LPG Sensor

Introduction

In this tutorial, we are using the MQ5 Gas sensor module (which is widely available in market). This module has two output possibilities – an analog out (A0) and a digital out (D0). The analog out can be used to detect Gas leakage and to measure volume of Gas leakage (by doing proper calculation of the sensor output inside program) in specific units (like parts per million). The digital out can be used to detect Gas leakage and hence trigger an alert system--an alarm.

There are many types of sensors in the MQ series range. Listed here are the different sensors

- MQ-2 - Methane, Butane, LPG, smoke
- MQ-3 - Alcohol, Ethanol, smoke
- MQ-4 - Methane, CNG Gas
- MQ-5 - Natural gas, LPG
- MQ-6 - LPG, butane gas
- MQ-7 - Carbon Monoxide
- MQ-8 - Hydrogen Gas
- MQ-9 - Carbon Monoxide, flammable gasses
- MQ131 - Ozone
- MQ135 - Air Quality (CO, Ammonia, Benzene, Alcohol, smoke)
- MQ136 - Hydrogen Sulfide gas
- MQ137 - Ammonia
- MQ138 - Benzene, Toluene, Alcohol, Acetone, Propane, Formaldehyde gas, Hydrogen
- MQ214 - Methane, Natural gas
- MQ216 - Natural gas, Coal gas

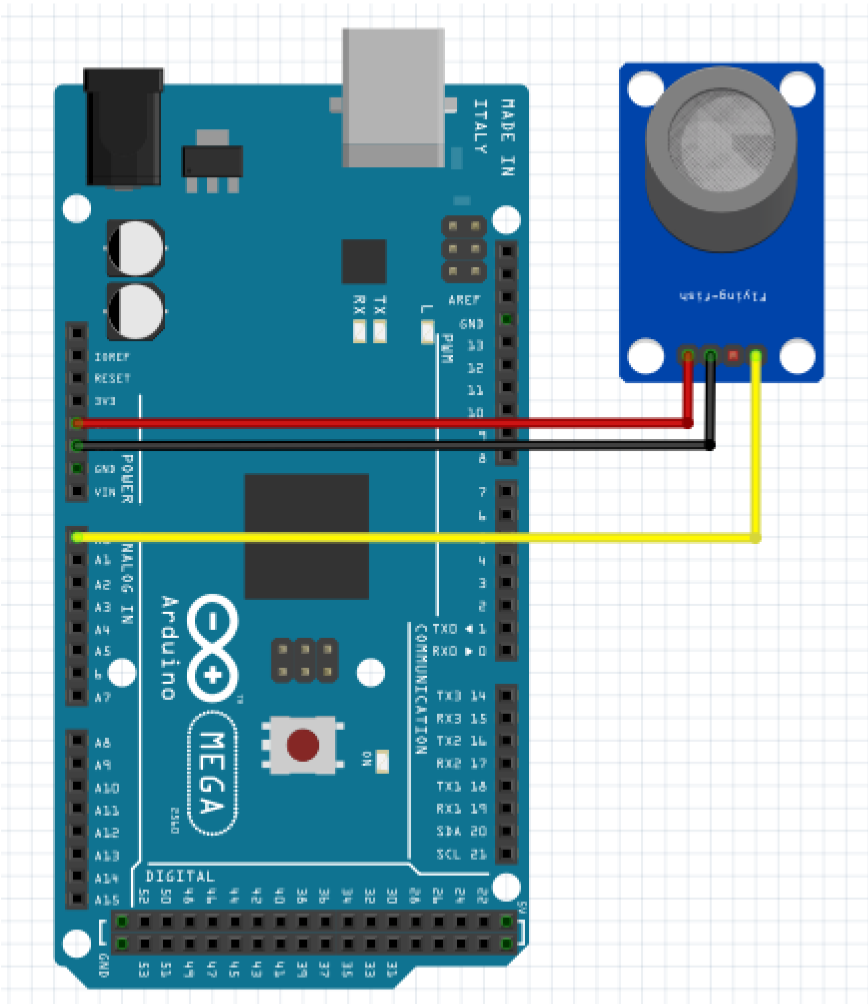
Interfacing with Arduino

The sensor is equipped with a Nickel-Chromium alloy heater that creates an ideal condition for the gas sensing layer made up of Tin-Dioxide. It is also important to have the sensor calibrated to get an accurate reading.

It is important to have the sensor pre-heated for about 24 hours to get an accurate reading. You can preheat the sensor simply by just applying power to it (5 volts).

Like the other sensors in your kit, this one also has a digital output – which outputs LOW when the trigger point is met and HIGH otherwise, an analog output and a trimmer resistor to adjust the trigger level of the sensor.

Lets see how the output of the sensor looks like. Wire your Arduino as follows



We connect power and ground. A0 to analog pin 0 and then we load the following program and run serial plotter

```
float voltage;
int adc;

void setup()
```

```
{
   Serial.begin(9600);
}

void loop()
{
   adc =  analogRead(0);
   voltage = adc * (5.0/1023.0);
   Serial.println(voltage);
   delay(200);
}
```

The program is straightforward and just throws the analog reading on Analog input 0.

So right now, the sensor is on free and clean air. Note its output. Now see when we expose the sensor to butane (light up a lighter, blow or just push the light lever, not starting the flame) and note that the output changes.

But how do we make sense of the data from the sensor? How can we put units to it? Shown here is the graph of how the sensor reacts to different concentrations of gas. Note that the sensor will not be able to detect the type of gas it is sensing.

Fig.2 sensitivity characteristics of the MQ-5

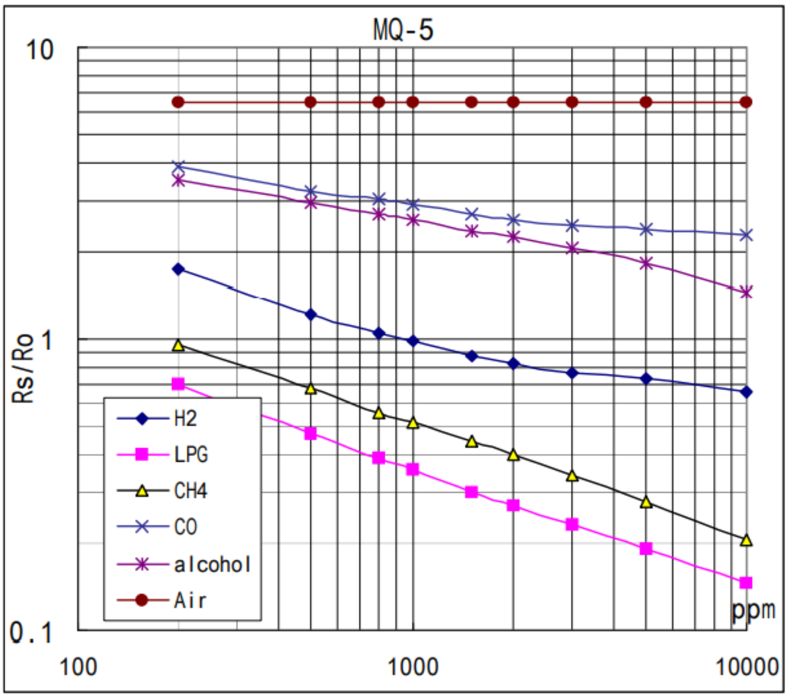


Fig.3 is shows the typical sensitivity characteristics of the MQ-5 for several gases.
in their: Temp: 20℃、
Humidity: 65%、
O_2 concentration 21%
RL=20k Ω
Ro: sensor resistance at 1000ppm of H_2 in the clean air.
Rs:sensor resistance at various concentrations of gases.

The output is determined by Rs/Ro. As the sensor react by changing its resistance to various concentrations of gas, Ro is the resistance in clean air and Rs is the resistance when reacting to different concentration of gasses. Also notice that the graph is in LOGARITHMIC scale. Meaning each line is per hundreds or per thousands.

Lets try to determine the value of Ro, using the same wiring as in the demo earlier (Gas_Calibration sketch).

```
void setup() {
```

```
  Serial.begin(9600);
}

void loop() {
  float sensor_volt;
  float RS_air;
  float R0;
  float sensor_value=0;

  Serial.println("Reading average");
  // measure 100 times, get the average
  for(int x = 0 ; x < 100 ; x++)
  {
    sensor_value = sensor_value + analogRead(A0);
    delay(10);
  }
  sensor_value = sensor_value/100.0; //averagev

  sensor_volt = sensor_value/(1024*5.0);
  RS_air = ((5.0-sensor_volt)/sensor_volt)*1000; // RL is 1k based on the module
  R0 = RS_air/6.5; // The ratio of RS/R0 is 6.5 in a clear air from the datasheet/graph shown

  Serial.print("sensor_value = ");
  Serial.print(sensor_value);

  Serial.print(" sensor_volt = ");
  Serial.print(sensor_volt);

  Serial.print(" R0 = ");
  Serial.println(R0);
  delay(1000);
}t
```

The value of R0 will vary from sensor to sensor. This is due to manufacturing tolerances etc. Therefore, it is important to always calibrate your sensor when using. R0 may vary very slightly.

Now, lets try getting the Rs (RS-RO Ratio sketch).

```
#define R0 1420.00 //this is the value you got from the earlier sketch

void setup() {
  Serial.begin(9600);
}

void loop() {

  float sensor_volt;
  float RS_gas; // Get value of RS in a GAS
  float ratio; // Get ratio RS_GAS/RS_air
  float sensor_value;

  sensor_value = analogRead(A0);

  sensor_volt=(float)sensor_value*(5.0/1023);
  RS_gas = ((5.0-sensor_volt)/sensor_volt)*1000; // RL is 1k from the circuit module
  ratio = RS_gas/R0;  // ratio = RS/R0

  Serial.print(" sensor_value = ");
  Serial.print(sensor_value);
  Serial.print(" sensor_volt = ");
  Serial.print(sensor_volt);
```

```
    Serial.print("  RS_gas = ");
    Serial.print(RS_gas);
    Serial.print("  RS/R0 = ");
    Serial.println(ratio);

    delay(1000);

}
```

The algorithm for getting RS is the same as earlier. However, as we already know R0, we could get the RS/R0 ratio. So go ahead and plugin the R0 value you got earlier.

Verify that as expected with clean air, we are getting a value of 6.5

Now try exposing the sensor to butane or lighter fumes and notice that the output changes. It is also important to note that it may take some a minute or two for the sensor to go back to the steady-state clean air value

Finally, you can also use the digital output of the sensor to trigger whenever a threshold gas level is met.

Helpful Links
https://wiki.seeedstudio.com/Grove-Gas_Sensor-MQ5/
https://www.circuitstoday.com/interfacing-mq5-lpg-sensor-to-arduino
https://biosci.mcdb.ucsb.edu/biochemistry/tw-lig/logarithms/logarithmic-axes.htm
https://blog.datawrapper.de/weeklychart-logscale/