

## ARDUINO TUTORIAL

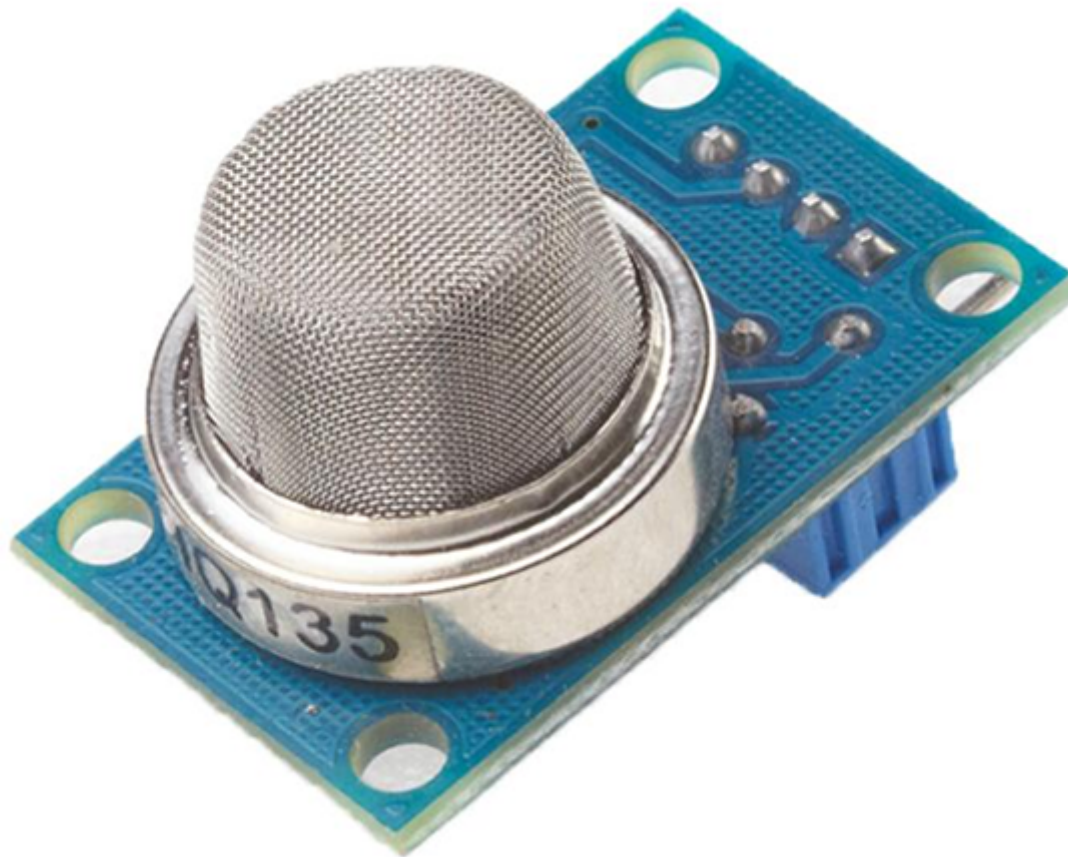
# MQ-135 Air Quality Sensor Tutorial



Roland Pelayo

January 29, 2020

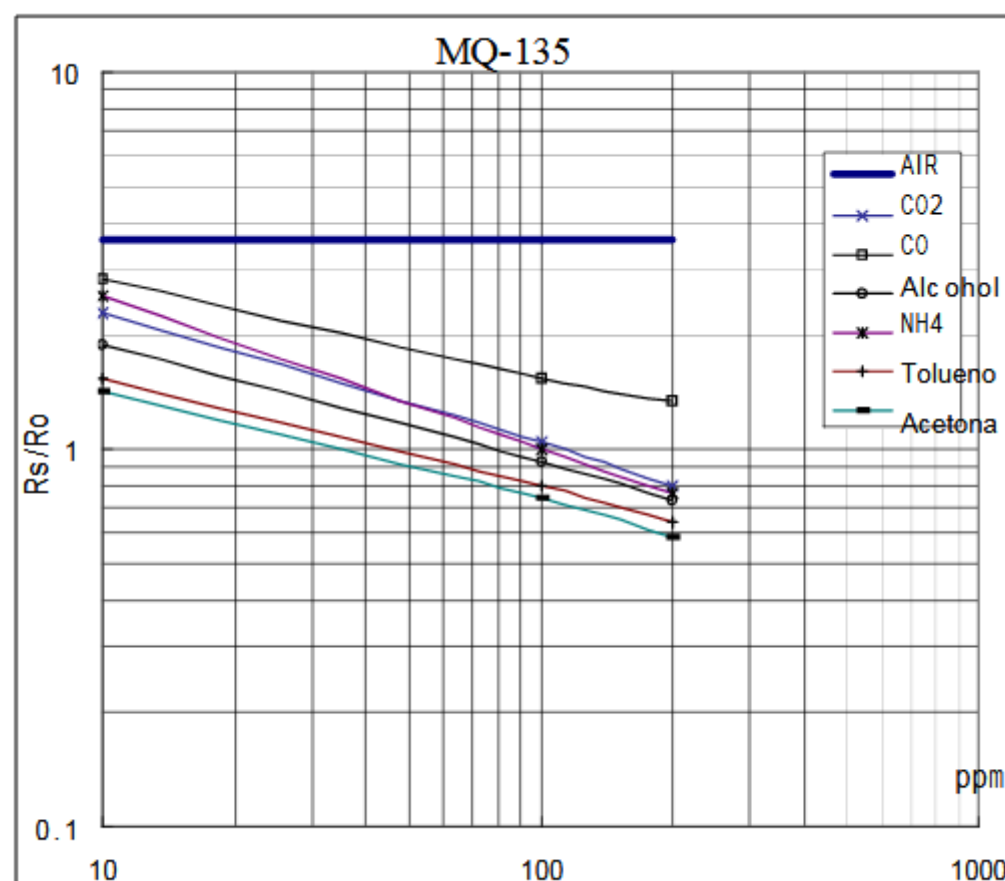
5 mins read



The MQ-135 “air quality” sensor is part of the MQ series of gas sensors that uses a chemical sensitive element. The resistance of such element varies when exposed to certain gases. The MQ-135 is sensitive to multiple gases including Benzene, Acetone, Alcohol and Ammonium.

## Getting Gas Concentration from MQ-135

Like all MQ gas sensors, the MQ-135 follows a sensitivity curve:

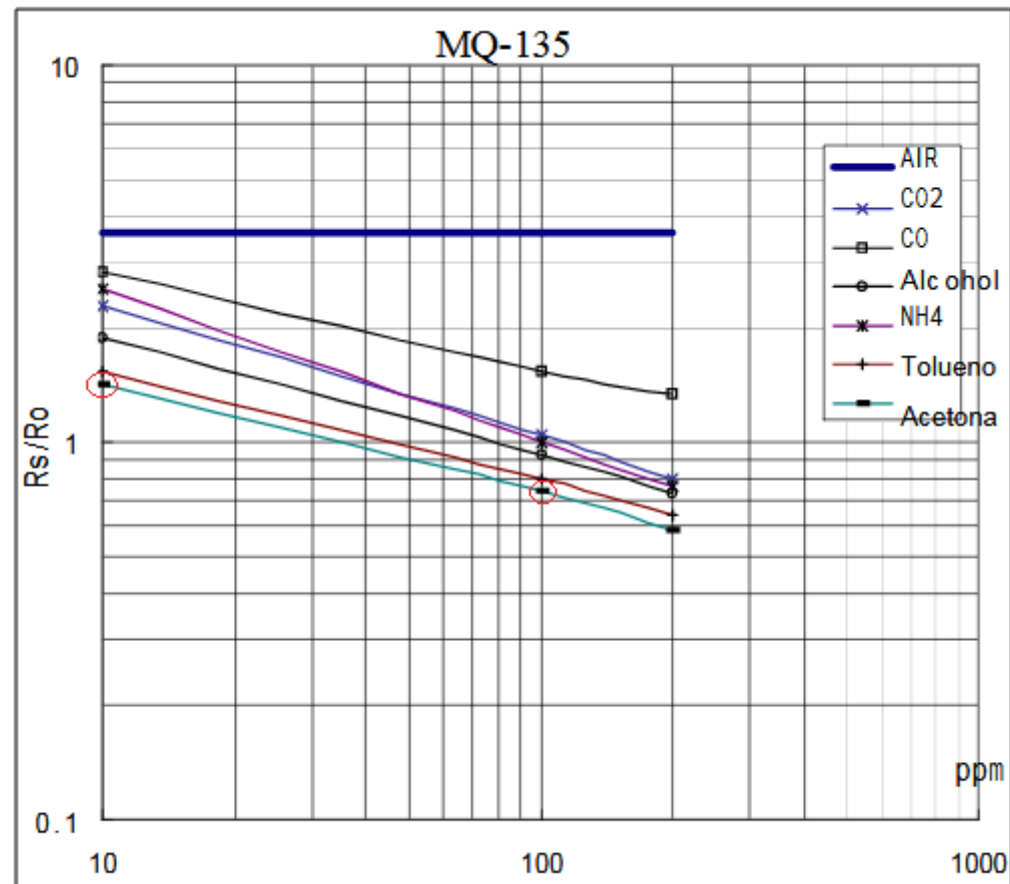




of the gas in parts per million (ppm), time the  $\gamma$  value to the  $R_s/R_0$  ratio. The value of the sensor resistance in clean air over the resistance of the sensor in various gases.

[the\_ad id="3059"]

Let's select acetone as our target gas. There are two clear intersections of the acetone line: one is when PPM is 100 or  $10^2$  and the other is when PPM is 10 or  $10^1$ :



At PPM = 10, the  $R_s/R_0$  ratio is right between 1 or  $10^0$  and 1.2589 or  $10^{0.1}$ . We can interpolate this as  $10^{0.05}$  hence at PPM=10,  $R_s/R_0$  should be approximately 1.122.

At PPM = 100, the  $R_s/R_0$  ratio is right between  $10^{-0.4}$  and  $10^{-0.3}$ . This should be  $10^{-0.35}$  or approximately 0.447.

Now that we have two points on the graph, we can use the two-point form of the line equation:

$$y - y_1 = \frac{y_2 - y_1}{x_2 - x_1} (x - x_1)$$

We set  $y_1 = 1.122$ ,  $x_1 = 10$  and  $y_2 = 0.447$ ,  $x_2 = 100$ . Our equation now for getting the concentration in PPM for acetone is:

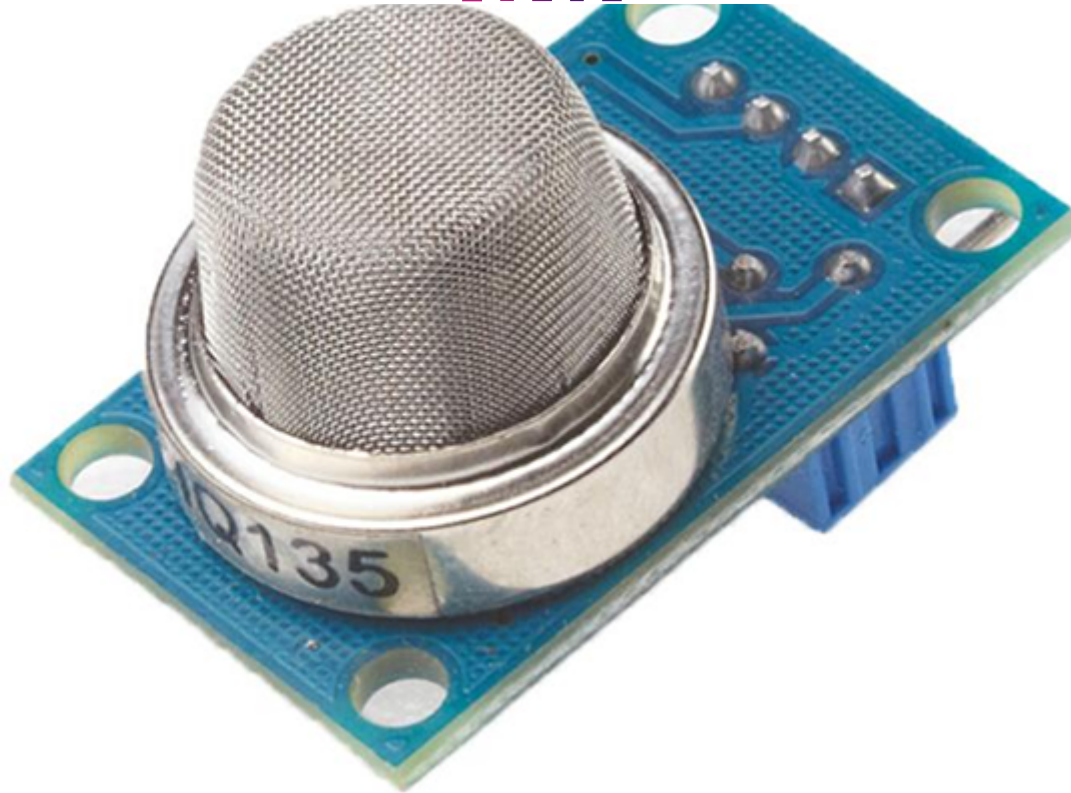
$$PPM_{acetone} = 159.6 - 133.33 \frac{R_s}{R_0}$$

The next step is to determine the resistance of the sensor in clean air  $R_0$ . This can be done by measuring the output voltage of the sensor and then deriving the resistance from that voltage.

## The MQ-135 Breakout Board

For this tutorial, I am using the MQ-135 sensor breakout board:



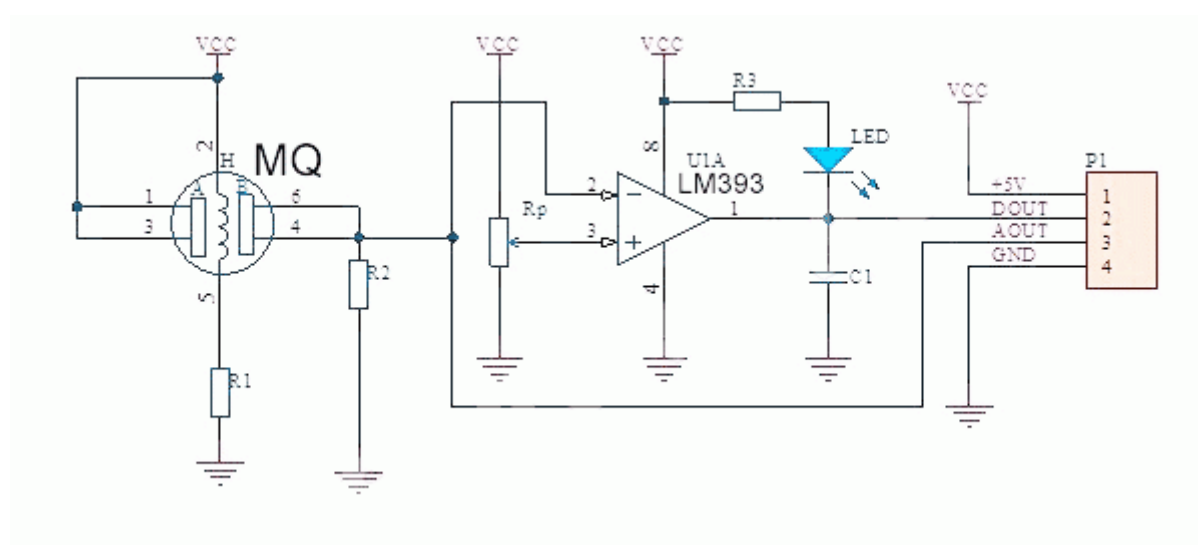


The MQ-135 breakout board, like other MQ boards, have four pins. The D0 pin goes high when sufficient levels of Benzene, Alcohol, Acetone or Ammonium are in the air. The exact concentration of these chemicals that triggers the D0 pin is unknown. Hence, measuring concentration is done through the A0 pin.

The A0 pin provides a voltage proportional to the concentration of the chemicals to which MQ-135 is sensitive. But before you can acquire the concentration in PPM, you must determine first the resistance of the sensor in clean air ( $R_0$ ).

## Measuring the Clean Air Resistance

Almost all MQ gas sensors follow this schematic:



Our aim is to measure the resistance of the sensor in clean air ( $R_0$ ). As shown, measuring the resistance between the A0 pin and GND would give us the value of resistor R2.  $R_0$  here creates a voltage divider with R2. Hence, if we can measure the output voltage at A0 in clean air,  $R_0$  is:

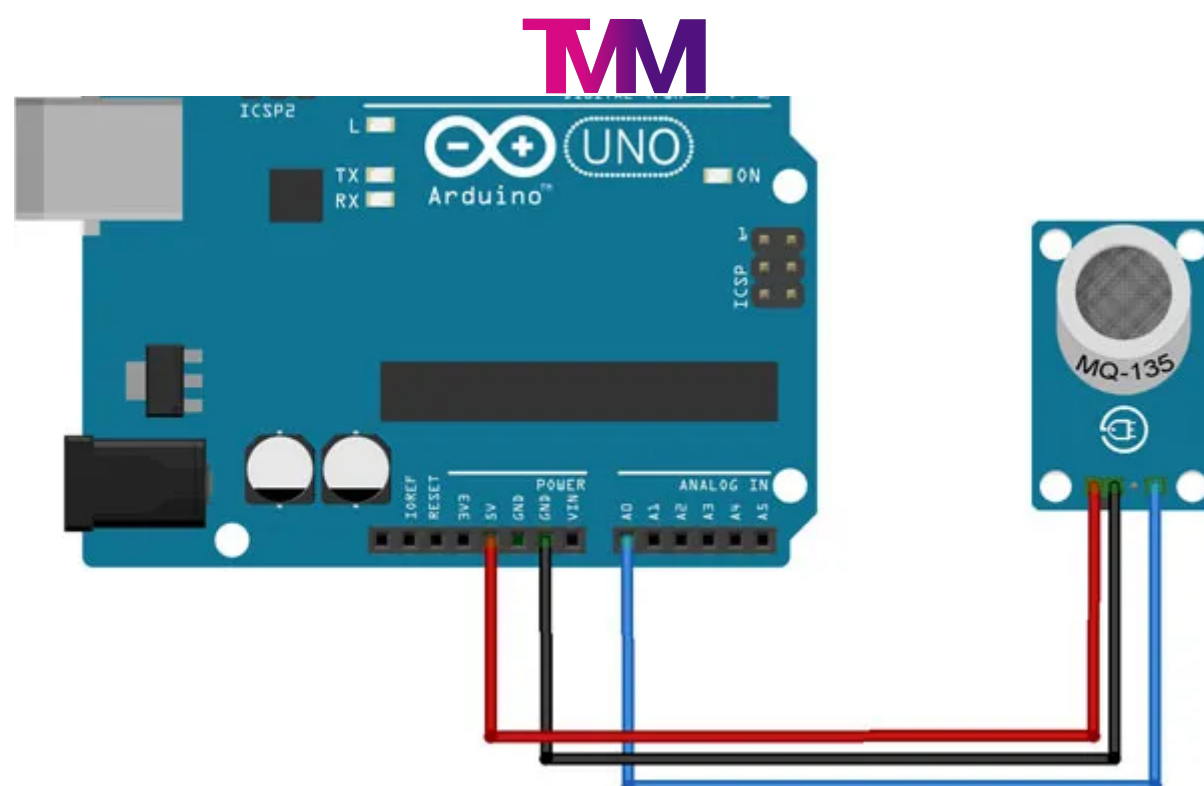
$$R_0 = \frac{R_2(1 - A_0)}{A_0}$$

Using an ohmmeter, measure the resistance between the A0 and GND to acquire R2. For example, the R2 on my module is approximately 1000 ohms.

[the\_ad id="3059"]

Now, wire up the diagram as shown:

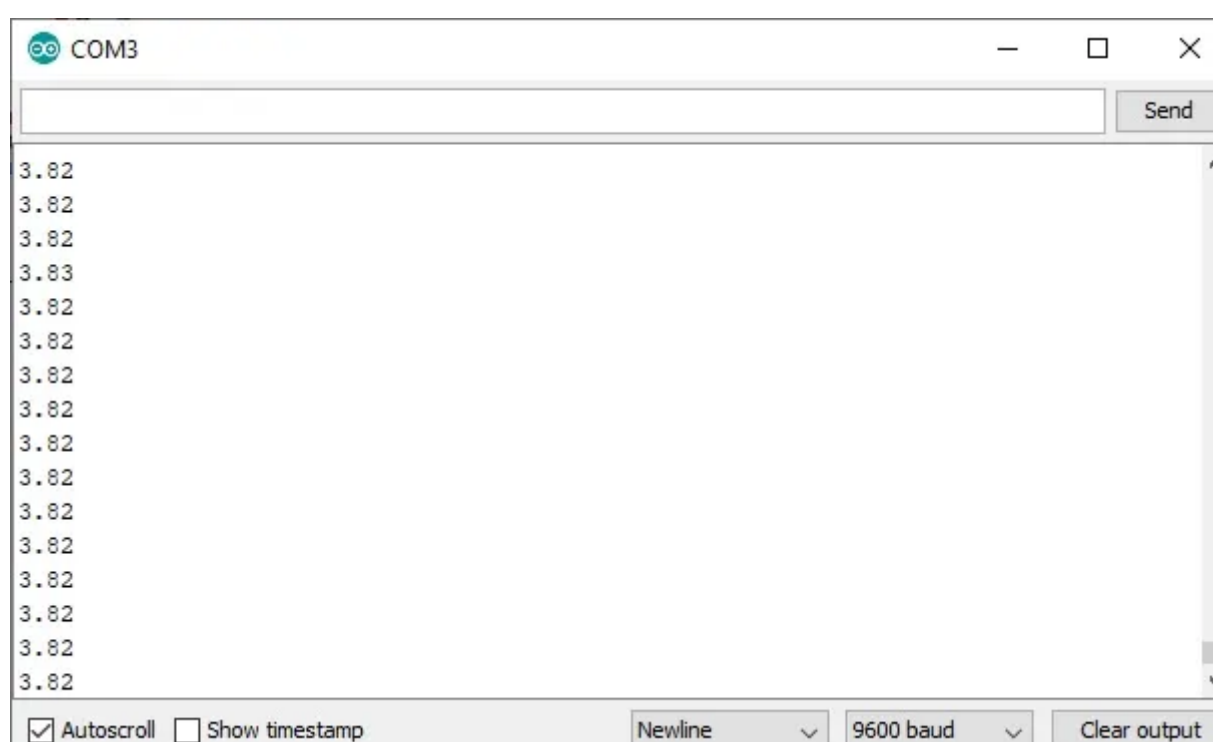




We will be using Arduino's analog to digital converter to determine voltage A0. Upload the sketch below:

```
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  // read the input on analog pin 0:  
  int sensorValue = analogRead(A0);  
  // convert to voltage:  
  float volts = sensorValue * 5;  
  volts = volts / 1023;  
  // print out the voltage:  
  Serial.println(volts);  
  delay(1); // delay in between reads for stability  
}
```

Open the serial monitor and the voltage is shown. Here's a screenshot of my serial monitor:





Since these MQ sensors, these mine have not been used for a long time and longer to preheat. Mine have been in storage for months, and it took almost 30 minutes to preheat. You will know it is ready when the values on the serial monitor is not drifting anymore. Mine was stable when the reading came down to 0.85.

Since I now know what my device's A0 at clean air is, I can now determine R0 using the formula:

$$R0 = \frac{R_2(1 - A_0)}{A_0}$$

My R2 is 1000, so that gives me an R0 of around 176 ohms.

## MQ-135 Acetone Sensor with Arduino

The concentration in acetone can now be determined using this sketch:

```
int R0 = 176;
int R2 = 1000;
float RS;
float PPM_acetone;

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

void loop() {
  // read the input on analog pin 0:
  int sensorValue = analogRead(A0);
  // convert to voltage:
  float volts = sensorValue * 5;
  volts = volts / 1023;
  // calculate RS
  RS = R2 * (1-volts);
  RS = RS/volts;
  // calculate acetone PPM
  PPM_acetone = 159.6 - 133.33*(RS/R0);
  // print out the acetone concentration:
  Serial.println(PPM_acetone);
  delay(1); // delay in between reads for stability
}
```

Recall that the formula for the concentration of acetone is:

$$PPM_{acetone} = 159.6 - 133.33 \frac{R_S}{R_0}$$







required to reverse the formula to get the concentration of acetone in PPM:

$$R_S = \frac{R_2(1 - A_0)}{A_0}$$

We use this value in the previous formula to get the concentration of acetone in PPM.

## Air Quality Trigger Device

Now if your project doesn't require that the concentration of the gases is known, you can use the MQ-135 as a simple switch. We modify the wiring diagram where the D0 pin now connects to the Arduino UNO instead of the A0 pin:

This is now the sketch:

```
int D0 = 2;

void setup() {
  Serial.begin(9600);
  pinMode(D0, INPUT);
}

void loop() {
  if(digitalRead(D0) == HIGH){
    Serial.println("Chemical detected!");
  }else{;}
  delay(100); // delay in between reads for stability
}
```

The sketch uses the serial monitor as before. This time, it will print "Chemical detected" when any of the MQ-135 chemicals is near.

Have any questions, reactions or suggestions? Kindly drop a comment below.

[Acetone](#)[Alcohol](#)[Ammonia](#)[Arduino UNO](#)[Mq](#)[Sensor](#)[Toluene](#)[Facebook](#)[Twitter](#)

### **Roland Pelayo**

Roland Pelayo started *TMM* in 2015. He is a firmware engineer who has over ten years of experience in developing electronic and microcontroller-based systems. Roland's designs include medical devices, security and automation, robots, emergency alert systems, and

