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. 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?**

<table>
<thead>
<tr>
<th>Parts</th>
<th>Materials</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Gas sensing layer</td>
<td>SnO₂</td>
</tr>
<tr>
<td>2 Electrode</td>
<td>Au</td>
</tr>
<tr>
<td>3 Electrode line</td>
<td>Pt</td>
</tr>
<tr>
<td>4 Heater coil</td>
<td>Ni-Cr alloy</td>
</tr>
<tr>
<td>5 Tubular ceramic</td>
<td>Al₂O₃</td>
</tr>
<tr>
<td>6 Anti-explosion network</td>
<td>Stainless steel gauze (304SS mesh)</td>
</tr>
<tr>
<td>7 Clamp ring</td>
<td>Copper plating Ni</td>
</tr>
<tr>
<td>8 Resin base</td>
<td>Boricite</td>
</tr>
<tr>
<td>9 Tube Pin</td>
<td>Copper plating Ni</td>
</tr>
</tbody>
</table>

**Figure 1: MQ-2 Gas Sensor**

**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. 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. The mesh is mounted to a bakelite base. Bakelite is a thermosetting plastic made from phenol and formaldehyde. It does not change shape and will not melt under heat, therefore, providing a very stable base for the sensor to rest.

The sensing mechanism is under the anti-explosion mesh and is connected to six legs. It is constructed of aluminum oxide (Al₂O₃) based ceramic and coated with tin dioxide (SnO₂) – 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 Al₂O₃ form a heating system to make sure the sensor is at working temperature and the platinum wires and SnO₂ form the sensing system.

When the gas interacts with the sensor, the heat from the system ionizes the gas. This ionization allows SnO₂ 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.

**Connecting the MQ 2**

---

---
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 order to calculate the concentration of gas from the signal instead of just a detection.

![MCP3008 A/D converter](image1)

**Figure 5: MCP3008 A/D converter**

![CYT1076 Level Converter](image2)

**Figure 6: CYT1076 Level Converter**

Connect the sensors as follows to the Raspberry Pi breadboard.

**MCP3008**

- **Pin 16 (VDD)** connects to your **positive rail**
- **Pin 15 (VREF)** 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 (DOUT)** connects to MISO on your Pi cobbler
- **Pin 11 (DIN)** connects to MOSI on your Pi cobbler
- **Pin 10 (CS)** connects to CE0 on your Pi cobbler
- **Pin 9 (DGND)** 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

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. 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 on 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, $RS = \frac{V_C - RL}{V_{out}} - RL$. 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
R0 can then be calculated using this equation, $R_0 = \frac{RS}{\text{Fresh air ratio value from datasheet}}$.

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 $m = \frac{\log(\frac{y}{y_0})}{\log(\frac{x}{x_0})}$ and $b = \log(y) - m\log(x)$. Once these values are obtained, the concentration of gas can now be calculated as $x(\text{ppm}) = 10^{\frac{\log(y) - b}{m}}$.

**Example calculation for LPG:**

At 1000ppm the values are (1000,0.8) and at 10000 the values are (10000, 0.27). So $m = \frac{\log(0.27/0.8)}{\log(10000/1000)} = 0.47$.

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

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:

```python
import time
import math
fromgpiozero import MCP3008

adc0 = MCP3008(channel=0)
```

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).

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.

*Figure 8: Sensitivity Diagram for MQ-2*
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 resistance in "clean air"
# 3.3 volts and 10 kΩ taken from datasheet
RSair = ((3.3*10)/sensorvolt)-10
# Calc sensor resistance 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 datasheet 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 ratio 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(LPGr, "LPGr")
print(LPGrp, "LPGrp")
# 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.

References