

this project, we will show what is MQ-7 Sensor and how to use it with the Arduino board.



### Preparations

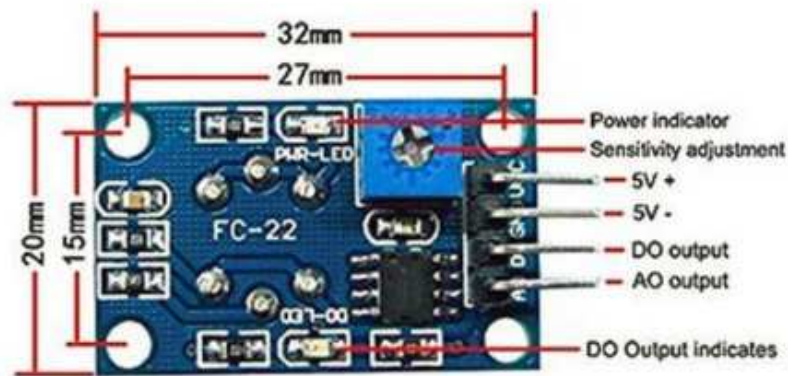
#### HARDWARE

- UNO Board (Fully compatible with Arduino UNO rev.3) x 1
- MQ-7 Sensor x 1
- Jumpers
- USB Cable x 1
- PC x 1

#### SOFTWARE

- Arduino IDE (version 1.6.4+)

### About MQ-7 Gas Sensor



## Description

### MQ-7 Semiconductor Sensor for Combustible Gas

Sensitive material of MQ-7 gas sensor is SnO<sub>2</sub>, which with lower conductivity in clean air. It make detection by method of cycle high and low temperature, and detect CO when low temperature (heated by 1.5V). The sensors conductivity is more higher along with the gas concentration rising. When high temperature (heated by 5.0V), it cleans the other gases adsorbed under low temperature. Please use simple electrocircuit, Convert change of conductivity to correspond output signal of gas concentration. MQ-7 gas sensor has high sensitivity to Carbon Monoxide. The sensor could be used to detect different gases contains CO, it is with low cost and suitable for different application.

## Character

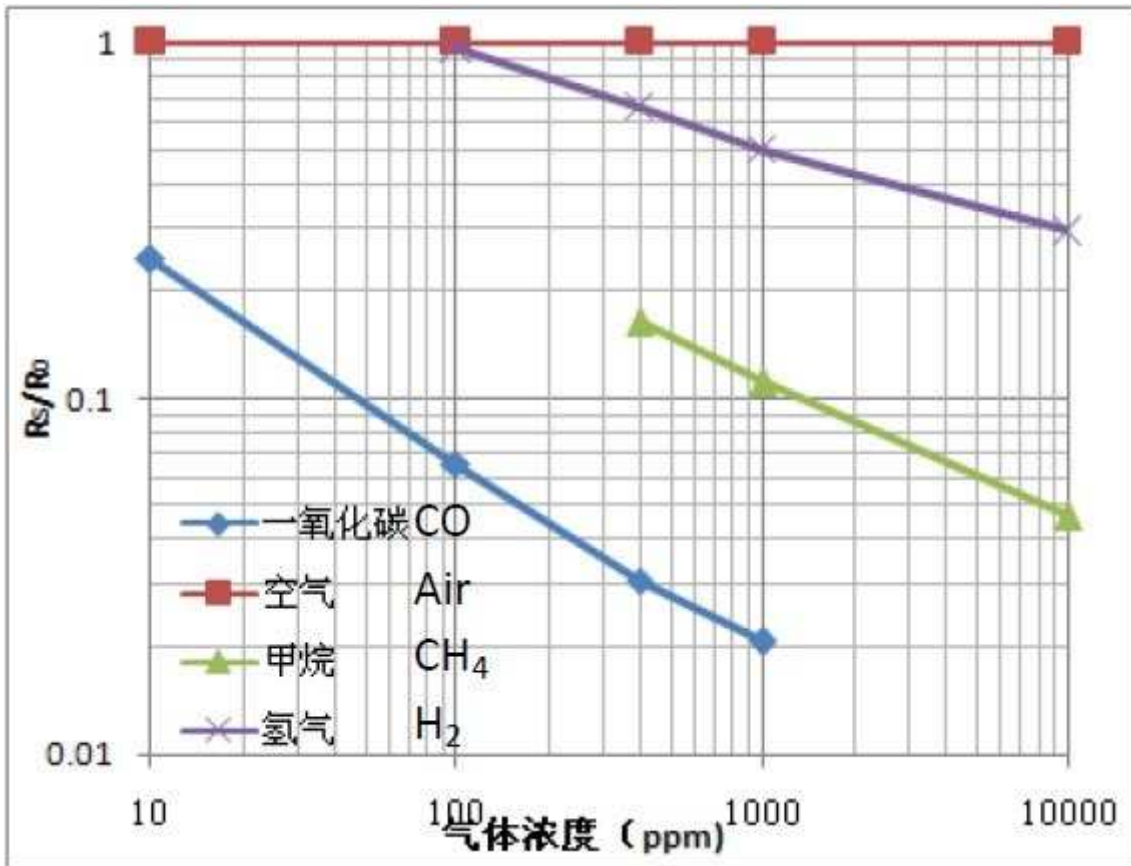
- High sensitivity to Combustible gas in wide range
- High sensitivity to Natural gas
- Fast response
- Wide detection range
- Stable performance, long life, low cost
- Simple drive circuit

## Technical Data

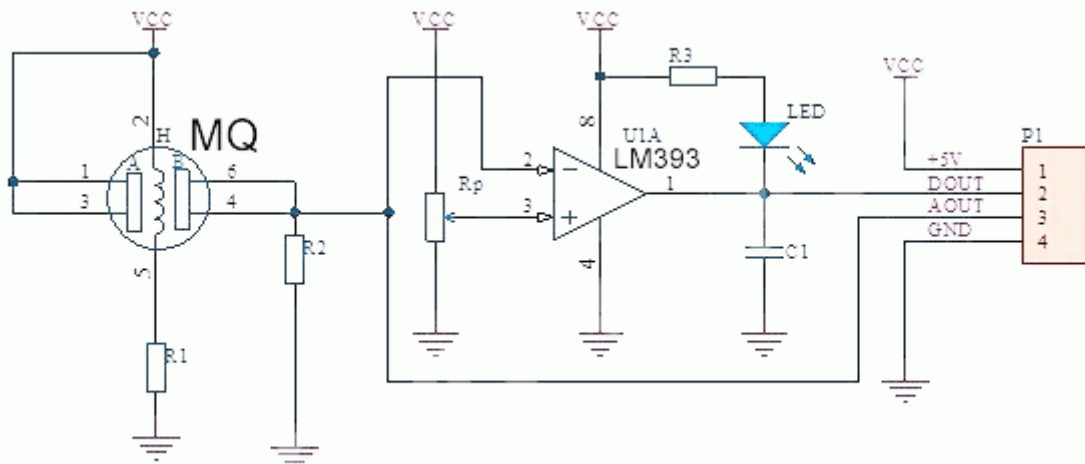
Model No.		MQ-7	
Sensor Type		Semiconductor	
Standard Encapsulation		Plastic	
Detection Gas		Carbon Monoxide	
Concentration		10-10000ppm CO	
Circuit	Loop Voltage	$V_c$	$\leq 10V$ DC
	Heater Voltage	$V_H$	$5.0V \pm 0.2V$ ACorDCHigh $1.5V \pm 0.1V$ ACorDCLow
	Heater Time	$T_L$	$60 \pm 1S$ High $90 \pm 1S$ Low
	Load Resistance	$R_L$	Adjustable
Character	Heater Resistance	$R_H$	$31\Omega \pm 3\Omega$ Room Tem.
	Heater consumption	$P_H$	$\leq 350mW$
	Sensing Resistance	$R_s$	$2K\Omega - 20K\Omega$ (in 100ppm CO )
	Sensitivity	$S$	$R_s(\text{in air})/R_s(100\text{ppm CO}) \geq 5$
	Slope	$\alpha$	$\leq 0.6(R_{300\text{ppm}}/R_{100\text{ppm CO}})$
Condition	Tem. Humidity	$20 \pm 265\% \pm 5\%RH$	
	Standard test circuit	$V_c: 5.0V \pm 0.1V$ $V_{HHigh}: 5.0V \pm 0.1V$ $V_{HLow}: 1.5V \pm 0.1V$	
	Preheat time	Over 48 hours	

#### Introduction to MQ-7 Carbon Monoxide Sensor

According to its datasheet, the MQ-7 carbon monoxide sensor detects 20 to 2000 ppm of CO in air. Here is its sensitivity characteristic curve:



This is a graph of  $R_s/R_0$  vs. gas concentration in ppm.  $R_s$  is the resistance of the sensor in target gas while  $R_0$  is the resistance in clean air. We will use this graph later when we create our code. This breakout board is more convenient as it converts resistance variations to voltage variations. Here is its schematic diagram:



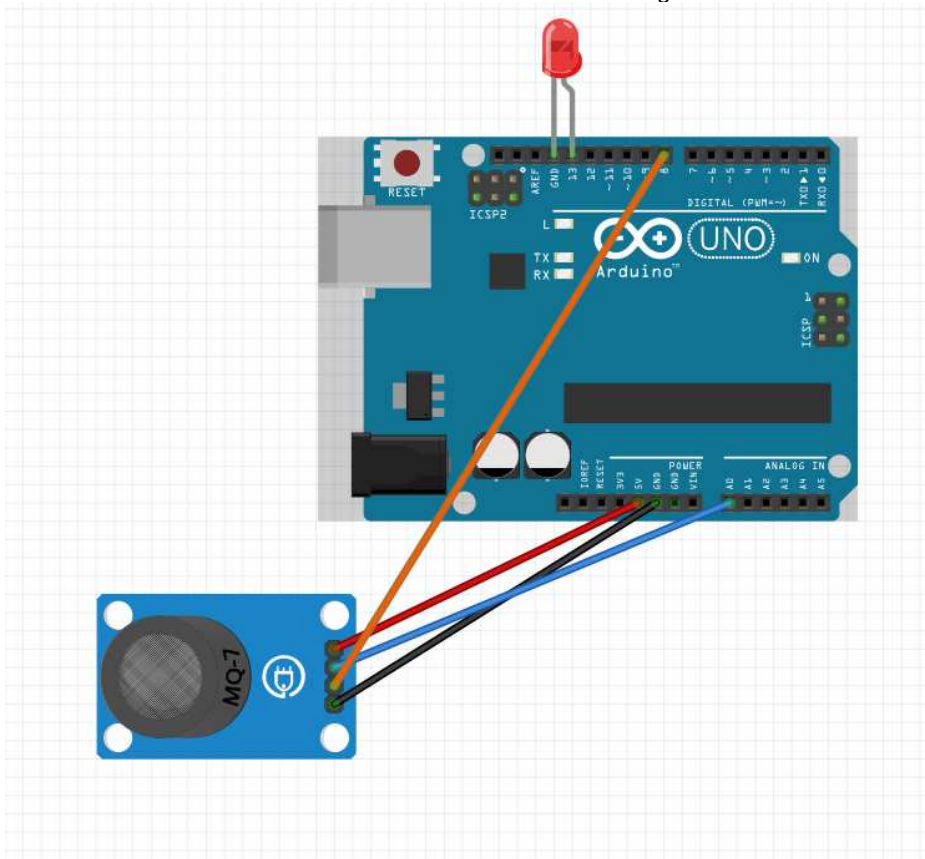
There are two ways to read the output from the MQ-7. One is through the DOUT pin which gives a high when the concentration threshold is reached and low otherwise. The threshold can be varied by adjusting the trimmer on the breakout board which is  $R_p$  in the schematic. Meanwhile, the AOUT pin gives varying voltage representing the CO concentration. We can convert the voltage reading to ppm if we look at the characteristic curve above. The relationship between concentration in ppm and  $R_s/R_0$  is:

$$ppm = 100 * \frac{\log(40) \cdot \frac{R_s}{R_0}}{\log(0.09)}$$

## Example

### CONNECTION

Connect the MQ 7 Sensro and the Arduino Board as below digram:



### ARDUINO SKETCH FOR MQ-7

#### SAMPLE CODE

```
CONST INT AOUTPIN=0; //THE AOUT PIN OF THE CO SENSOR GOES INTO ANALOG  
PIN A0 OF THE ARDUINO
```

```
CONST INT DOUTPIN=8; //THE DOUT PIN OF THE CO SENSOR GOES INTO DIGITAL  
PIN D8 OF THE ARDUINO
```

```
CONST INT LEDPIN=13; //THE ANODE OF THE LED CONNECTS TO DIGITAL PIN D13  
OF THE ARDUINO
```

```
INT LIMIT;

INT VALUE;

VOID SETUP() {

SERIAL.BEGIN(115200); //SETS THE BAUD RATE

PINMODE(DOUTPIN, INPUT); //SETS THE PIN AS AN INPUT TO THE ARDUINO

PINMODE(LEDPIN, OUTPUT); //SETS THE PIN AS AN OUTPUT OF THE ARDUINO

}

VOID LOOP()

{

VALUE= ANALOGREAD(AOUTPIN); //READS THE ANALAOG VALUE FROM THE CO
SENSOR'S AOUT PIN

LIMIT= DIGITALREAD(DOUTPIN); //READS THE DIGITAL VALUE FROM THE CO
SENSOR'S DOUT PIN

SERIAL.PRINT("CO VALUE: ");

SERIAL.PRINTLN(VALUE); //PRINTS THE CO VALUE

SERIAL.PRINT("LIMIT: ");

SERIAL.PRINT(LIMIT); //PRINTS THE LIMIT REACHED AS EITHER LOW OR HIGH
(ABOVE OR UNDERNEATH)

DELAY(100);

IF (LIMIT == HIGH){

DIGITALWRITE(LEDPIN, HIGH); //IF LIMIT HAS BEEN REACHED, LED TURNS ON AS
STATUS INDICATOR
```

```
}  
  
ELSE{  
  
DIGITALWRITE(LEDPIN, LOW); //IF THRESHOLD NOT REACHED, LED REMAINS OFF  
  
}  
  
}
```

The first block of code defines all the pin connections of the sensor and the LED. Since the AOUTpin connects to analog pin A0, it is initialized to 0. Since the DOUTpin connects to digital pin D8, it is initialized to 8. Since the LED connects to digital pin D13, it is initialized to 13. 2 variables, *limit* and *value*, are also declared. These will be used to store the value of the analog pin AOUT and digital pin DOUT.

The next block of code sets the baud rate and declares the DOUTpin as input and the ledPin as output. This is because the sensor is an input to the arduino for the arduino to read and process the sensor value. And the LED is an output will serves an indicator if the sensor has detected alcohol.

The next block of code reads the sensor pin AOUT and stores the value in the integer *value*. It also reads the sensor pin DOUT and stores the value in the integer *limit*. We then print the alcohol value, which will be a numeric value ranging from either 0 (no alcohol detected) to 1023 (maximum level of carbon monoxide that can be read). We will aslo print the limit which will either be HIGH or LOW. If the CO detected is under the threshold level, the value of limit returned will be low. If the CO detected is above the threshold, the value of limit returned will be HIGH.

If the value is HIGH, the LED will turn on. If the value is low, the LED will remain off.