Bachelor of Engineering (Computer Science & Engineering)

NAME-AKASH DATTA

UID-17BCS4510

SUB-PROJECT REPORT

TOPIC-IOT Based air Pollution Detection Monitoring system with arduino

DISCOVER . LEARN . EMPOWER

CERTIFICATE OF APPROVAL

The project report titled “ Air quality sensing and monitoring ” prepared by Akash datta UID-17BCS4510 ,is hereby approved and certified as a creditable study in technological subjects performed in a way sufficient for its acceptance for fulfilment of the degree for which it is submitted.

It is to be understood that by this approval, the undersigned do not, necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein, but approve the project only for the purpose for which it is submitted.

ACKNOWLEDGEMENT

It is a great privilege for us to express our profound gratitude to our respected teacher Mr. Deeksha kumari mam supervisor of my project from the college of chandigarh university , for his constant guidance, valuable suggestions, supervision and inspiration throughout the course work without which it would have been difficult to complete the work within scheduled time.

We are also indebted to the Head of the Department, Mr. Aman kausik sir, Computer science & Engineering IOT , Chandigarh university for permitting us to pursue the project.
We would like to take this opportunity to thank all the respected teachers of this department for being a perennial source of inspiration and showing the right path at the time of necessity.

1. INTRODUCTION

As our project is based on IoT, let us throw some light on the topic of IoT itself.

The Internet of Things (IoT) is a system of interrelated computing devices, mechanical and digital machines, objects animals, or people that are provided with unique identifiers (UIDs) and the ability to transfer data over a network wit requiring human-to-human or human-to-computer interaction.

APPLICATIONS OF IOT:

A growing portion of IoT devices are created for consumer use, including connected vehicles, home automation, wearable technology, connected health, and appliances with remote monitoring capabilities.

In the consumer market, IoT technology is most synonymous with products pertaining to the concept of the “smart home”

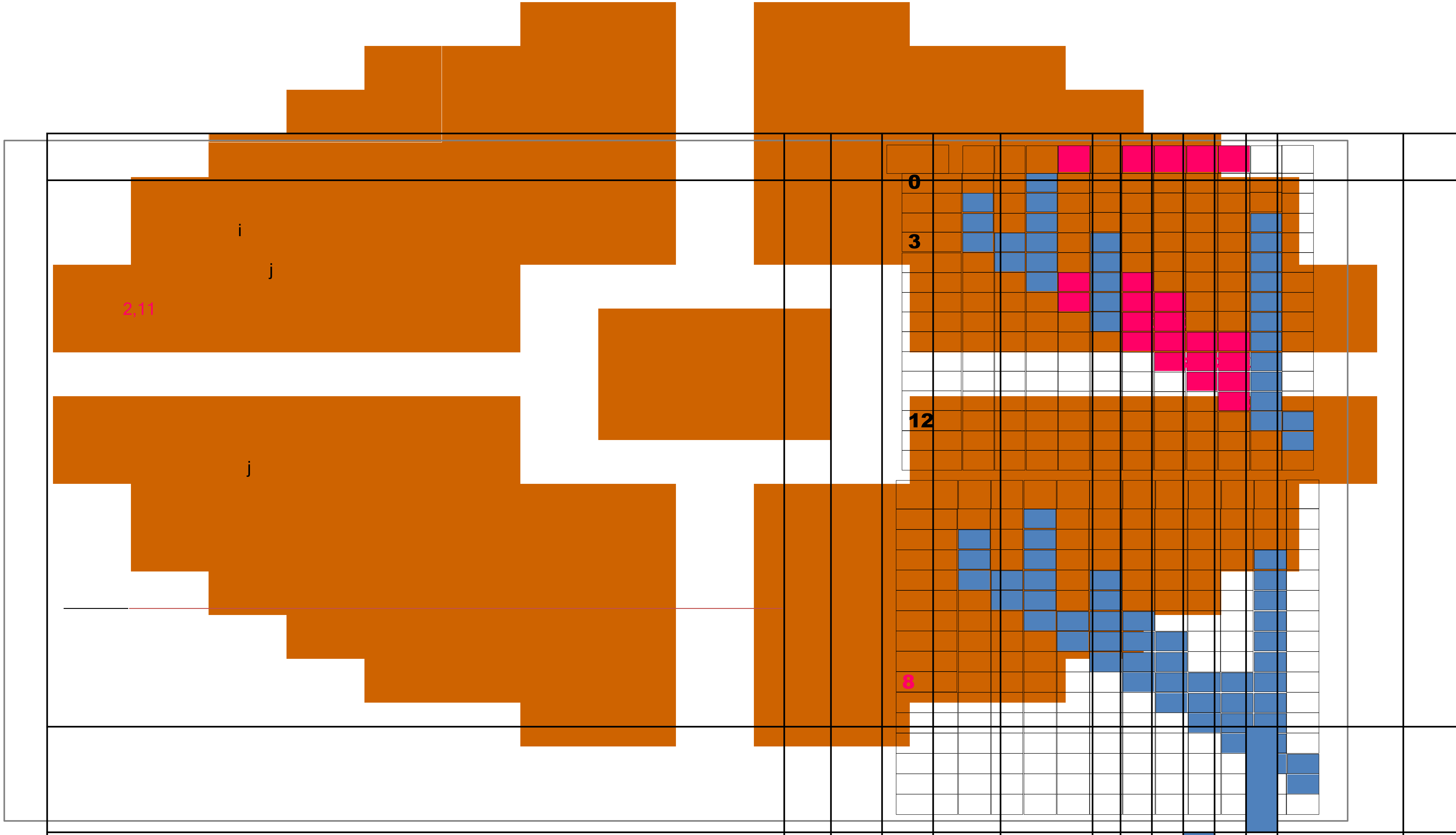
Therefore, using IoT, the aim of this project is to monitor the air quality level in the surrounding of our device using Arduino and MQ135 gas sensor and show the results in PPM units

Abstract

Air pollution has become a common phenomenon everywhere. Specially in the urban areas, air pollution is a real-life problem. A lot of people get sick only due to air pollution. In the urban areas, the increased number of petrol and diese vehicles and the presence of industrial areas at the outskirts of the major cities are the main causes of air pollution. T problem is seriously intensified in the metropolitan cities. Also, the climate change is now apparent. The governments around the world are taking every measure in their capacity. Many European countries have aimed to replace petrol a diesel vehicles with the electric vehicles by 2030. Even India has aimed to do so by 2025. The use of coal for electrici generation is now going to be a thing of past. The nations are now focusing to generate energy from nuclear reactors and the renewable resources like solar energy, wind energy and hydroelectric power.

It is now important to monitor air pollution in real time in most of the urban areas. This project is aimed at developing a IOT device which can monitor air pollution in real time and log data to a remote server. Remote monitoring was facilitated using classical motes in the past, which has some pitfalls like limited memory, processing speed and compl programming strategies. By using Internet of Things and recording sensor data to a remote server, the limitations of memory in the monitoring devices and manual collection of data from the installed devices can be overcome. The IOT also helps monitoring the data in real time.

The air pollution monitoring device developed in this project is based Arduino UNO. The Arduino board connects with ThingSpeak platform using ESP8266 Wi-Fi Module. As the cities usually have Wi-Fi hotspots at most of the places, s the device can be easily installed near any hotspot for its operation. The ThingSpeak is a popular IOT platform which easy to use and program. The sensor used for monitoring the air pollution is MQ-135 gas sensor. The sensor data is also displayed on a character LCD interfaced in the monitoring IOT device.
The sensing of data and sending it to the ThingSpeak server using Wi-Fi module is managed by the Arduino Sketch. The Arduino sketch is written, compiled and loaded to the Arduino board using Arduino IDE.


i
j
2,11
j
0
3
12
8

Proposed System

Now in this project, we are using the locally available gas sensors for observing polluted gases like Carbon monoxide (CO), Carbon dioxide (CO2), and parameters like temperature, humidity. By using this method people can view the level of pollution through a wireless system. It reduced cost, is reliable, and is comfortable for any place where we are monitoring the gases.

Goals and Objectives

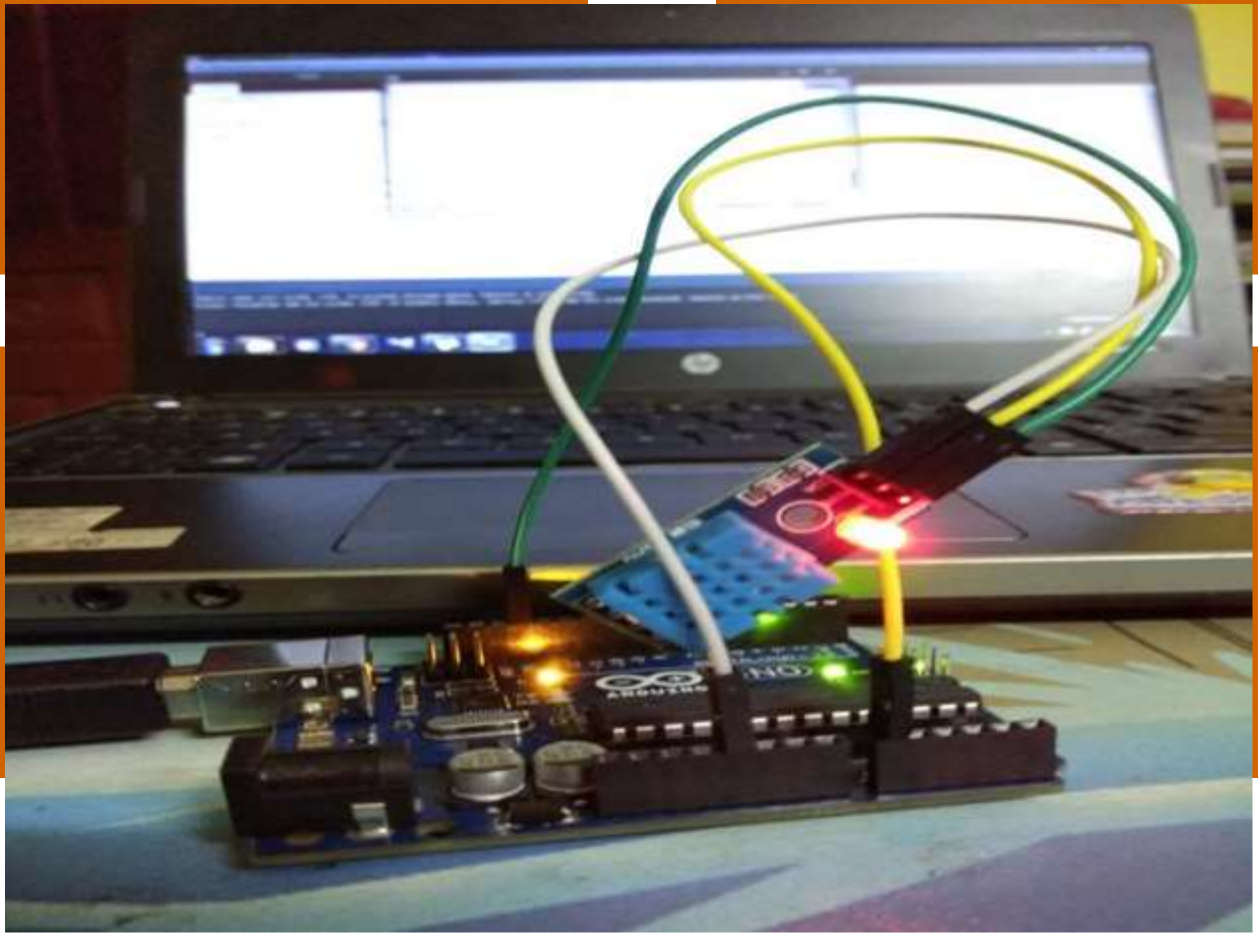
- quality of air can be checked indoors as well as outdoor.
- Detecting a wide range of physical parameters.
- Indoor air quality monitoring.
- Industrial perimeter monitoring.
- Roadside pollution monitoring.
- To make this data available to the common man.

Project Setup
Connections:

MQ135’s voltage and ground are connected to +5V and 0V and the analog output pin is connected to analog pin A0 o
Arduino Uno.

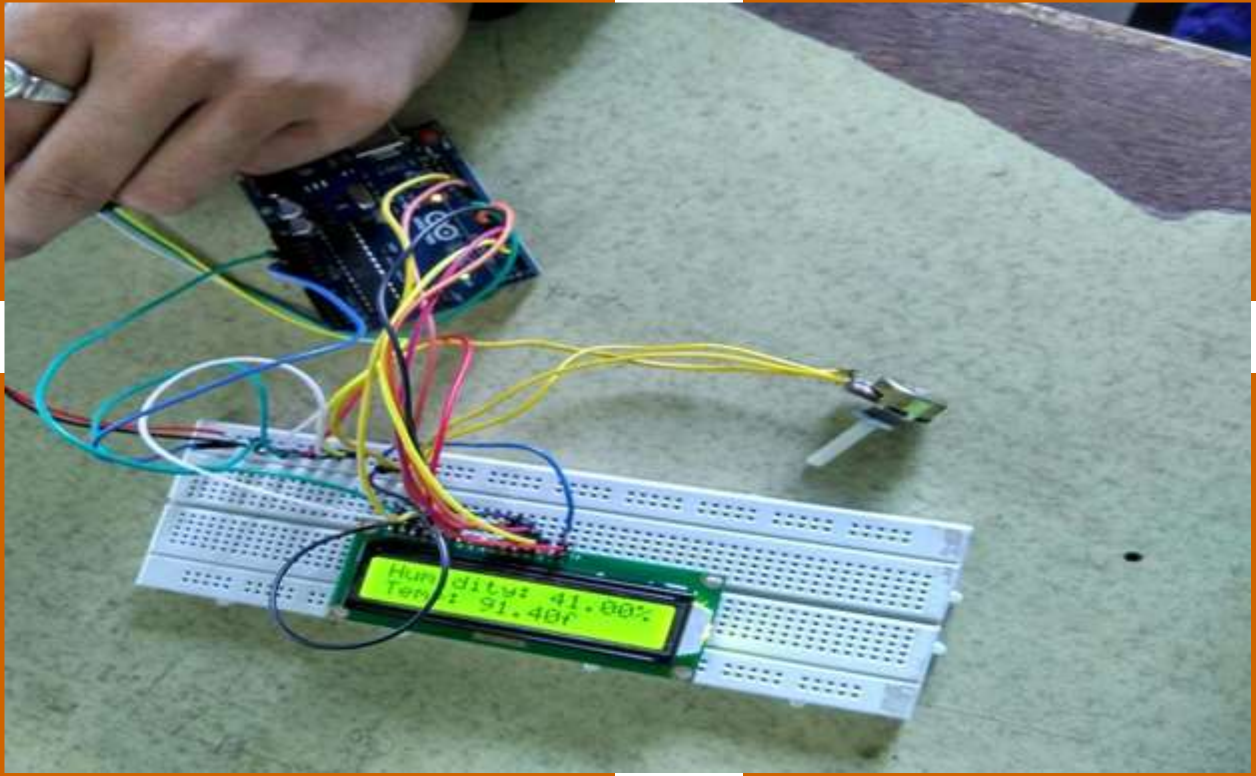
LCD RS pin to digital pin 12, Enable pin to digital pin 11, D4 pin to digital pin 5, D5 pin to digital pin 4, D6 pin to digital
pin 3, D7 pin to digital pin 2, R/W pin to ground, VSS pin to ground, VCC pin to 5V, 10K resistor ends to +5V and grou
and wiper to LCD VO pin.

i,k



The Analog pin of the MQ-135 sensor is connected to the analog pin of the Arduino UNO.

ikkj



Project Resources

HARDWARE REQUIREMENTS:

- Air Quality sensor (MQ 135)
- Potentiometer
- 16x2 LCD Panel
- Arduino Uno
- Wires

SOFTWARE REQUIREMENTS:

- Arduino (Version 1.8.2)

THINGSPEAK website

e

SYSTEM ANALYSIS AND DESIGN

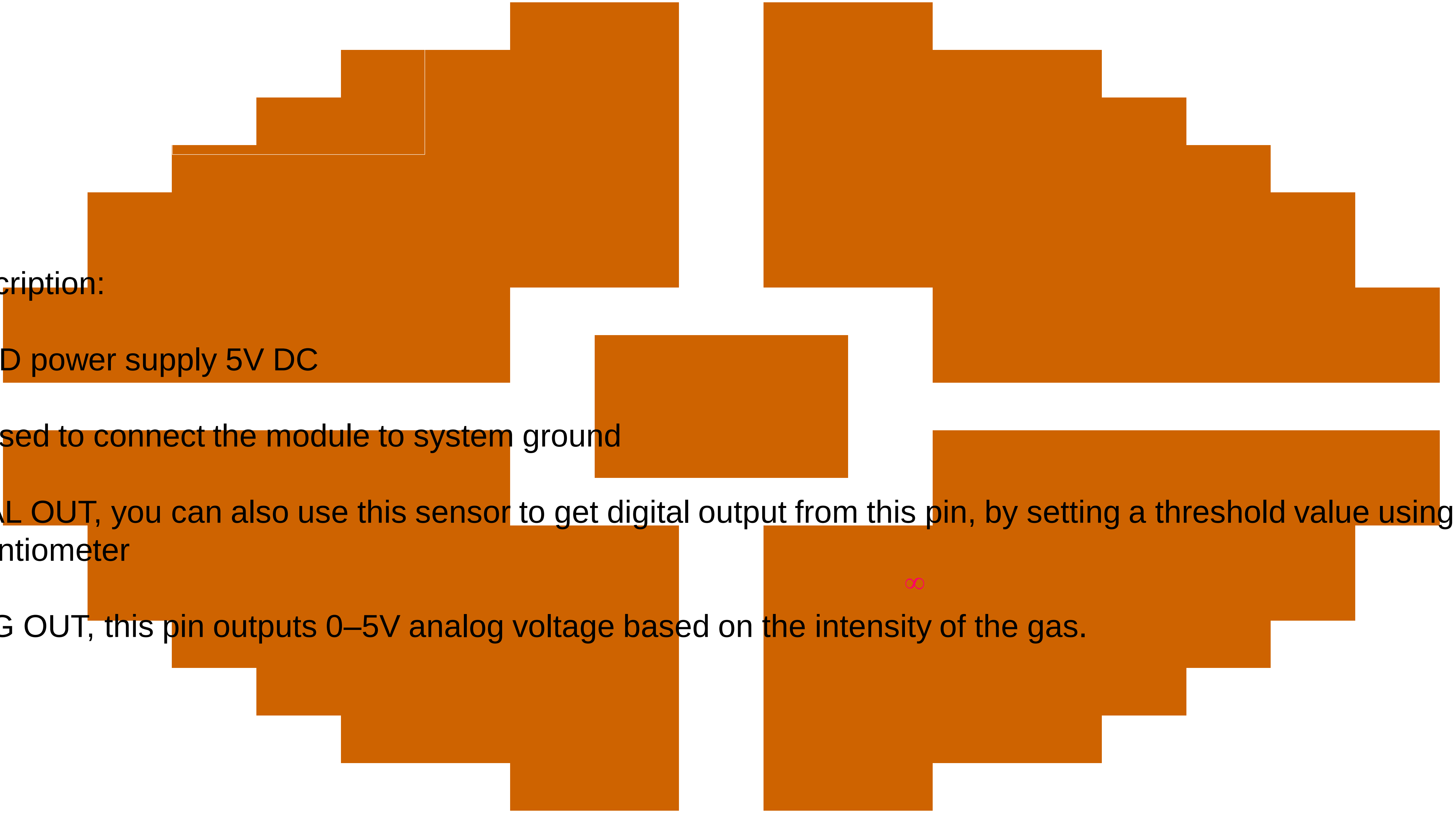
COMPONENT DESCRIPTION:

Air Quality Sensor (MQ135):-

Product Description:

Air quality click is suitable for detecting ammonia (NH_3), nitrogen oxides (NO_x) benzene, smoke, CO_2, and other harmful or poisonous gases that impact air quality. The MQ-135 sensor unit has a sensor layer made of tin dioxide (SnO_2), an inorganic compound that has lower conductivity in clean air than when polluting gases are present. To calibrate Air quality, use the onboard potentiometer to adjust the load resistance on the sensor circuit.

R



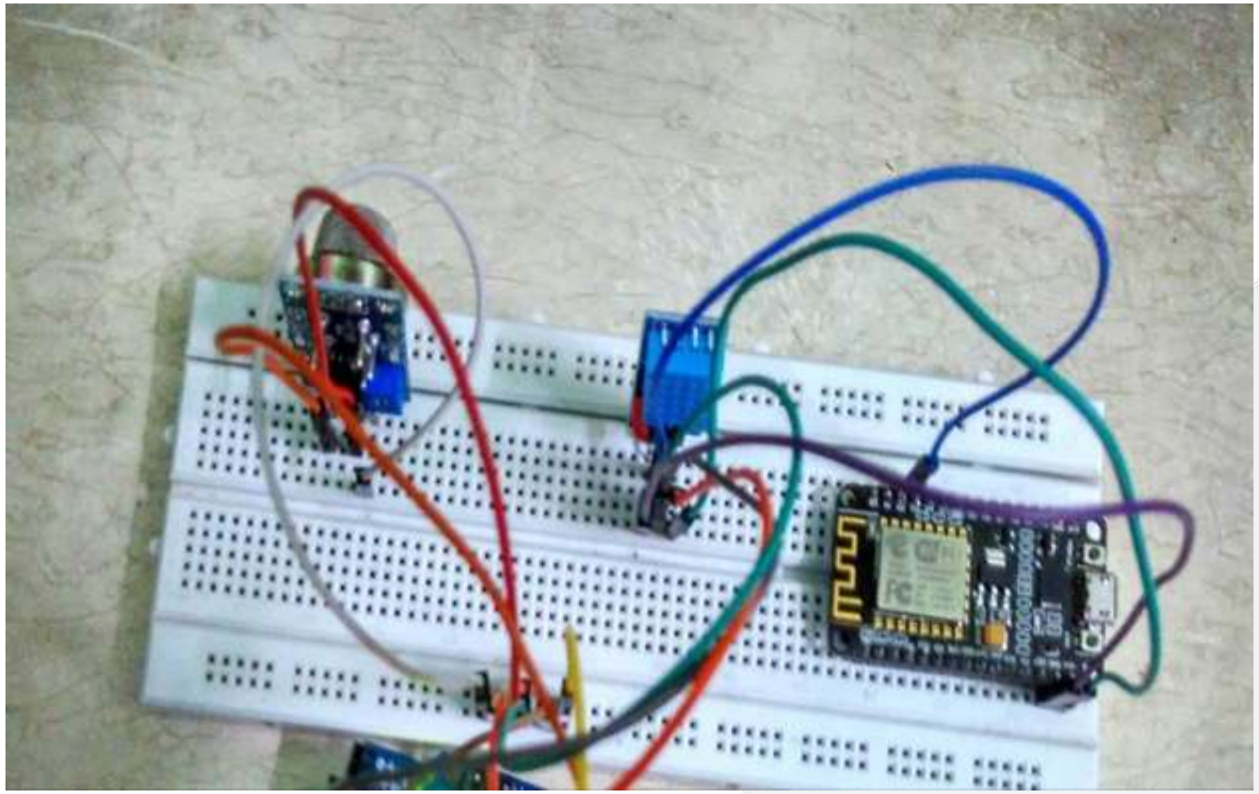
Pin Description:

· the VDD power supply 5V DC

· GND used to connect the module to system ground

· DIGITAL OUT, you can also use this sensor to get digital output from this pin, by setting a threshold value using the potentiometer

ANALOG OUT, this pin outputs 0–5V analog voltage based on the intensity of the gas.



Potentiometer: -
Product Description:

A potentiometer is a three-terminal resistor with a sliding or rotating contact that forms an adjustable voltage divide
only two terminals are used, one end and the wiper, it acts as a variable resistor or rheostat. The measuring
instrument called a potentiometer is essentially a voltage divider used for measuring electric potential (voltage); the
component is an implementation of the same principle, hence its name.

16X2 LCD Panel:-
Product Description:

A liquid-crystal display (LCD) is a flat-panel display or another electronically modulated optical device that uses the ligh
modulating properties of liquid crystals. Liquid crystals do not emit

[1]

light directly, instead of using a backlight or reflector to produce images in color or monochrome. LCDs are available to
display arbitrary images (as in a general-purpose computer display) or fixed images with low information content, which
can be displayed or hidden, such as preset words, digits, and seven-segment displays.

· Connect pin 1 (VEE) to the ground.

· Connect pin 2 (VDD or VCC) to the 5V.

· Connect pin 3 (V0) to the middle pin of the 10K potentiometer and connect the other two ends of the potentiometer to the VCC and the GND. The potentiometer is used to control the screen contrast of the LCD. A potentiometer of values other than 10K will work too.

· Connect pin 4 (RS) to pin 12 of the Arduino.

· Connect pin 5 (Read/Write) to the ground of Arduino. This pin is not often used so we will connect it to the ground.

· Connect pin 6 (E) to pin 11 of the Arduino. The RS and E pin are the control pins that are used to send data and characters.

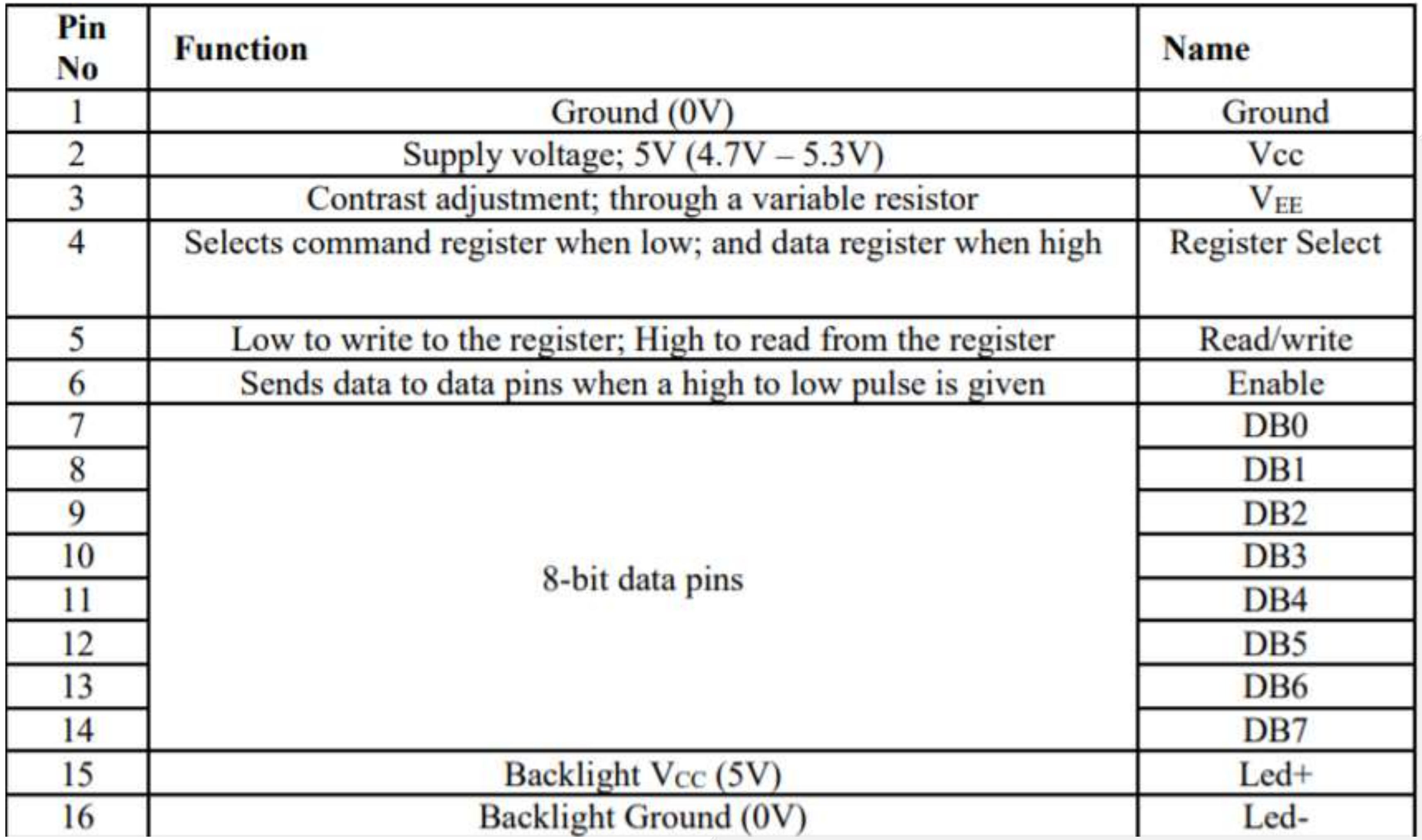
· The following four pins are data pins that are used to communicate with the Arduino.

· Connect pin 11 (D4) to pin 5 of Arduino.

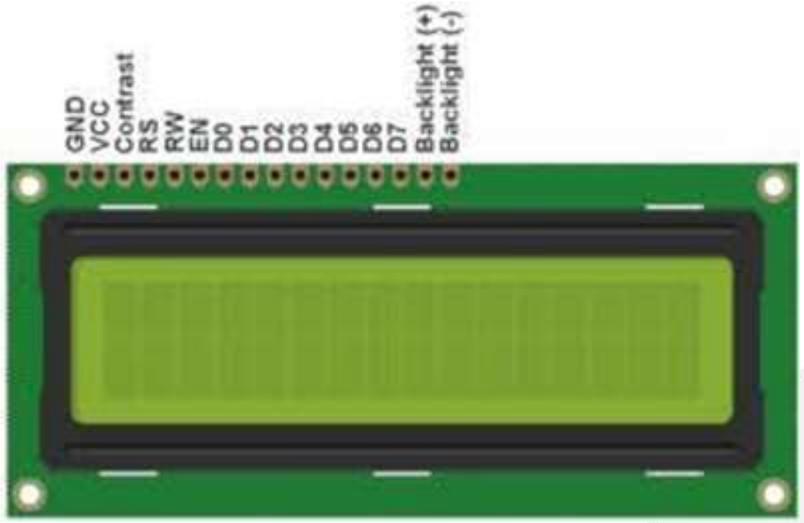
· Connect pin 12 (D5) to pin 4 of Arduino.

· Connect pin 13 (D6) to pin 3 of Arduino.

· Connect pin 14 (D7) to pin 2 of Arduino.

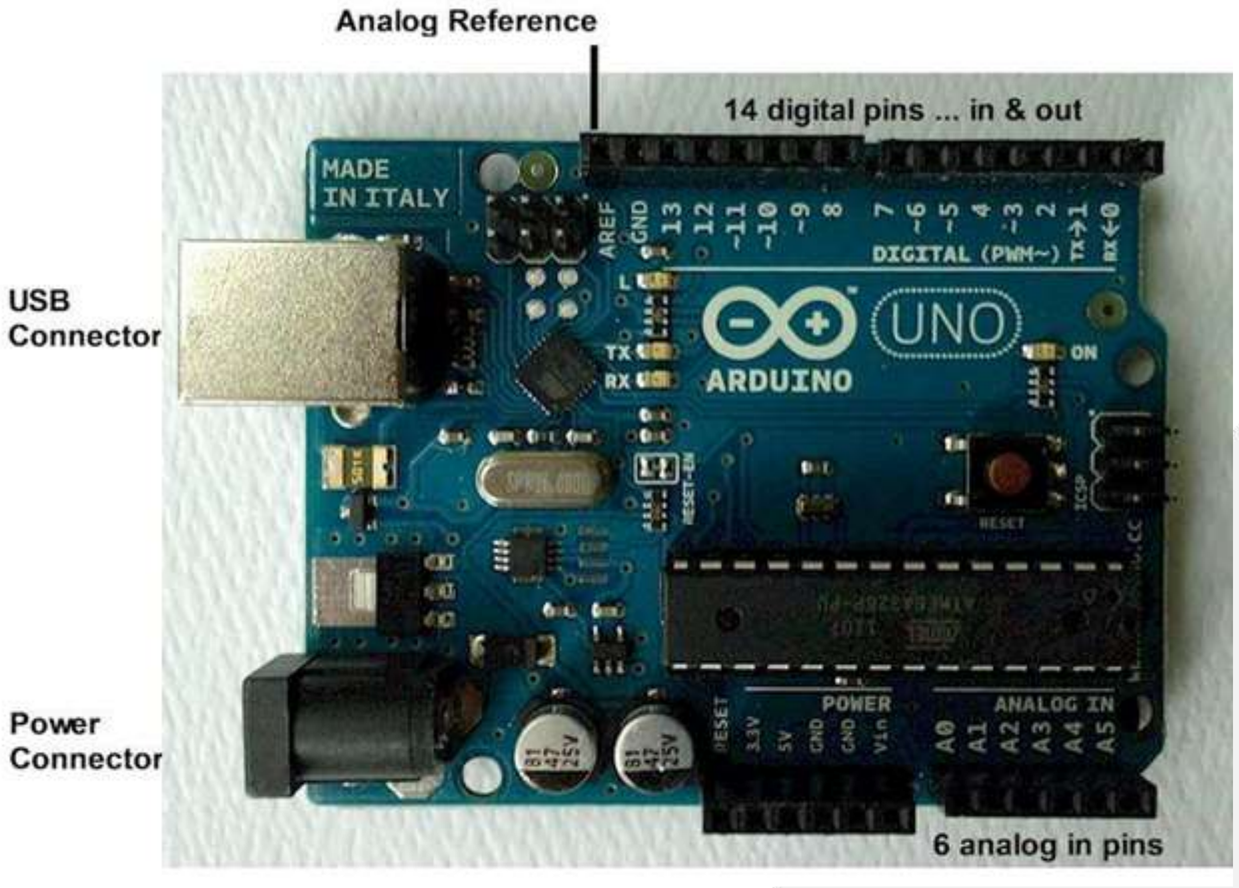


Pin No	Function	Name
1	Ground (0V)	Ground
2	Supply voltage; 5V (4.7V – 5.3V)	Vcc
3	Contrast adjustment; through a variable resistor	V_{EE}
4	Selects command register when low; and data register when high	Register Select
5	Low to write to the register; High to read from the register	Read/write
6	Sends data to data pins when a high to low pulse is given	Enable
7	8-bit data pins	DB0
8		DB1
9		DB2
10		DB3
11		DB4
12		DB5
13		DB6
14		DB7
15	Backlight V_{CC} (5V)	Led+
16	Backlight Ground (0V)	Led-


GND
VCC
Contrast
RS
RW
EN
D0
D1
D2
D3
D4
D5
D6
D7
Backlight (+)
Backlight (-)

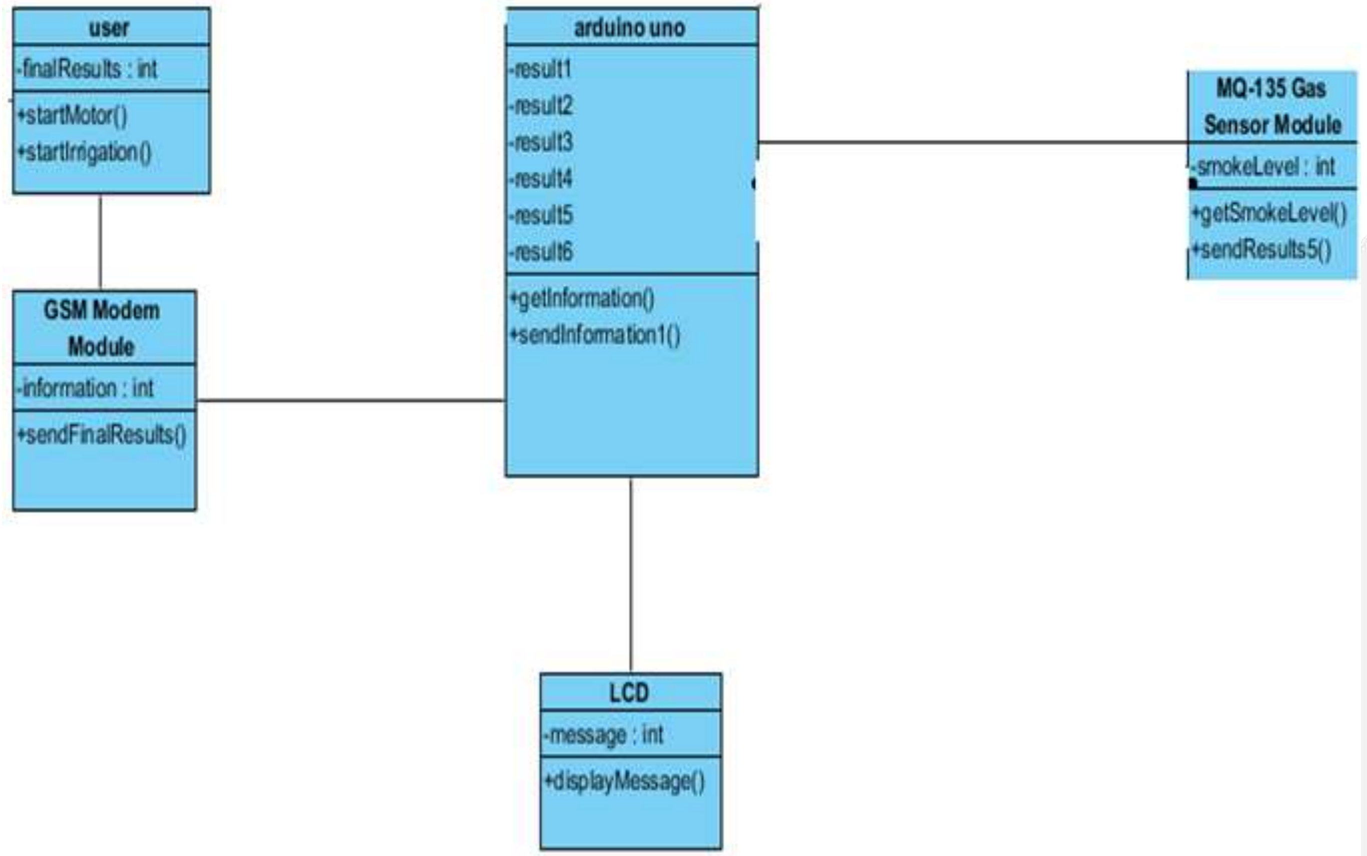
Arduino Uno:-
Product Description:

Arduino is an open-source computer hardware and software company, project, and user community that designs an
manufactures single-board microcontrollers and microcontroller kits for building digital devices and interactive objec
that can sense and control objects in the physical world.



rduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and nalog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The oards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also sed for loading programs from personal computers. The microcontrollers are typically programmed using a dialect of eatures from the programming languages C and C++. In addition to using traditional compiler toolchains, the Arduino roject provides an integrated development environment (IDE) based on the Processing language project

Power	Vin, 3.3V, 5V, GND	Vin: Input voltage to Arduino when using an external power source. 5V: Regulated power supply used to power microcontroller and other components on the board. 3.3V: 3.3V supply generated by on-board voltage regulator. Maximum current draw is 50mA. GND: ground pins.


user
-finalResults : int
+startMotor()
+startIrrigation()
arduino uno
-result1
-result2
-result3
-result4
-result5
-result6
+getInformation()
+sendInformation1()
MQ-135 Gas Sensor Module
-smokeLevel : int
+getSmokeLevel()
+sendResults5()
GSM Modem Module
-information : int
+sendFinalResults()
LCD
-message : int
+displayMessage()

User Interface

Arduino is an open-source computer hardware and software company, project, and user community that designs and manufactures single-board microcontrollers and microcontroller kits for building digital devices and interactive objects that can sense and control objects in the physical world. Arduino board designs use a variety of microprocessors and controllers. The boards are equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The boards feature serial communications interfaces, including Universal Serial Bus (USB) on some models, which are also used for loading programs from personal computers. The microcontrollers are typically programmed using a dialect of features from the programming languages C and C++. In addition to using traditional compiler toolchains, the Arduino project provides an integrated development environment (IDE) based on the Processing language project.

Code Explanation:
Before beginning the coding for this project, we need to first Calibrate the MQ135 Gas sensor. There are lots of calculations involved in converting the output of the sensor into a PPM value. we are using the Library for MQ135, yo can download and install this MQ135

Using this library, you can directly get the PPM values, by just using the below two lines:

```
MQ135 gasSensor = MQ135(A0);
float air quality = gasSensor.getPPM();
```

But before that, we need to calibrate the MQ135 sensor, for calibrating the sensor upload the below-given code and let it run for 12 to 24 hours, and then get the RZERO value.

```
#include "MQ135.h"
void setup (){
Serial.begin (9600);
}
void loop() {
MQ135 gasSensor = MQ135(A0); // Attach sensor to pin A0
float rzero = gasSensor.getRZero();
Serial.println (rzero);
delay (1000);
}
```

After getting the RZERO value. Put the RZERO value in the library file you downloaded “MQ135.h”: #define RZERO 494.63

Now we can begin the actual code for our Air quality monitoring project.

In the code, first of all, we have defined the libraries and the variables for the Gas sensor and the LCD. By using the Software Serial Library, we can make any digital pin as TX and RX pin. In this code, we have made Pin 9 as the RX p and pin 10 as the TX pin for the ESP8266. Then we have included the library for the LCD and have defined the pins f the same. We have also defined two more variables: one for the sensor analog pin and the other for storing air qualit value.

```
#include <SoftwareSerial.h>
#define DEBUG true
SoftwareSerial esp8266(9,10);
#include <LiquidCrystal.h>
LiquidCrystal lcd(12,11, 5, 4, 3, 2);
const int sensorPin= 0;
int air_quality
```

Then we will declare pin 8 as the output pin where we have connected the buzzer. lcd.begin(16,2) command will start the LCD to receive data and then we will set the cursor to first-line and will print the ‘ ’. Then we will set the cursor on the second line and will print ‘Sensor Warming’.

```
pinMode(8, OUTPUT);
lcd.begin(16,2);
lcd.setCursor (0,0);
lcd.print (" ");
lcd.setCursor (0,1);
lcd.print ("Sensor Warming ");
delay(1000);
```

Then we will set the baud rate for the serial communication. Different ESP’s have different baud rates so write it according to your ESP’s baud rate. Then we will send the commands to set the ESP to communicate with the Arduino and show the IP address on the serial monitor.

```
Serial.begin(115200);
esp8266.begin(115200);
sendData("AT+RST\r\n",2000,DEBUG);
sendData("AT+CWMODE=2\r\n",1000,DEBUG);
sendData("AT+CIFSR\r\n",1000,DEBUG);
sendData("AT+CIPMUair_quality=1\r\n",1000,DEBUG);
sendData("AT+CIPSERVER=1,80\r\n",1000,DEBUG);
pinMode(sensorPin, INPUT);
lcd.clear();
```

The following code will print the data on the LCD. We have applied various conditions for checking air quality, and LCD will print the messages according to conditions and the buzzer will also beep if the pollution goes beyond 1000 PPM.

```
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);
```

ADVANTAGES:

- Sensors are easily available.
- Simple, compact, easy to handle.
- Sensors have long life and less cost.
- Quality of air can be checked indoor as well as outdoor.
- Detecting a wide range of physical parameters including temperature ,humidity and carbon dioxide.

APPLICATIONS:

- Indoor air quality monitoring.
- Industrial perimeter monitoring.
- Roadside pollution monitoring.
- To make this data available to common man.

How the circuit works –

The device developed in this project can be installed near any Wi-Fi hotspot in a populated urban area. As the device is powered, the Arduino board loads the required libraries, flashes some initial messages on the LCD screen and start sensing data from the MQ-135 sensor. The sensitivity curve of the sensor for different combustible gases is already mentioned above. The sensor can be calibrated so that its analog output voltage is proportional to the concentration of polluting gases in PPM. The analog voltage sensed at the pin A0 of the Arduino is converted to a digital value by using the in-built ADC channel of the Arduino. The Arduino board has 10-bit ADC channels, so the digitized value ranges from 0 to 1023. The digitized value can be assumed proportional to the concentration of gases in PPM. The read value is first displayed on LCD screen and passed to the ESP8266 module wrapped in proper string through virtual serial function. The Wi-Fi module is configured to connect with the ThingSpeak IOT platform. ThingSpeak is an IOT analytics platform service that allows to aggregate, visualize and analyze live data streams in the cloud. ThingSpeak provides instant visualizations of data posted by the IOT devices to ThingSpeak server. With the ability to execute MATLAB code in ThingSpeak one can perform online analysis and processing of the data as it comes in.

The Wi-Fi module can be connected with the ThingSpeak server by sending AT commands from the module. The module first test the AT startup by sending the following command –

```
AT
```

The command is passed by the controller to the Wi-Fi module using software serial function. In response to the command 'AT', the platform must respond with 'OK' if the cloud service is running. Then, the AT command to view the version information is passed as follow –

```
AT + GMR
```

In response to this command, the IOT platform must respond by sending back the version information, sdk version and the time bin is compiled. Next, the AT command to set the connection to Wi-Fi mode is sent as follow –

```
AT + CWMODE = 3
```

By setting the parameter in CWMODE to 3, the Wi-Fi connection is configured to SoftAP as well as station mode. This AT command can in fact take three parameters as follow –

1 – set Wi-Fi connection to station mode
2 – set Wi-Fi connection to SoftAP mode
3 – set Wi-Fi connection to SoftAP + station mode

In response to this command, the IOT platform must send back the string indication the Wi-Fi connection mode set. Now the AT command to reset the module is sent as follow –
AT + RST
In response to this command, the Wi-Fi module must restart and send back a response of ‘OK’. After resetting the module, AT command to setup multiple connections is enabled by sending the following command –

AT + CIPMUX=1

This AT command can take two parameters – 0 for setting single connection and 1 for setting multiple connections. Next, the command to connect with the Access Point (AP) is passed which takes two parameters where first parameter is the SSID of the registered cloud service on ThingSpeak and the other parameter is the password to login the cloud service.

```
AT+CWJAP=”EngineersGarage”,”egP@$$w0rd?
```

Now, the AT command to get local IP address is passed as follow –

```
AT + CIFSR
```

In response to this command, the local IP address of the Wi-Fi connection is sent back by the module. Now, the module is ready to establish TCP IP connection with the ThingSpeak server. The controller reads the sensor data and store it in a string variable. The TCP IP connection is established by sending the following AT command –

```
AT + CIPSTART = 4, “TCP”, “184.106.153.149”, 80
```

The AT + CIPSTART command can be used to establish a TCP connection, register an UDP port or establish an SSL connection. In the above command, it is used to establish a TCP IP connection. For establishing a TCP-IP connection, the command takes four parameters where first parameter is link ID which can be a number between 0 to 4, second parameter is connection type which can be TCP or UDP, third parameter is remote IP address or IP address of the cloud service to connect with and last parameter is detection time interval for checking if the connection is live. If the last parameter is set to 0, the TCP keep-alive feature is disabled otherwise a time interval in seconds range from 1 to 7200 can be passed as parameter. In response to this command, the server must respond with 'OK' if connection is successfully established otherwise it should respond with message 'ERROR'.

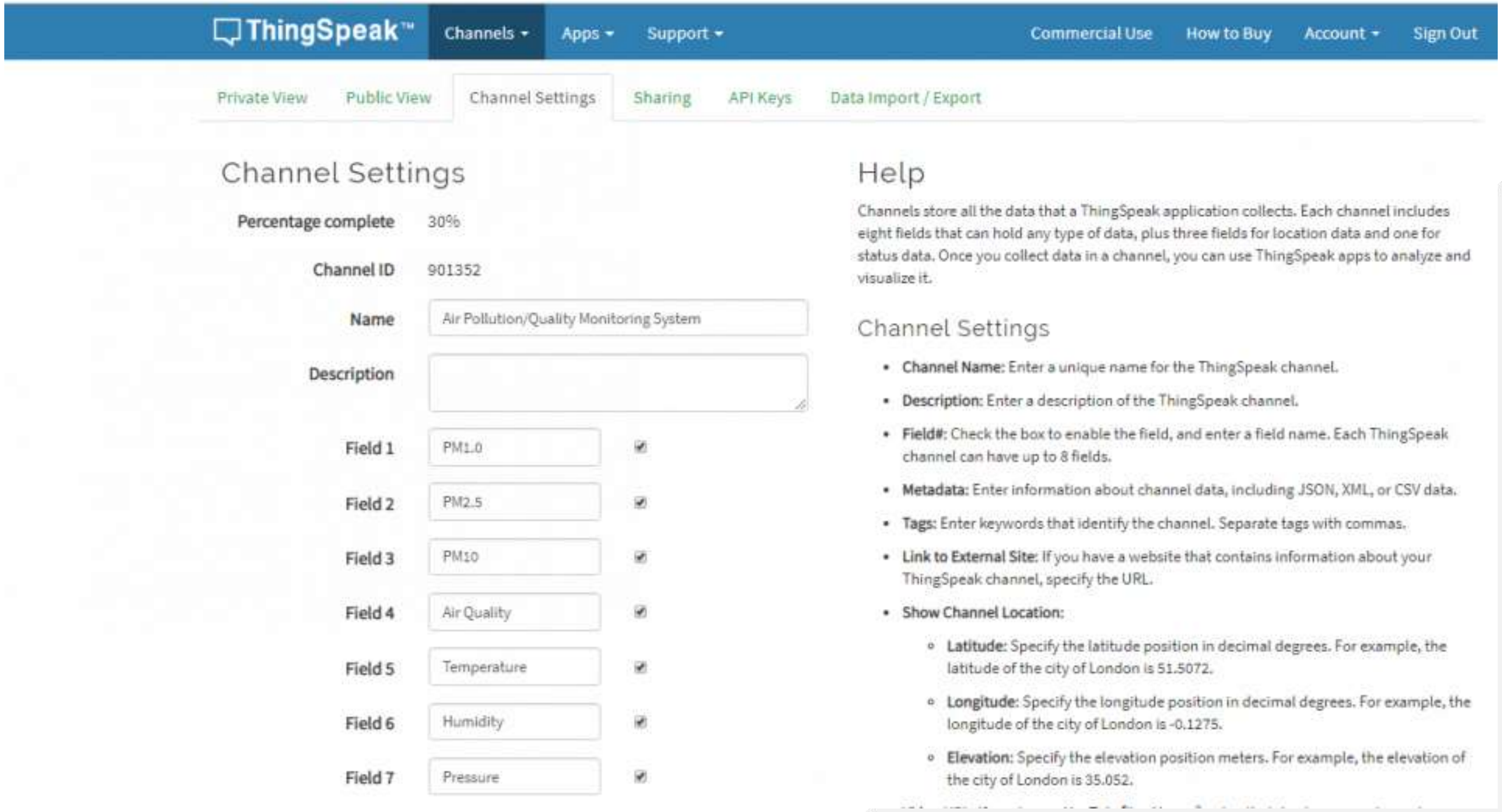
Now when the connection with the server is successfully established and the controller has read the sensor value, it can send the data to the cloud using the following command –

```
AT + CIPSEND = 4
```

This command takes four parameters, where first parameter is the link ID which can be a number between 0 to 4, second parameter is data length which can be maximum 2048 bytes long, third parameter is remote IP in case of an UDP connection and remote port number in case of UDP connection. The third and fourth parameter are optional and used only in case of UDP connection with the server. Since, the TCP IP connection is established, these parameters are not used. The command is followed by a string containing the URL having the field names and values passed through the HTTP GET method. In this project, a string containing the URL having API Key and the sensor value as the field and value is passed. The passed field and its value are logged on the cloud server. It is important to pass the API key in this URL as one of the field-value in order to connect with the registered cloud service. The Air quality measured by sensor can now be monitored and recorded through the thingspeak IOT plat form through the Wi-Fi module.

Setting up Thingspeak

We need to setup Thingspeak account to monitor the data from Nodemcu ESP8266-12E online. To setup thingspeak visit https://thingspeak.com/. Here you need to create an account.



Channel Settings

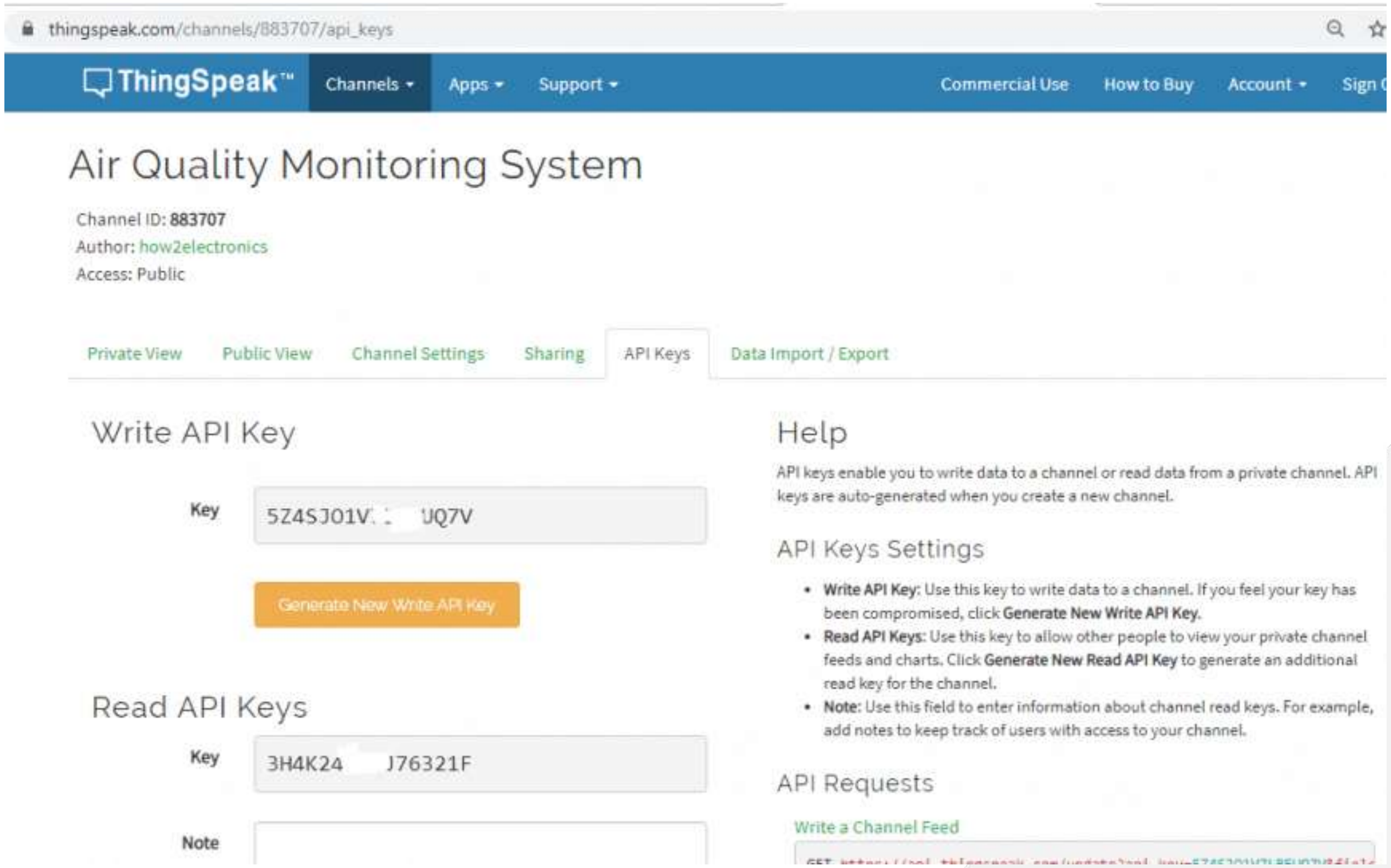
Percentage complete	30%	
Channel ID	901352	
Name	Air Pollution/Quality Monitoring System	
Description		
Field 1	PM1.0	✓
Field 2	PM2.5	✓
Field 3	PM10	✓
Field 4	Air Quality	✓
Field 5	Temperature	✓
Field 6	Humidity	✓
Field 7	Pressure	✓

Help

Channels store all the data that a ThingSpeak application collects. Each channel includes eight fields that can hold any type of data, plus three fields for location data and one for status data. Once you collect data in a channel, you can use ThingSpeak apps to analyze and visualize it.

Channel Settings

- **Channel Name:** Enter a unique name for the ThingSpeak channel.
- **Description:** Enter a description of the ThingSpeak channel.
- **Field#:** Check the box to enable the field, and enter a field name. Each ThingSpeak channel can have up to 8 fields.
- **Metadata:** Enter information about channel data, including JSON, XML, or CSV data.
- **Tags:** Enter keywords that identify the channel. Separate tags with commas.
- **Link to External Site:** If you have a website that contains information about your ThingSpeak channel, specify the URL.
- **Show Channel Location:**
  - **Latitude:** Specify the latitude position in decimal degrees. For example, the latitude of the city of London is 51.5072.
  - **Longitude:** Specify the longitude position in decimal degrees. For example, the longitude of the city of London is -0.1275.
  - **Elevation:** Specify the elevation position meters. For example, the elevation of the city of London is 35.052.


thingspeak.com/channels/883707/api_keys

Air Quality Monitoring System

Channel ID: **883707**
Author: how2electronics
Access: Public

Write API Key

Key: 5Z4SJ01V. . UQ7V

Read API Keys

Key: 3H4K24 J76321F

Note:

Help

API keys enable you to write data to a channel or read data from a private channel. API keys are auto-generated when you create a new channel.

API Keys Settings

- **Write API Key:** Use this key to write data to a channel. If you feel your key has been compromised, click **Generate New Write API Key.**
- **Read API Keys:** Use this key to allow other people to view your private channel feeds and charts. Click **Generate New Read API Key** to generate an additional read key for the channel.
- **Note:** Use this field to enter information about channel read keys. For example, add notes to keep track of users with access to your channel.

API Requests

Write a Channel Feed

```
GET https://api.thingspeak.com/update?api_key=5Z4SJ01VZLBEUQ7V&fiel
```

2. LITERATURE SURVEY

Gravimetric Method

• Particulate Matter(PM10):

Determine the amount of particulate material (PM10) present in ambient air, the materials required are repairable fine particles sampler, glass fiber clean paper of0length 8X10 inch, stability weight container, mechanical volumetric flow manage flow calculator machine and peak loading orifice equipment [4] Method : Initially the clean paper is inspected for join holes using a rigid stand then the movable particles are to be detached using a spongy brush continued by naming or applying the clean paper with a unique experimentation code for future reference The weight of0the clear paper is taken before sampling (Wi) Then the clear paper is conditioned in the conditioning room with the temperature maintained within20-30° C

and 40-50%qualified humidity. An sealed desiccators for twenty four periods and closing weight of the clear paper (Wf) is taken.
estimate:

C PM100µg/m3= (Wf–Wi) x 106/ V
And
C PM10 = consideration of Nitrogen dioxide V = Capacity of air tried (m3)

• Particulate Matter (PM2.5):

To discover the quantity of particulate substance (PM2.5) current in ambient midair, the supplies required are Repairable dirt analyst, mesh daily, equilibrium and heaviness case. Method: In this way an electrically motorized air tester magnets in air at a endless rapidity or the current rate (16.7lpm) kept by a current organizer which is fixed to a microchip into which a specially calculated particle-size divider is attached (hurricanes) wherever the delayed particulate material during the PM2.5 scope variety is parted used for a 47 mm polytetra fluoroethy lene (PTFE) strainer completed a declared example time. Then apiece strainer is considered earlier than and following the taster gathering towards discover the mesh improvement stylish unpaid toward the particulate material present during the ambient air which is then calculated while the whole collection of the poised atoms into the PM2.5 volume choices and alienated in the genuine capacity of air tasted and is expressed in µg/m3units. The computer chip reads and supplies five -minute medians of ambient fever, ambient pressure, strainer infection and volumetric flow rate and also calculates the regular fevers and weight, total volumetric flow.

Calculation:

• Equation to analyze the weight of excellent particulate substance composed on top of a Teflon clear:

M2.5= (Mf–Mi) mgx 103µg

where ,

M2.5 =whole mass offine particulate together through example time (µg)

Mf0 =last weightof0the hardened clear following example group(mg)

Mi =first weight of0the hardened clear previous to example set (mg)

103=element translation actor used for milligrams (mg) to micrograms (µg)

• Arena annals of0PM2.5 testers are obligator y o deliver capacities of0the whole size of0ambient air transient finished the technician (V)in cubic meter on the real infections and pressures unhurried throughout sample. The next formulary is used if V is not accessible straight after the technicianV= Q avg x t x 103m3

Anywhere,

V=whole instance worth (m3)

Q avg = even flow degree ended the complete length of the specimen old-fashioned (L/min) t = retro of sample dated (min)

103=section modification influence designed for liters(L) into cubicmeters (m3)

Effect would be unproven based PM 2.5 value.

Using RaspberryPi:
This system is designed using Raspberry pi computer. Numerous sensors are used in this project to determine various environmental parameters such as Automobile Monoxide Particulate substance Automobile Dioxide high temperature, moisture and stress haze sensors are interfaced Arduino, Microcontroller and Raspberry pi computer which are in turn interfaced with Arduino Uno all the way through the USB cable . The information which is detected by these devices is unceasingly logged analyzed intended and transmitted finished Raspberry pi in the direction of the cloud platform above the. The sensors DSM501A is a PM sensor whose production is of the kind PWM pulse, which is second-hand meant for calculating the particulate substance, DHT22 and BMP180 sensor shave digital productivity which are use for designed calculating parameters such as high temperature, moisture and force.

ZigBee and Web System Based:
This system collects information mainly through the Zig Bee complex and display information in a fixed receiving terminal. This new method is presented in this paper, the collected information is transferred to the Internet through the nod of Zig Bee. In order to allow users to be able to observe the information though the internet, the information would be uploaded to the public Internet platform, the user could directly observe the information from the web browser. System set up a tree complex based on Zig Bee wireless communication technology, through the sensor nodes to collect and to form a sensor information frames, packed into the gateway device, to achieve environmental information monitoring

CAEXMOS:
The Cellular phone Air Excellence Monitoring System(CAEXMOS) provides a cellular phone air excellence monitoring system that utilizes touching vehicles competentof0gas sensors to manage a large region The sensor join consists of0a microcontroller an onboard universal Positioning System (UPS) element and a position of0ozone (O3) Can nitrogen dioxide (NO2) attentions.
The lump is Bluetooth enabled consequently it can be able to drive the information to the entry in automobile When the automobile moves the machine example the sensors very little and supplies the information tagged through a position at what time the automobile moves t o a Wi-Fi hotspot the entrance in the automobile will broadcast the information to attend and the information is developed and published lying on the sensor Maportal CAEXMOS gives a complete evidence regarding air quality and pollutant dispersion within t he area But this examining scheme cannot instantly drive the examining information reverse.

Wireless Sensor Complex:
Wireless sensor complex skill to understand and confirmation monitoring information to realize mechanically air viewing tasks. the hardware side special types of0sensors and OctopusII wireless communication modular integrated to behavior Wireless communication below the ZigBee procedure The reverse end stage controlled by the Lab examination plan successfully0communicates with client during transfer them SMS mail. It as well supplies a great quantity of information into the information base via the MySQL course so that professional scan set up a calculation model toxic waste transmission based on the information. In adding the real examining information reveals minute extent pollution environment in Gong Guan around concerning The information can be applied to address the subjects such as the crashesof0motorcycles in the unavailable quickness on air advantage and the organization involving the top supervising information for Environmental Protection Administration (EPA) and the information composed by our projected single monitoring multifaceted. Additionally to complete existent point in time monitoring we imagine the informationof0CO concentration should be demonstrated on the cellular phone communication machines such as PDA elegant mobile phones ,tablet and computer for enchanting safety measure to continue air pollution in check.

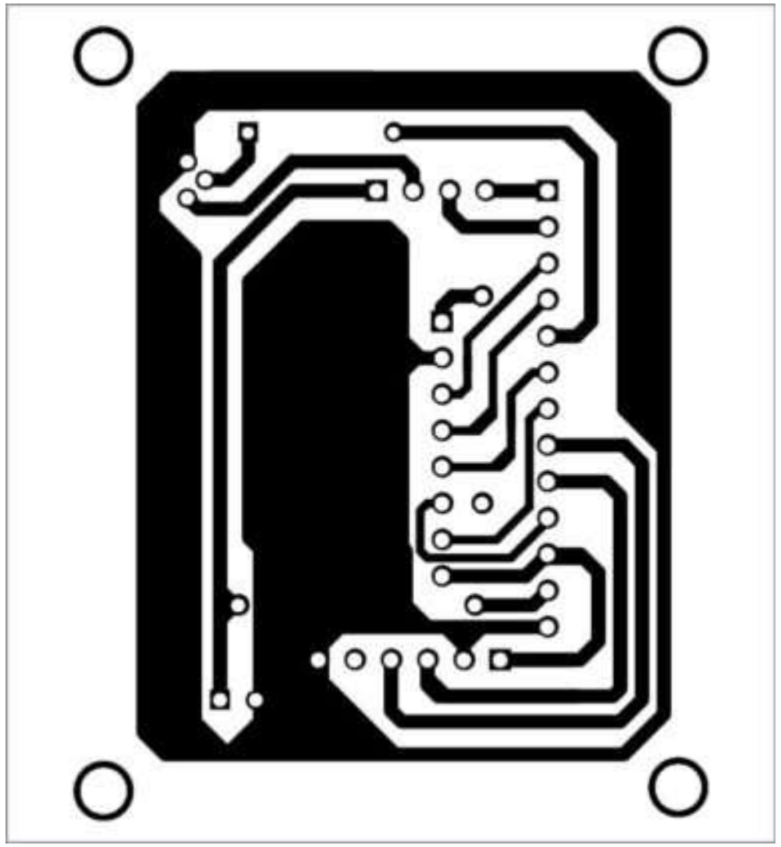
3. Methodology

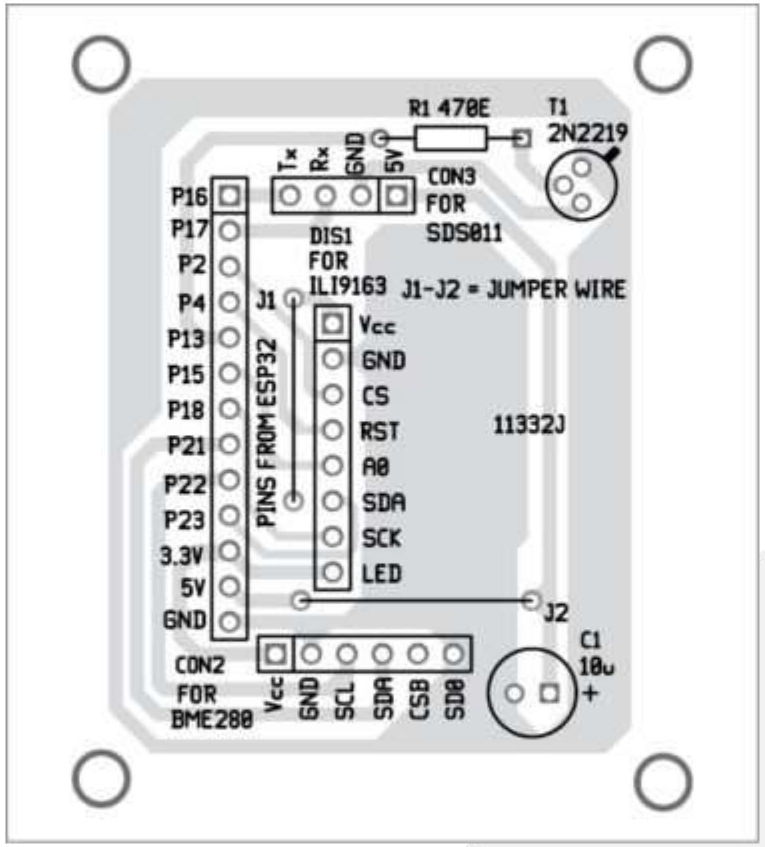
We are implementing this method in our project which results into real time standalone air quality calculator parameters. A simplified Block figure of the planned system is shown in the Fig.2. ESP32 Node MCU is the controller which is regulatory our system. The sensors such as SDS011, BME 280, MQ-135 are used to measure several atmospheric parameters.

These are divided into 5 layers. The primary layer is the ecological parameters which are obtained by extent. The next layer was the school work of the physiognomies and topographical of the instruments. The third coating is the, identifying, gauging, decision creation. The fourth coating is the device information attainment. The fifth coat is the ambient intellect environment. The device collected information when worked by the microcontroller and furthered it over the internet for study via the ESP32 Wi-Fi module. Employers will able to television slow restrictions on their smart earphones or web browser, Fig.1 shows the flow chart of how the information flows from devices to the web.

Construction and testing

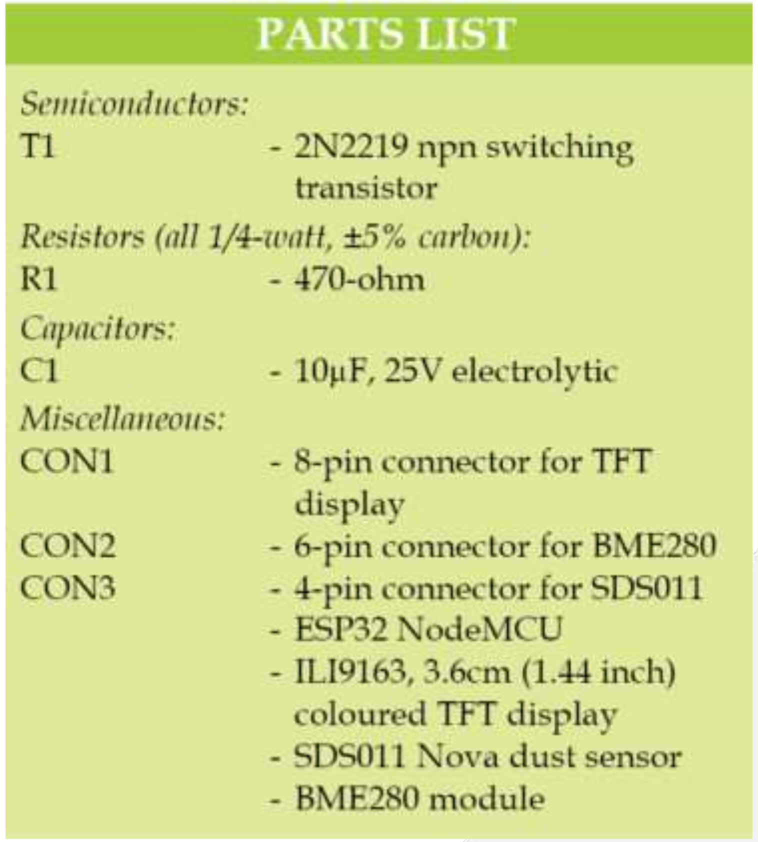
An actual-size PCB layout for the air quality monitoring system is shown in Fig. 7 and its components layout in Fig.
Mount various modules on the PCB. Switch on the power supply to ESP32.





d

If everything is correct, you will see the readings on the TFT display ofair quality monitoring system. The same value
will appear on www.thingspeak.com



PARTS LIST

Semiconductors:	
T1	- 2N2219 npn switching transistor
Resistors (all 1/4-watt, ±5% carbon):	
R1	- 470-ohm
Capacitors:	
C1	- 10µF, 25V electrolytic
Miscellaneous:	
CON1	- 8-pin connector for TFT display
CON2	- 6-pin connector for BME280
CON3	- 4-pin connector for SDS011
	- ESP32 NodeMCU
	- ILI9163, 3.6cm (1.44 inch) coloured TFT display
	- SDS011 Nova dust sensor
	- BME280 module

IMPLEMENTATION AND RESULTS
The comprehensive implementation details can be visualized using flowchart as shown in Figure 3.
Phase 1: Detection of air pollutants level
In this section, air pollution detection kit is developed using IoT. This is built in two steps:
Step 1: the first step deals with the collection of data from gas sensors connected to Arduino board and information is sent to a
cloud platform (i.e., Ubidots) that stores it.
Step 2: the second step portrays accessing this information utilizing Android platform.
For this purpose, the data is generated by gas sensors (viz. Carbon monoxide, Carbon dioxide, Methane) that read concentration of
gas in the region. The details about gas sensors used have been tabulated in Table 1

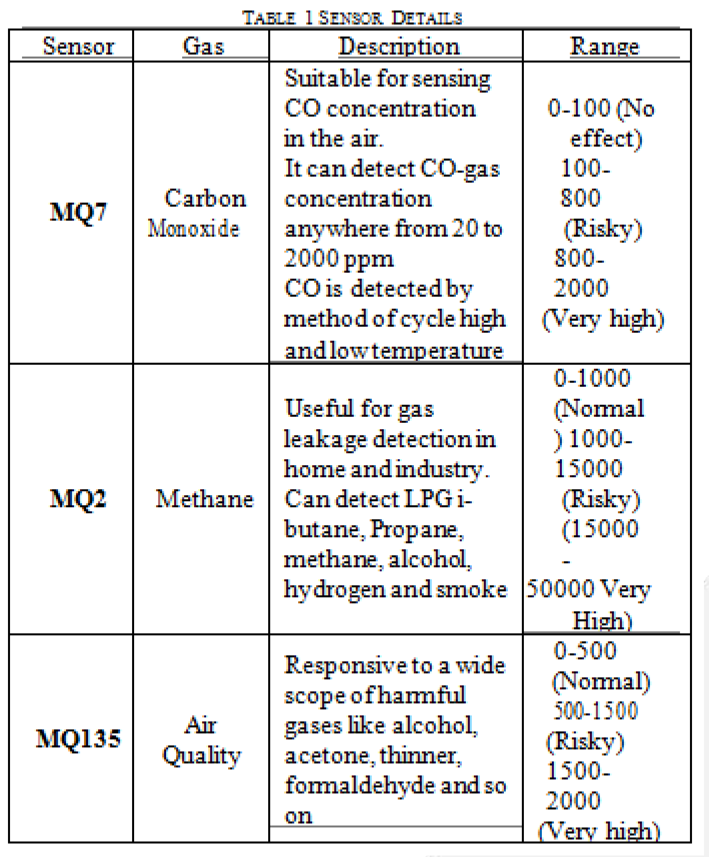
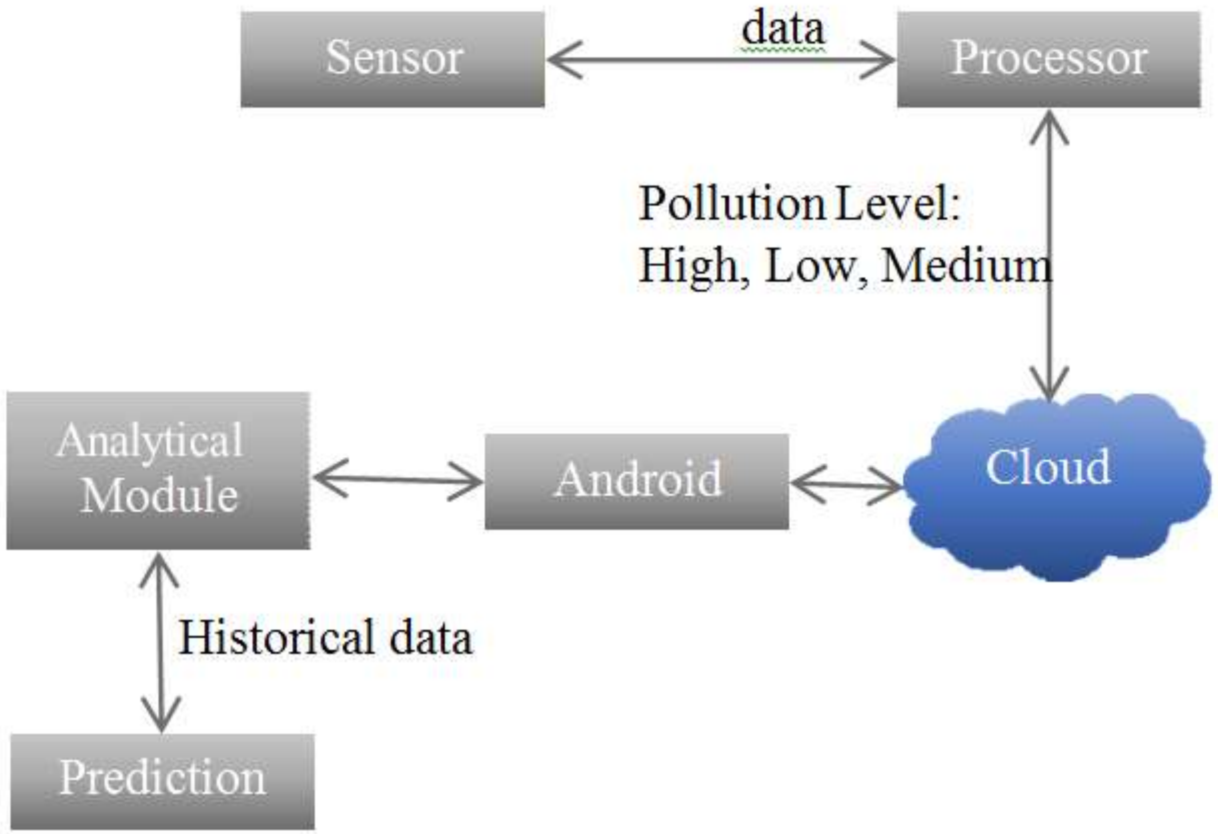
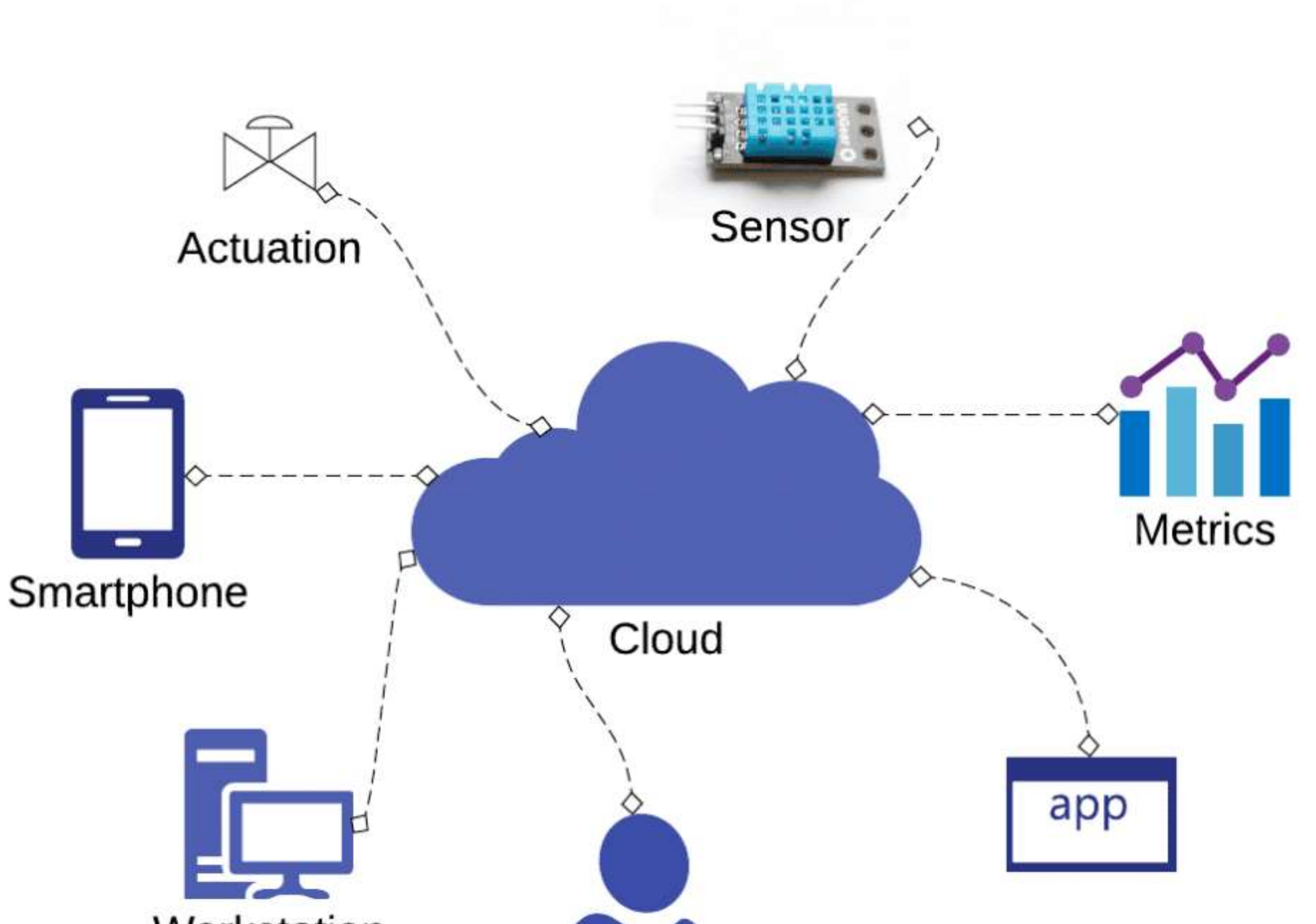


TABLE 1 SENSOR DETAILS

Sensor	Gas	Description	Range
MQ7	Carbon Monoxide	Suitable for sensing CO concentration in the air. It can detect CO-gas concentration anywhere from 20 to 2000 ppm CO is detected by method of cycle high and low temperature	0-100 (No effect) 100-800 (Risky) 800-2000 (Very high)
MQ2	Methane	Useful for gas leakage detection in home and industry. Can detect LPG i-butane, Propane, methane, alcohol, hydrogen and smoke	0-1000 (Normal) 1000-15000 (Risky) (15000 - 50000 Very High)
MQ135	Air Quality	Responsive to a wide scope of harmful gases like alcohol, acetone, thinner, formaldehyde and so on	0-500 (Normal) 500-1500 (Risky) 1500-2000 (Very high)


Sensor
data
Processor
Pollution Level:
High, Low, Medium
Analytical Module
Android
Cloud
Historical data
Prediction

Step 1: Connecting Arduino to Cloud
In step 1, IoT Kit is constructed in which the data generated by sensors is sent to the cloud, where it is processed and displayed to
the user in the appropriate form. The overview of phase 1 can be represented as shown in Figure 4.



First, the Arduino sketch is setup, and the connections are drawn. A program in Arduino is written that can find air quality based
on gas sensor readings. For example, a methane sensor (MQ-4) is used to sense methane gas in the air. The output of the MQ-4
sensor is displayed as an analog value on the COM screen of Arduino.
The MQ-4 sensor is connected to the Arduino board. Arduino, in turn, uses an ESP8266 Wi-Fi module to connect to the network
to send data.
Algorithm 1 for Designing App module and connecting past information to real-time current location information prediction.
Input: Measure pollution and air quality measures and condition prediction
Output: IoT- APP with all features pollution control and alert system

Algorithm

Step 1: Read the values from sensor

Step 2: The value sent for Arduino for processing value higher to threshold process then step 3

Step 3: AQI classified as high then step 7

Step 4: Value less than are equal to threshold AQI classified as Normal then goto step 7

Step 5: Value is less than min/max expected then step 6

Step 6: Sensor is damaged

Step 7: Data is stored appropriate format based on (Latitude, Longitude) the pollution at the place is determined then goto step 13
and 14

Step 8: Android application design components are divided in to phases step 9 and step 10

Step 9: To find the pollution level at particular area then step 11

Step 10: To know the path is taken is pollution safe then step 12

Step 11: Using GPS, (Latitude, Longitude) of the place is determined

Step 12: Source and destination is entered and intermediate place are determined

Step 13: Data displayed in appropriate form with help of current and past stored data

Step 14: Pollution path is drawn using suitable color conversion

In above mentioned algorithm 1 are contain two scenarios are there one is real-time pollution, air quality, and CO2 sensor
deployment to Arduino board for collecting current pollution state. To processing an android app design with first, fetching current
location and predicting current and future status of pollution and air quality using coloring system it is shown clearly in table 2.

IoT- To ensure that all the sensors are working as expected and the values are being retrieved correctly, it is essential to check each
of these conditions in the Arduino sketch by an appropriate algorithm. A sample algorithm for a CO sensor in the Arduino sketch
is shown below:

Algorithm 2 for monitoring CO (Sample Sensor)
Input : Pin numbers
Output : CO Level in the environment
Setting
Variables Used ɵ: - C o-level in the atmosphere
µ: - The value of the same gas clean air.
Ψ: - The value above which the pollution is considered high
α: - The time after which the user wants to re-read the value again from the sensor

Step 1: Read the CO value with delay of α. This data is logged to an excel file (So that they can be sent to a cloud).
Step 2: if (ɵ >ψ && ɵ< μ)
Then the value is sent to the cloud and is marked as normal in the range column
Step 3: if (ɵ<ψ)
Then send value to the system with value as high
Step 4: Sensor is not in correct working condition. Kindly re-Check it.
Once the correct sensor values are retrieved, they are sent to the cloud. In Arduino, an Arduino HTTP client is created that calls
a JSON service passing the data to cloud and stored in the cloud. After creating the Arduino HTTP client, a project is configured in
Ubidots using the Ubidots web interface so that the client can send data along with an authentication token. In the Ubidots web
interface, variables are created to store gas sensor values. Once the variables are configured, they can be used to send data. In the
cloud, data is stored in the following format:

- Location ID
- Latitude
- Longitude
- Timestamp
- Concentration of CO
- Concentration of CH_4
- Concentration of CO_2
- AQI

For each Location ID where sensors are placed, the corresponding latitude and longitude and the concentration of various gases

Step 2: Connecting Ubidots to Android

In the previous step, Arduino was connected to the cloud (i.e., Ubidots) such that sensor data is sent from the Arduino board to Ubidots. In this step, the developed Android application IoT- receives sensor data sent by Arduino using Ubidots services.

The following steps are undertaken in order to accomplish these tasks:

•Develop Android client: To handle the HTTP connection to the Ubidots server, an Android client is developed. For security
purposes, the authentication token is used.

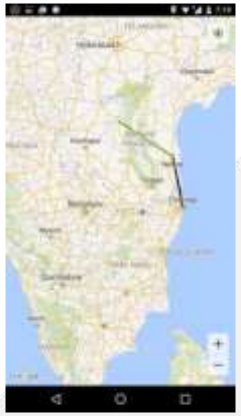
•Handle JSON format to read data: After requesting remote services by the android application, a JSON response is received
that is analyzed to extract data, i.e., the variable value (sensor information), timestamp, and AQI.

Step 1: Draw route from source to destination

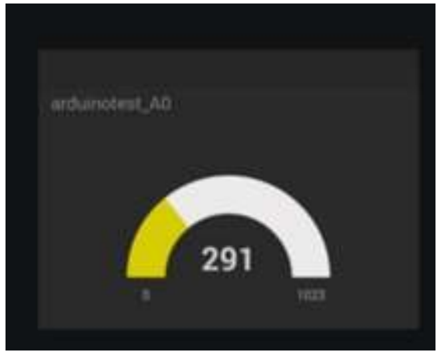
For a user traveling from a source to a destination, the pollution level of the entire route is predicted and a warning is displayed if the pollution level is too high so that the user can re- route his journey.

The data collected from the sensors and other trusted websites is made as are placed in a large database. When the user
enters his destination of travel, the IoT- android application first converts the address into corresponding latitude/longitude form.

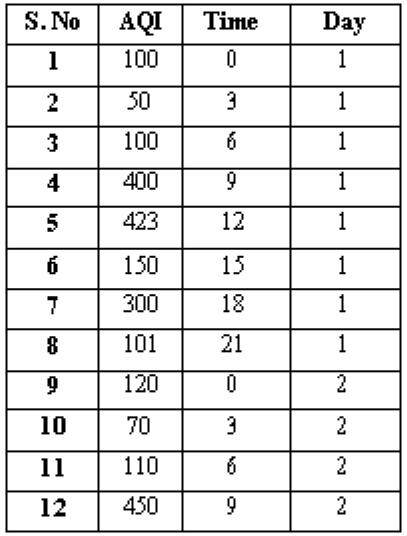
The latitude and longitude are searched in the cloud-based database. Intermediate places between the starting and finishing location
are also displayed. Suitable colors as shown in Table 2 are used to indicate the pollution level on the map. The route is drawn as
shown in Figure 6.



Step 2: Create charts to display data collected by IoT
Based on data retrieved by the sensor, using external libraries charts are drawn by android app as shown in Figure 7.



Step 3: Prediction and Analysis
Historical data can be used to predict pollution levels for subsequent days.
A dataset is maintained so that an evaluation of consecutive days can be done. Suppose AQI is mapped to 7 consecutive
days at a particular time such that on
Day 1: From time 12:00-15:00 the AQI is 423 (High)
Day 2: From time 12:00-15:00 the AQI is 500 (High)
Similarly, for the next five days, the AQI is very high from 12:00-15:00. Therefore, we can predict that the pollution level on the
eighth day at the same place and time will be high (i.e. ~400-500). Therefore, if the system is capable of logging daily data, it can
be used to warn the user not to travel during that time. This same data can be represented in the form of scatter plots and histogram
in order to make the analysis easier. In Table 3 the high concentration of dots in the time duration 12-15 shows that during this time
particular region is highly polluted.



S. No	AQI	Time	Day
1	100	0	1
2	50	3	1
3	100	6	1
4	400	9	1
5	423	12	1
6	150	15	1
7	300	18	1
8	101	21	1
9	120	0	2
10	70	3	2
11	110	6	2
12	450	9	2

Energy efficiency: the proposed model works with small devices, such as sensors, Arduino board, Ubidots (for Cloud)., which all
are low energy devices. Comparing with traditional pollution monitoring systems, it is more efficient and consume less energy.

CONCLUSION

Pollution in earlier days was negligible. Currently, however, pollution is increasing day-by-day because of various reasons such as
industrial growth, development of automobile industries, and chemical industries. Therefore, to reduce the level of pollution from
such sources and to protect humans and the environment from harmful gasses, this air pollution kit was developed that helps a
person to detect, monitor, and test air pollution in a given area. The kit has been integrated with the mobile application IoT- that
helps the user in predicting the pollution level of their entire route. Further, data logging can be used to predict AQI levels. This
proposed air pollution monitoring kit along with the integrated mobile application can be helpful to people suffering from respiratory
diseases. The app had following features, indices of air quality for a specific city using real-time computation, air quality daily
forecasts, timing outdoor activities for different recommendation of generation, air quality dips related to health risks, specific
reports for air quality measures based on locations, air quality maps generation. The proposed system faces with computational
complexity particularly when we are dealing with big sensor data. One solution could be using fog computing, instead of cloud
computing to reduce computation complexity and enhance the performance of the system. We can also implement zero tolerance
fast big data real-time stream analytical tools to process such a complex system.

[1] U. Varshney, Pervasive healthcare computing: EMR/EHR, wireless and health monitoring. Springer Science & Business
Media, 2009.
[2] J. J. Caubel, T. E. Cados, and T. W. Kirchstetter, “A New Black Carbon Sensor for Dense Air Quality Monitoring
Networks,” Sensors, vol. 18, no. 3, p. 738, 2018.
[3] L. Morawska, P. Thai, X. Liu, A. Asumadu-Sakyia, G. Ayoko, A. Bartonova, A. Bedini, F. Chai, B. Christensen, and M.
Dunbabin, “Applications of low-cost sensing technologies for air quality monitoring and exposure assessment: how far have they
gone?,” Environ. Int., 2018.
[4] D. Santi, E. Magnani, M. Michelangeli, R. Grassi, B. Vecchi, G. Pedroni, L. Roli, M. C. De Santis, E. Baraldi, and M.
Setti, “Seasonal variation of semen parameters correlates with environmental temperature and air pollution: A big data analysis
over 6 years,” Environ. Pollut., vol. 235, pp. 806–813, 2018.
[5] L. Spinelle, M. Gerboles, M. G. Villani, M. Aleixandre, and F. Bonavitacola, “Field calibration of a cluster of low-cost
commercially available sensors for air quality monitoring. Part B: NO, CO and CO2,” Sensors Actuators B Chem., vol. 238, pp.
706–715, 2017.
[6] N. Castell, F. R. Dauge, P. Schneider, M. Vogt, U. Lerner, B. Fishbain, D. Broday, and A. Bartonova, “Can commercial
low-cost sensor platforms contribute to air quality monitoring and exposure estimates?,” Environ. Int., vol. 99, pp. 293–302, 2017.
[7] S. Gaglio, G. Lo Re, G. Martorella, D. Peri, and S. D. Vassallo, “Development of an IoT environmental monitoring
application with a novel middleware for resource constrained devices,” in Proceedings of the 2nd Conference on Mobile and
Information Technologies in Medicine (MobileMed 2014), 2014.

THANK YOU