Air quality monitor Using IOT

INTRODUCTION:

The world we live in is being rapidly automated and emerging technologies like Cloud, Internet of Things, and so forth are being continuously integrated into concepts such as Smart Cities to provide a high level of comfort to the residents with minimum human intervention . A major challenge faced by corporations of developed cities is to control and regulate air quality. With the advent of modern air quality monitoring and pollution control systems, a novel prediction framework aids the process of finding effective solutions to complex problems. This project focuses on investigating the correlation between air quality and weather and building a prediction model based on the results of the exploratory analysis of historical weather and pollution data.

PROBLEMS:

There are a number of concerns about the risks in the growth of IoT technologies and products, especially in the areas of privacy and security, and consequently, industry and governmental moves to address these concerns have begun, including the. development of international and local standards, guidelines, and regulatory frameworks.
IoT devices are a part of the larger concept of home automation, which can include lighting, heating and air conditioning, media and security systems and camera systems. Long-term benefits could include energy savings by automatically ensuring lights and electronics are turned off or by making the residents in the home aware of usage.
A smart toilet seat that measures blood pressure, weight, pulse and oxygen levels. A smart home or automated home could be based on a platform or hubs that control smart devics and appliances. For instance, using Apple's HomeKit, manufacturers can have their home products and accessories controlled by an application in iOS devices such as the iPhoneand the Apple Watch. This could be a dedicated app or iOS native applications such as Siri.

PROJECT OBJECTIVE:

Air pollution is one of the biggest threats to the present-day environment. Everyone is being affected by air pollution day by day including humans, animals, crops, cities, forests and aquatic ecosystems. Besides that, it should be controlled at a certain level to prevent the increasing rate of global warming. This project aims to design an IOT-based air pollution monitoring system using the internet from anywhere using a computer or mobile to monitor the air quality of the

surroundings and environment. There are various methods and instruments available for the measurement and monitoring quality of air. The IoT-based air pollution monitoring system would not only help us to monitor the air quality but also be able to send alert signals whenever the
air quality deteriorates and goes down beyond a certain level.

COMPONENTS USED IN PROJECT:

Hardware Components

1.NodeMCU V3
2.DHT11 Sensor Module
3.MQ-135 Gas Sensor Module
4.Veroboard(KS100)
5.Breadboard
6.Connecting Wires
7.AC-DC Adapters
8.LEDs emitting green, yellow and red colours
9.Resistors

SOFTWARE COMPONENTS

1. ThinkSpeak Cloud
2. Arduino IDE

Brief Description of the Components

NodeMCU V3:

NodeMCU V3 is an open-source ESP8266 development kit, armed with the CH340G USB-TTL Serial chip. It has firmware that runs on ESP8266 Wi-Fi SoC from Espressif Systems. Whilst cheaper, CH340 is super reliable even in industrial applications. It is tested to be stable on all supported platforms as well. It can be simply coded in Arduino IDE. It has a very low current consumption between 15 µA to 400 mA.
The pinout Diagram of NodeMC3 is shown in Fig. 2.1.

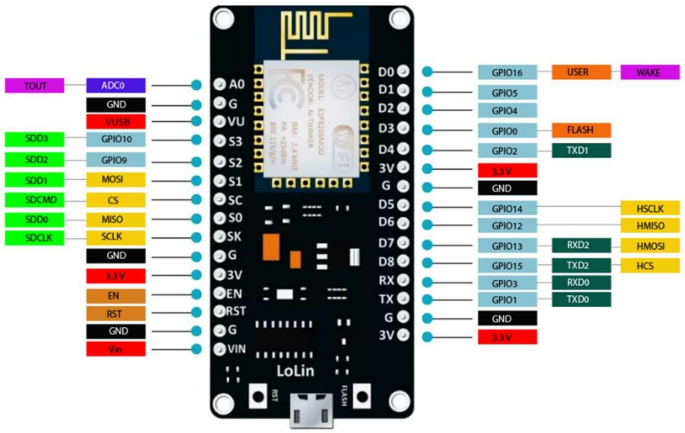


Fig. 2.1 (Pinout Diagram of NodeMCU V3)

DHT11 Sensor Module:

The DHT11 is a temperature and humidity sensor that gives digital output in terms of voltage. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air.
As shown in Fig. 2.2, we need to supply a voltage of 5V (DC) to the Vcc pin and ground it to the GND pin. The sensor output can be easily read from the Data pin in terms of voltage (in digital mode).
Humidity Measurement: The humidity sensing capacitor has two electrodes with a moisture-holding substrate as a

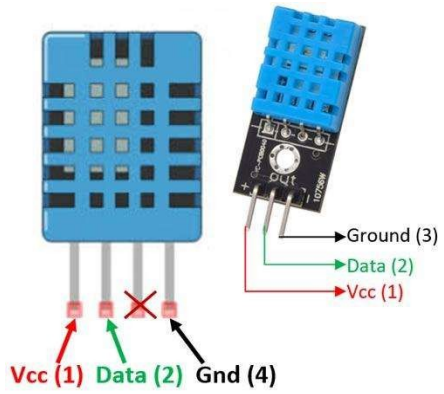
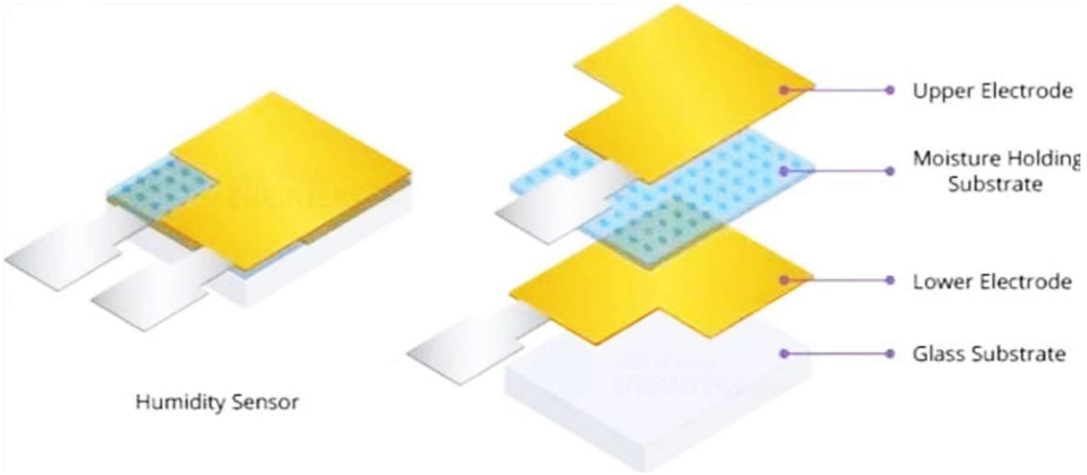


Fig 2.2 (Pinout Diagram of DHT11sensor)

dielectric between them as shown in Fig 2.3. Change in the capacitance value occurs with the change in humidity levels. The IC measure, process these changed resistance values and then converts them into digital form.

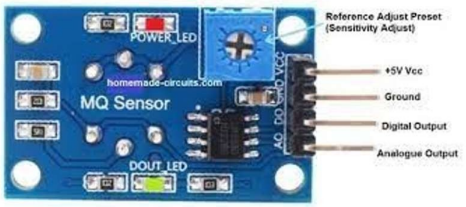


Temperature Measurement: For measuring the temperature, the DHT11 sensor uses a negative temperature coefficient thermistor, which causes a decrease in its resistance value with an increase in temperature. To get a wide range of resistance values, the sensor is made up of semiconductor ceramics or polymers.

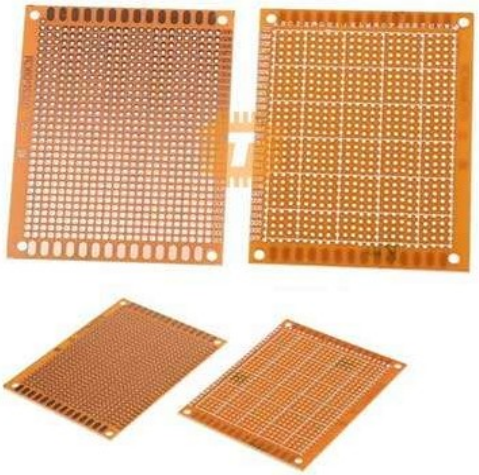
MQ-135 Gas Sensor Module:

The material of MQ135 is SnO2, it is a special material: when exposed to clean air, it is hardly being conducted, however, when put in an environment with combustible gas, it has a pretty performance of conductivity. Just make a simple electronic circuit, and convert the change of conductivity to a corresponding output signal. MQ135 gas sensor is sensitive to

Ammonia, Sulphide, Benzene steam, smoke and other harmful gases. Used for family, surrounding environment noxious gas detection device, apply to ammonia, aromatics, sulphur, benzene vapor, and other harmful gases/smoke, gas detection, tested concentration range: 10 to 1000ppm. In a normal environment, the environment which doesn't have detected gas set the sensor's output voltage as the reference voltage, the analog output voltage will be about 1V, when the sensor detects gas, harmful gas concentration increases by 20ppm per voltage increase by 0.1V.



Veroboard (KS100):



Veroboard is the original prototyping board. Sometimes referred to as 'stripboard' or 'matrix board' these offer total flexibility for hard wiring discrete components. Manufactured from a copper clad laminate board or Epoxy based substrate, it is offered in both single and double-sided formats. Vero boards are available in a wide range of board sizes and in both imperial and metric pitch – Veroboard is an ideal base for circuit construction and offers even greater

adaptability using our range of terminal pins and assemblies. As with other stripboards, in using
Fig 2.5 Veroboard

Veroboard, components are suitably positioned and soldered to the conductors to form the required circuit. Breaks can be made in the tracks, usually around holes, to divide the strips into multiple electrical nodes enabling increased circuit complexity. This type of wiring

board may be used for initial electronic circuit development, to construct prototypes for bench testing or in the production of complete electronic units in small quantities.

AC-DC Power Adapter:

An AC-DC power supply or adapter is an electrical device that obtains electricity from a grid-based power supply and converts it into a
different current, frequency, and voltage. AC-DC power supplies are necessary to provide the right power that an electrical component needs. The AC- DC power supply delivers electricity to devices that would typically run-on batteries or have no other power source.

LED (Red, Green & Yellow):

A light-emitting diode (LED) is a semiconductor light source that emits light when current flows through it. Electrons in the semiconductor recombine with electron holes, releasing energy in the form of photons. The colour of the light (corresponding to the energy of the photons) is determined by the energy required for electrons to cross the band gap of the semiconductor. White light is obtained by using multiple semiconductors or a layer of light-emitting phosphor on the semiconductor device. LEDs have many advantages over incandescent light sources, including lower power consumption, longer lifetime, improved physical robustness, smaller size, and faster switching. In exchange for these generally favourable attributes, disadvantages of LEDs
include electrical limitations to low voltage and generally to DC (not AC) power, inability to provide steady illumination from a pulsing DC or an AC electrical supply source, and lesser maximum operating temperature and storage temperature. In contrast to LEDs, incandescent
lamps can be made to intrinsically run at virtually

any supply voltage, can utilize either AC or DC
Fig 2.7LEDs

current interchangeably, and will provide steady illumination when powered by AC or pulsing DC even at a frequency as low as 50 Hz. LEDs usually need electronic support components to function, while an incandescent bulb can and usually does operate directly from an unregulated DC or AC power source.

RESISTOR:

A resistor is a passivetwo-terminalelectrical component that implements electrical resistance as a circuit element. In electronic circuits, resistors are used to reduce current flow, adjust signal levels, to divide voltages, bias active elements, and terminate transmission lines, among other uses. High-power resistors that can dissipate many watts of electrical power as heat may be used as part of motor

controls, in power distribution systems, or as test loads for generators. Fixed resistors have resistances

Fig 2.8 Resistors

that only change slightly with temperature, time or operating voltage.

ARDUINO IDE:

The Arduino IDE is open-source software, which is used to write and upload code to the Arduino boards. The IDE application is suitable for different operating systems such as Windows, Mac OS X, and Linux. It supports the programming languages C and C++. Here, IDEstands for Integrated Development Environment. The program or

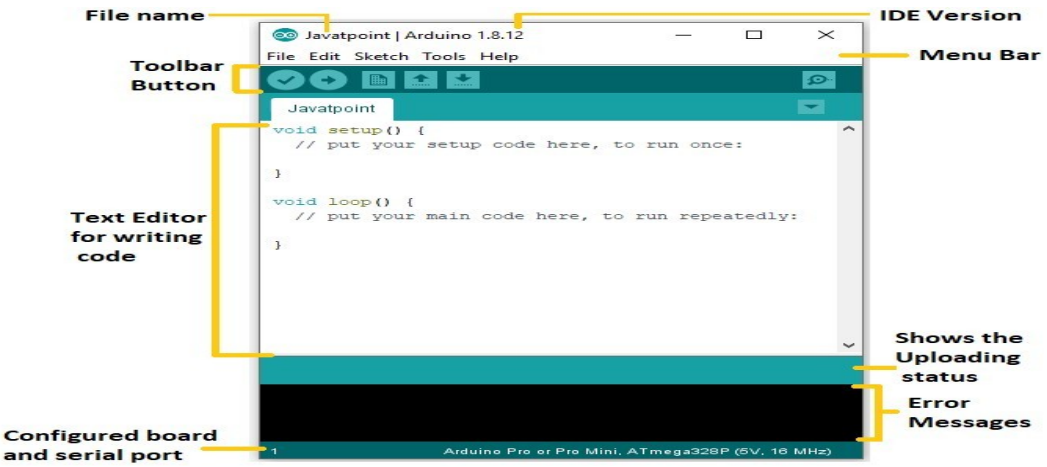


Fig 2.9Arduino IDE

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

ThingSpeak Cloud:

ThingSpeak is open-source software written in Ruby which allows users to communicate with internet-enabled devices. It facilitates data access, retrieval and logging of data by providing an API to both the devices and social network websites. ThingSpeak was originally launched by ioBridge in 2010 as a service in support of IoT applications. ThingSpeak has integrated support from the numerical computing software MATLAB from MathWorks, allowing ThingSpeak users

Fig 2.10 ThingSpeak Cloud

to analyse and visualize uploaded data using MATLAB without requiring the purchase of a MATLAB license from MathWorks.

Working Procedures:

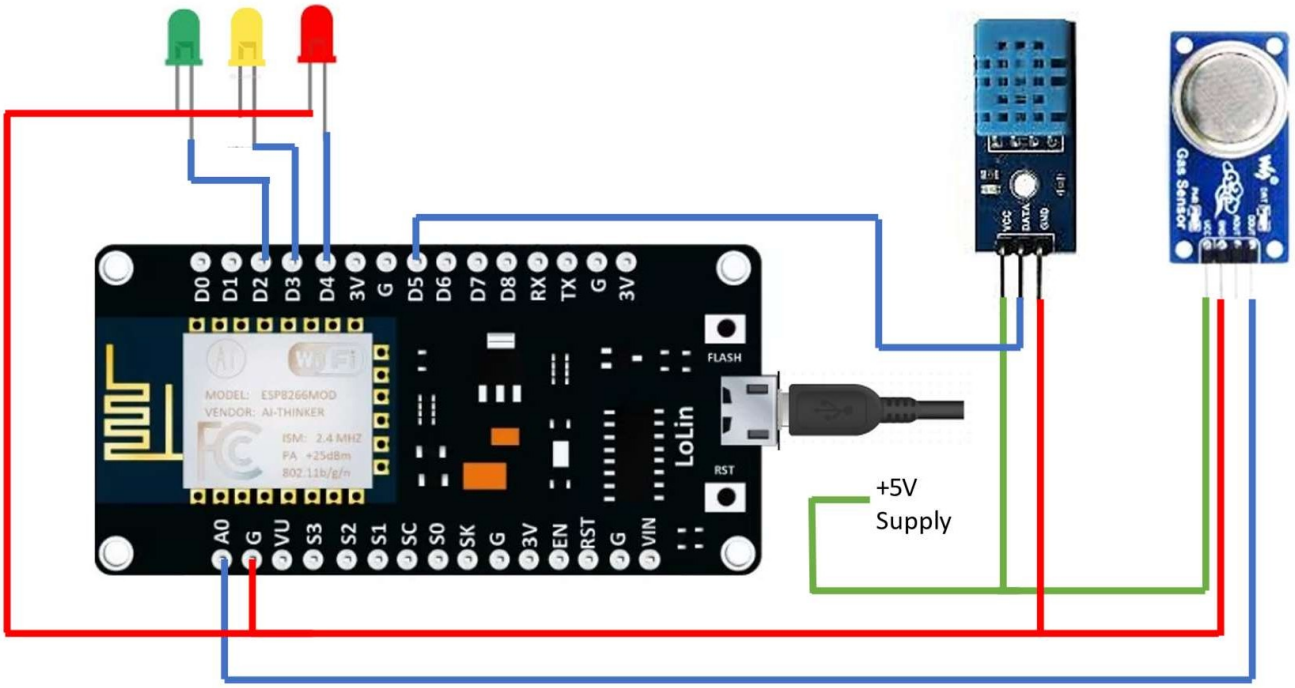
NodeMCU plays the main controlling role in this project. It has been programmed in a manner, such that, it senses the sensory signals from the sensors and shows the quality level via led indicators. The DHT11 sensor module is used to measure the temperature and the humidity of the surroundings.With the help of the MQ-135 gas sensor module, air quality is measured in ppm. These data are fed to the ThinkSpeak cloud over the internet. We have also provided LED indicators to indicate the safety levels.

STEP 1. Firstly, the calibration ofthe MQ-135 gas sensor module is done. The sensor is set to preheat for 24 minutes. Then the software code is uploaded to the NodeMCU followed by the hardware circuit to calibrate the sensor has been performed.

STEP 2. Then, the DHT11 sensor is set to preheat for 10 minutes.

STEP 3. The result of calibration found in STEP 1 is used to configure the final working code.
STEP 4. The final working code is then uploaded to the NodeMCU. STEP 5. Finally, the complete hardware circuit is implemented.
The software codes and the hardware circuits are described in the following chapters.

SCHEMATIC DIAGRAM:



PROGRAM SETUP:

```
void setup()
{
Serial.begin(9600); //Baud rate

pinMode(A0,INPUT);
}

void loop()
{
float sensor_volt; //Define variable for sensor voltage float RS_air; //Define variable for sensor resistance float R0; //Define variable for R0
float sensorValue=0.0; //Define variable for analog readings Serial.print("Sensor Reading = "); Serial.println(analogRead(A0));

for(int x = 0 ; x < 500 ; x++) //Start for loop
{
sensorValue = sensorValue + analogRead(A0); //Add analog values of sensor 500 times
```

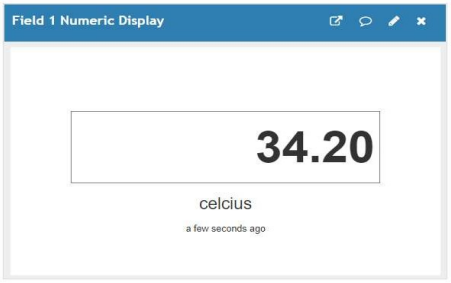
```
}
sensorValue = sensorValue/500.0; //Take average of readings sensor_volt =
sensorValue*(5.0/1023.0); //Convert average to voltage RS_air =
((5.0*1.0)/sensor_volt)-1.0; //Calculate RS in fresh air
R0 = RS_air/3.7; //Calculate R0

Serial.print("R0 = "); //Display "R0" Serial.println(R0); //Display value of R0
delay(1000); //Wait 1 second

}
```

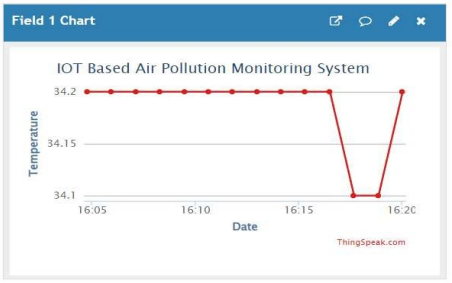
Execution of the Main Program:

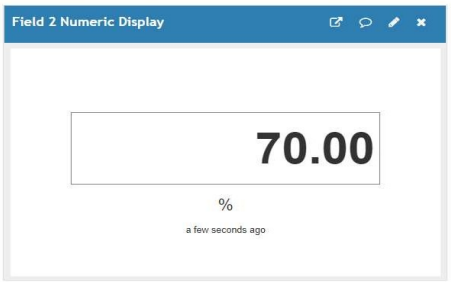
```
#include <ESP8266WiFi.h> #include <DHT.h>
#include <ThingSpeak.h>

DHT dht(D5, DHT11);
#define LED_GREEN D2 #define LED_YELLOW D3 #define LED_RED D4 #define
MQ_135 A0
int ppm=0;
float m = -0.3376; //Slope float b = 0.7165; //Y-Intercept
float R0 = 3.12; //Sensor Resistance in fresh air from previous code WiFiClient client;
long myChannelNumber = 123456; // Channel id

const char myWriteAPIKey[] = "API_Key";

void setup() {
// put your setup code here, to run once:
Serial.begin(9600); pinMode(LED_GREEN,OUTPUT);
pinMode(LED_YELLOW,OUTPUT); pinMode(LED_RED,OUTPUT);
pinMode(MQ_135, INPUT); WiFi.begin("WiFi_Name", "WiFi_Password");
while(WiFi.status() != WL_CONNECTED)
{
delay(200); Serial.print(".");
}
Serial.println();
Serial.println("NodeMCU is connected!"); Serial.println(WiFi.localIP());
dht.begin(); ThingSpeak.begin(client);
}

void loop() {
float sensor_volt; //Define variable for sensor voltage float RS_gas; //Define variable
for sensor resistance float ratio; //Define variable for ratio
int sensorValue;//Variable to store the analog values from MQ-135 float h;
float t;
float ppm_log; //Get ppm value in linear scale according to the the ratio value float
ppm; //Convert ppm value to log scale
```

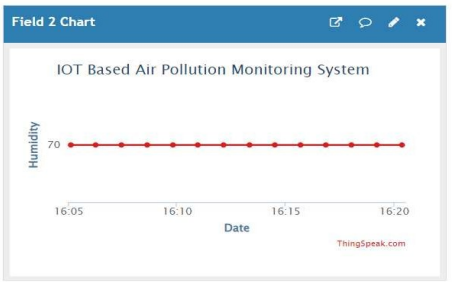
```
h = dht.readHumidity(); delay(4000);
t = dht.readTemperature(); delay(4000);


sensorValue = analogRead(gas_sensor); //Read analog values of sensor sensor_volt =
sensorValue*(5.0/1023.0); //Convert analog values to voltage RS_gas =
((5.0*1.0)/sensor_volt)-1.0; //Get value of RS in a gas
ratio = RS_gas/R0; // Get ratio RS_gas/RS_air
ppm_log = (log10(ratio)-b)/m; //Get ppm value in linear scale according to the ratio
value ppm = pow(10, ppm_log); //Convert ppm value to log scale

Serial.println("Temperature: " + (String) t); Serial.println("Humidity: " + (String) h);

Serial.println("Our desired PPM = "+ (String) ppm);

ThingSpeak.writeField(myChannelNumber, 1, t, myWriteAPIKey); delay(20000);
ThingSpeak.writeField(myChannelNumber, 2, h, myWriteAPIKey); delay(20000);
ThingSpeak.writeField(myChannelNumber, 3, ppm, myWriteAPIKey); delay(20000);

if(ppm<=100)
{
digitalWrite(LED_GREEN,HIGH); digitalWrite(LED_YELLOW,LOW);
digitalWrite(LED_RED,LOW);
}
else if(ppm<=200)
{
digitalWrite(LED_GREEN,LOW); digitalWrite(LED_YELLOW,HIGH);
digitalWrite(LED_RED,LOW);
}
else
{
digitalWrite(LED_GREEN,LOW); digitalWrite(LED_YELLOW,LOW);
digitalWrite(LED_RED,HIGH);
}
delay(2000);}
```

Result:

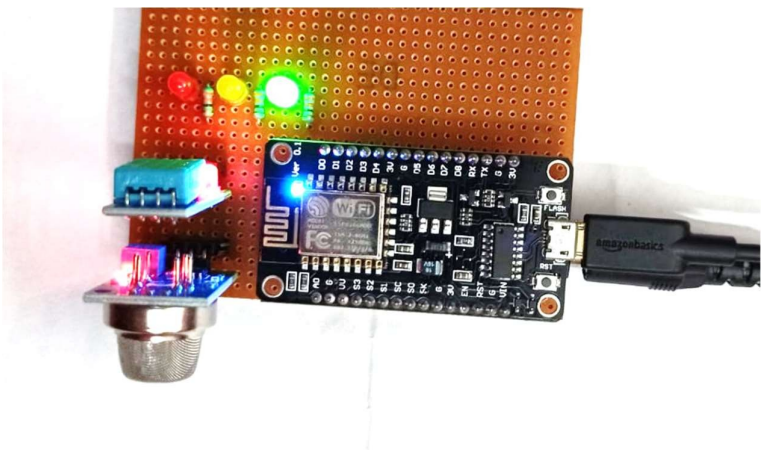








PROTOTYPE:



Data Analysis:

Name	String Index	Description
DEW	[0-4]	The observed dew point temperature value with the following properties **Min:** -0982 **Max:** +0368 **Units:** Degrees Celsius **Scaling Factor:** 10
DEW	[5-6]	The code that denotes a quality status of the reported dew point temperature.
TMP	[0-4]	The observed temperature value with the following properties: **Min:** -0932 **Max:** +0618 **Units:** Degrees Celsius **Scaling Factor:** 10
TMP	[5-6]	The code that denotes a quality status of the reported temperature.
WIND	[0-2]	The angle, measured in a clockwise direction, between true north and the direction from which the wind is blowing. **Min:** 001 **Max:** 360 **Units:** Angular Degrees **Scaling Factor:** 1 **Missing:** 999
WIND	[4]	The code that denotes the quality status of a reported wind direction angle.
WIND	[6]	The code that denotes the character of the wind observation, which is a qualitative metric to indicate calm or strong winds.
WIND	[8-11]	The observed wind speed which is the rate of horizontal travel of air past a fixed point. **Min:** 0000 **Max:** 0900 **Units:** meters per second **Scaling Factor:** 10
WIND	[13]	The code that denotes the quality status of the reported wind speed.

Application for project:

Air quality monitoring projects can be used in various applications. Here are some examples:

1. **Environmental monitoring**: Air quality monitoring systems can be used to monitor the air quality in the environment and detect pollutants such as **Carbon Monoxide**, **Nitrogen Oxide**, and **Ammonia** . This information can be used to identify areas with high levels of pollution and take steps to reduce pollution levels.

2. **Health monitoring**: Air quality monitoring systems can be used to monitor the air quality in hospitals, clinics, and other healthcare facilities. This information can be used to ensure that the air quality is safe for patients and staff .

3. **Smart homes**: Air quality monitoring systems can be used in smart homes to monitor the air quality and alert homeowners when the air quality drops below a certain level. This can help homeowners take steps to improve the air quality in their homes .

4. **Industrial applications**: Air quality monitoring systems can be used in industrial applications to monitor the air quality in factories, warehouses, and other industrial facilities. This information can be used to ensure that workers are not exposed to harmful pollutants .

5. **Agriculture**: Air quality monitoring systems can be used in agriculture to monitor the air quality in greenhouses and other agricultural facilities. This information can be used to ensure that plants are not exposed to harmful pollutants .

Conclusion:

In this project IoT based on measurement and display of Air Quality Index (AQI), Humidity and Temperature of the atmosphere have been performed. From the information obtained from the project, it is possible to calculate Air Quality in PPM. The disadvantage of the MQ135 sensor is that specifically it can't tell the Carbon Monoxide or Carbon Dioxide level in the atmosphere, but the advantage of MQ135 is that it is able to detect smoke, CO, CO2, NH4, etc harmful gases.

After performing several experiments, it can be easily concluded that the setup is able to measure the air quality in ppm, the temperature in Celsius and humidity in percentage with considerable accuracy. The results obtained from the experiments are verified through Google data. Moreover, the led indicators help us to detect the air quality level around the setup. However, the project experiences a drawback that is it cannot measure the ppm values of the pollutant components separately. This could have been improved by adding gas sensors for different pollutants. But eventually, it would increase the cost of the setup and not be a necessary provision to monitor the air quality. Since it's an IOT-based project, it will require a stable internet connection for uploading the data to the ThinkSpeak cloud. Therefore, it is possible to conclude that the designed prototype can be utilized for air quality, humidity and temperature of the surrounding atmosphere successfully.