

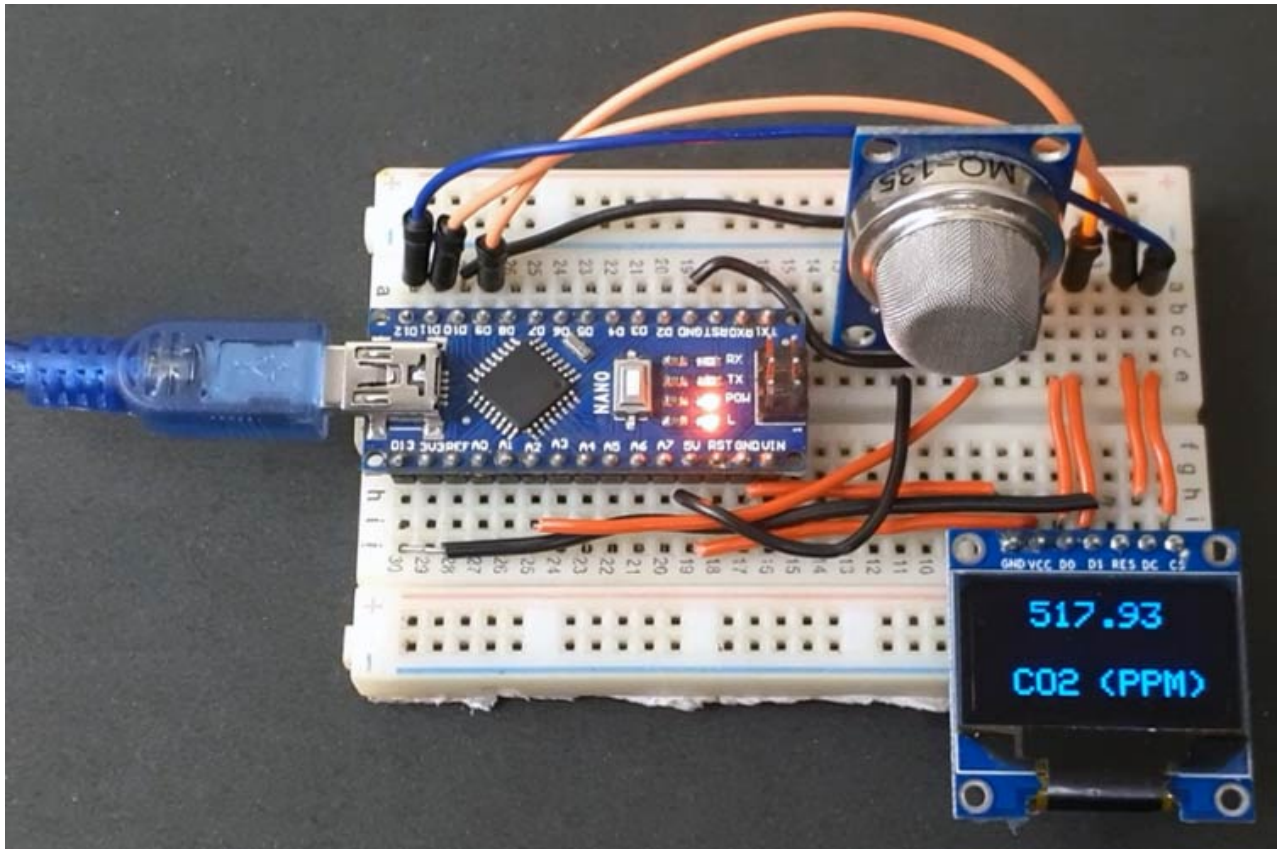
Measuring CO2 Concentration in Air using Arduino and MQ-135 Sensor

circuitdigest.com/microcontroller-projects/interfacing-mq135-gas-sensor-with-arduino-to-measure-co2-levels-in-ppm

November 17, 2020

Arduino

By Ashish Choudhary Nov 17, 2020 2



The earth's Atmospheric CO2 level is increasing day by day. The global average atmospheric carbon dioxide in 2019 was 409.8 parts per million and in October-2020 it is 411.29. Carbon dioxide is a key greenhouse gas and responsible for about three-quarters of emissions. So **CO2 level monitoring** has also started to gain importance.

In our previous project, we used the Gravity Infrared CO2 sensor to measure the CO2 concentration in air. In this project, we are going to use an MQ-135 sensor with Arduino to measure the CO2 concentration. The measured CO2 concentration values will be displayed on the OLED module and last we will also compare the **Arduino MQ-135 sensor** readings with Infrared CO2 sensor readings. Apart from CO2, we have also measured the concentration of LPG, Smoke, and Ammonia gas using Arduino.

Components Required

- Arduino Nano
- MQ-135 Sensor

- Jumper Wires
- 0.96" SPI OLED Display Module
- Breadboard
- 22KΩ Resistor

0.96" OLED Display Module

OLED (Organic Light-Emitting Diodes) is a self light-emitting technology, constructed by placing a series of organic thin films between two conductors. A bright light is produced when an electric current is applied to these films. OLEDs are using the same technology as televisions, but have fewer pixels than in most of our TVs.



For this project, we are using a Monochrome 7-pin SSD1306 0.96" [OLED display](#). It can work on three different communications Protocols: SPI 3 Wire mode, SPI four-wire mode, and I2C mode. You can also learn more about the [basics of OLED display and its types](#) by reading the linked article. The pins and its functions are explained in the table below:

Pin Name	Other Names	Description
Gnd	Ground	Ground pin of the module

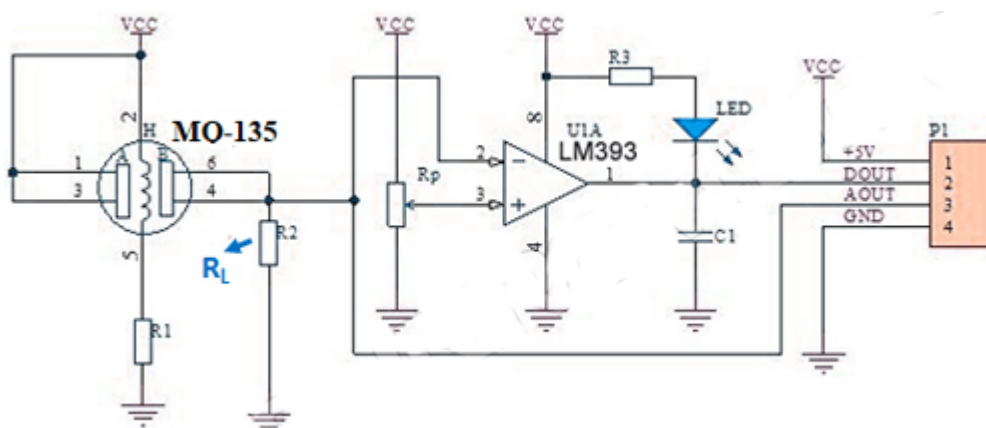
Vdd	Vcc, 5V	Power pin (3-5V tolerable)
SCK	D0,SCL,CLK	Acts as the clock pin. Used for both I2C and SPI
SDA	D1,MOSI	Data pin of the module. Used for both IIC and SPI
RES	RST, RESET	Resets the module (useful during SPI)
DC	A0	Data Command pin. Used for SPI protocol
CS	Chip Select	Useful when more than one module is used under SPI protocol

OLED Specifications:

- OLED Driver IC: SSD1306
- Resolution: 128 x 64
- Visual Angle: >160°
- Input Voltage: 3.3V ~ 6V
- Pixel Colour: Blue
- Working temperature: -30°C ~ 70°C

Preparing the MQ-135 Sensor

MQ-135 Gas Sensor is an air quality sensor for detecting a wide range of gases, including NH₃, NO_x, alcohol, benzene, smoke, and CO₂. MQ-135 sensor can either purchased as a module or just as a sensor alone. In this project, we are using an MQ-135 sensor module to measure CO₂ concentration in PPM. The circuit diagram for the MQ-135 board is given below:



The load resistor R_L plays a very important role in making the sensor work. This resistor changes its resistance value according to the concentration of gas. According to the [MQ-135 datasheet](#), the load resistor value can range anywhere from 10K Ω to 47K Ω . The

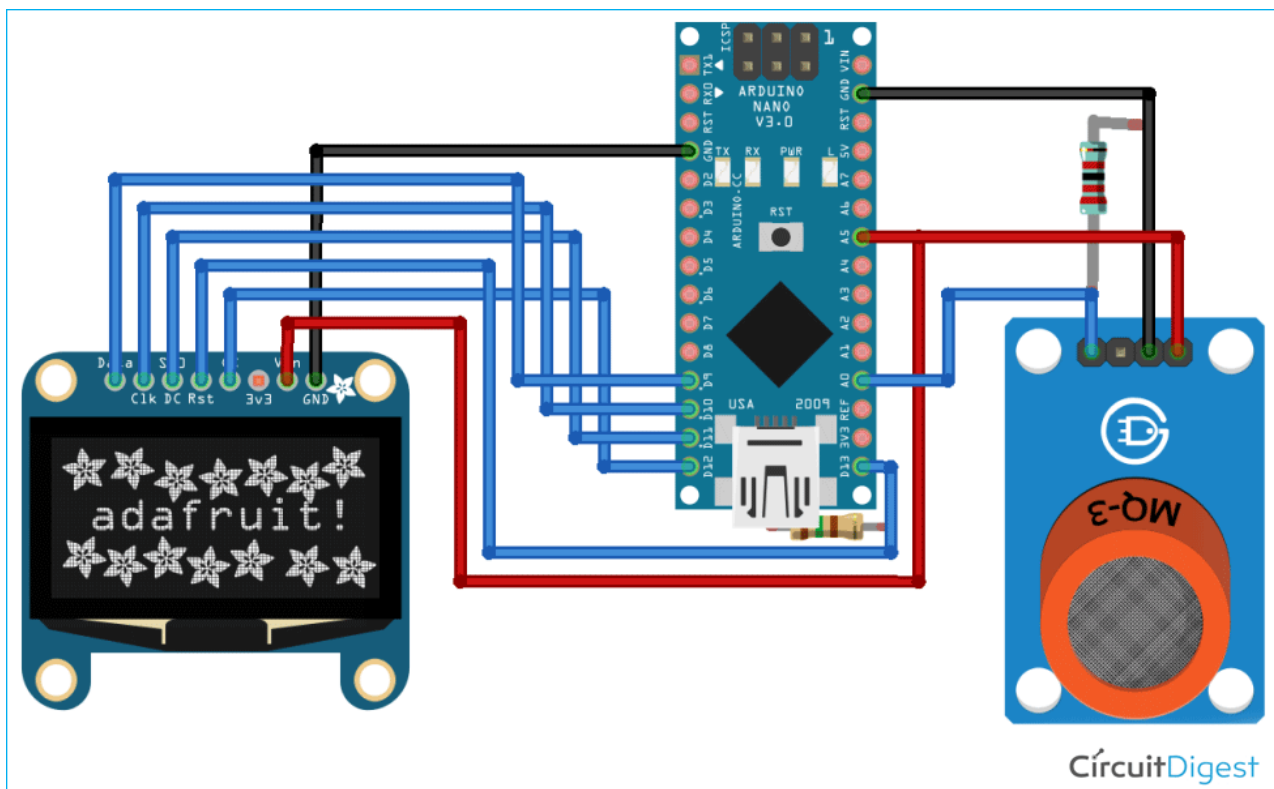
datasheet recommends that you calibrate the detector for 100ppm NH₃ or 50ppm Alcohol concentration in air and use a value of load resistance (R_L) of about 20 K Ω . But if you track your PCB traces to **find the value of your R_L** in the board, you can see a 1K Ω (102) load resistor.



So to measure the appropriate CO₂ concentration values, you have to replace the 1K Ω resistor with a 22K Ω resistor.

Circuit Diagram to Interface MQ135 with Arduino

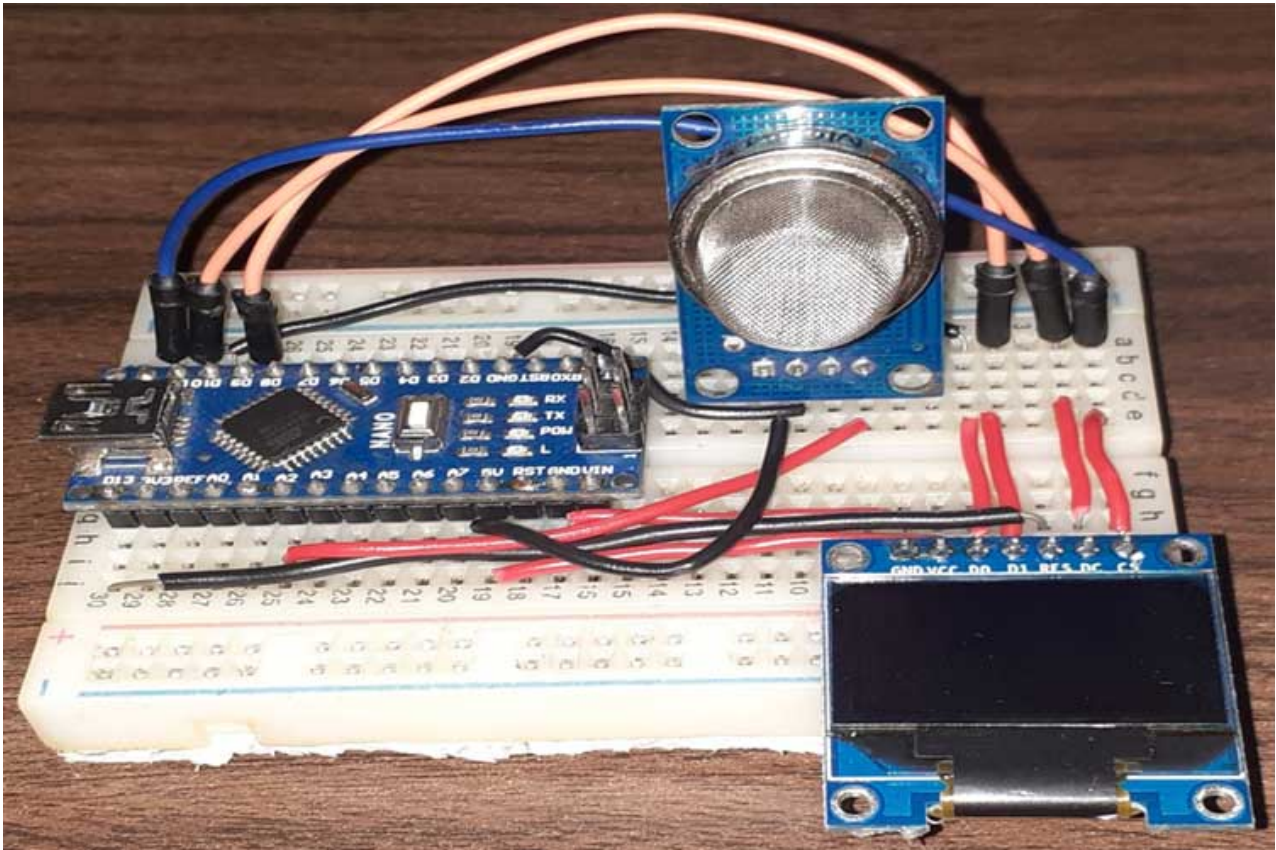
The complete schematics to connect **MQ-135 Gas Sensor with Arduino** is given below:



The circuit is very simple as we are only connecting the MQ-135 Sensor and OLED Display module with Arduino Nano. MQ-135 Gas Sensor and OLED Display module both are powered with +5V and GND. The Analog Out pin of the MQ-135 sensor is connected to the A0 pin of Arduino Nano. Since the OLED Display module uses SPI communication, we have established an SPI communication between the OLED module and Arduino Nano. The connections are shown in the below table:

S.No	OLED Module Pin	Arduino Pin
1	GND	Ground
2	VCC	5V
3	D0	10
4	D1	9
5	RES	13
6	DC	11
7	CS	12

After connecting the hardware according to the circuit diagram, the Arduino MQ135 sensor setup should look something like below:



Calculating the R_0 Value of MQ135 Sensor

Now that we know the value of R_L , let's proceed on how to calculate the R_0 values in clean air. Here we are going to use [MQ135.h](#) to measure the CO₂ concentration in the air. So first download the MQ-135 Library, then preheat the sensor for 24 hours before reading the R_0 values. After the preheating process, use the below code to read the R_0 values:

```
#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);
}
```

Now once you got the R_0 values, Go to **Documents > Arduino > libraries > MQ135-master folder** and open the **MQ135.h** file and change the RLOAD & RZERO values.

```
///The load resistance on the board
#define RLOAD 22.0
///Calibration resistance at atmospheric CO2 level
#define RZERO 5804.99
```

Now scroll down and replace the ATMOCO₂ value with the current Atmospheric CO₂ that is **411.29**

```
///  
//Atmospheric CO2 level for calibration purposes  
#define ATMOC02 397.13
```

Code to Measure CO2 Using Arduino MQ135 Sensor

The complete code for interfacing MQ-135 Sensor with Arduino is given at the end of the document. Here we are explaining some important parts of the **MQ135 Arduino code**.

The code uses the **Adafruit_GFX**, and **Adafruit_SSD1306**, and **MQ135.h** libraries. These libraries can be downloaded from the Library Manager in the Arduino IDE and install it from there. For that, open the Arduino IDE and go to **Sketch < Include Library < Manage Libraries**. Now search for Adafruit GFX and install the **Adafruit GFX library by Adafruit**.

Adafruit DotStarMatrix

by Adafruit

Adafruit_GFX-compatible library for DotStar grids Adafruit_GFX-compatible library for DotStar grids

[More info](#)

Adafruit GFX Library

by Adafruit Version 1.9.0 **INSTALLED**

Adafruit GFX graphics core library, this is the 'core' class that all our other graphics libraries derive from. Install this library in addition to the display library for your hardware.

[More info](#)

Similarly, install the **Adafruit SSD1306 libraries by Adafruit**. MQ135 library can be downloaded from [here](#).

After installing the libraries to Arduino IDE, start the code by including the needed libraries files.

```
#include "MQ135.h"  
#include <SPI.h>  
#include <Adafruit_GFX.h>  
#include <Adafruit_SSD1306.h>
```

Then, define the OLED width and height. In this project, we're using a 128×64 SPI OLED display. You can change the **SCREEN_WIDTH**, and **SCREEN_HEIGHT** variables according to your display.

```
#define SCREEN_WIDTH 128  
#define SCREEN_HEIGHT 64
```

Then define the SPI communication pins where OLED Display is connected.

```
#define OLED_MOSI    9  
#define OLED_CLK    10  
#define OLED_DC      11  
#define OLED_CS      12  
#define OLED_RESET   13
```

Then, create an Adafruit display instance with the width and height defined earlier with the SPI communication protocol.

```
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, OLED_MOSI, OLED_CLK,
OLED_DC, OLED_RESET, OLED_CS);
```

After that, define the Arduino pin where the MQ-135 sensor is connected.

```
int sensorIn = A0;
```

Now inside the **setup()** function, initialize the Serial Monitor at a baud rate of 9600 for debugging purposes. Also, Initialize the OLED display with the **begin()** function.

```
Serial.begin(9600);
display.begin(SSD1306_SWITCHCAPVCC);
display.clearDisplay();
```

Inside the **loop()** function, first read the signal values at the Analog pin of Arduino by calling the **analogRead()** function.

```
val = analogRead(A0);
Serial.print ("raw = ");
```

Then in the next line, call the **gasSensor.getPPM()** to calculate the PPM values. The PPM values are calculated using the Load resistor, R_0 , and reading from the analog pin.

```
float ppm = gasSensor.getPPM();
Serial.print ("ppm: ");
Serial.println (ppm);
```

After that, set the text size and text colour using the **setTextSize()** and **setTextColor()**.

```
display.setTextSize(1);
display.setTextColor(WHITE);
```

Then in the next line, define the position where the text starts using the **setCursor(x,y)** method. And print the CO2 Values on OLED Display using the **display.println()** function.

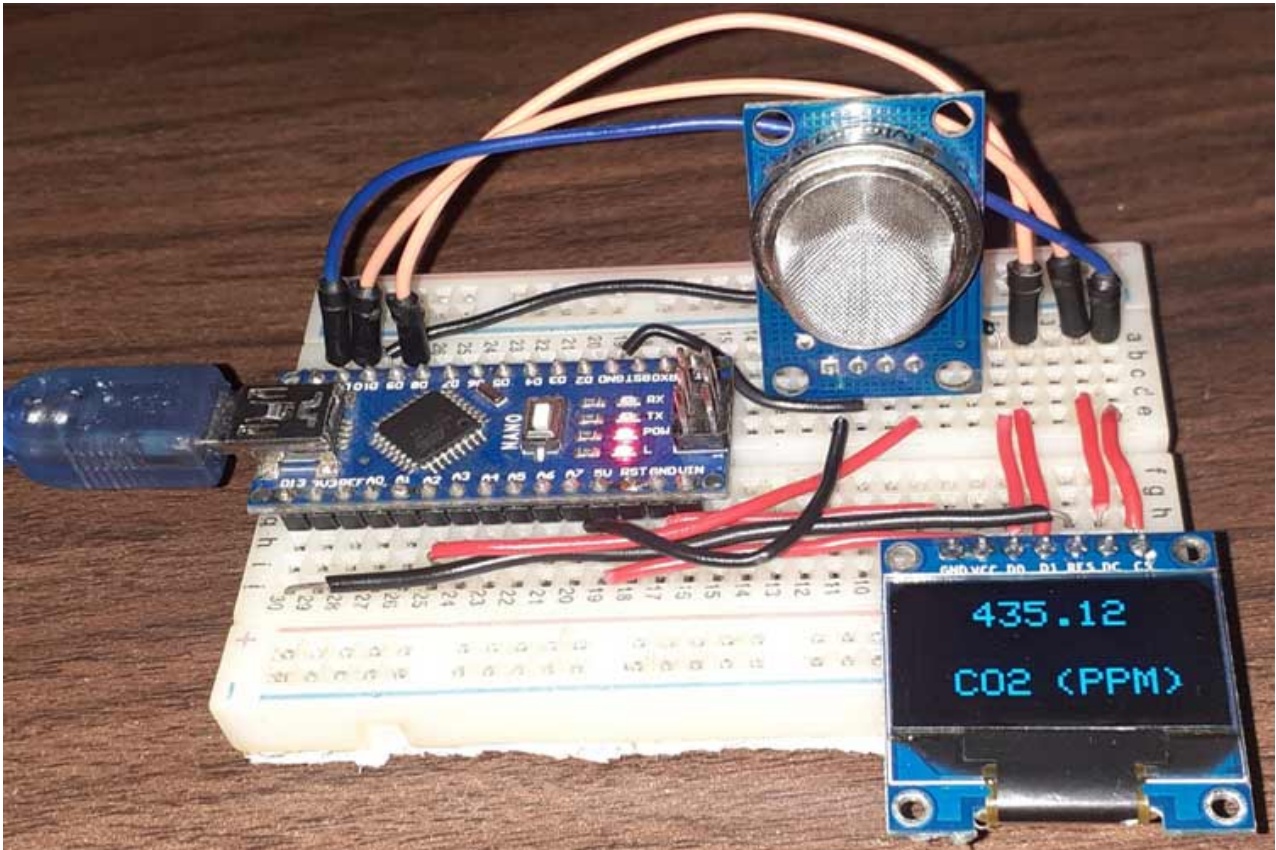
```
display.setCursor(18,43);
display.println("CO2");
display.setCursor(63,43);
display.println("(PPM)");
display.setTextSize(2);
display.setCursor(28,5);
display.println(ppm);
```

And in the last, call the **display()** method to display the text on OLED Display.

```
display.display();
display.clearDisplay();
```

Testing the Interfacing of MQ-135 Sensor with Arduino

Once the hardware and code are ready, it is time to test the sensor. For that, connect the Arduino to the laptop, select the Board and Port, and hit the upload button. Then open your serial monitor and wait for some time (preheat process), then you'll see the final data. The Values will be displayed on OLED display as shown below:



This is how an MQ-135 sensor can be used to measure accurate CO2 in the air. The complete **MQ135 Air Quality Sensor Arduino Code** and working video are given below. If you have any doubts, leave them in the comment section.

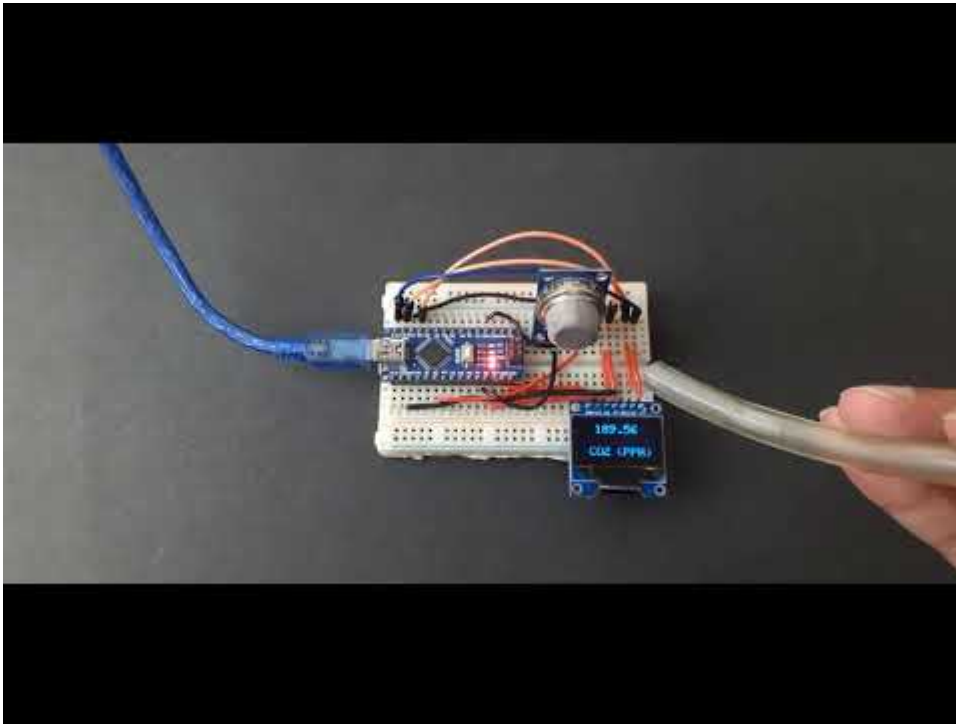
Code

```

/*
 * Interfacing MQ135 Gas Sensor with Arduino
 * Author: Ashish
 * Website: www.circuitdigest.com
 * Date: 11-11-2020
 */
// The load resistance on the board
#define RLOAD 22.0
#include "MQ135.h"
#include <SPI.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#define SCREEN_WIDTH 128 // OLED display width, in pixels
#define SCREEN_HEIGHT 64 // OLED display height, in pixels
// Declaration for SSD1306 display connected using software SPI (default case):
#define OLED_MOSI 9
#define OLED_CLK 10
#define OLED_DC 11
#define OLED_CS 12
#define OLED_RESET 13
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT,
  OLED_MOSI, OLED_CLK, OLED_DC, OLED_RESET, OLED_CS);
MQ135 gasSensor = MQ135(A0);
int val;
int sensorPin = A0;
int sensorValue = 0;
void setup() {
  Serial.begin(9600);
  pinMode(sensorPin, INPUT);
  display.begin(SSD1306_SWITCHCAPVCC);
  display.clearDisplay();
  display.display();
}
void loop() {
  val = analogRead(A0);
  Serial.print ("raw = ");
  Serial.println (val);
  // float zero = gasSensor.getRZero();
  // Serial.print ("rzero: ");
  //Serial.println (zero);
  float ppm = gasSensor.getPPM();
  Serial.print ("ppm: ");
  Serial.println (ppm);
  display.setTextSize(2);
  display.setTextColor(WHITE);
  display.setCursor(18,43);
  display.println("CO2");
  display.setCursor(63,43);
  display.println("(PPM)");
  display.setTextSize(2);
  display.setCursor(28,5);
  display.println(ppm);
  display.display();
  display.clearDisplay();
  delay(2000);
}

```

Video



Watch Video At: <https://youtu.be/PQZ8cj2m2FY>