for monitoring weather conditions in rugged wildland fire environments is presented. FireWxNet provides the fire fighting community the ability to safely and easily measure and view fire and weather conditions over a wide range of locations and elevations within forest fires.

2. PEGASUS FLX™ PORTABLE WEATHER STATION[2]
   Author: Columbia Weather Station
   High resolution wind direction sensor and up to 160 mph wind speed range
   Digital sensors calibrated for accurate readings, yet tough enough for emergency environments.
   Quite expensive to implement in all remote areas.

3. A LOW COST ZIGBEE BASED AUTOMATIC WIRELESS WEATHER STATION WITH GUI AND WEB HOSTING FACILITY[3]
   Authors: Nitant Sabharwal, Rajesh Kumar, Abhishek Thakur, Jitender Sharma
   In this paper we have proposed and developed a low cost hardware module based on Arduino Uno Board, which measures the meteorological data, including air temperature, atmospheric pressure, relative humidity, dew point temperature, wind speed and wind direction. It sends this information to the GUI running on a PC through Zigbee wireless link.
   Lacks fire alert and pollution monitoring system.

4. A wireless sensor network air pollution monitoring system in Industrial areas.[4]
   Authors: Kavi K. Khedo, Rajiv Perseedoss and Avinash Mungur
   This paper proposes an industrial air pollution monitoring system based on wireless sensor network system that enables sensor data to be delivered within time constraints so that appropriate observations can be made or actions taken. Obtaining these accurate real-time results in-situ allows regulatory agency to take necessary action whenever pollution occurs.
   Lacks Weather forecasting System.

5. SenseWeather: Sensor-Based Weather Monitoring System for Kenya.[5]
   Authors: Muthoni MASINDE, Antoine BAGULA , Muthama NZIOKA.
   In this paper, we have described the design and development of SenseWeather; an application that uses sensor-based weather station to complement the sparse network of professional weather stations in Kenya. The system is designed to integrate weather readings from the stations with parameters (temperature, pressure, relative humidity, precipitation, soil moisture, wind speed and wind direction) recorded by the sensors.

2.1 Motivation

A weather Reporting station is that facility on land or sea, which has the instruments and devices for observing and measuring atmospheric parameters to provide the information for weather forecasts. Weather monitoring has been very important to man over centuries. Weather parameters are the basic tools which can be used to predict the weather. Automatic wireless weather stations make use of digital sensors and microcontroller technology to take the measurements. Automatic wireless weather stations are small in size and easy to install and they also save the manpower.

3 Proposed System :

3.1 Introduction of propose system and architecture

- The proposed system measures the real time weather parameters automatically.
- It uses low cost high precision calibrated digital sensors for accurate data measurements.
- This system can record the data and host it on the internet for remote viewing of weather conditions.
- And it generates the short term local alerts based on current weather and pollution conditions.

Architecture

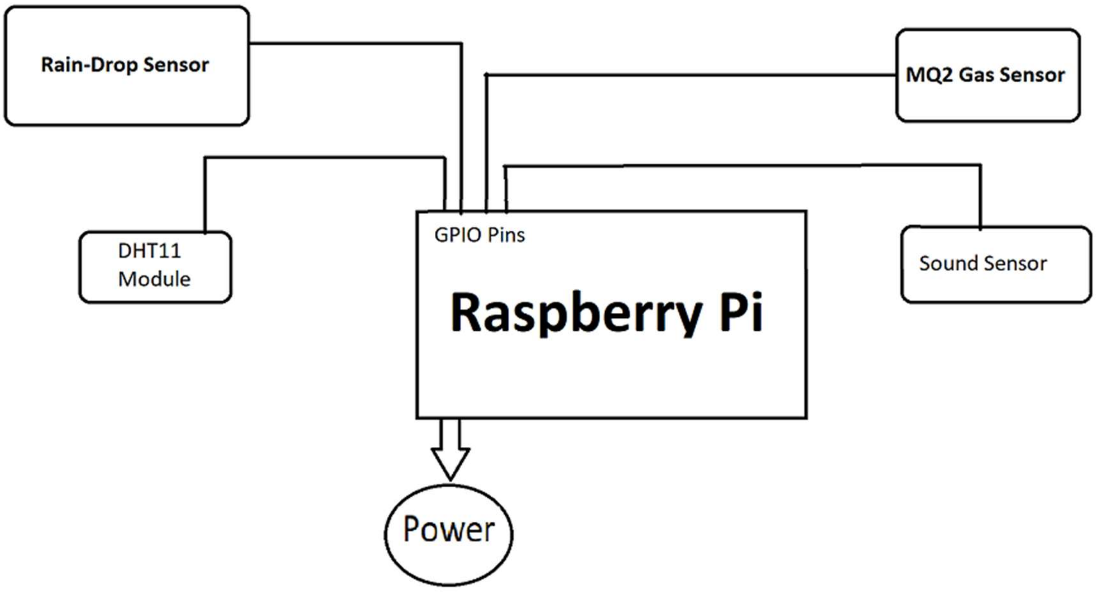

Fig.1 Architecture

3.2 Hardware and Software Requirement

Raspberry pi 3b

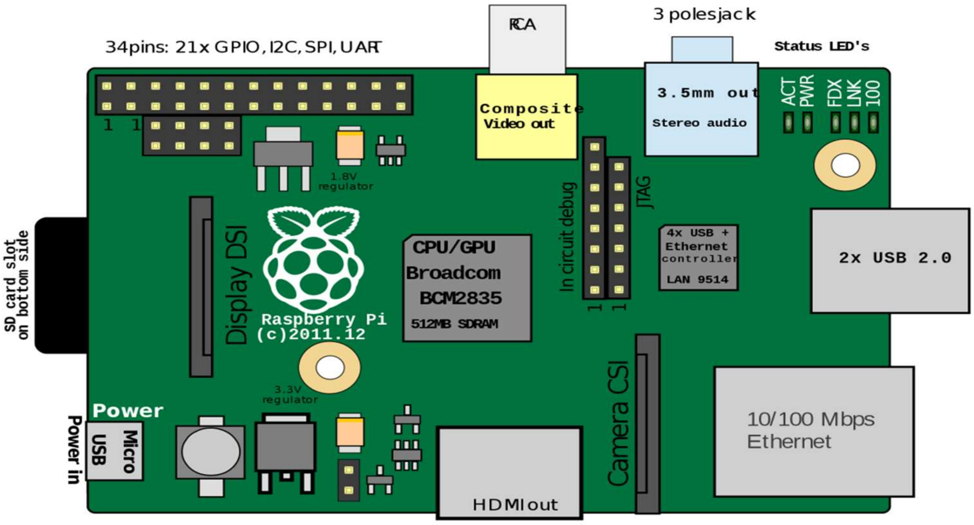

Fig.2 RaspberryPi

Fig.3.1 MQ2 Gas Sensor

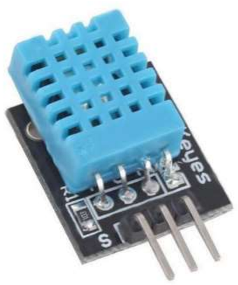

Fig.3.2 DHT11 Module

Fig.3.3 Rain Drop Sensor

Fig.3.4 Sound Sensor

1. DHT11 Module:

   DHT11 is a Digital Sensor consisting of two different sensors in a single package. The sensor contains an NTC (Negative Temperature Coefficient) Temperature Sensor, a Resistive-type Humidity Sensor and an 8-bit Microcontroller to convert the analog signals from these sensors and produce a Digital Output.

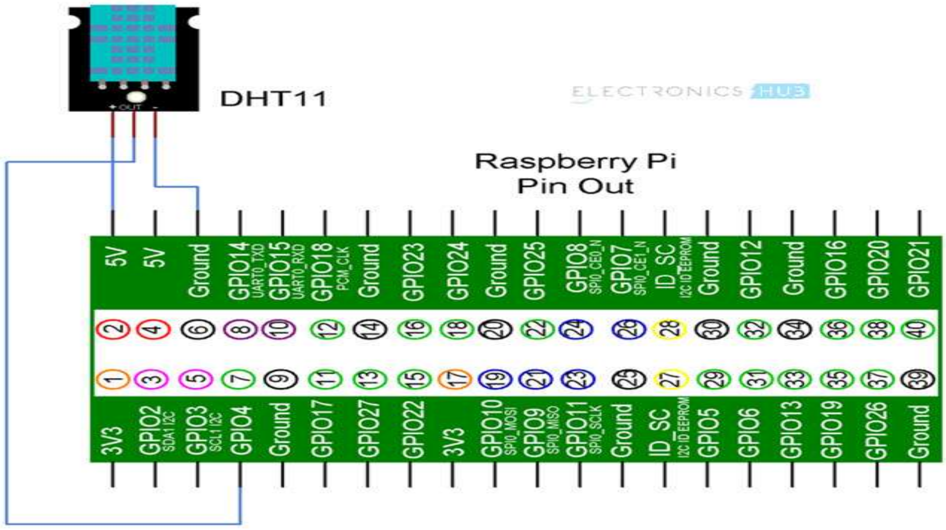

Fig.3.5: DHT11 Circuit Diagram

2. Rain Drop Sensor

   Raindrop sensor is basically a board on which nickel is coated in the form of lines. It works on the principal of resistance. Rain Sensor module allows to measure moisture via analog output pins and it provides a digital output when a threshold of moisture exceeds. The module is based on the LM393 op amp. It includes the electronics module and a printed circuit board that “collects” the rain drops. As rain drops are collected on the circuit board, they create paths of parallel resistance that are measured via the op amp. The sensor is a resistive dipole that shows less resistance when wet and more resistance when dry. When there is no rain drop on board it increases the Resistance so we gets high voltage according to V=IR. When rain drop present it reduces the resistance because water is a conductor of electricity and presence of water connects nickel lines in parallel so reduces resistance and reduces voltage drop across it.

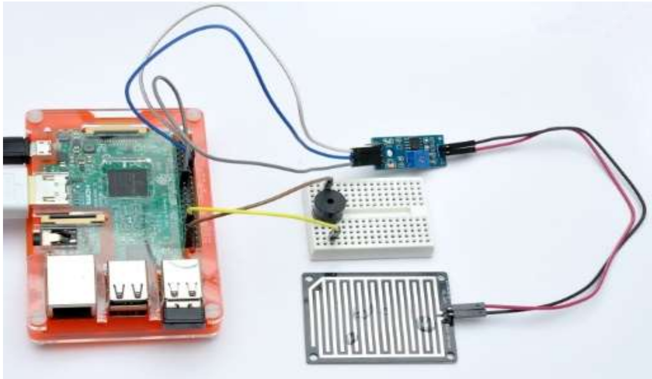

Fig.3.6: Raindrop Sensor Circuit Diagram

3. MQ2 Gas Sensor

The MQ-2 smoke sensor reports smoke by the voltage level that it outputs. The more smoke there is, the greater the voltage that it outputs. Conversely, the less smoke that it is exposed to, the less voltage it outputs. The MQ-2 also has a built-in potentiometer to adjust the sensitivity to smoke. By adjusting the potentiometer, you can change how sensitive it is to smoke, so it's a form of calibrating it to adjust how much voltage it will put out in relation to the smoke it is exposed to.

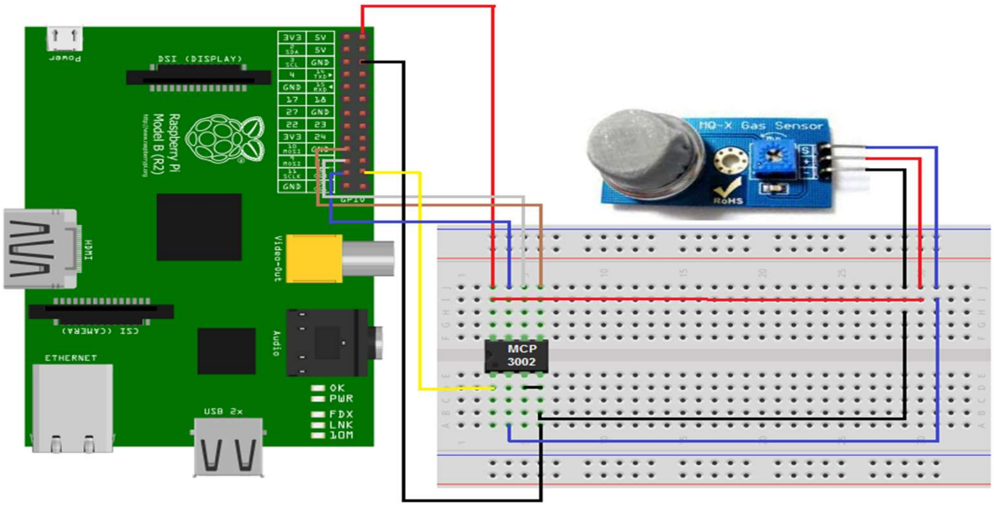

Fig. 3.7: MQ2 Gas Sensor Circuit Diagram

4. Sound Sensor

The sound sensor is one type of module used to notice the sound. Generally, this module is used to detect the intensity of sound. The applications of this module mainly include switch, security, as well as monitoring. The accuracy of this sensor can be changed for the ease of usage. This sensor employs a microphone to provide input to buffer, peak detector and an amplifier. This sensor notices a sound, & processes an o/p voltage signal to a microcontroller. After that, it executes required processing. This sensor is capable to determine noise levels within DB's or decibels at 3 kHz 6 kHz frequencies approximately wherever the human ear is sensitive. In smartphones, there is an android application namely decibel meter used to measure the sound level.

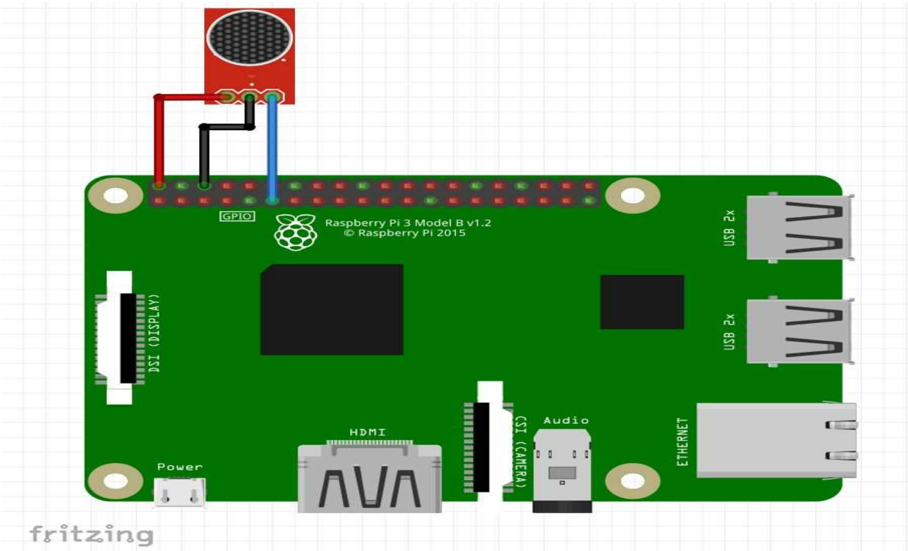

Fig. 3.8: Sound Sensor Circuit Diagram

Software:

- Raspbian OS:

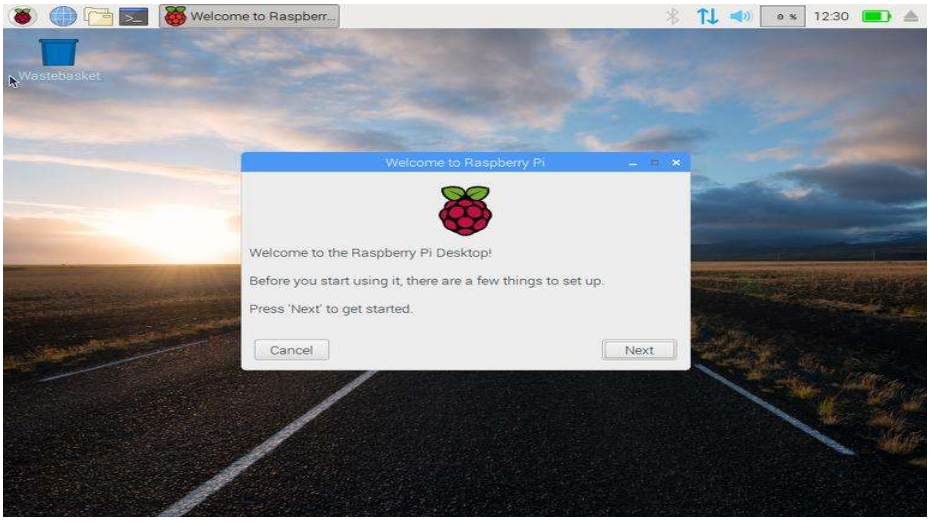

Fig.3.9 Raspbian OS

- Blynk:

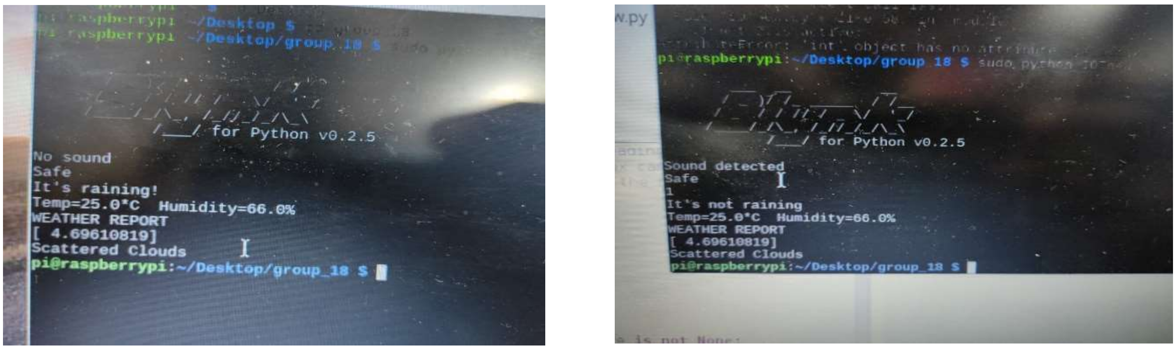

Fig 3.10: Blynk Output

- VNC Viewer
- Putty

4 Implementation

4.1 Circuit Design

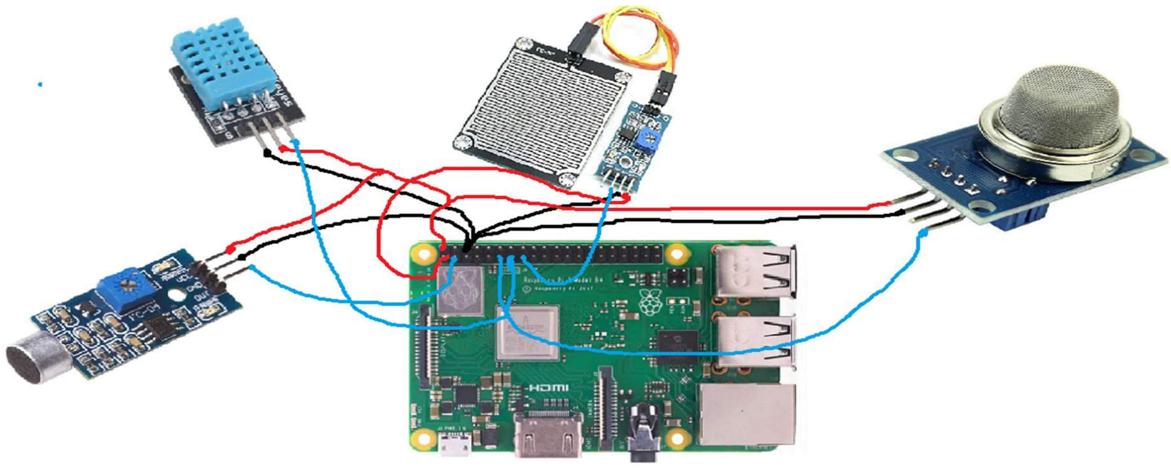

Fig. 4.1: Project Circuit Diagram

4.2 Connection Details

DHT11 Module, Rain-Drop Sensor, MQ2 Gas Sensor, Sound Sensor is connected to Raspberry Pi.

DHT11 Module consists 3 pins: Vcc, Data, Ground.
Vcc of DHT11 is connected to 5V pin of Raspberry Pi
Data pin is connected to GPIO pin #27
Ground pin is connected to GND of RPi

Rain-Drop Sensor Module consists of 4 pins: Vcc, Ground, AO, DO
Its Vcc is connected to 3.3V output pin of Raspberry Pi
Ground pin is connected to ground of RPi
AO pin is left open
DO pin is connected to GPIO pin #22

MQ2 Gas Sensor Module also has 4 pins: DOUT, AOUT, Ground, Vcc
DOUT pin is connected to GPIO pin #17 of RaspberryPi
AOUT pin is left open as we are not using analog output
Ground pin is connected to ground of RPi
Vcc is connected to 5.5V of RPi

Sound Sensor consists of 3 pins: Vcc, Ground, OUT
Vcc of this sensor is connected to 5V o/p of Raspberry Pi
Ground is connected to ground pin of RPi
OUT pin is connected to GPIO Pin #2 of Raspberrypi

Blynk is used to notify the user about the weather conditions.
Blynk combines a cloud platform with applications that put things, people, and data at the heart of business operations.

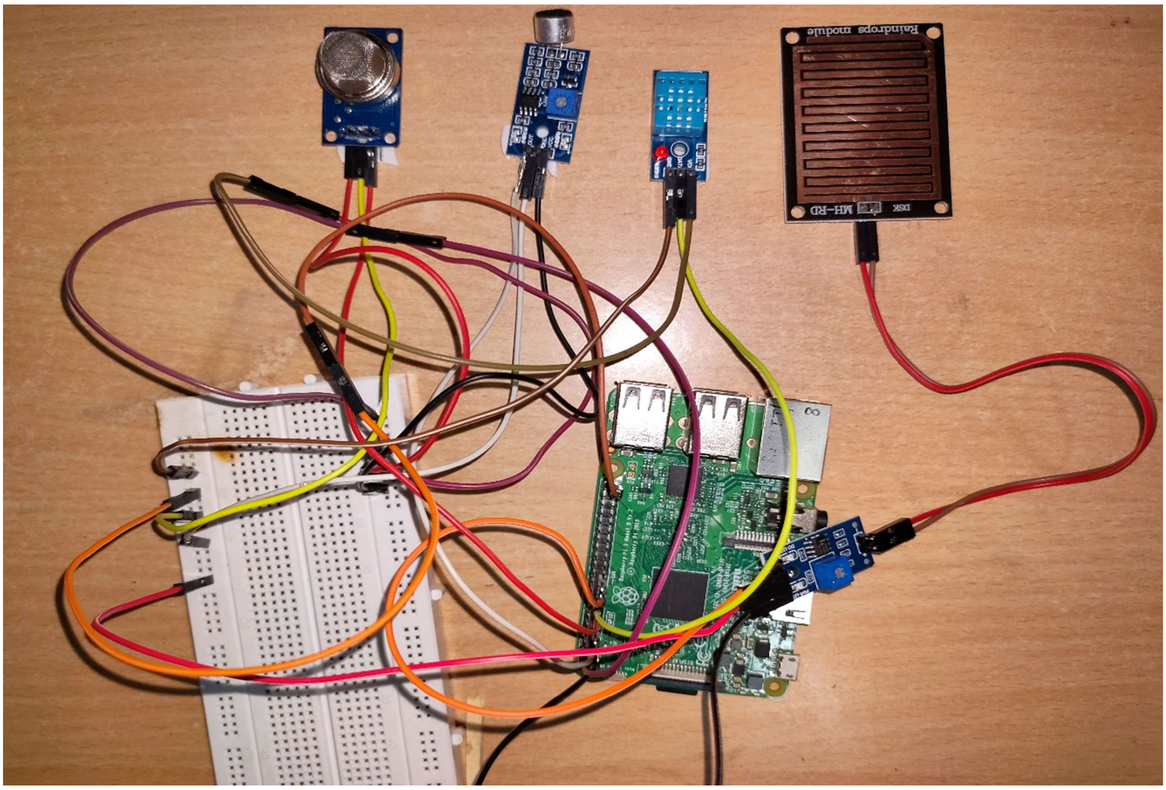

Fig. 4.2: Circuit Connections

4.3 Code

```
import RPi.GPIO as GPIO
import time
import Adafruit_DHT
from sklearn import linear_model
from sklearn import preprocessing
import pandas as pd
import blynklib

pin=27
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(2,GPIO.IN) #sound
GPIO.setup(3,GPIO.OUT) #sound/output
GPIO.setup(27,GPIO.IN) #temperature DHT11
GPIO.setup(17,GPIO.IN) #gas
GPIO.setup(22,GPIO.IN) #raindrop
GPIO.setup(4,GPIO.OUT) #buzzer
sensor = Adafruit_DHT.DHT11
df=pd.read_csv("Weather3.csv")
X = df[['HUMIDITY', 'TEMPERATURE']]
y = df['WEATHER FORECAST']


reg = linear_model.LinearRegression()
reg.fit(df[['HUMIDITY', 'TEMPERATURE']], df['WEATHER FORECAST'])
BLYNK_AUTH='IpOfhJ5EehVIZb86tB7EYXFA_MsennTg'
blynk=blynklib.Blynk(BLYNK_AUTH)


while(True):
    blynk.run()
      x=GPIO.input(2)
      if x==1:
              GPIO.output(3,GPIO.LOW)
              print("Sound detected")
              blynk.notify("Too much sound in the zone!!!")
      else:
              print('No sound')

    x=GPIO.input(17)

    if x==0 :
```

```
            blynk.notify("Fire Alert!!!")
            print("Fire Alert!!!")
            GPIO.output(4,GPIO.HIGH)
            time.sleep(1)
            GPIO.output(4,GPIO.LOW)
        else :
            print("Safe ")



        # Try to grab a sensor reading.  Use the read_retry method which will retry up
        # to 15 times to get a sensor reading (waiting 2 seconds between each retry).
        humidity, temperature = Adafruit_DHT.read_retry(sensor, pin)

        # Note that sometimes you won't get a reading and
        # the results will be null (because Linux can't
        # guarantee the timing of calls to read the sensor).
        # If this happens try again!



        z=GPIO.input(22)
        if not z.is_active:
            blynk.notify("It's raining!")
            print("It's raining!")
        else :
            print("It's not raining")

        if humidity is not None and temperature is not None:
            print('Temp={0:0.1f}*C  Humidity={1:0.1f}%'.format(temperature,
humidity))
            fahrenheit=float((temperature*1.8)+32)  #formula

            X = df[['HUMIDITY', 'TEMPERATURE']]

            y = df['WEATHER FORECAST']

            le = preprocessing.LabelEncoder()#making LabelEncoder function variable

            df = df.apply(le.fit_transform)#this is used to convert string values into
integer values

            reg = linear_model.LinearRegression()

            reg.fit(df[['HUMIDITY', 'TEMPERATURE']], df['WEATHER
FORECAST'])
```

```
        print("WEATHER REPORT")
        print(reg.predict([[humidity,fahrenheit]]))


        if(float(reg.predict([[humidity,fahrenheit]]))>=0 and
float(reg.predict([[humidity,fahrenheit]]))<1):
            try:
                blynk.notify('Light rain')
                print("Light rain")
            except KeyboardInterrupt:
                print("Exit")
                GPIO.cleanup()

        elif(float(reg.predict([[humidity,fahrenheit]]))>=1 and
float(reg.predict([[humidity,fahrenheit]]))<2):
            try:
                blynk.notify('Broken Clouds')
                print("Broken Clouds")
            except KeyboardInterrupt:
                print("Exit")
                GPIO.cleanup()

        elif (float(reg.predict([[humidity, fahrenheit]]))>=2 and
float(reg.predict([[humidity, fahrenheit]])) < 3):
            try:
                blynk.notify('Proximity Shower Rain')
                print("Proximity Shower Rain ")
            except KeyboardInterrupt:
                print("Exit")
                GPIO.cleanup()

        elif (float(reg.predict([[humidity, fahrenheit]]))>=3 and
float(reg.predict([[humidity, fahrenheit]])) < 4):
            try:
                blynk.notify('Sky is Clear')
                print("Sky is Clear")
            except KeyboardInterrupt:
                print("Exit")
                GPIO.cleanup()

        elif (float(reg.predict([[humidity, fahrenheit]]))>=4 and
float(reg.predict([[humidity, fahrenheit]])) < 5):
            try:
                blynk.notify('Scattered Clouds')
                print("Scattered Clouds")
            except KeyboardInterrupt:
```

```
                print("Exit")
                GPIO.cleanup()

        elif (float(reg.predict([[humidity, fahrenheit]])) >= 5 and
float(reg.predict([[humidity, fahrenheit]])) < 6):
            try:
                blynk.notify('Few Clouds')
                print("Few Clouds")
            except KeyboardInterrupt:
                print("Exit")
                GPIO.cleanup()

        elif (float(reg.predict([[humidity, fahrenheit]])) >= 6 and
float(reg.predict([[humidity, fahrenheit]])) < 7):
            try:
                blynk.notify('Squalls')
                print("Squalls")
            except KeyboardInterrupt:
                print("Exit")
                GPIO.cleanup()

        elif (float(reg.predict([[humidity, fahrenheit]])) >= 7 and
float(reg.predict([[humidity, fahrenheit]])) < 8):
            try:
                blynk.notify('Overcast Clouds')
                print("Overcast clouds ")
            except KeyboardInterrupt:
                print("Exit")
                GPIO.cleanup()

        elif (float(reg.predict([[humidity, fahrenheit]])) >= 8 and
float(reg.predict([[humidity, fahrenheit]])) < 9):
            try:
                        blynk.notify('Heavy Snow')
                print("Heavy Snow")
            except KeyboardInterrupt:
                print("Exit")
                GPIO.cleanup()

        elif (float(reg.predict([[humidity, fahrenheit]])) >= 9 and
float(reg.predict([[humidity, fahrenheit]])) < 10):
            try:
                blynk.notify('Mist')
                print("Mist")
            except KeyboardInterrupt:
                print("Exit")
```

```
                    GPIO.cleanup()

            elif (float(reg.predict([[humidity, fahrenheit]])) >= 10 and
float(reg.predict([[humidity, fahrenheit]])) < 11):
                try:
                    blynk.notify('Haze')
                    print("Haze")
                except KeyboardInterrupt:
                    print("Exit")
                    GPIO.cleanup()

            elif (float(reg.predict([[humidity, fahrenheit]])) >= 11 and
float(reg.predict([[humidity, fahrenheit]])) < 12):
                try:
                    blynk.notify('Fog')
                    print("Fog")
                except KeyboardInterrupt:
                    print("Exit")
                    GPIO.cleanup()

            else:
                print("not predicted")
        else:
            print('Failed to get reading. Try again!')

        time.sleep(10)

        break
```

5 Result

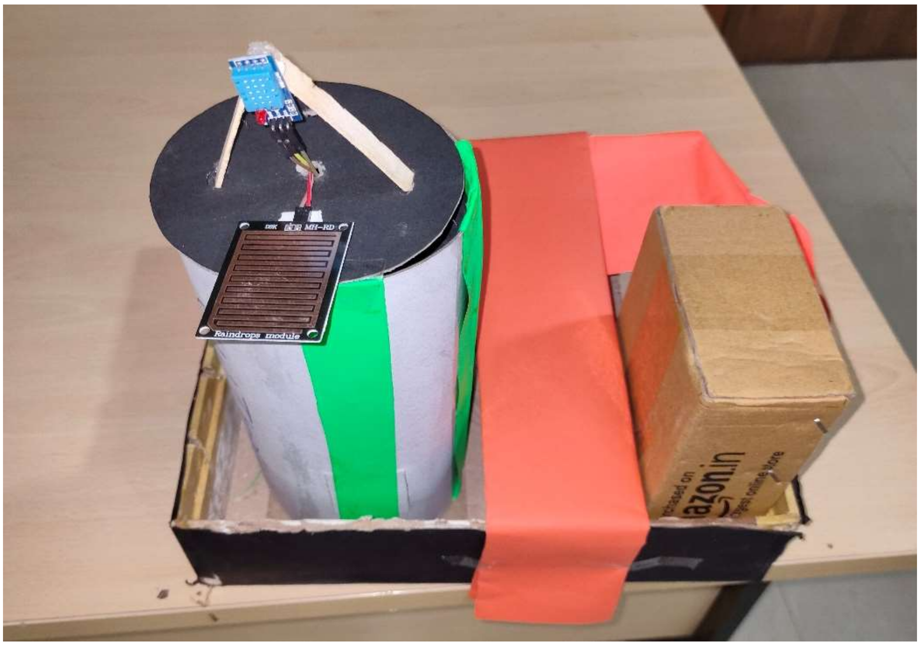

Fig. 5.1: Project Model

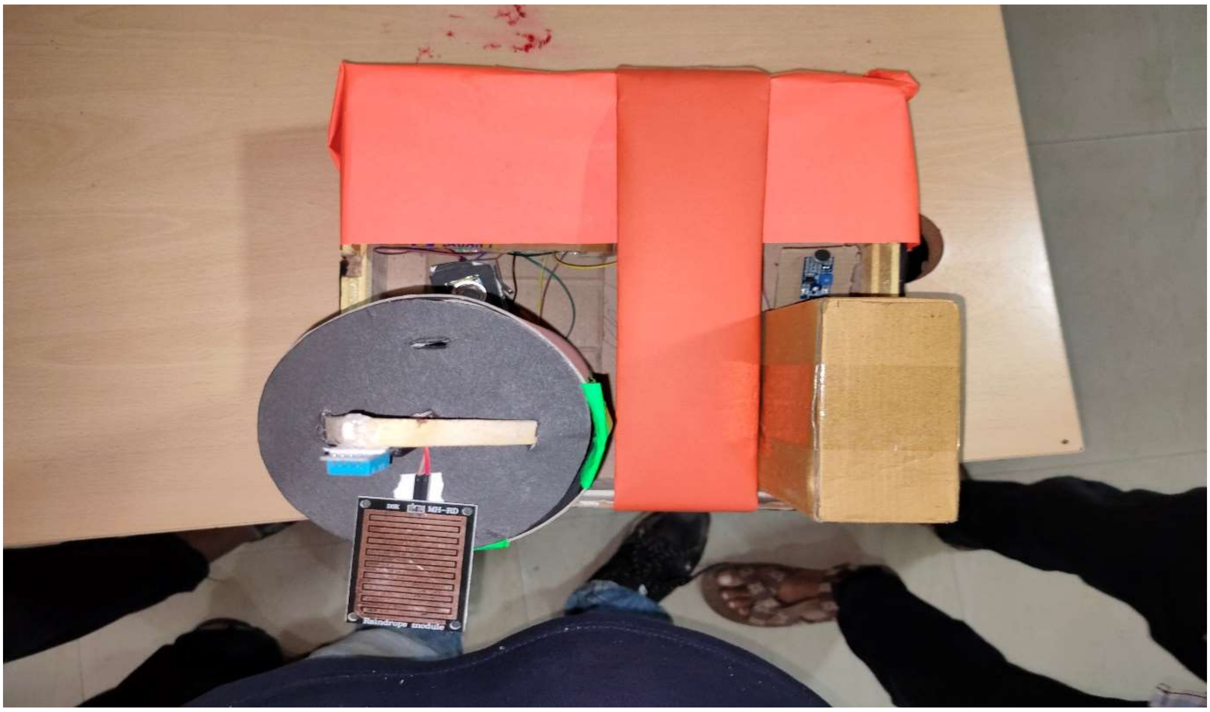

Fig. 5.2: Project Model

6 Conclusion and Future Scope

6.1 Conclusion

We have developed and set-up a low cost IOT and Raspberry-Pi based automatic weather & Pollution monitoring system with high accuracy. Data logging facility is working and it saves the data in a file compatible with MS Excel. System is able to generate alerts using Blynk System. The officials will be notified about the weather conditions.

In this way, an efficient weather reporting and pollution monitoring system has been developed.

6.2 Future scope

The features which can be developed in the future are:

One can implement a few more sensors and connect it to the satellite as a global feature of this system. Long time (> 7days) weather forecasting by using IoT with Machine Learning Technology. It can also be implemented in hospitals or medical institutes for the research & study in "Effect of Weather on Health and Diseases", hence to provide better precaution alerts. It can also proved to be useful in Agriculture for prediction of weather to avoid any loss of farmers.

6.3 Benefits to the society

Weather Reporting system have a wide range of applications. IOT based Weather Reporting system using raspberry pi can be used in Agricultural studies, Weather Analysis, predicting of future effects, etc. Pollution Monitoring system can be used in Hospitals where pollution control is very crucial factor. It can also be used to take certain measures in particular area where pollution is exceeding its limits.

7 References :

Papers:

- FireWxNet: A Multi-Tiered Portable Wireless System for Monitoring Weather Conditions in Wildland Fire Environments∗

- PEGASUS FLX™ PORTABLE WEATHER STATION, Columbia Weather Station.

- A LOW COST ZIGBEE BASED AUTOMATIC WIRELESS WEATHER STATION WITH GUI AND WEB HOSTING FACILITY

- A wireless sensor network air pollution monitoring system in Industrial areas.

- SenseWeather: Sensor-Based Weather Monitoring System for Kenya , Muthoni MASINDE, Antoine BAGULA , Muthama NZIOKA.

Websites:

- http://www.columbiaweather.com/Pegasus EX-Brochure

- https://microcontrollerslab.com/iot-weather-reporting-system-using-pic-microcontroller/

- https://nevonprojects.com/iot-weather-reporting-system/