PRESENTED BY
BATCH 2

SMART AIR POLLUTION MONITORING SYSTEM

INSTITUTION: NSRIT
COMPANY: TIERRA AUTOMATIONS

INTRO

- Polluted air became the major issue in cities these days due to toxic gases released by human activities which directly effects human health. This needs special attention further.

- **The system we developed uses the gas sensors to detect the presence of harmful gases present in the air and constantly transmits data.**

TABLE OF CONTENTS

PROJECT OBJECTIVES

This device uses a gas sensor which detects harmful gases and calculates the air quality around the particular area.

The main objective is to monitor, analyze and log data about air quality to a remote server and maintain the data over the internet.

PROJECT OBJECTIVES

HARDWARE COMPONENTS

SENSOR

MQ-135 gas sensor

MICROCONTROLLER

ESP8266

DISPLAY

I2C LCD

SOFTWARE COMPONENTS

CODE

ARDUINO IDE

BLYNK SOFTWARE

about the components

MQ 135 gas sensor

- This gas sensor is used to detect various gases such as alcohol ,benzene , carbon dioxide ,nitrogen dioxide ,etc.. This sensor gives the output in analog format .
- This requires some pre-heating before it could give accurate results. This MQ 135 gas sensor is different from other sensors because this is highly sensitive to Ammonia ,Sulfide and also sensitive to smoke and other harmful gases and the other reason is it is affordable.
- **MQ 135 configuration:**

MQ 135	ESP8266
VCC	3V
GND	GND
A0	A0

about the components

MICRO CONTROLLER ESP8266

- The NodeMCU (Node MicroController Unit) is an open-source software and hardware development environment built around an inexpensive System-on-a-Chip (SoC) called the ESP8266
- It contains the crucial elements of a computer: CPU, RAM, networking (WiFi), and even a modern operating system and SDK. That makes it an excellent choice for Internet of Things (IoT) projects of all kinds.
- The ESP8266 12-E chip comes with 17 GPIO pins. Not all GPIOs are exposed in all ESP8266 development boards, some GPIOs are not recommended to use, and others have very specific functions.

I2C LCD DISPLAY

Actually ,LCD I2C is composed of a normal LCD ,an I2Cmodule and a potentiometer.

pin out :

LCD I2C uses I2C interface ,so it has 4 pins:

- GND Pin: Needs to be connected to GND(0v).
- VCC Pin :The power supply for the LCD needs to be connected to the VV(5V).
- SDA Pin: I2C data signal
- SCL Pin: I2C clock signal

I2C CONFIGURATION:

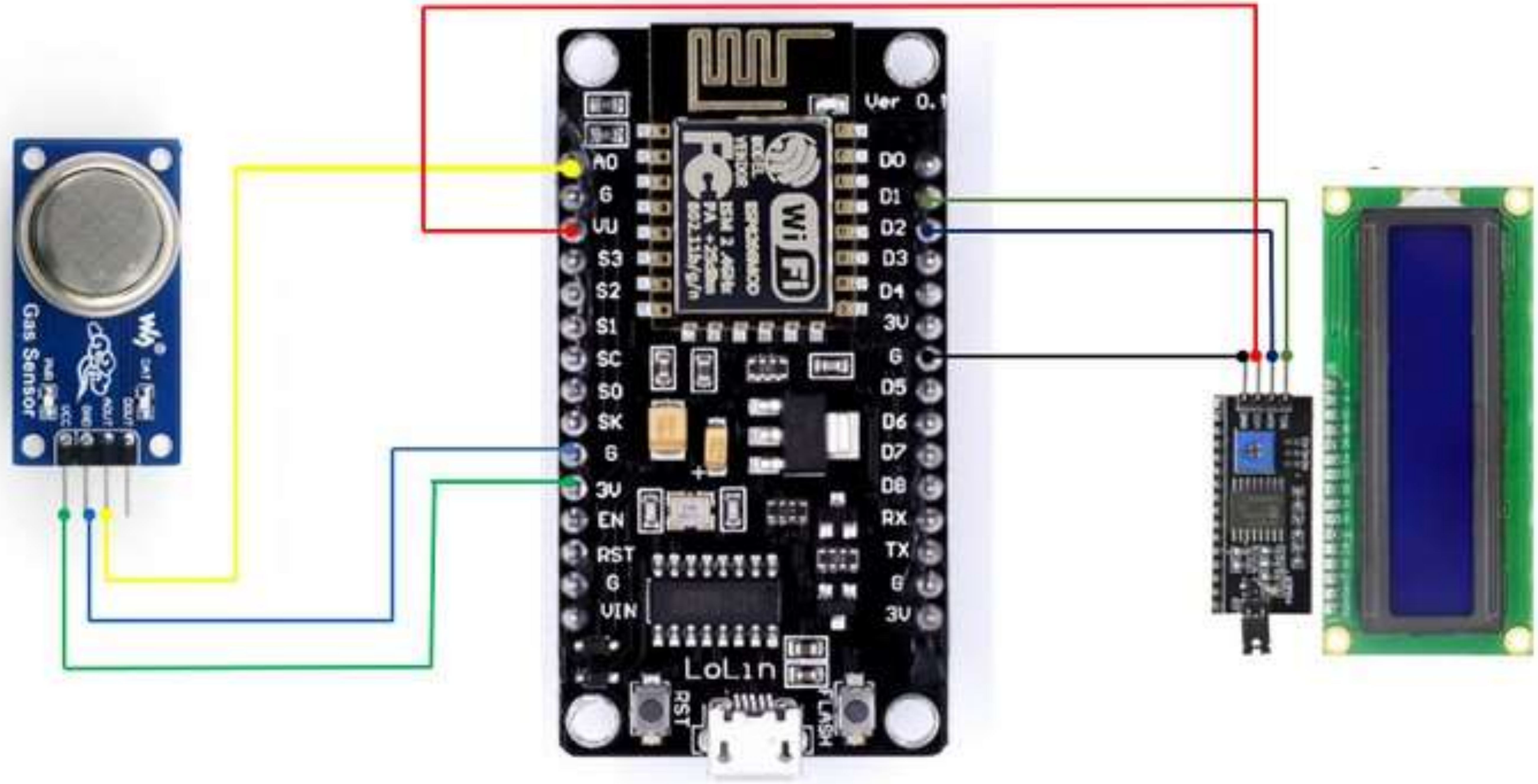
SDA Pin: D2
SCL Pin: D1

since D1 and D2 are best pins

Blynk was designed for the Internet of Things. It can control hardware remotely, it can display sensor data, it can store data, vizualize it and do many other cool things.

- Blynk App - allows to you create amazing interfaces for your projects using various widgets we provided.
- Blynk Server - responsible for all the communications between the smartphone and hardware. You can use our Blynk Cloud or run your private blynk server locally. It's open-source, could easily handle thousands of devices and can even be launched on a Raspberry Pi.

CIRCUIT DESIGN



code:-

```
#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,DEBUG); // 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",1000,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;
}
```

OVERALL PROJECT

SMART AIR POLLUTION MONITORING SYSYTEM

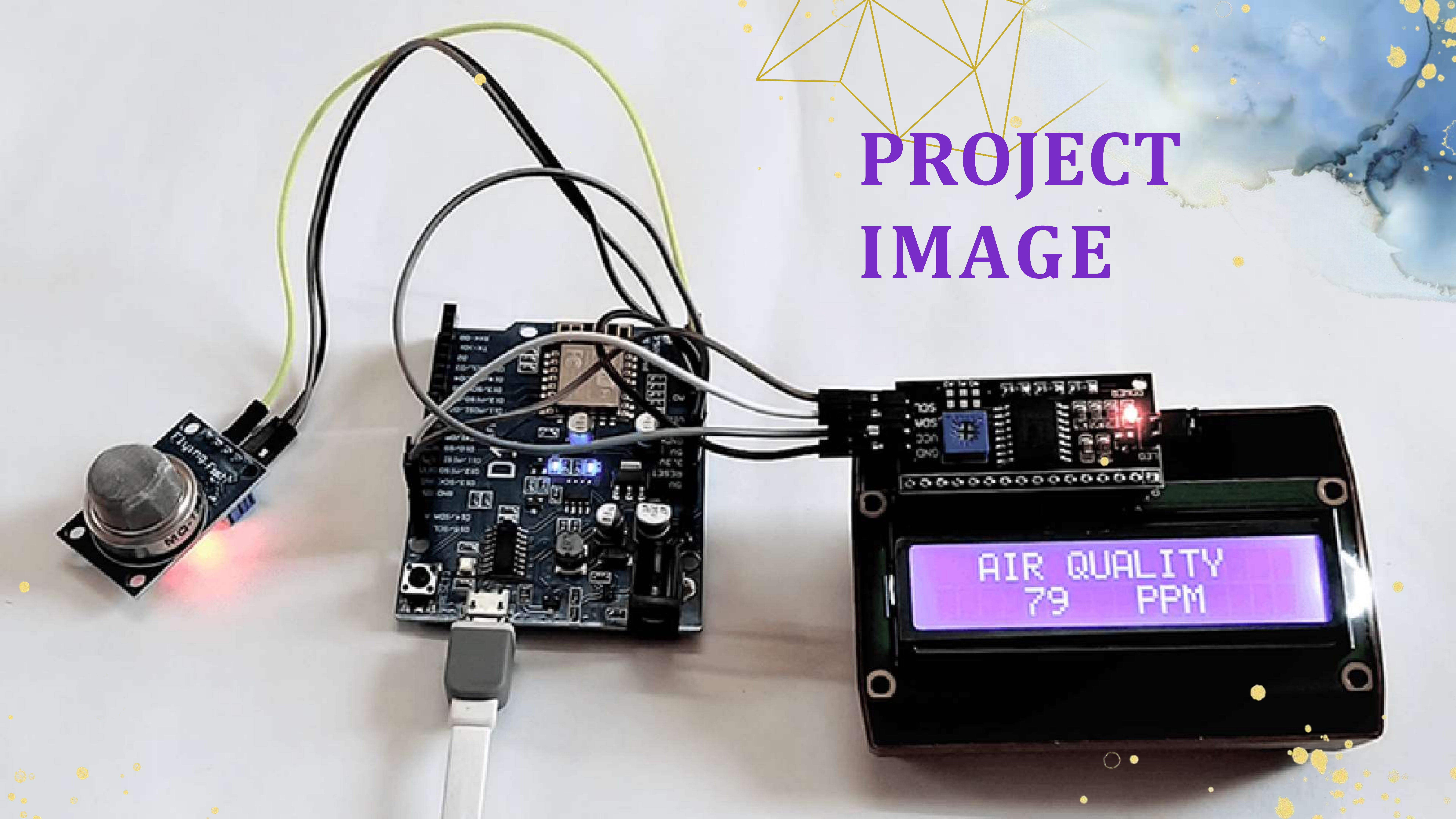
- The developed device is set up in a particular environment for which we are checking the air quality.
- The device has a Wi-Fi module through which it connects to the network.
- We would be dumping the code using Arduino IDE software.
- In the code itself the configurations and details of the Blynk App are given so that the device is now also connected to the Blynk software.
- Now after all the setup, we can now view the Air Quality (ppm) either in the device (I2C LCD) or in the Blynk Dashboard.

THE RESULT

The air Quality value in ppm can be viewed from time to time in the I2C LCD display board. If the ppm value is far more than safe, we can see an alert displayed on the board.

In the Blynk we create a datastream that displays the Air Quality in ppm over a meter and if there are harmful gases we would be sent a notification on our mobile phones through the app.

PROJECT IMAGE



PROJECT CONCLUSION

Using this device we can monitor the air quality over a particular area, city or an industry.

As the quality of the air is updated to us from time to time, we could take necessary steps to prevent the pollution.

This particular project can be implemented as an idea in Smart cities.

FURTHER IMPROVEMENTS:

We can improve the project further by adding few other features, Some of them are listed here:

- Other gas sensors like MQ-6, MQ-4 can be added to the device to detect other harmful gases also.
- An alarm or a Buzzer can be connected so that in case of any alert we would be receiving a sound signal.
- Another feature can be said as an extension to the Blynk in which the software sends a mail to the personal mail id in case of any alert.

Our Team

ALICE CROSBY

KIRANMAI

PRASEETHA

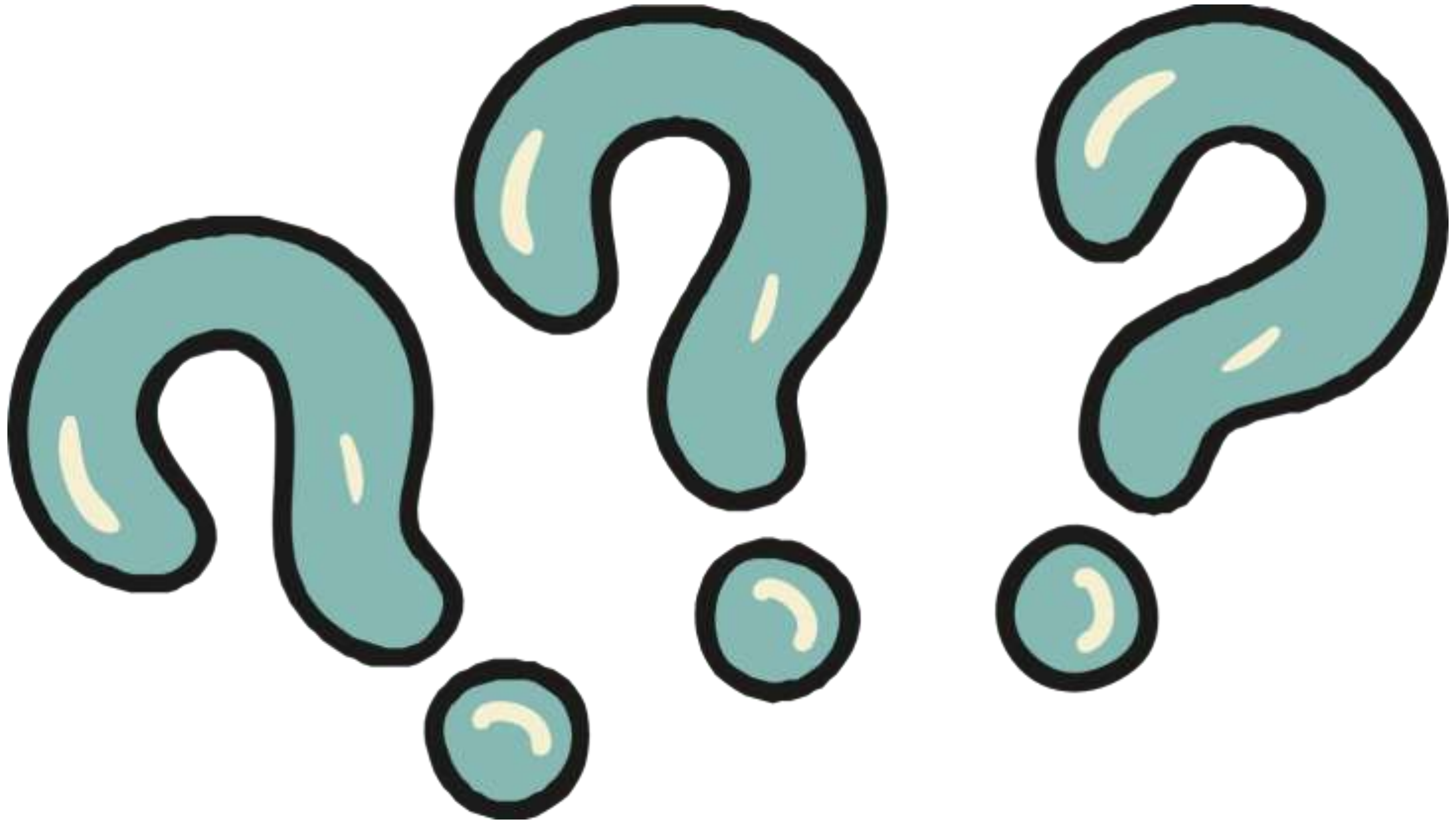
GOUS IMAM

NARESH

DINESH

CHANDARA RAO

VENKATA SAI



ANY QUERIES ?


Thank
You!