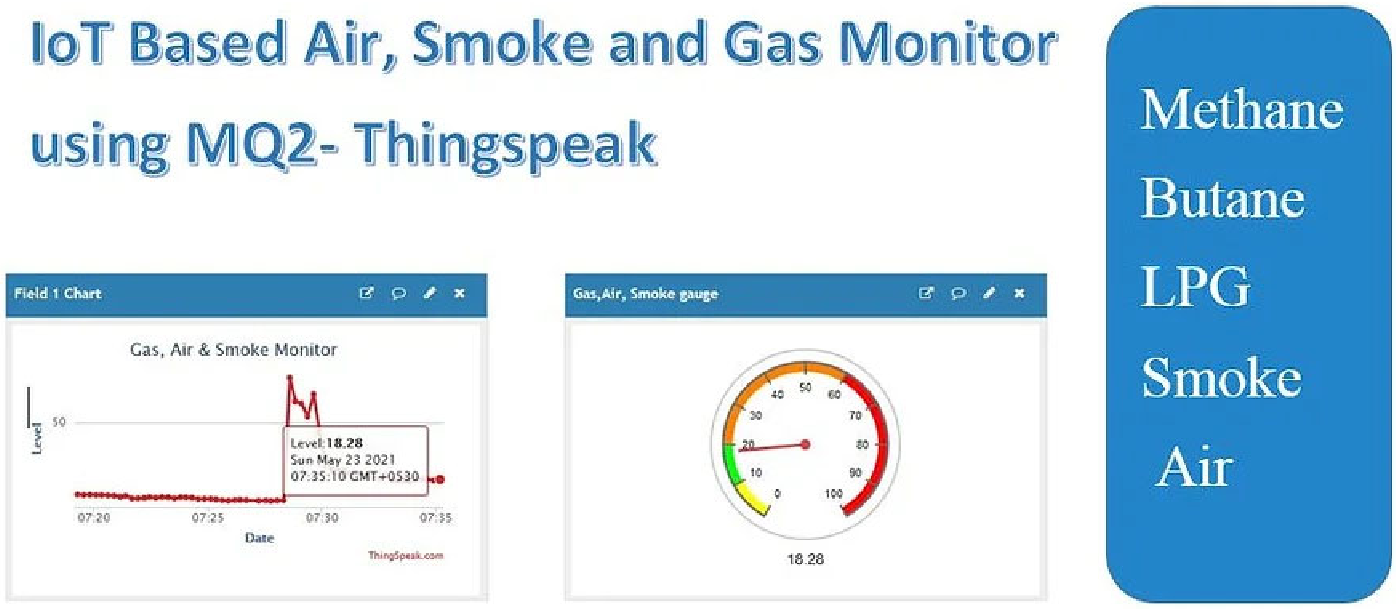
Ramesh G May 23, 2021 4 min read

IoT Based Air, Smoke and Gas Monitor using MQ2 and Thingspeak Platform

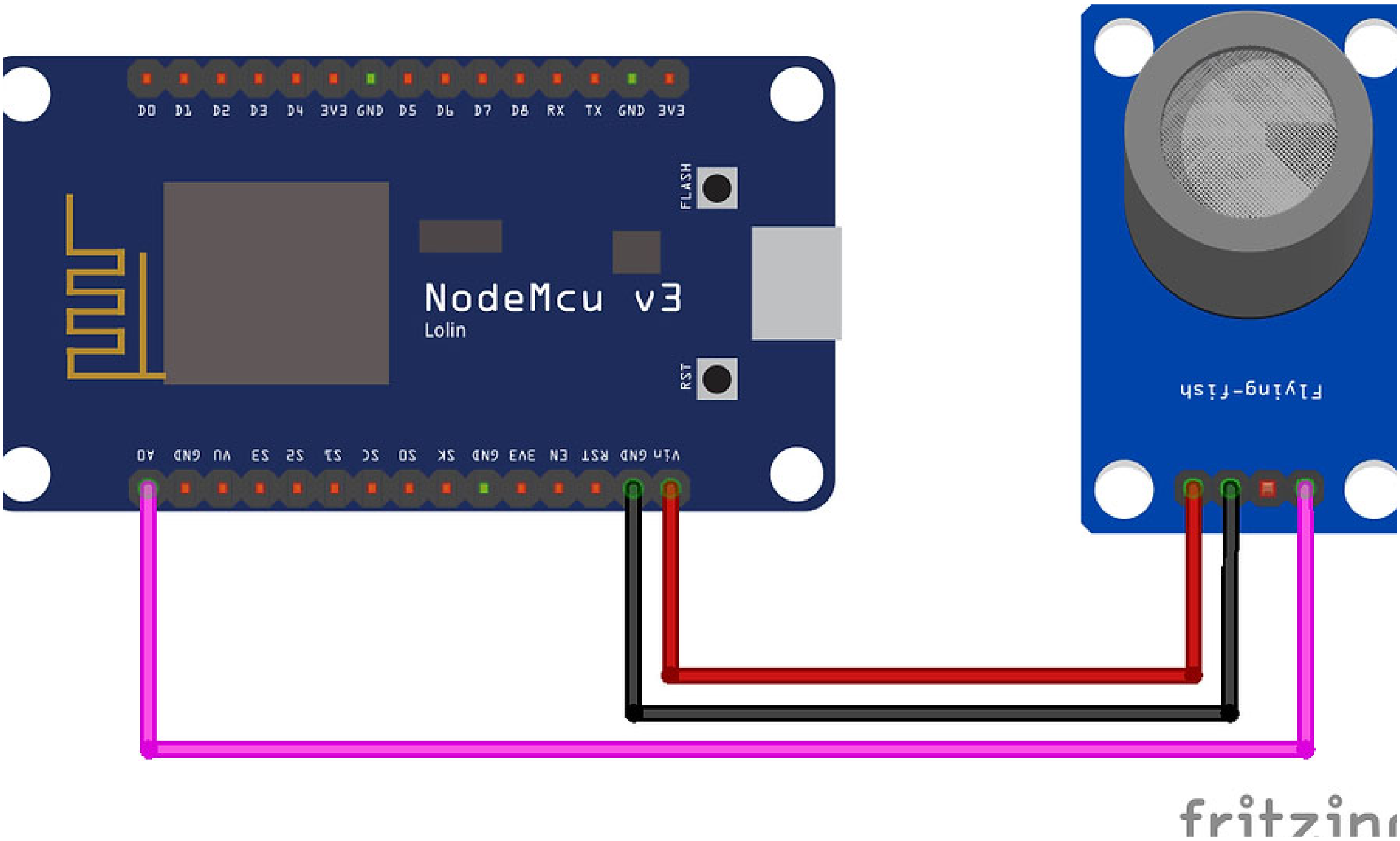
Updated: May 31, 2021

Here to make Smoke and Gas Level Monitor Using MQ-2 Gas Sensor Module, thingspeak server. The measured the quantity of gas% in the air data can be monitored and stored in thingspeak server and data send and accessible over the internet.



In this tutorial, we are going to discuss how to interface the MQ2 smoke detector sensor with NodeMCU. The MQ2 sensor is a powerful sensor in order to measure different types of gases like Butane, Methane, LPG, Smoke, Alcohol, Propane, Hydrogen concentrations in the air.

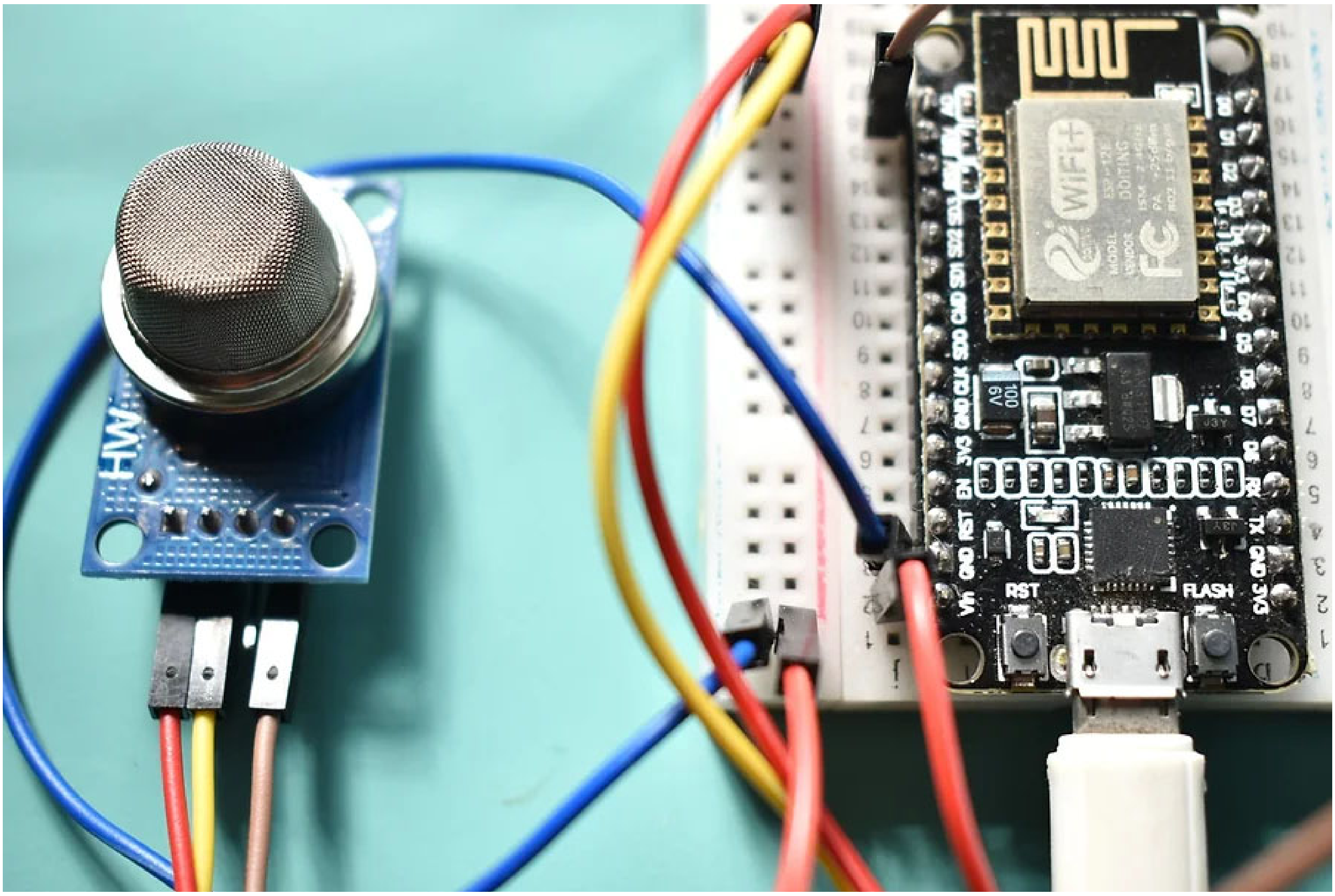
Circuit diagram


D0 D1 D2 D3 D4 3V3 GND D5 D6 D7 D8 RX TX GND 3V3
FLASH
NodeMcu v3
Lolin
RST
A0 GND VU S3 S2 S1 SC S0 SK GND 3V3 EN RST GND Vin
Flying-fish
fritzing

Components Required

ESP8266 12e- NodeMCU - 1 no

MQ2 Sensor - 1 no



MQ2 Sensor

Gas sensor is a device which detects the concentration level of the gases present in the atmosphere. This project contains interfacing of MQ2 sensor with Arduino and its code.

When it comes to measuring or detecting a particular Gas the MQ series Gas sensors are the most commonly used ones. These sensors can either be purchased as a module or as just the sensor alone. If you are trying to only detect (not measuring ppm) the presence of a gas then you can buy it as a module since it comes with an op-amp comparator and a digital out pin. But if you planning to measure the ppm of a gas it is recommended to buy the sensor alone (without module).

Where to use MQ-2 Gas sensor:

The MQ-2 Gas sensor can detect or measure gasses like LPG, Alcohol, Propane, Hydrogen, CO and even methane. The module version of this sensor comes with a Digital Pin which makes this sensor to operate even without a microcontroller and that comes in handy when you are only trying to detect one particular gas. When it comes to measuring the gas in ppm the analog pin has to be used, the analog pin also TTL driven and works on 5V and hence can be used with most common microcontrollers.

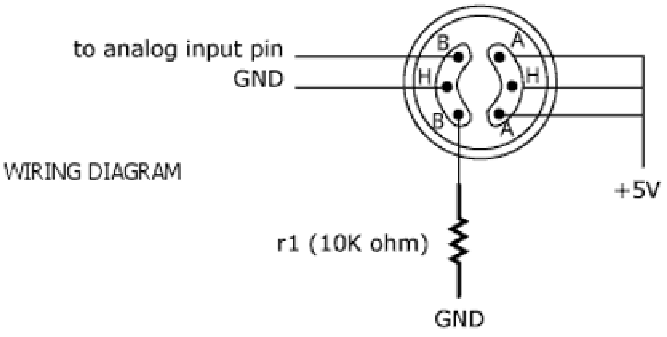
So if you are looking for a sensor to detect or measure gasses like LPG, Alcohol, Propane, Hydrogen, CO and even methane with or without a microcontroller then this sensor might be the right choice for you.

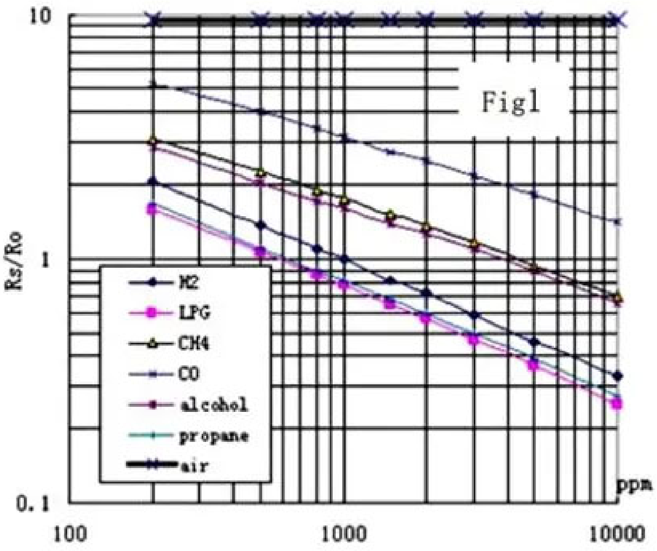
How to use MQ-2 Sensors to detect gas:

Using an MQ sensor it detects a gas is very easy. You can either use the digital pin or the analog pin to accomplish this. Simply power the module with 5V and you should notice the power LED on the module to glow and when no gas it detected the output LED will remain turned off meaning the digital output pin will be 0V. Remember that these sensors have to be kept on for pre-heating time (mentioned in features above) before you can actually work with it. Now, introduce the sensor to the gas you want to detect and you should see the output LED to go high along with the digital pin, if not use the potentiometer until the output gets high. Now every time your sensor gets introduced to this gas at this particular concentration the digital pin will go high (5V) else will remain low (0V).

You can also use the analog pin to achieve the same thing. Read the analog values (0-5V) using a microcontroller, this value will be directly proportional to the concentration of the gas to which the sensor detects. You can experiment with this values and check how the sensor reacts to different concentration of gas and develop your program accordingly.

How to use the MQ-2 sensor to measure PPM:





If you are looking for some accuracy with your readings then measuring the PPM would be the best way to go with it. It can also help you to distinguish one gas from another. So to measure PPM you can directly use a module. A basic wiring for the sensor from datasheet is shown side.

The procedure to measure PPM using MQ sensor is the same but few constant values will vary based on the type of MQ sensor used. Basically, we need to look into the (Rs/Ro) VS PPM graph given in the datasheet (also shown below).

The value of Ro is the value of resistance in fresh air and the value of Rs is the value of resistance in Gas concentration. First, you should calibrate the sensor by finding the values of Ro in fresh air and then use that value to find Rs using the formulae

f sensor(Rs): Rs=(V

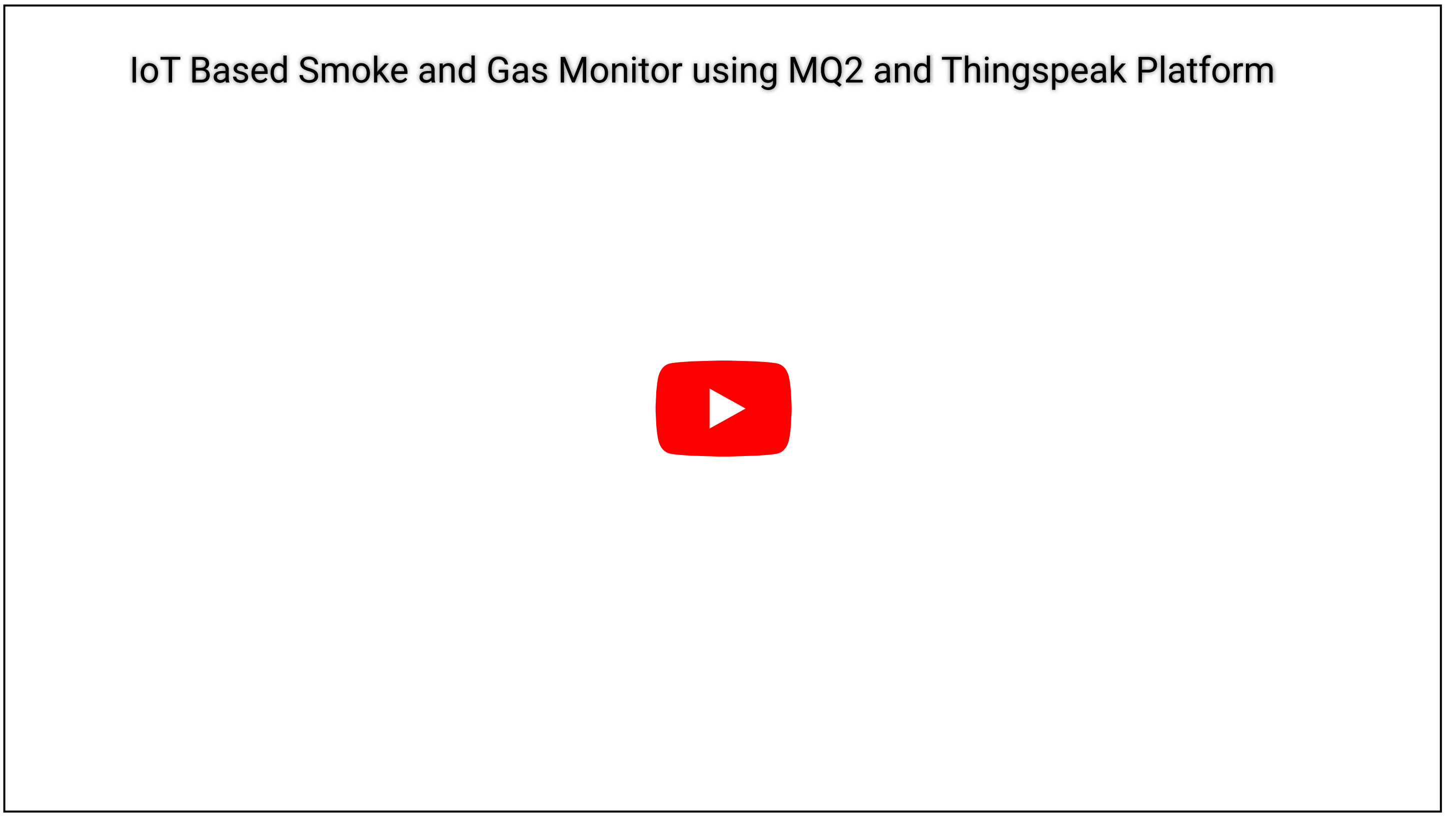
Once we calculate Rs and Ro we can find the ratio and then use the graph shown above we can calculate the equivalent value of PPM for that particular gas.

Applications:

- Detects or measure Gases like LPG, Alcohol, Propane, Hydrogen, CO and even methane
- Air quality monitor
- Gas leak alarm
- Safety standard maintenance
- Maintaining environment standards in hospitals

NodeMCU

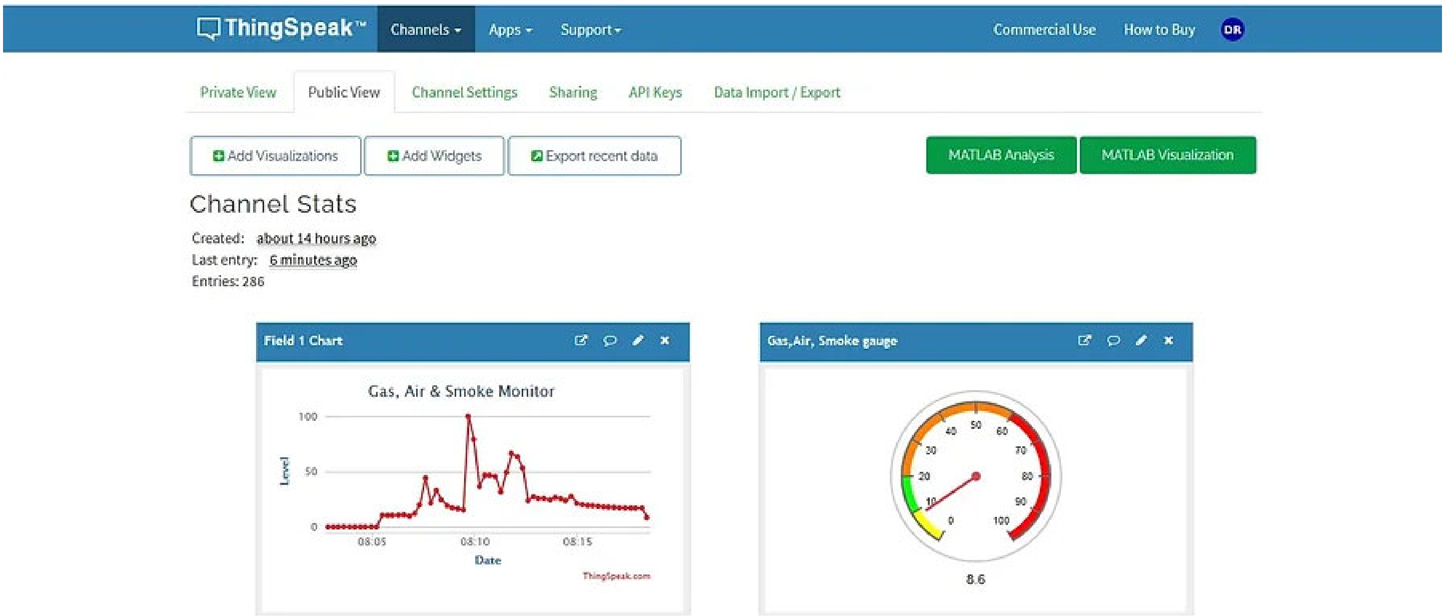
NodeMCU ESP8266-12E MCU is a development board with one analogue and many general-purpose input output (GPIO) pins. It has 4MB flash memory, and can operate at a default clock frequency of 80MHz. In this project, Analogl pin A0 of NodeMCU is used to read Data of MQ2 Sensor.


IoT Based Smoke and Gas Monitor using MQ2 and Thingspeak Platform

Thingspeak

ThingSpeak is an IoT analytics platform service that allows you to aggregate, visualize, and analyze live data streams in the cloud. You can send data to ThingSpeak from your devices, create instant visualization of live data, and send alerts.

https://thingspeak.com/ and create an account.



Channel Stats

Created: about 14 hours ago
Last entry: 6 minutes ago
Entries: 286

Then, upload the code to your NodeMCU board. Make sure you have selected the right board and COM port. Also, make sure you've inserted your **WiFi Credentials** in the code.

After a successful upload, open the Serial Monitor at a baud rate of 115200. Press the "**EN/RST**" button on the ESP8266 board.

Arduino code

Subscribe and Get code.

```
#include <ESP8266WiFi.h>

//Wifi Cedentials
const char* ssid = "xxxxxxxxxxxxxxxxx";  // your SSID
const char* password = "xxxxxxxxxxxxxxx"; // your wIFi password

//Thingspeak

String apiKey = "xxxxxxxxxxxx"; // API key
const char* server = "api.thingspeak.com";




WiFiClient client;

float Level;

void setup()
{
Serial.begin(115200);
delay(10);
Serial.println("Connecting to ");
Serial.println(ssid);
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED)
{
delay(500);
Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");
}
void loop()
{
float reading = analogRead(A0);
Level = ((reading/1023)*100);
if (isnan(Level))
{
Serial.println("Failed to read from MQ-2 sensor!");
return;
}
if (client.connect(server, 80)) // "184.106.153.149" or api.thingspeak.com
{
   String postStr = apiKey;
   postStr +="&field1=";
   postStr += String(Level);
   postStr += "\r\n\r\n";

   client.print("POST /update HTTP/1.1\n");
   client.print("Host: api.thingspeak.com\n");
   client.print("Connection: close\n");
   client.print("X-THINGSPEAKAPIKEY: " + apiKey + "\n");
   client.print("Content-Type: application/x-www-form-urlencoded\n");
   client.print("Content-Length: ");
   client.print(postStr.length());
   client.print("\n\n");
   client.print(postStr);
Serial.print("Gas, Smoke, Air Level: ");
Serial.println(Level);
Serial.println("Data Send to Thingspeak");
}
delay(500);
client.stop();
Serial.println("Waiting...");

// thingspeak needs minimum 15 sec delay between updates.
delay(1500);
}
```