International Research Journal of Education and Technology

Peer Reviewed Journal

ISSN 2581-7795

AIR POLLUTION MONITORING SYSTEM

S Malini[#1], P Minnala [#2] M ,A Monisha [#3,] P Vimala [#4]

[#1,2,3,4] *UG student, Department of Electronics and Communication Engineering*

[#1,2,3,4]*Peri Institute of Technology, Chennai, India.*

ABSTRACT

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. In this system, NodeMCU plays the main controlling role. 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. Besides the harmful gases (such as CO2, CO, smoke, etc) temperature and humidity can be monitored through the temperature and humidity sensor by this system. Sensor responses are fed to the NodeMCU which displays the monitored data in the ThingSpeak cloud which can be utilized for analyzing the air quality of that area. The following simple flow diagram (as shown in Fig. 1) indicates the working mechanism of the IoT-based Air Pollution Monitoring System.

Keywords: IOT, Air, Pollution, NodeMCU

I . INTRODUCTION

The Internet of Things (IoT) describes the network of physical objects—“things”—that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the internet. These devices range from ordinary household objects to sophisticated industrial tools. The field has evolved due to the convergence of multiple technologies, including ubiquitous computing, commodity sensors, increasingly powerful embedded systems, and machine learning.

Traditional fields of embedded systems, wireless sensor networks, control systems, automation (including home and building automation), independently and collectively enable the Internet of things. In the consumer market, IoT technology is most synonymous with products pertaining to the concept of the "smart home", including devices and appliances (such as lighting fixtures, thermostats, home security systems, cameras, and other home appliances) that support one or more common ecosystems, and can be controlled via devices associated with that ecosystem, such as smartphones and smart speakers. IoT is also used in healthcaresystems.

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 thehome aware of usage.

Significant numbers of energy-consuming devices (e.g. lamps, household appliances, motors, pumps, etc.) already integrate Internet connectivity, which can allow them to communicate with utilities not only to balance power generation but also helps optimize the energy consumption as a whole. These devices allow for remote control by users, or central management via a cloud-based interface, and enable functions like scheduling (e.g., remotely powering on or off heating systems, controlling ovens, changing lighting conditions, etc.).The smart grid is a utility-side IoT application; systems gather and act on energy and power related information to improve the efficiency of the production and distribution of electricity. Using advanced metering infrastructure (AMI) Internet- connected devices, electric utilities not only collect data from end-users but also manage distribution automation devices like transformers.

In this project, we have tried to implement the concept of IoT to monitor the temperature, humidity and air quality of the surroundings

II LITERATURE SURVEY

The explanation of the Air Quality Index (AQI) and its standard ranges are described in [1]. From 0-100 ppm the atmosphere is safe for living. If the ppm level increases above 100 then it moves out of the safety zone. If the ppm value rises above 200 then it becomes extremely dangerous for human life.

The DHT11 sensor module is used tomeasure the temperature and the humidity of the surroundings [2]. The MQ-135 gas sensor is used to measure the air quality of the surroundings [3]. It can be calibrated with respect to fresh air, alcohol, carbon dioxide, hydrogen and methane. In this project, it has been calibrated with respect to fresh air [9], [10].

In [4] the controlling action of NodeMCU hasbeen described. This research has shown the uses of C++ as the programming language for scripting the software code. It has an inbuilt Wi-Fi module which allows the project to implement IoT easily. Arduino IDE is used to implement the coding part of the project [5], [8]. ThingSpeak cloud is used for the cloud service. It has a free version which requires a delay of 15 seconds to upload an entry in the cloud [6], [7]. As this project uses two sensors, both of them have internal heater elements andwithdraw more power(P=V*I), so though bothsensors are turned ON, their output voltage levels vary and show unpredictable values due to insufficient power drive. So, we used a separate power supply for the sensors as NodeMCU alone is not sufficient to drive twosensors [9].

III COMPONENTS & HARDWARE ASSEMBLY

A. Hardware Components

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

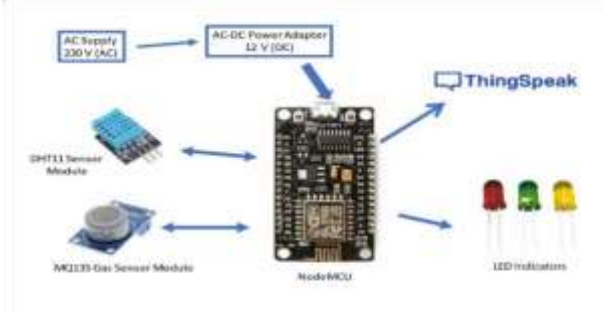


Figure 1: Components

B. Software Components

1. ThinkSpeak Cloud
2. Arduino IDE

C. Hardware Connection

The following steps were performedto execute the Hardware connection

STEP 1: The Vcc pin of the MQ-135 gassensor module and DHT11 sensor module was connected via Veroboard with an adapter delivering around 5V.

STEP 2: The Gnd pin of the MQ-135 gas sensor module, DHT11 sensor module and the cathode of the LED indicators was connected via Veroboard with the Gnd pin of the NodeMCU.
STEP 3: The analog DATA pin of the MQ-135 gas sensor module was connected with the A0 Pin of the NodeMCU.
STEP 4: The DATA pin of the DHT11 sensor module was connected with the D0 pin of the NodeMCU.
STEP 5: The anode of the three LED indicators (green, yellow, and red) were connected to the D2, D3, and D4 pins of the NodeMCU respectively.
STEP 6: The software code to execute the project was then uploaded to the NodeMCU. STEP 7: The setup was then powered with 9V DC via AC-DC adapter.It can be now turned ON/OFF as per the requirements. Fig 1 represents the circuit diagram of the setup.

The following steps were performed topreheat the DHT11 sensor module:

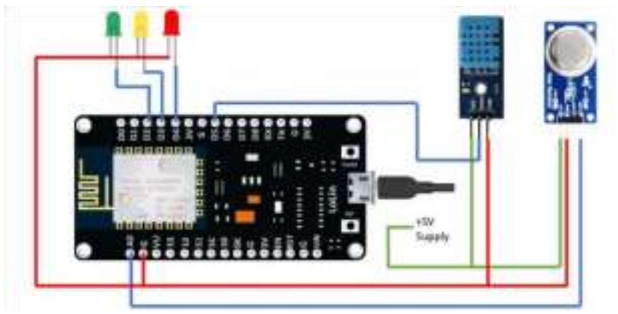


Figure 2: Circuit Diagram of the HardwareConnection

IV SYSTEM ARCHITECTURE

A. Software code for Calibration ofMQ135 Sensor

```
void setup()
{
Serial.begin(9600); //Baud
ratepinMode(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 averageto 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 R0delay(1000); //Wait 1 second

}

B. Execution of the MainProgram

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

DHT dht(D5, DHT11);
#define LED_GREEN D2 #defineLED_YELLOW D3
#define LED_RED D4 #define MQ_135 A0int ppm=0; float m = -0.3376;
//Slope float b = 0.7165; //Y-
Intercept
float R0 = 3.12; //Sensor Resistance in freshair 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 ratioint 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 RSin 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); //Convertppm 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);}
```

C. Working Procedure

NodeMCU plays the main controllingrole 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 inppm. 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 hardwarecircuits are described in the following chapters.

VI RESULTS ANDDISCUSSION

A. Experiment 1

Aim: To demonstrate the working of the system in a warm and humid outdoor atmosphere.

Experimental Condition: The experiment was performed on a warm sunny day in a local outdoor area.

Observations in ThingSpeak Cloud:

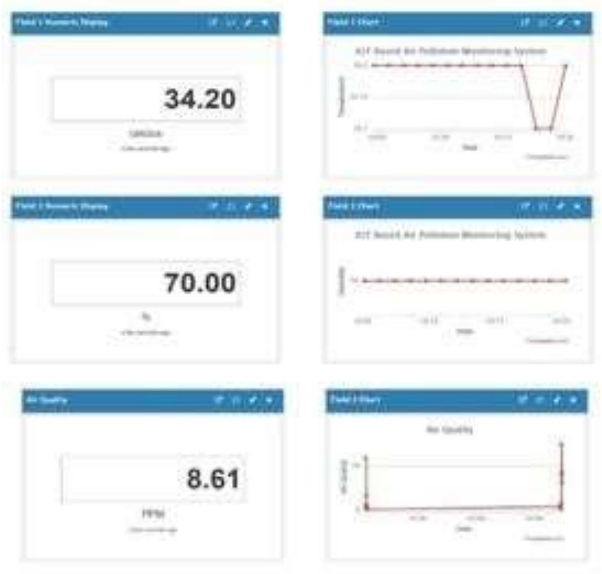


Figure 3: Observation ofExperiment 1

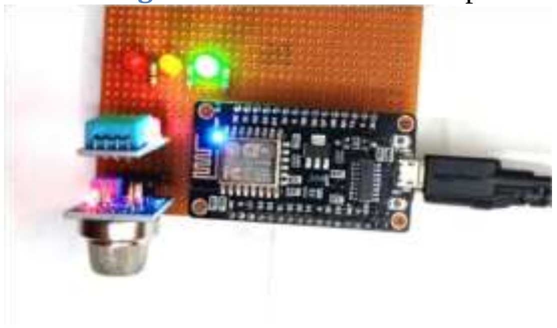
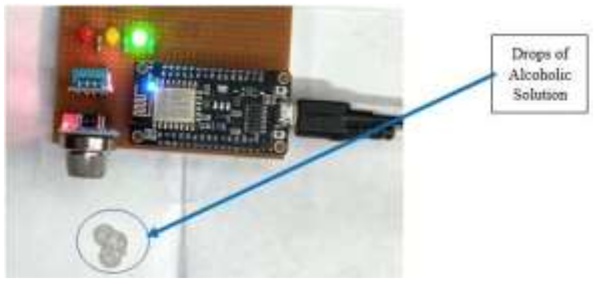


Figure 4: Setup of Experiment 1

Conclusion: We have taken the reference from the Samsung mobile weather app for verifying the values. It matched with a +1.20 error with the temperature data, +5 error with the humidity data and +0.11 error with the PPM data. Hence, we can conclude that the setup has measured the temperature and humidity around the setup area successfully.

B. Experiment 2

Aim: To demonstrate the working of the system in the presence of alcoholic gases.



Experimental Condition: The experiment was performed indoor in the presence of alcoholic gases. Drops of an alcoholic mixture (hand sanitiser) were used to produce alcoholic vapours.

Observations in ThingSpeak Cloud:

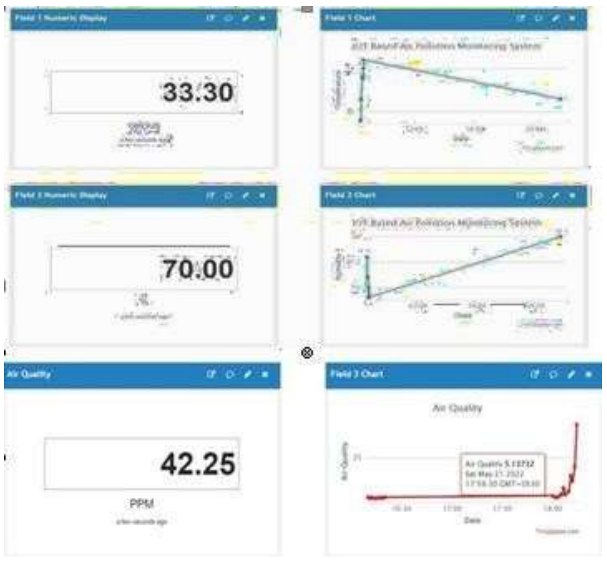


Figure 5: Observation of Experiment 2

Figure 6: Setup of Experiment 2

Conclusion:

We can observe from the results that the presence of alcohol vapours near the setup can be easily detected by the system. We have taken the reference from the Samsung mobile weather app for verifying the values. It matched with a +1.30 error with the temperature data, +5 error with the humidity data and +0.25 error with the PPM data. Hence, it can be concluded that we can detect the presence of alcoholic vapours with the help of this monitoring system.

C. Experiment 3

Aim: To demonstrate the working of the system in smoky conditions.

Experimental Condition: The experiment was performed in the presence of smoke coming from an incense stick placed near the setup.

Observations in ThinkSpeak Cloud

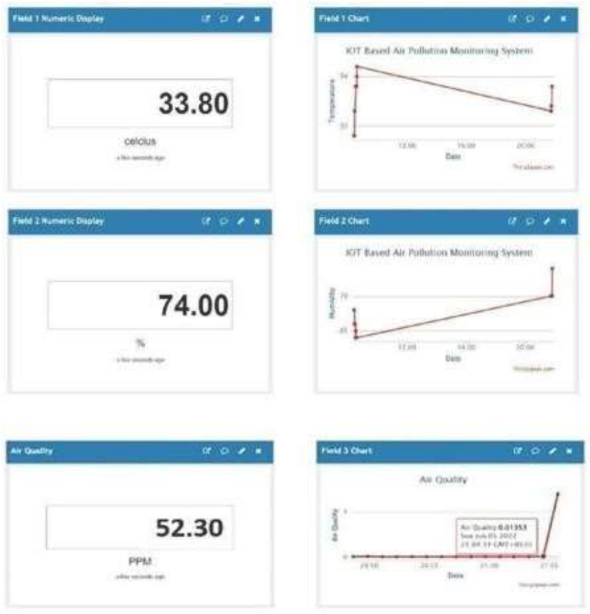


Figure 7: Observation of Experiment 3

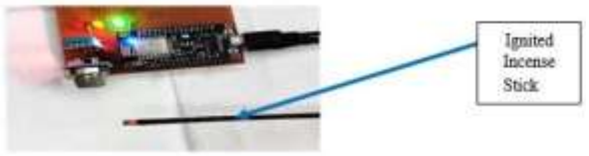
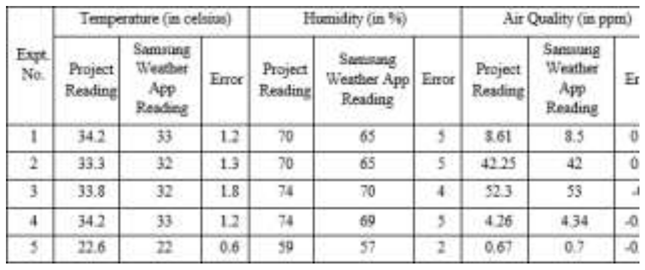


Figure 8: Setup of Experiment 3
Conclusion:

We can observe from the results that the presence of smoke near the setup can be easily detected by the system. We have taken the reference from the Samsung mobile weather app for verifying the values. It matched with a +1.80 error with the temperature data, +4 error with the humidity data and -0.7 error with the PPM data. Hence, it can be concluded that we can detect the presence of smoke with the help of this monitoring system.



Expt. No.	Temperature (in celsius)			Humidity (in %)			Air Quality (in ppm)		
	Project Reading	Samsung Weather App Reading	Error	Project Reading	Samsung Weather App Reading	Error	Project Reading	Samsung Weather App Reading	Er
1	34.2	33	1.2	70	65	5	8.61	8.5	0
2	33.3	32	1.3	70	65	5	42.25	42	0
3	33.8	32	1.8	74	70	4	52.3	53	-
4	34.2	33	1.2	74	69	5	4.26	4.34	-0
5	22.6	22	0.6	59	57	2	0.67	0.7	-0

Figure 9: Setup of Experiment 3

VII 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, itis possible to conclude that the designed prototype can be utilized for air quality, humidity and temperature of the surrounding atmosphere successfully.

REFERENCES

[1] https://gaslab.com/blogs/articles/carbon-monoxide-levels

[2] tps://www.instructables.com/Measuring-Humidity-Using-Sensor-DHT11

[3] https://pdf1.alldatasheet.com/datashee t-pdf/view/1307647/WINSEN/MQ135.html

[4] https://components101.com/development-boards/nodemcu-esp8266-pinout-featuresand-datasheet

[5] https://www.arduino.cc[

6] https://thingspeak.com

[7] Pasha, S. (2016). ThingSpeak based sensing and monitoring system for IoT with Matlab Analysis. International Journal of New Technology and Research, 2(6).

[8] Kumar, N. S., Vuayalakshmi, B., Prarthana, R. J., & Shankar, A. (2016, November). IOT based smart garbage alert system using Arduino UNO. In 2016 IEEE Region 10 Conference (TENCON) (pp. 1028-1034). IEEE.

[9] IoT based Air Quality monitoring system using MQ135 & MQ7 with Machine Learning analysis by Kinnera Bharath Kumar Sai M.Tech CSE VIT University, Vellore Subhaditya Mukherjee B.Tech CSE VIT University, Vellore Dr. Parveen Sultana H Associate Professor Department of CSE, VIT University.

[10] https://www.codrey.com/electronic-circuits/how-to-use-mq-135-gas-sensor
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.