

IOT PHASE 3

AIR QUALITY MONITORING

OVER VIEW:

An IoT-based air and sound pollution monitoring system is implemented using a network of sensors, connectivity technologies, and data analytics platforms.

Air quality sensors are deployed in strategic locations to measure pollutant levels such as particulate matter, gases, and volatile organic compound(VOCs)

COMPONENTS:

REQUIRED COMPONENTS:

1.ARDUINO UNO

2.16X2 LCD SCREEN

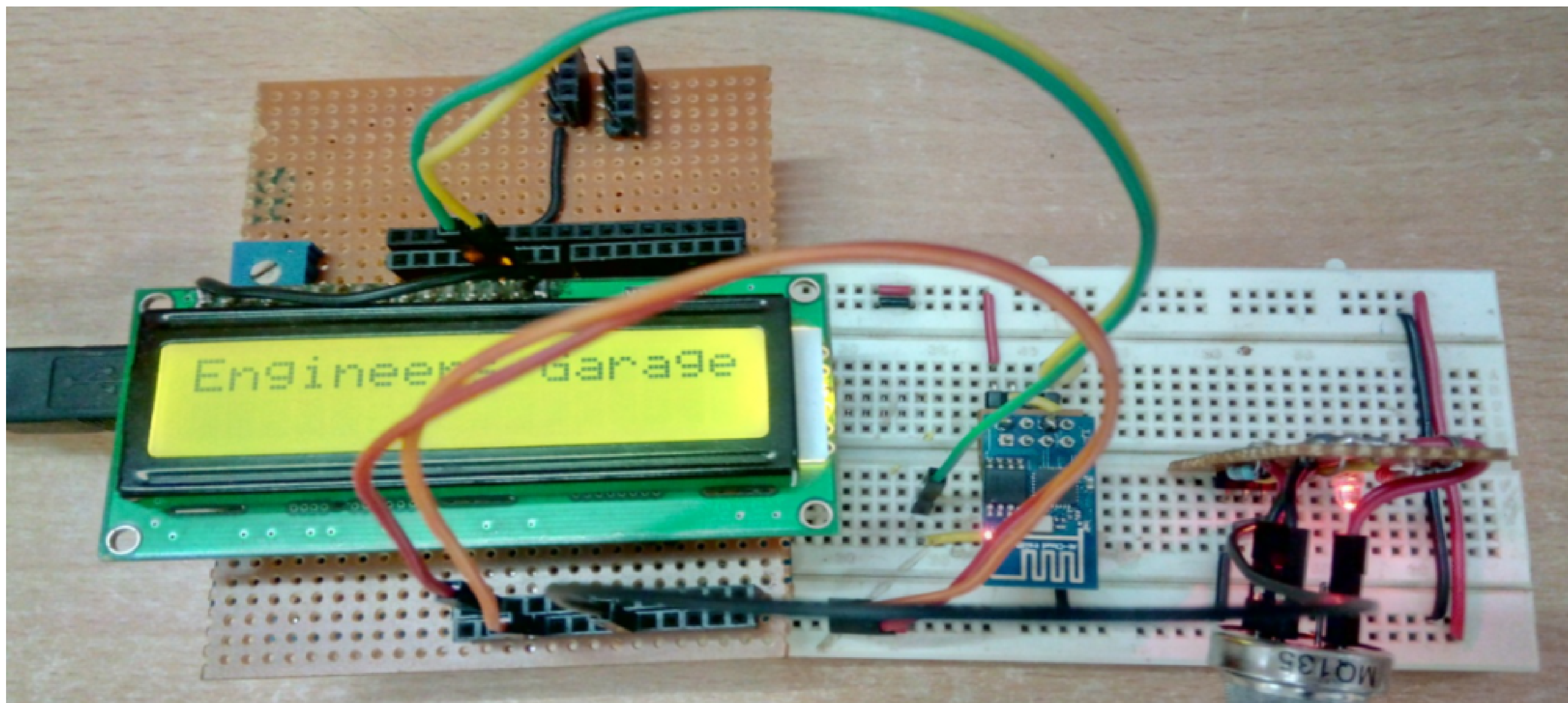
3.MQ135 SENSOR

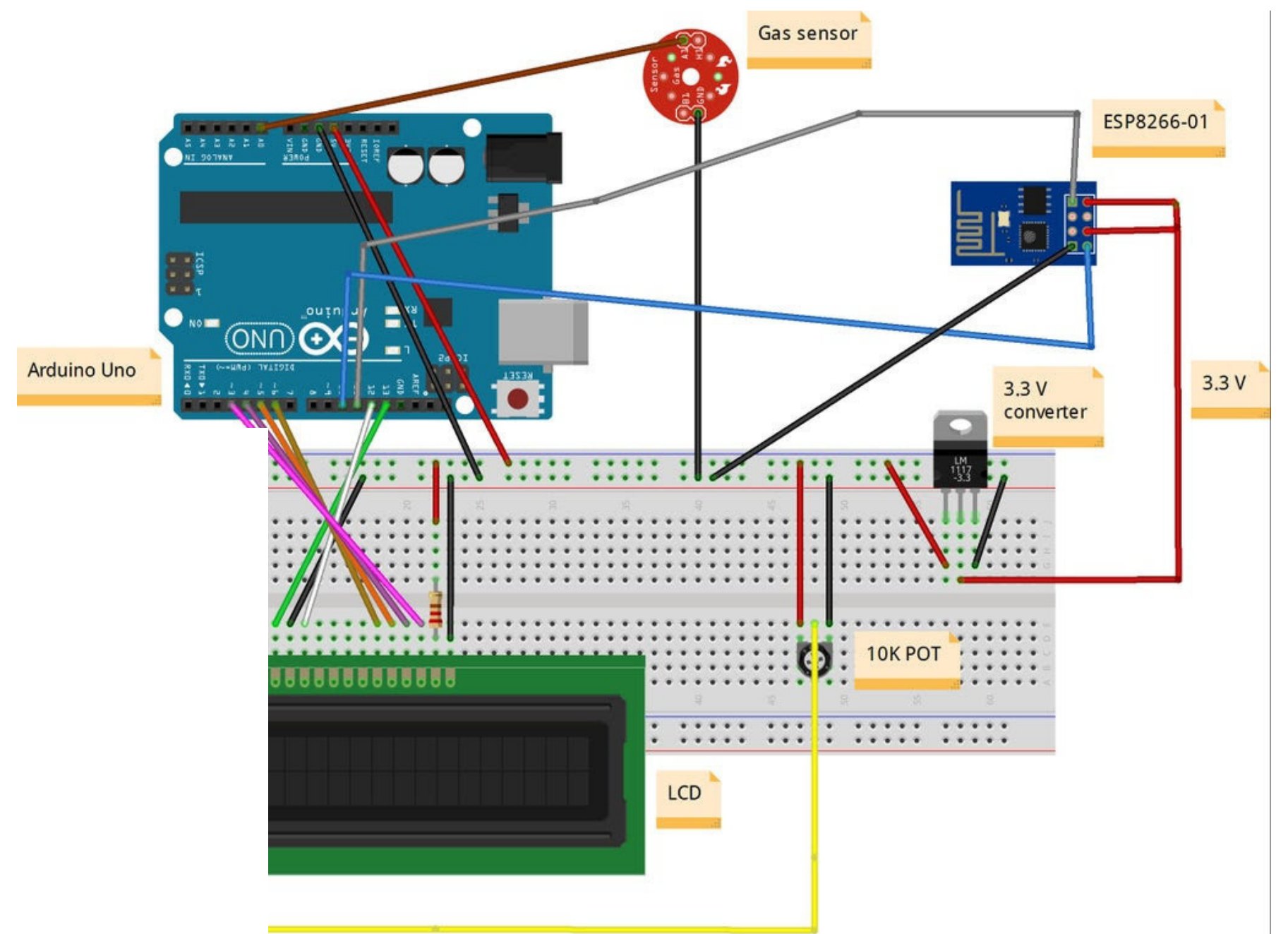
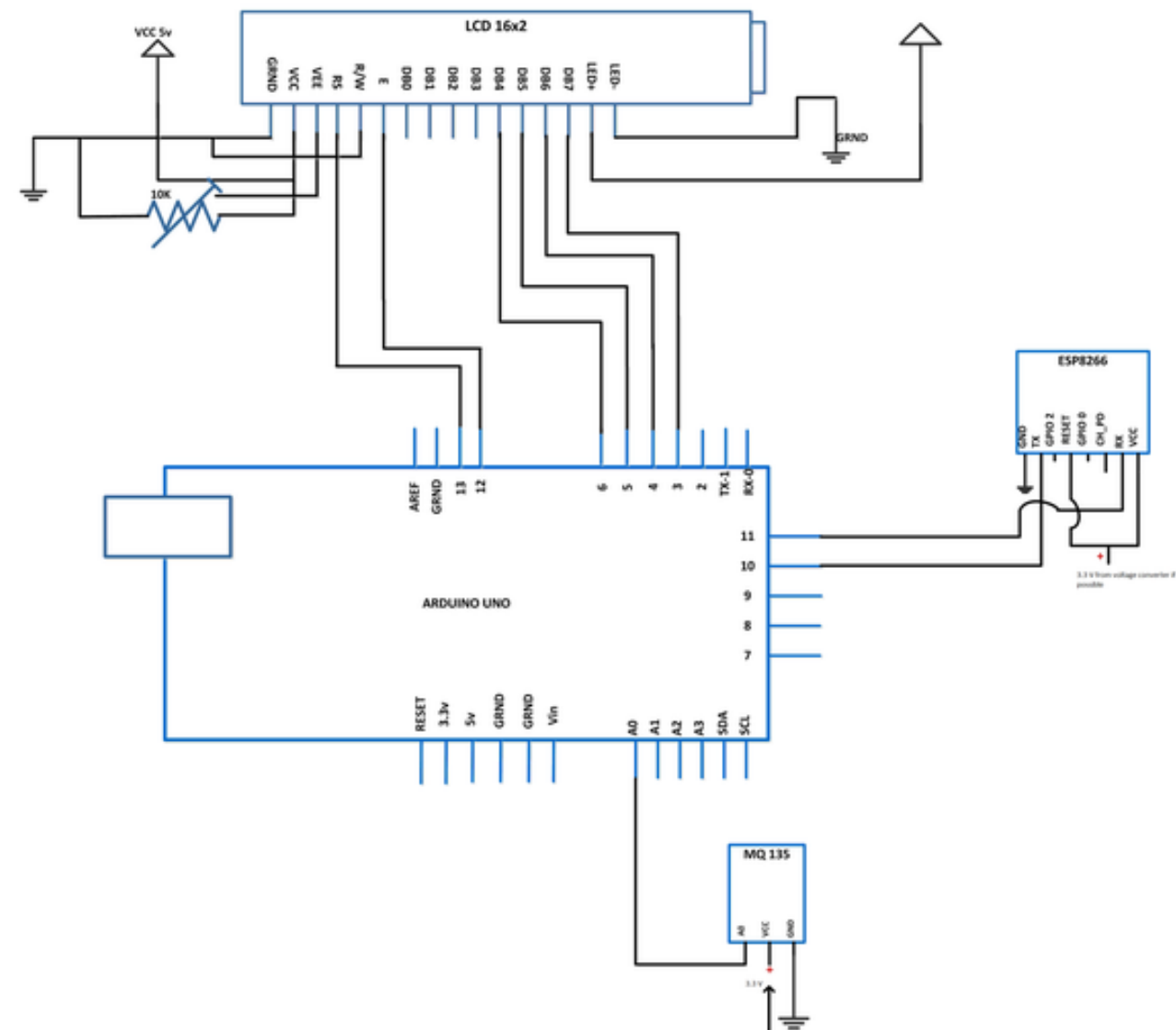
4.ESP8266 MODULE

5.WIFI

6.CONNECTING WIRES

Circuit Connections —

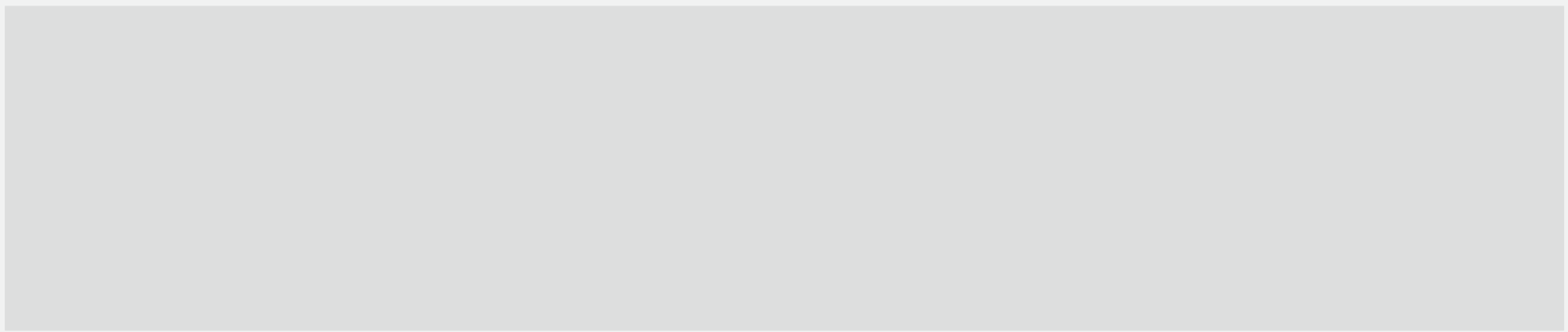
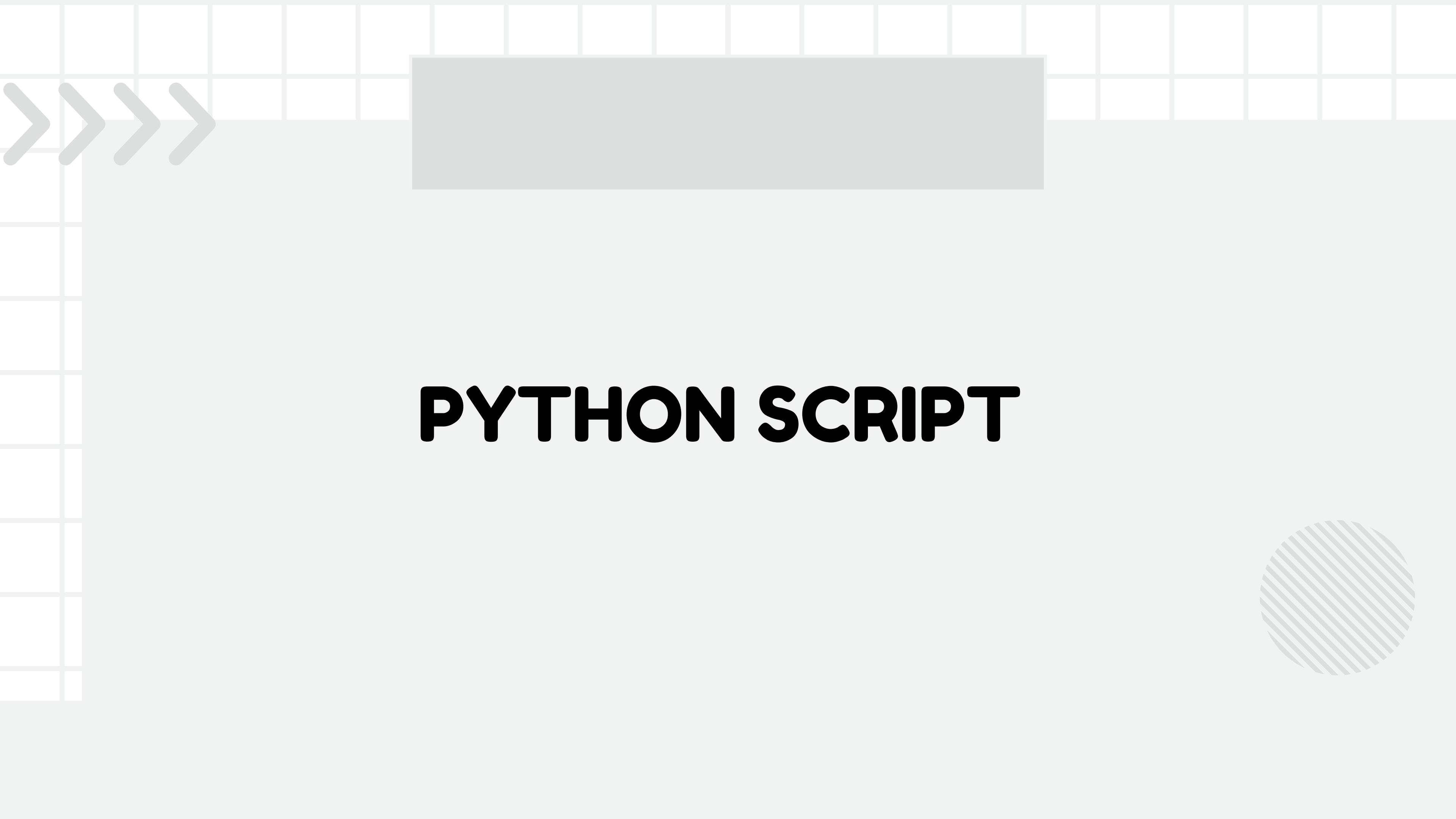




fritzing

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.



PYTHON SCRIPT




```
//PROGRAM TO
#include <SOFTWARESERIAL.H>
#include <LIQUIDCRYSTAL.H>
LIQUIDCRYSTAL LCD(13, 12, 6, 5, 4, 3)
FLOAT T=0;
CHAR DATA = 0;
// REPLACE WITH YOUR CHANNEL'S
THINGSPEAK API KEY
STRING APIKEY = "8NBNB4VQ9F2EEWQM";
// CONNECT 10 TO TX OF SERIAL USB
// CONNECT 11 TO RX OF SERIAL USB
SOFTWARESERIAL SER(10,11); // RX, TX
// THIS RUNS ONCE

VOID SETUP()
  // ENABLE DEBUG SERIAL
  //SERIAL.BEGIN(9600);
  // ENABLE SOFTWARE SERIAL
  SER.BEGIN(9600);
  LCD.BEGIN(16, 2);
  LCD.SETCURSOR(0,0);
  LCD.PRINT("ENGINEERS GARAGE");
  LCD.SETCURSOR(0,1);
  LCD.PRINT("          ");
  DELAY(3000);
  LCD.CLEAR();
  LCD.SETCURSOR(0,0);
  LCD.PRINT("    IOT AIR");
  LCD.SETCURSOR(0,1);
  LCD.PRINT("QUALITY MONITOR");
  DELAY(3000);
```

```
// pinMode(12, INPUT);
  // reset ESP8266 WiFi
  connection AT+CIPMUX=1
  AT+CWJAP
    ser.println("AT");
    delay(1000);
    ser.println("AT+GMR");
    delay(1000);
    ser.println("AT+CWMODE=3");
    delay(1000);
    ser.println("AT+RST");
    delay(5000);
    ser.println("AT+CIPMUX=1");
    delay(1000);
    String
    cmd="AT+CWJAP="Engineers
    Garage","egP@$w0rd?";
    ser.println(cmd);
    delay(1000);
    ser.println("AT+CIFSR");
    delay(1000);
```



```
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("    WIFI");
    lcd.setCursor(0,1);
    lcd.print("  CONNECTED"); }
    // the loop
    void loop()
    {
        delay(1000);
        t = analogRead(A0);
        Serial.print("Airquality = ");
        Serial.println(t);
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("  SENDING DATA");
        lcd.setCursor(0,1);
        lcd.print("    TO CLOUD");
        esp_8266();
    }
```

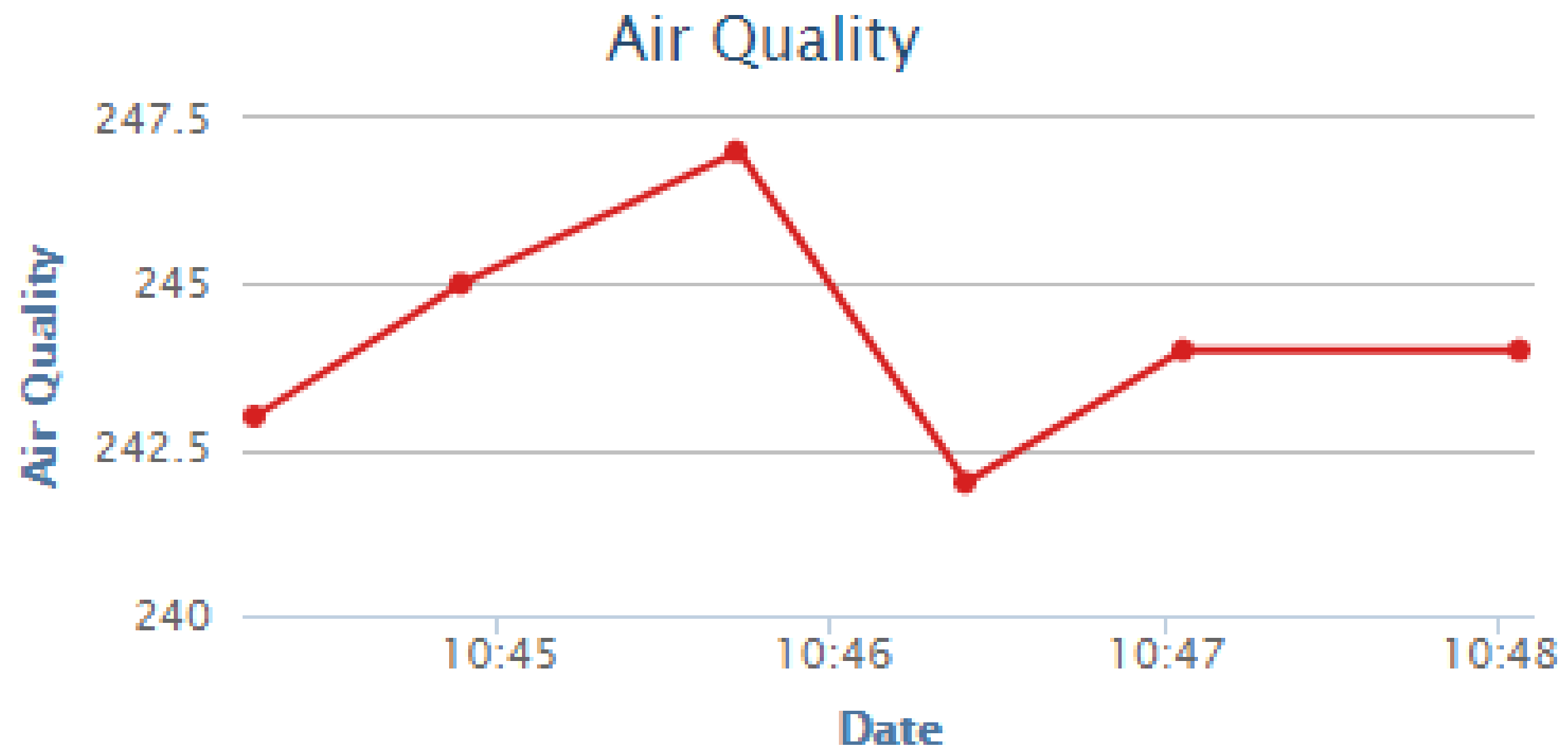
```
void esp_8266()
{
    // TCP connection
    AT+CIPSTART=4,"TCP","184.106.153.14
        9",80
    String cmd =
    "AT+CIPSTART=4,\"TCP\",\"";
    cmd += "184.106.153.149"; //
    api.thingspeak.com
    cmd += "\",80";
    ser.println(cmd);
    Serial.println(cmd);
    if(ser.find("Error"))
    {
        Serial.println("AT+CIPSTART
        error");
        return;
    }
```



- // prepare GET string GET
https://api.thingspeak.com/update?
api_key=LHAG4NSIYJ5UWS6U&field1=0rnrn
String getStr = "GET /update?api_key=";
- getStr += apiKey;
- //getStr += "&field1=";
- //getStr += String(h);
- getStr += "&field1=";
- getStr += String(t);
- getStr += "rnrn";
- // send data length
- cmd = "AT+CIPSEND=4,";
- cmd += String(getStr.length());
- ser.println(cmd);
- Serial.println(cmd);
- delay(1000);
- ser.print(getStr);
- Serial.println(getStr);
- // thingspeak needs 15 sec delay between updates
- delay(16000);
- }
-
- ###
-

Software
Output

Field 1 Chart



ThingSpeak.com