MQ 2 Gas Sensor

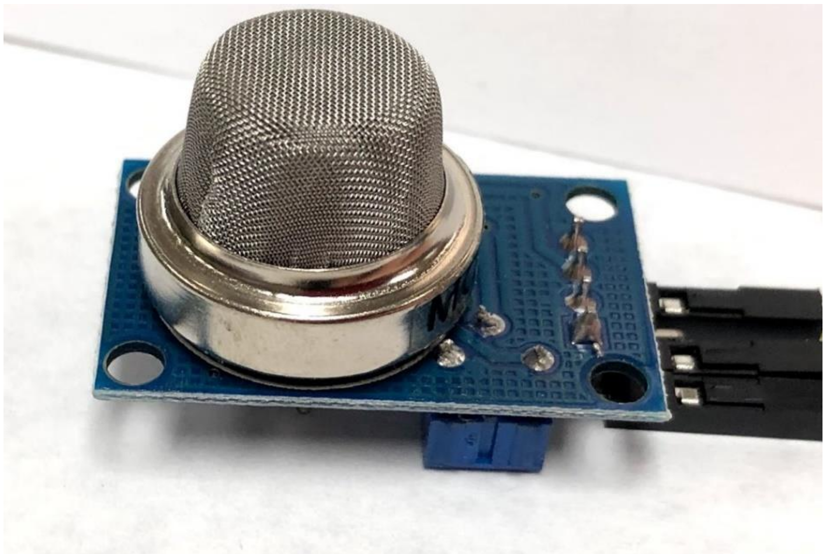

Figure 1: MQ-2 Gas Sensor

The MQ 2 gas sensor is used to detect or monitor the concentration and/or presence of combustible gases in the air. It features a simple drive circuit, stable, long life, fast response, and a wide scope.[1] Due to its high sensitivity to hydrogen, LPG (liquid petroleum gas), methane, carbon monoxide, alcohol, smoke, and propane; this sensor is traditionally used to help detect gas leaks in many family and industrial practices. This walkthrough will describe how the MQ 2 sensor functions and how to set up the sensor for use on a Raspberry Pi.

How does the MQ 2 work?

	Parts	Materials
1	Gas sensing layer	SnO_2
2	Electrode	Au
3	Electrode line	Pt
4	Heater coil	Ni-Cr alloy
5	Tubular ceramic	Al_2O_3
6	Anti-explosion network	Stainless steel gauze (SUS316 100-mesh)
7	Clamp ring	Copper plating Ni
8	Resin base	Bakelite
9	Tube Pin	Copper plating Ni

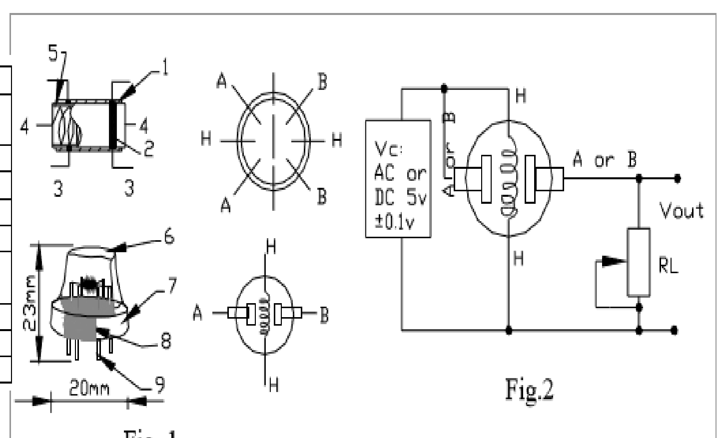

Figure 2: Schematic of MQ-2 Gas Sensor from Datasheet

The MQ 2 sensor is a metal oxide semiconductor (MOS), also known as a chemiresistor.[2] MOS sensors measure the change in resistance when gases are present. This type of sensor requires the gas to encounter the sensor for a chemical reaction to occur. This chemical reaction causes a change in resistance, which is then measured to detect the presence or concentration of the gas. However, the MQ 2 sensor is not specific towards one gas. Therefore, a calculated concentration of gas may not accurately represent the mixture in the air.

The mechanism to sensing the change in resistance lies within the structure of the sensor. The MQ 2 sensor contains an anti-explosion network. This is particularly important due to the heating element inside the sensor and its potential contact with a flammable gas. This network is composed of two layers of stainless-steel mesh. This mesh also functions to filter our particulates from contaminating and dirtying the sensing mechanism.[3] The mesh is mounted to a bakelite base. Bakelite is a thermosetting plastic made from phenol and formaldehyde.[4] It does not change shape and will not melt under heat, therefore, providing a very stable base for the sensor to rest.

Figure 3: Anti-Explosion Mesh on MQ-2 gas sensor

The sensing mechanism is under the anti-explosion mesh and is connected to six legs. It is constructed of aluminum oxide (Al_2O_3) based ceramic and coated with tin dioxide (SnO_2) – acts as the gas sensing layer - to form a ceramic tube. Inside this ceramic tube is a nickel-chromium heating coil (Ni-Cr). The Ni-Cr coil is connected to two of the six legs surrounding the tube. The other four legs, composed of platinum wires, send the small changing currents as output signals from the sensor. In summary, the Ni-Cr coil and the Al_2O_3 form a heating system to make sure the sensor is at working temperature and the platinum wires and SnO_2 form the sensing system. When the gas interacts with the sensor, the heat from the system ionizes the gas. This ionization allows SnO_2 to absorb the gas. This adsorption causes a change in the resistance on the sensor. The output legs then send this resistance as an output signal to the microcontroller.[3,5]

1. **Why is the MQ-2 sensor sensitive to LPG, methane, carbon monoxide, alcohol, smoke, and propane?**

2. **How does the sensor read in the presence of gas?**

3. **Can the MQ-2 sensor tell the difference between LPG, methane, carbon monoxide, alcohol, smoke, and propane?**

Connecting the MQ 2

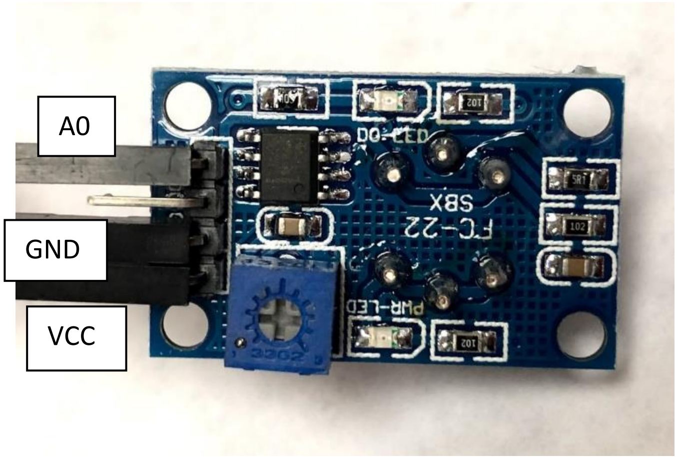

Figure 4: Connections used to connect MQ-2 to Breadboard

The MQ 2 is mounted on a breakout board to allow for easy connection to a breadboard and microcontroller. The breakout board has a sensitivity adjustor that allows for calibration based on a potentiometric threshold on the digital pin (D0), this is mainly used for detection limit purposes, such as a fire. This set up will utilize the analog pin only for this setup, it supplies the output voltage in proportion to the concentration of gas present. The breakout board also contains a pin for the VCC that supplies a 5V output from the Raspberry Pi (microcontroller) and a GND that grounds the sensor on the Raspberry Pi.

In order to connect the MQ 2 sensor to the Raspberry Pi, two other sensors are needed. A level converter is needed to take the 5V output signal to a 3.3V signal and an A/D converter to be able to read the signal in digital form. It is necessary to step the voltage down because a raspberry pi can only read signals at 3.3V without potential for damage. The A/D converter changes the analog signal to a digital signal in or order to calculate the concentration of gas from the signal instead of just a detection.

Figure 6: MCP3008 A/D converter

Figure 5: TXB0108 Level Converter

Connect the sensors as follows to the Raspberry Pi breadboard.

MCP3008

Pin 16 (V_{DD}) connects to your **positive rail**
Pin 15 (V_{REF}) connects to your **positive rail**
Pin 14 (AGND) connects to your **ground rail**
Pin 13 (CLK) connects to **SCLK** on your Pi cobbler
Pin 12(D_{OUT}) connects to **MISO** on your Pi cobbler
Pin 11(D_{IN}) connects to **MOSI** on your Pi cobbler
Pin 10 (CS) connects to **CE0** on your Pi cobbler
Pin 9(D_{GND}) connects to your **ground rail.**

CYT1076 Red level converter

LV to Pi **3.3V Positive Rail**
HV to Pi **5V Positive Rail**
HV side GND to **GND on Pi**
LV side GND to **GND on Pi**

MQ-2

VCC to Pi **5V Positive Rail**
GND to **GND on Pi**
A0 to **CYT1076 HV1**
CYT1076 LV1 to **A0 on MCP3008**

10K resistor from **A0** to **GND**

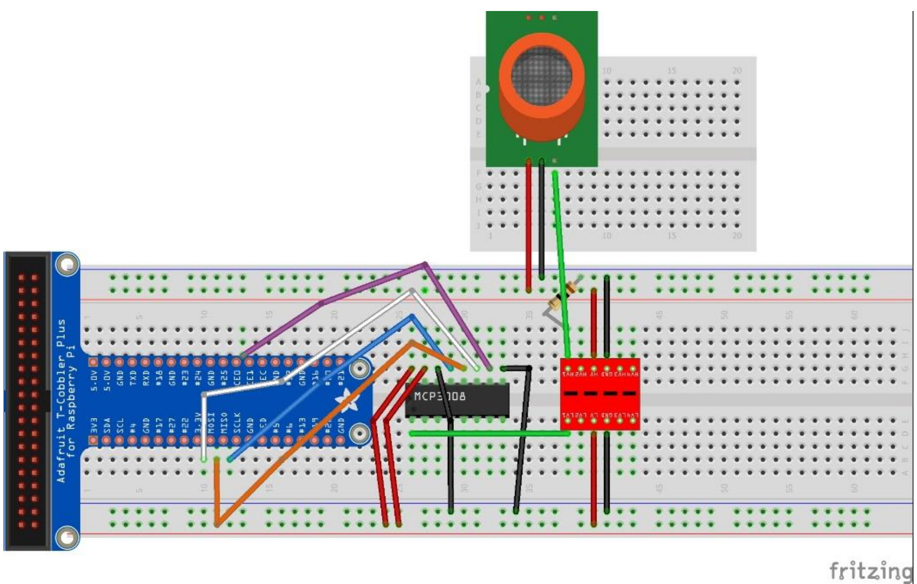

Figure 7: Schematic of MQ-2 Hookup

4. **How would you connect multiple MQ-2 sensors to the breadboard?**

5. **How many data streams can be connected on the MCP3008 and CYT1076 sensors?**

Coding the MQ-2

Coding for the MQ-2 sensor depends entirely on how it's going to be used. To use it for detection only requires a very simple code that will not be covered here, but can be found at *Last Minute Engineers* for the Arduino.[3] This walk-through will cover how to code the MQ-2 to calculate ppm concentrations of gas using a fresh air calibration.

The concentration (ppm) of gases is calculated based of the resistance ratio (RS/R0). RS is the measured change in resistance when the sensing mechanism detects gas and R0 is the stable sensor resistance in fresh air or no gas presence. Using Ohm's law and the sensor schematic, $RS = \frac{VC-RL}{Vout} - RL$. VC is the voltage current (in this case 5V from the pi), Vout is the output voltage (measured analog/digital value), and RL is the load resistance (this set up is at 10K). R0 can then be calculated using this equation, R0 = RS/Fresh air ratio value from datasheet.

In order to convert the digital signal to concentration units, the datasheet chart is used again. A simple calibration line has y = mx + b, however, the MQ-2 gas sensor is not linear. It follows a log-log scale so a bit more calculation is needed. So the y = mx + b equation can be converted to log(y) = m*log(x) + b. Now using the chart, the slope and intercept can be calculated, where $m = \frac{log\left({}^{y}/_{y0}\right)}{log\left({}^{x}/_{x0}\right)}$ and $b = \log(y) - m * \log(x)$. Once these values are obtained, the concentration of gas can now be calculated as $x(ppm) = 10^{[\log(y)-b]/m}$. (Remember that y is equal to RS/R0.

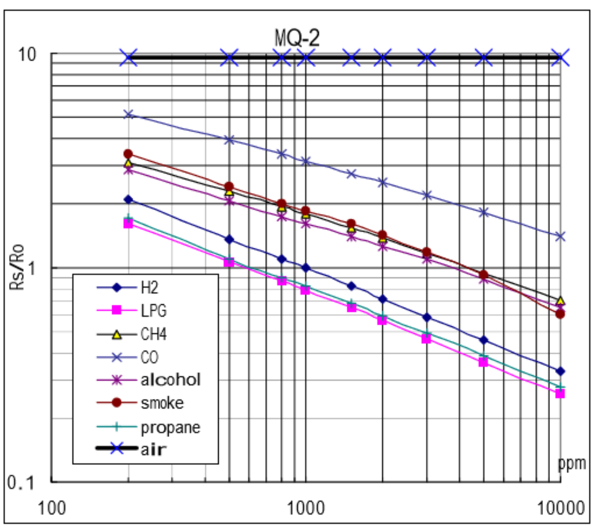

Figure 8: Sensitivity Diagram for MQ-2

Example calculation for LPG:

At 1000ppm the values are (1000,0.8) and at 10000 the values are (10000, 0.27). So

$$m = \frac{log\left({}^{0.27}/_{0.8}\right)}{log\left({}^{10000}/_{1000}\right)} = 0.47$$

And

$$b = \log(0.8) + 0.47 * \log(1000) = 1.31.$$

6. **Calculate the slope and y-intercept for two of the other gases in the sensitivity diagram.**

7. **Calculate the concentration of those gases if the RS/R0 value is equal to 1, 6, and 10.**

Once all the parameters have been calculated, coding for the sensor can begin. The first part will be to code for the fresh air calibration to obtain the RS and R0 value in clean air. The second part of the code will detect the presence of gas and output a ppm concentration reading. Coding is as follows in the Thonny IDE:

```
import time
import math
from gpiozero import MCP3008
```

These are the libraries needed to run the MCP3008 and to calculate the ppm concentrations of the gas. (These libraries should already be installed on Raspberry Pi when purchased).

```
#location of analog signal from MQ2 sensor
adc0 = MCP3008(channel=0)
```

In order to obtain the digital output values, the correct location must be set in the code. The MQ-2 sensor was connected to the first channel on the MCP3008, which is set up as channel 0.

```
sensorValue=0
x=0
#loops analog signal to get average of value for air calibration
for x in range(0, 500):
    sensorValue = sensorValue + adc1.value
    #loops 500 times
    x=x+1
```

The sensorValue is the digital signal value coming from the MQ-2 sensor readings. In order to calibrate in fresh air and average of 500 readings is taken. The initial value starts at 0 to allow for addition of each reading to one another.

```
#Gets average of digital value
sensorValue1 = sensorValue/500
#calculates the sensing resitance in "clean air"
#3.3 volts and 10 RL taken from datasheet
RSair = ((3.3*10)/sensorvolt)-10
#Calc sensor restistance in clean air from RS using air
#value
R0 =RSair/9.9
```

This part of the code averages the readings and converts to the RS value in air. From this voltage we can calculate the R0 in fresh air.

```
while True:
    #slope and intercept values calculated from calibration
    #on data sheet y=mx+b === log(y) = mlog(x) + b
    LPGm = -0.47
    LPGb = 1.31

    #digital value from detection
    sensorValue = adc1.value
    #RS calc from detection
    RSgas = ((3.3*10)/sensorvolt)-10
    #ratio from detection
    ratio1 = (RSgas/R0)
    #log of ration to calc ppm
    ratio = math.log10(ratio1)
```

Using the R0 value calculated from above, this part of the code measures in real time for gas. Once the ratio is obtained, the concentration of gas can be calculated.

```
    LPGratio = (ratio-LPGb)/LPGm
    LPGppm = math.pow(10,LPGratio)
    LPGperc = LPGppm/10000

    print(LPGperc, "LPG%")
    print(LPGppm, "LPGppm")
    #loops every 5 seconds
    time.sleep(5)
```

To see the values, simply use a print statement. The sensor can cycle as many times as desired.

Homework:

Create a code to allow three MQ-2 sensors to run simultaneously and have them calibrated for three different gases. Have the sensors take a measurement every 10 minutes for 6 hours and save the ppm results to a spreadsheet. Test your sensors with a lighter to see if there is an increase in any of the gases.

Write a code to then graph the results.

Turn in code and data plot.

The values obtained from this calibration are approximate and are not meant to be exact values.

References

(1) Arduino Playground - MQGasSensors https://playground.arduino.cc/Main/MQGasSensors (accessed Feb 7, 2019).
(2) Chemiresistor. *Wikipedia*; 2019.
(3) lmewp1908. In-Depth: How MQ2 Gas/Smoke Sensor Works? & Interface It with Arduino. *Last Minute Engineers*, 2018.
(4) Is it Bakelite? https://www.realorrepro.com/article/Is-it-Bakelite (accessed Feb 6, 2019).
(5) EngineersGarage. Insight - Learn the Working of a Gas Sensor https://www.engineersgarage.com/insight/how-gas-sensor-works (accessed Feb 6, 2019).