

# Interfacing Gas Sensor (MQ2) With AVR

After basic peripheral interfaces with AVR, it's time to interface some sensors with it. In this tutorial we will interface Gas sensor MQ2 with Atmega 32 breakout. We will require a [Smoke Sensor \(MQ2\) Board](#).

## Contents

- [1 Basics](#)
- [2 Hook Up](#)
- [3 Code](#)
- [4 Output Demo](#)
- [5 Downloads](#)

## Basics

The gas sensor MQ2 suitable for detecting of LPG, i-butane, propane, methane, alcohol, Hydrogen, smoke etc. Since It is highly sensitive and gives fast response, we can take measurements as soon as possible. This sensor can be used for gas leakage detection.

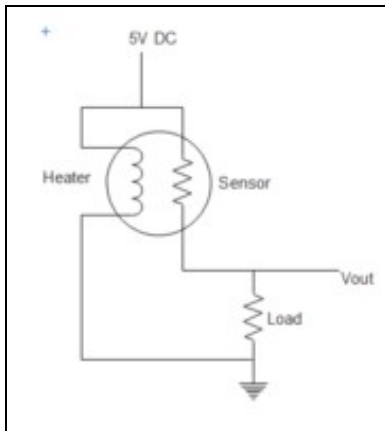


fig 1: MQ-2 sensor schematic

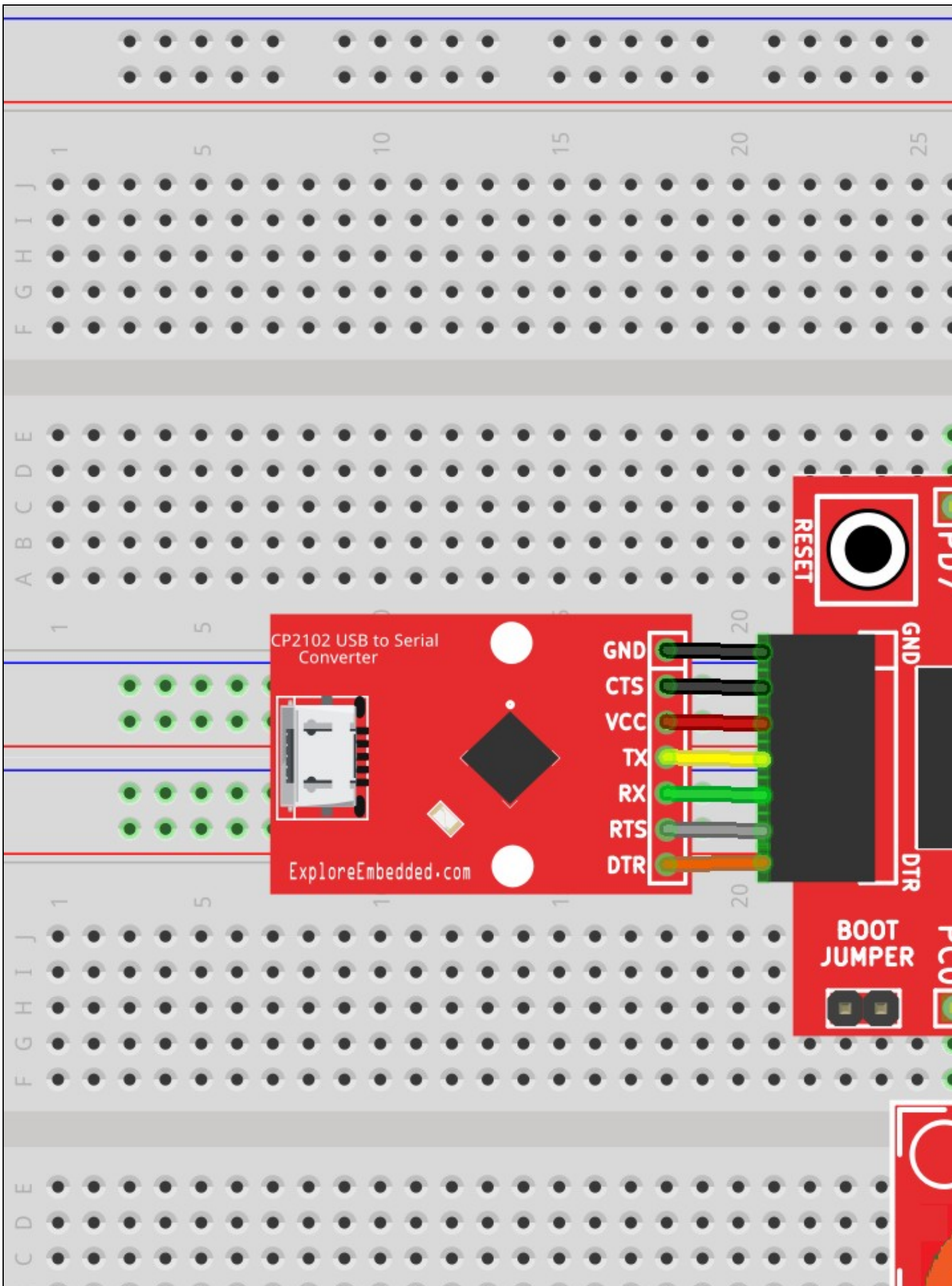
At normal condition, sensor resistor will be high so voltage drop across the load will be low and it will be a constant. If sensor senses flammable gases, resistance of sensor will drop. That means more current will flow from load resistor. So the voltage across it increases. This output voltage increases with increase in concentration of gas in air. The sensitivity of the gas sensor can be adjusted using potentiometer.

Refer [MQ-2 datasheet](#) for detailed information.

This [Smoke Sensor \(MQ2\) Board](#) has analog as well as digital output. For this tutorial we will use analog output. Analog output pin needs to be connect ADC channel 0 of atmega 32 breakout as shown in hook up.

## Hook Up

- As MQ-2 sensor has heater inside, it is prefer to give power to the sensor from separate source.
- For stable operation sensor requires around 24 hour preheating.
- We can use Ultra AVR Developement Board, Starter AVR or Atmega32 Breakout.



# Code

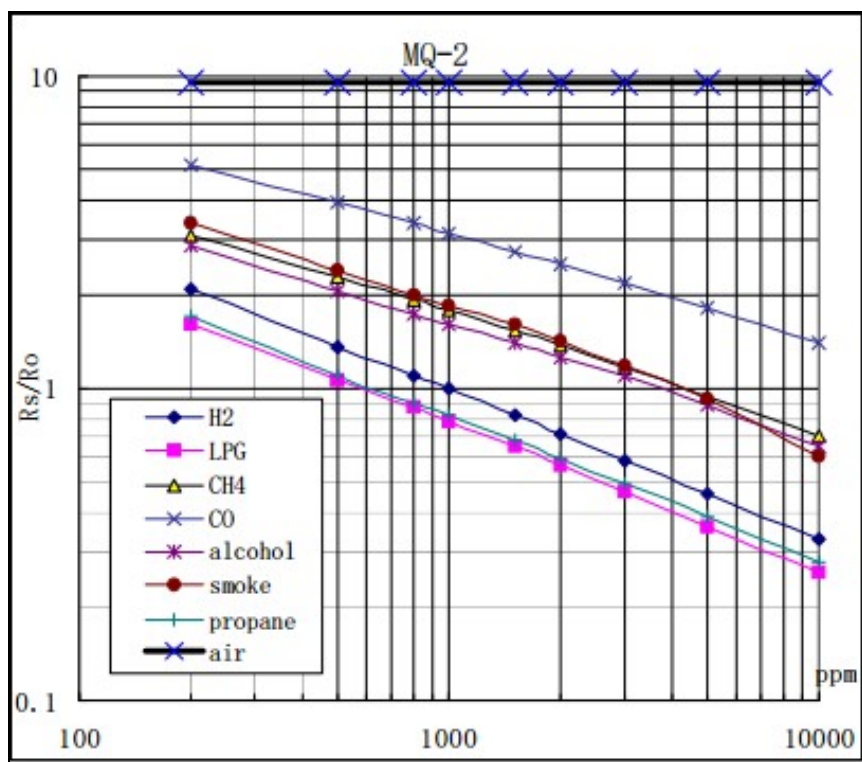
Resistance value of MQ-2 is difference to various kinds and various concentration gases. So, When using this components, sensitivity adjustment is very necessary.

The sensor resistor  $R_S$  and load resistor  $R_L$  form a voltage divider. Based on the chart provided in the [MQ-2 datasheet](#),  $R_S$  in clean air under given temperature and humidity is a constant. The ratio of  $R_S/R_0$  in clean air is 9.8 as described in datasheet. We will first calibrate the sensor. place sensor in clean air. We will get  $R_0$  value by dividing it by  $R_S/R_0$  value in clean air.

Now once  $R_0$  is derived, the targeted gas can be sensed using  $R_S/R_0$  ratio as shown in below program.

- Initially place the sensor in clean air and reset the controller. It will calibrate the sensor and give the message as calibration done.
- Then place the sensor in smoke affected area, it will detect the smoke and will output a gas concentration in ppm and also sensor LED will glow.
- For calculating the concentration of gas in ppm take two points from the curve of particular gas from the graph shown below. Then calculate a slope of that line.

For eg. Take two points from LPG curve point 1 is (log 200, log 1.6) and point 2 is (log 1000, log 0.26). These points are in logarithmic scale so take log of points and calculate the slope using formula:  $\text{Slope} = (y_2 - y_1) / (x_2 - x_1)$  One point from above and slope is used in format (x,y, slope) in a program.

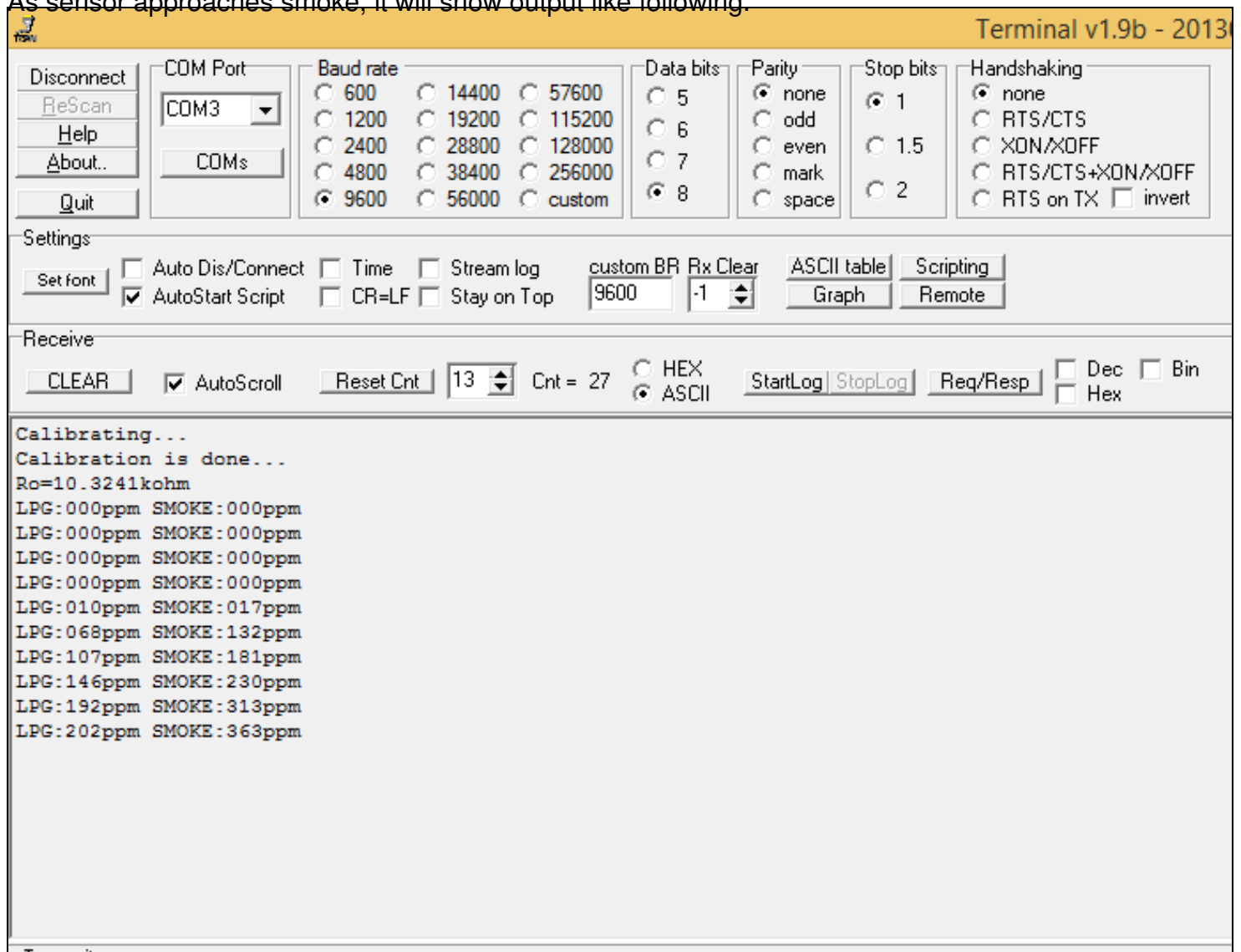


- In a graph the x-axis is  $R_S/R_0$  ratio and y-axis is concentration of gas in ppm.

We have to find (Gas concentration,  $R_S/R_0$  ratio) point. In clean air we will get a  $R_S/R_0$  ratio. Now using above slope,  $R_S/R_0$  ratio and first point mentioned above can be used for calculation of concentration of gas using below formula. As it is logarithmic coordinate, power 10 is taken to convert it to non logarithmic value.  $x_2 = (((y_2 - y_1) / \text{slope}) + x_1)$  Gas concentration =  $10^{\{((\log(R_S/R_0) - (y_1)) / \text{slope}) + x_1\}}$  ppm

# Output Demo

As sensor approaches smoke, it will show output like following.



The screenshot shows the 'Terminal v1.9b - 2013' application window. The interface includes a menu bar with 'Disconnect', 'ReScan', 'Help', 'About..', and 'Quit'. Below the menu is a 'Settings' section with various configuration options: 'COM Port' (set to COM3), 'Baud rate' (set to 9600), 'Data bits' (set to 8), 'Parity' (set to none), 'Stop bits' (set to 1), and 'Handshaking' (set to none). There are also checkboxes for 'Auto Dis/Connect', 'Time', 'Stream log', 'custom BR', 'Rx Clear', 'ASCII table', 'Scripting', 'AutoStart Script', 'CR=LF', and 'Stay on Top'. The 'Receive' section has a 'CLEAR' button, 'AutoScroll' checkbox, 'Reset Cnt' button, a 'Cnt' value of 27, and radio buttons for 'HEX' and 'ASCII'. There are also buttons for 'StartLog', 'StopLog', 'Req/Resp', and checkboxes for 'Dec' and 'Bin'. The main terminal area displays the following output:

```
Calibrating...
Calibration is done...
Ro=10.3241kohm
LPG:000ppm SMOKE:000ppm
LPG:000ppm SMOKE:000ppm
LPG:000ppm SMOKE:000ppm
LPG:000ppm SMOKE:000ppm
LPG:010ppm SMOKE:017ppm
LPG:068ppm SMOKE:132ppm
LPG:107ppm SMOKE:181ppm
LPG:146ppm SMOKE:230ppm
LPG:192ppm SMOKE:313ppm
LPG:202ppm SMOKE:363ppm
```

## Downloads

Download the complete project folder from the below link:

<https://github.com/ExploreEmbedded/AVR-MCU-Breakout-Board/archive/master.zip>

Have a opinion, suggestion , question or feedback about the article let it out here!

Please enable JavaScript to view the [comments powered by Disqus](#).