**Phase5 :Project documentation and submission**

**Objectives:**

The main objective of these Air quality monitoring system is to record the concentration levels of atmospheric pollutants in order to define air quality levels and establish action plans if high levels of contamination are detected.

To Design and develop a low-cost air pollution monitoring system. The proposed system alert workers, notify data of polluted area that has been checked for the immediate surroundings. The proposed system has various parameters such as Air Quality, Temperature and Humidity sensors with ESP32 microcontroller which collects and upload data into the cloud using ESP32 Wi-Fi module. The data is transmitted to the cloud platform using MQTT(Message queuing Telemetry Transport) protocol and alerts the user through an application. The proposed system has various applications like in industry where the pollution levels check of dangerous gasses is paramount. Further pollution information is used to alert the workers about the air quality in their surroundings are not good for health

**IOT Device Setup:**

**Requirements :**

1. 1.MQ135 Gas sensor

2.Arduino Uno

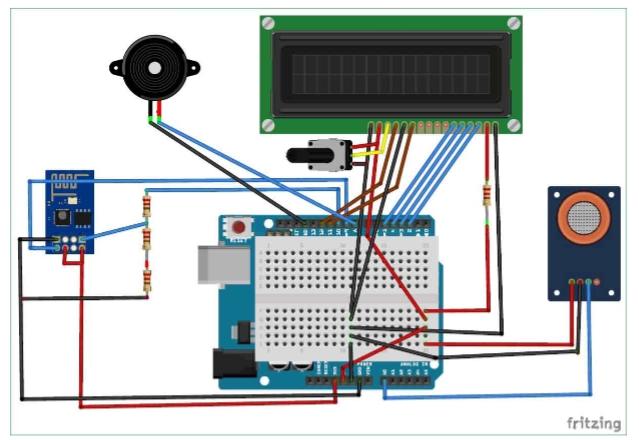
1. 3.Wi-Fi module ESP8266
2. 4.16X2 LCD
3. 5.Breadboard
4. 6.10K potentiometer
5. 7.1K ohm resistors
6. 8.220 ohm resistor
7. 9.Buzzer

**Circuit diagram and Explanation:**

First of all we will connect the **ESP8266 with the Arduino**. ESP8266 runs on 3.3V and if you will give it 5V from the Arduino then it won’t work properly and it may get damage. Connect the VCC and the CH\_PD to the 3.3V pin of Arduino. The RX pin of ESP8266 works on 3.3V and it will not communicate with the Arduino when we will connect it directly to the Arduino. So, we will have to make a voltage divider for it which will convert the 5V into 3.3V. This can be done by connecting three resistors in series like we did in the circuit. Connect the TX pin of the ESP8266 to the pin 10 of the Arduino and the RX pin of the esp8266 to the pin 9 of Arduino through the resistors.

Then we will connect the **MQ135 sensor with the Arduino**. Connect the VCC and the ground pin of the sensor to the 5V and ground of the Arduino and the Analog pin of sensor to the A0 of the Arduino.

Connect a buzzer to the pin 8 of the Arduino which will start to beep when the condition becomes true.



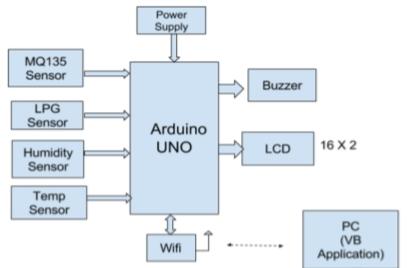
**Working and Explanation:**

The MQ135 sensor can sense NH3, NOx, alcohol, Benzene, smoke, CO2 and some other gases, so it is perfect gas sensor for our **Air Quality Monitoring Project**. When we will connect it to Arduino then it will sense the gases, and we will get the Pollution level in PPM (parts per million). MQ135 gas sensor gives the output in form of voltage levels and we need to convert it into PPM. So for converting the output in PPM, here we have used a library for MQ135 sensor.

Sensor was giving us value of 90 when there was no gas near it and the safe level of air quality is 350 PPM and it should not exceed 1000 PPM. When it exceeds the limit of 1000 PPM, then it starts cause Headaches, sleepiness and stagnant, stale, stuffy air and if exceeds beyond 2000 PPM then it can cause increased heart rate and many other diseases.

When the value will be less than 1000 PPM, then the LCD and webpage will display “Fresh Air”.  Whenever the value will increase 1000 PPM, then the buzzer will start beeping and the LCD and webpage will display “Poor Air, Open Windows”. If it will increase 2000 then the buzzer will keep beeping and the LCD and webpage will display “Danger! Move to fresh Air”.

**Block diagram:**

****

**Code:**

By Using MQ135 library,

#include "MQ135.h"

#include <SoftwareSerial.h>

#define DEBUG true

SoftwareSerial esp8266(9,10); // This

makes pin 9 of Arduino as RX pin and

pin 10 of Arduino as the TX pin

const int sensorPin= 0;

int air\_quality;

#include <LiquidCrystal.h>

LiquidCrystal lcd(12,11, 5, 4, 3, 2);

void setup() {

pinMode(8, OUTPUT);

lcd.begin(16,2);

lcd.setCursor (0,0);

lcd.print ("circuitdigest ");

lcd.setCursor (0,1);

lcd.print ("Sensor Warming ");

delay(1000);

Serial.begin(115200);

esp8266.begin(115200); // your esp's

baud rate might be different

sendData("AT+RST\r\n",2000,DEBUG);

// reset module

sendData("AT+CWMODE=2\r\n",1000,DEB

UG); // configure as access point

sendData("AT+CIFSR\r\n",1000,DEBUG)

; // get ip address

sendData("AT+CIPMUair\_quality=1\r\n

",1000,DEBUG); // configure for

multiple connections

sendData("AT+CIPSERVER=1,80\r\n",10

00,DEBUG); // turn on server on port

80

pinMode(sensorPin,

INPUT); //Gas sensor will be

an input to the arduino

lcd.clear();

}

void loop() {

MQ135 gasSensor = MQ135(A0);

float air\_quality =

gasSensor.getPPM();

if(esp8266.available()) // check if

the esp is sending a message

{

if(esp8266.find("+IPD,"))

{

delay(1000);

int connectionId =

esp8266.read()-48; /\* We are

subtracting 48 from the output

because the read() function returns

the ASCII decimal value and the first

decimal number which is 0 starts at

48\*/

String webpage = "<h1>IOT Air

Pollution Monitoring System</h1>";

webpage += "<p><h2>";

webpage+= " Air Quality is ";

webpage+= air\_quality;

webpage+=" PPM";

webpage += "<p>";

if (air\_quality<=1000)

{

webpage+= "Fresh Air";

}

else if(air\_quality<=2000 &&

air\_quality>=1000)

{

webpage+= "Poor Air";

}

else if (air\_quality>=2000 )

{

webpage+= "Danger! Move to Fresh

Air";

}

webpage += "</h2></p></body>";

String cipSend = "AT+CIPSEND=";

cipSend += connectionId;

cipSend += ",";

cipSend +=webpage.length();

cipSend +="\r\n";

sendData(cipSend,1000,DEBUG);

sendData(webpage,1000,DEBUG);

cipSend = "AT+CIPSEND=";

cipSend += connectionId;

cipSend += ",";

cipSend +=webpage.length();

cipSend +="\r\n";

String closeCommand =

"AT+CIPCLOSE=";

closeCommand+=connectionId; //

append connection id

closeCommand+="\r\n";

sendData(closeCommand,3000,DEBUG

);

}

}

lcd.setCursor (0, 0);

lcd.print ("Air Quality is ");

lcd.print (air\_quality);

lcd.print (" PPM ");

lcd.setCursor (0,1);

if (air\_quality<=1000)

{

lcd.print("Fresh Air");

digitalWrite(8, LOW);

}

else if( air\_quality>=1000 &&

air\_quality<=2000 )

{

lcd.print("Poor Air, Open Windows");

digitalWrite(8, HIGH );

}

else if (air\_quality>=2000 )

{

lcd.print("Danger! Move to Fresh Air");

digitalWrite(8, HIGH); // turn the LED on

}

lcd.scrollDisplayLeft();

delay(1000);

}

String sendData(String command, const int timeout, boolean debug)

{

String response = "";

esp8266.print(command); // send the read character to the esp8266

long int time = millis();

while( (time+timeout) > millis())

{

while(esp8266.available())

{

// The esp has data so display its output to the serial

window

char c = esp8266.read(); //

read the next character.

response+=c;

}

}

if(debug)

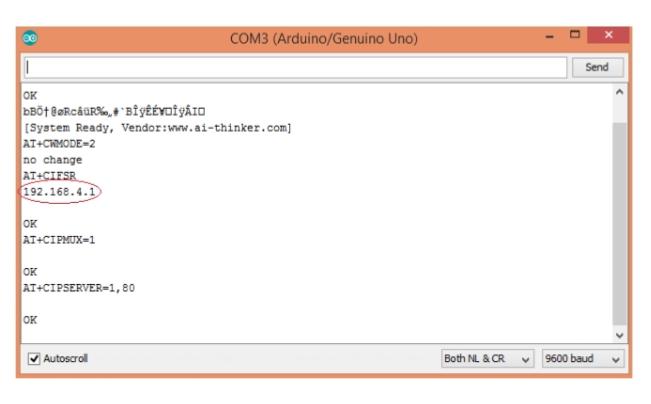
{

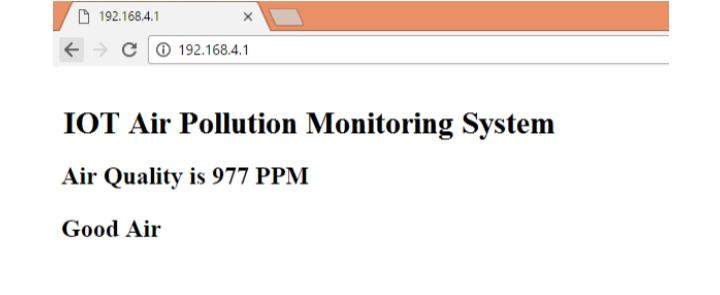
Serial.print(response);

}

return response;

**Output:**

****

****

**Think speak 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

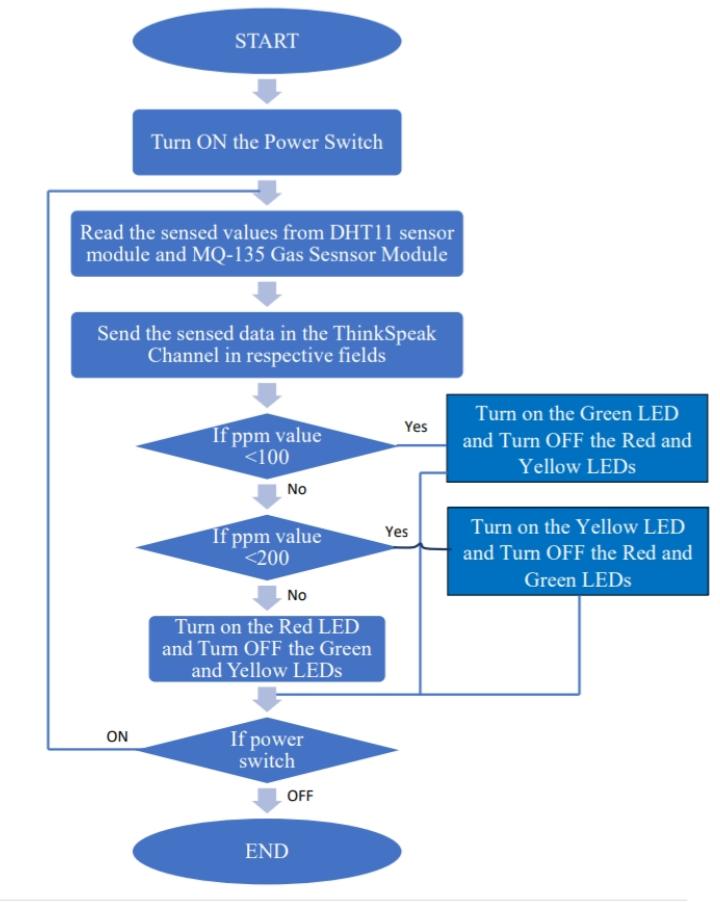
to analyse and visualize uploaded data using MATLAB without requiring the purchase of a

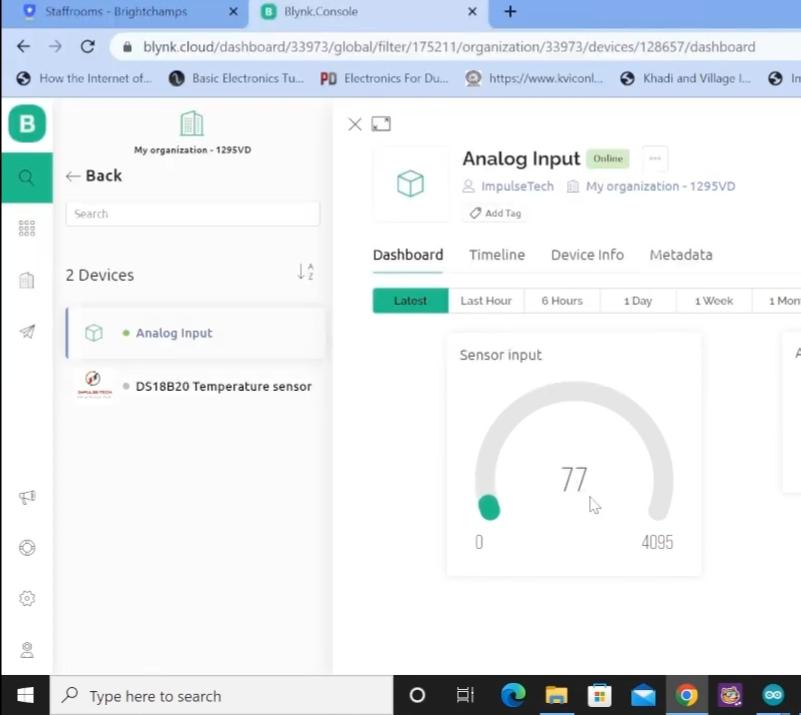
MATLAB license from MathWorks.

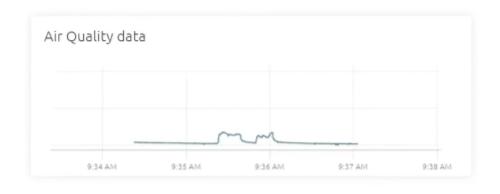


****

**Working Algorithm:**

****

****

****

**Applications:**

1) Industrial perimeter monitoring

2) Indoor air quality monitoring.

3) Site selection for reference monitoring stations.

4) Making data available to users.

**Advantages:**

1) Easy to Install

2) Updates On mobile phone directly

3) Accurate Pollution monitoring

4) Remote location monitoring

**Conclusion:**

The system to monitor the air of environment using

Arduino microcontroller, IOT Technology is proposed

to improve quality of air. With the use of IOTtechnology enhances the process of monitoring various aspects of environment such as air quality monitoring issue proposed in this paper. Here, using the MQ135 and MQ6 gas sensor gives the sense of different type of dangerous gas