

DETECTOR DE GASES CON ARDUINO Y LA FAMILIA DE SENSORES MQ

Fuente: <https://www.luisllamas.es/arduino-detector-gas-mq/>

Versión: 06-12-19

¿QUÉ ES UN SENSOR DE GASES MQ?

Los sensores de gases MQ son una familia de dispositivos diseñados para detectar la presencia de distintos componentes químicos en el aire. Podemos conectar estos dispositivos a un autómata o procesador como Arduino.

Existe una gran variedad de sensores MQ. Cada modelo está diseñado para detectar una o más sustancias, pensadas para un uso específico, como por ejemplo detección gases inflamables, calidad del aire o detección de alcohol en aire respirado.

Los sensores de gases MQ suelen proporcionarse con una placa de medición estándar con el comparador LMC662 o similar, que permite obtener la lectura tanto como un valor analógico, como un valor digital cuando se supera un cierto umbral regulado a través de un potenciómetro ubicado en la placa.

Los sensores de gases deben ser calibrados antes de obtener una medida precisa. Aun calibrados estos sensores no disponen de la garantía necesaria para formar parte de un sistema de seguridad.

No usar estos sensores en aplicaciones de las que dependa la seguridad de personas o equipamientos.

Pese a sus limitaciones, los sensores de gases tipo MQ son muy usados en proyectos de electrónica casera con Arduino. Por ejemplo, podemos hacer encender o apagar un ventilador en función de la calidad del aire, hacer un pequeño detector de alcoholemia, o una alarma que suene al detectar humos.

PRECIO

Los sensores de gas MQ son dispositivos relativamente baratos. El precio varía de un modelo a otro pero, en general, oscila entre 1 a 1.5€, en vendedores internacionales de eBay y AliExpress.



¿CÓMO FUNCIONA UN SENSOR DE GASES?

Los sensores MQ están compuestos por un sensor electro-químico que varía su resistencia al estar en contacto con las sustancias.

Los sensores de gases son dispositivos con alta inercia, es decir, la respuesta necesita tiempos largos para estabilizarse tras un cambio de concentración de los gases medidos. Ello es debido a la necesidad física de que el gas abandone el material sensible, lo cual es un proceso lento.

Todos los modelos MQ disponen de un calentador necesario para elevar la temperatura del sensor, y que sus materiales adquieran la sensibilidad. Mientras el calentador no alcance la temperatura de funcionamiento, la lectura del sensor no será fiable.

El tiempo de calentamiento depende de cada modelo de sensor. En la mayoría de modelos es suficiente para con unos pocos minutos pero algunos modelos requieren hasta 12 y 48 horas hasta obtener mediciones estable.

Por otro lado, cada modelo necesita su propia tensión para alimentar el calentador. En muchos modelos esta tensión es de 5V, pero algunos modelos tienen condicionantes especiales para la alimentación.

El consumo de los sensores MQ puede ser elevado debido al calor necesario para funcionar el calentador, que puede llegar hasta 800 mW en algunos modelos. Esto es superior a la potencia que puede suministrar el regulador de Arduino, por lo que será necesario proporcionar una fuente de alimentación externa.

A continuación, tenéis una tabla de resumen con los distintos modelos de sensores disponibles, los gases a los que son sensibles, y algunos datos sobre el calentador.

No obstante, consultar detalladamente el Datasheet de cada sensor MQ particular antes de emplearlo para detallar sus especificaciones técnicas, especialmente la tensión de alimentación del calentador, el tiempo de calentamiento, y la curva de sensibilidad del sensor.

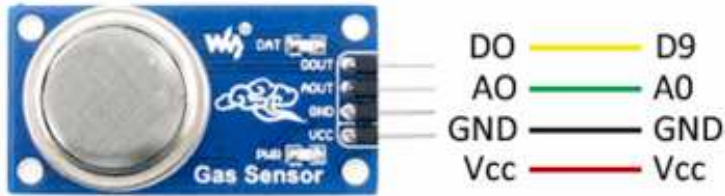
Modelo	Sustancias detectadas	Calentador
MQ-2	Metano, butano, GLP, humo	5V
MQ-3	Alcohol, Etanol, humo	5V
MQ-303A	Alcohol, etanol, humo	0.9V
MQ-4	Metano, gas natural comprimido (GNP)	5V
MQ-5	Gas natural, GLP	5V
MQ-6	Butano, GLP	5V
MQ-306A	Butano, GLP	0.9V
MQ-7	Monóxido de carbono	Alternado 5V y 1.4V
MQ-307A	Monóxido de carbono	Alternado 0.2 y 0.9V
MQ-8	Hidrógeno	5V
MQ-9	Monóxido de carbono, gases inflamables	Alternado 5V y 1.5V
MQ-309A	Monóxido de carbono, gases inflamables	Alternado 0.2 y 0.9V
MQ-131	Ozono	6V
MQ-135	Benceno, alcohol, humo, calidad del aire	5V
MQ-136	Ácido sulfhídrico	5V
MQ-137	Amoniaco	5V
MQ-138	Benceno, tolueno, alcohol, acetona, propano, formaldeido, hidrógeno	5V
MQ-214	Metano, gas natural	5V
MQ-216	Gas natural, gas carbón	6V
MG-811	Dióxido de carbono	6V
AQ-104	Calidad del aire *	
AQ-2	Gases inflamables, humo	
AQ-3	Alcohol, Benceno	
AQ-7	Monóxido de carbono	

* Conviene amplificación

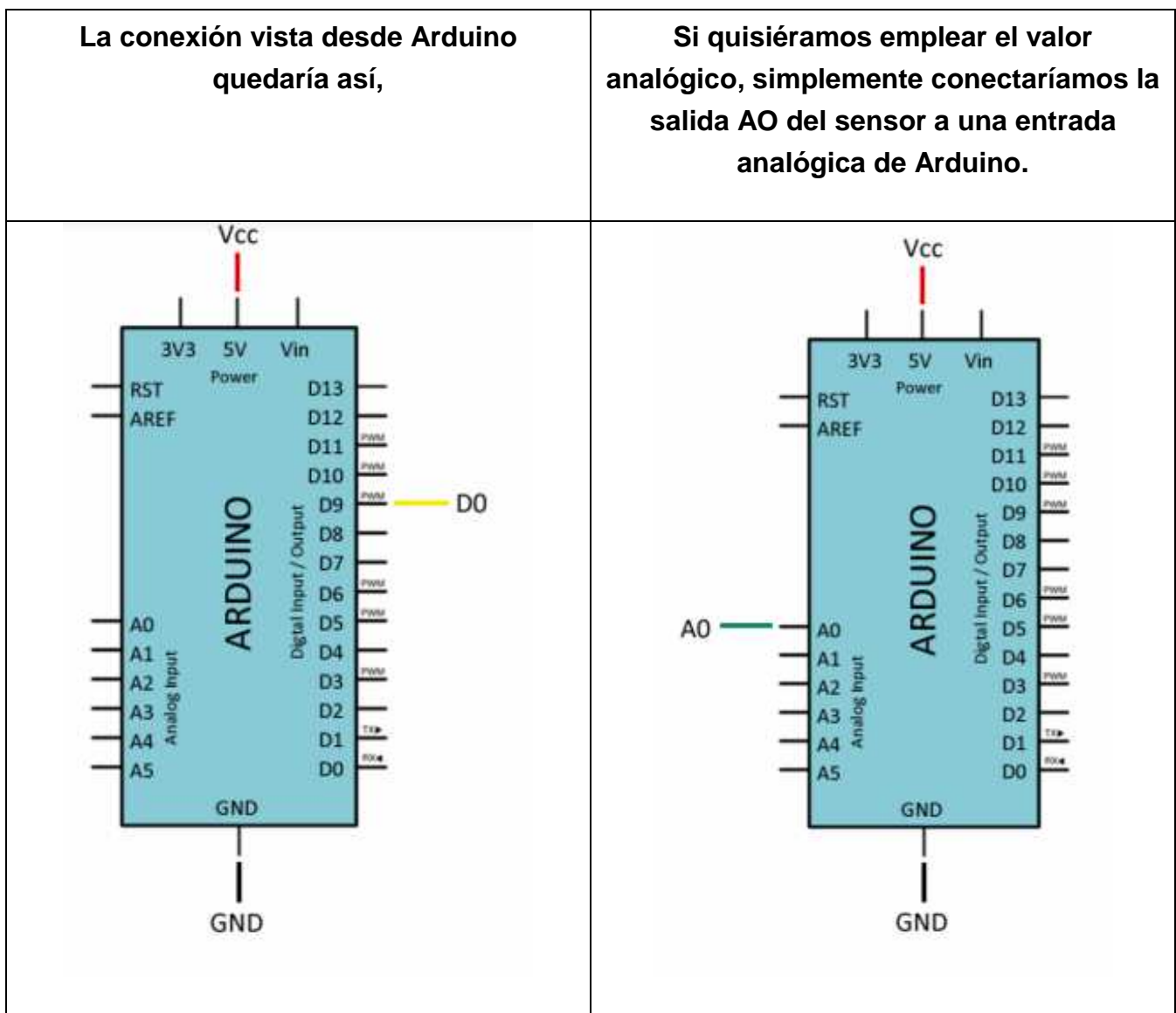
ESQUEMA DE MONTAJE (GENERAL)

El esquema eléctrico es sencillo. Alimentamos el módulo conectando GND y 5V a los pines correspondientes de Arduino.

Ahora si queremos usar la lectura digital, conectamos la salida DO a una de las entradas digitales de Arduino.



Opcionalmente, podemos calibrar el umbral de disparo de la salida digital con el potenciómetro instalado en el módulo.



EJEMPLOS DE CÓDIGO

LECTURA DIGITAL

El siguiente ejemplo muestra la lectura digital del sensor. El código es muy, simplemente utilizamos una entrada digital para comprobar el estado del sensor, empleando el puerto serie para informar de la detección. En un ejemplo real, realizaríamos las acciones oportunas ante una detección.

```

const int MQ_PIN = 2;
const int MQ_DELAY = 2000;

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

void loop()
{
  bool state= digitalRead(MQ_PIN);

  if (!state)
  {
    Serial.println("Deteccion");
  }
  else
  {
    Serial.println("No detectado");
  }
  delay(MQ_DELAY);
}

```

LECTURA ANALÓGICA

En el siguiente ejemplo realizamos la lectura analógica del sensor. Al igual que el anterior, es código es muy sencillo. Simplemente empleamos una entrada analógica cualquiera para leer la salida analógica del sensor, y mostramos el resultado por puerto serie.

```

const int MQ_PIN = A0;
const int MQ_DELAY = 2000;

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

void loop()
{
  int raw_adc = analogRead(MQ_PIN);
  float value_adc = raw_adc * (5.0 / 1023.0);

  Serial.print("Raw:");
  Serial.print(raw_adc);
  Serial.print(" Tension:");
  Serial.println(value_adc);

  delay(MQ_DELAY);
}

```

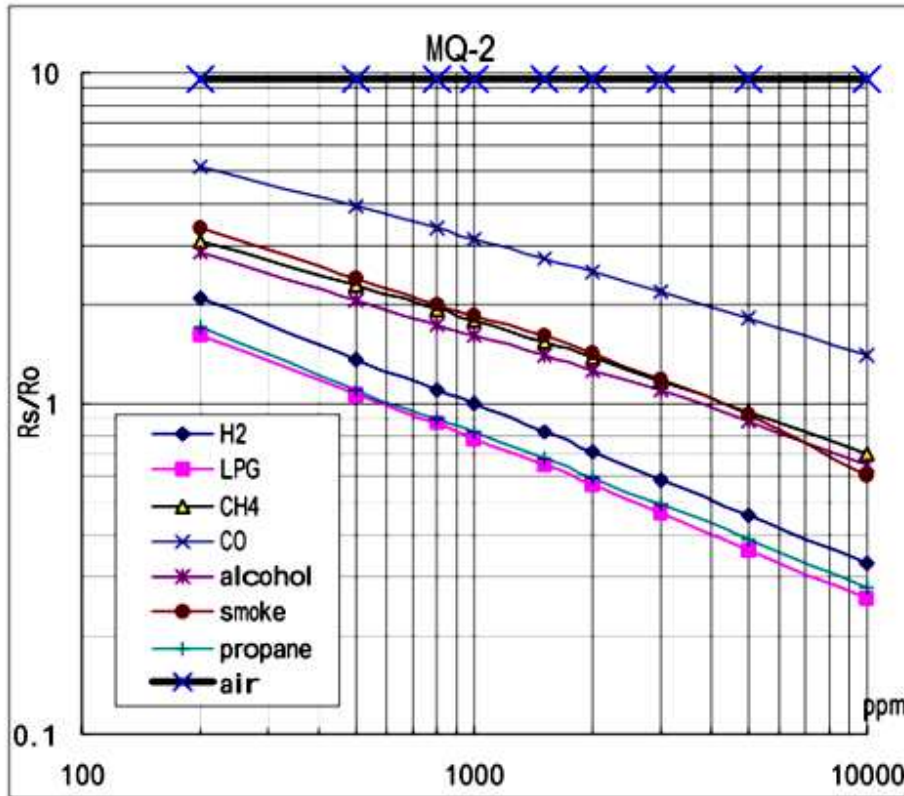
LEER LA CONCENTRACIÓN

En el siguiente ejemplo, empleamos la lectura analógica para determinar la concentración del gas normalmente en ppm (partes por millón), pero algunos sensores usan otras unidades como mg/L o bpm (partes por billón)

El Datasheet de cada sensor proporciona unas gráficas que permiten obtener la concentración del gas a partir de la relación entre la resistencia del sensor R0 y la

resistencia medida R_s . También es necesario conocer la resistencia R_I empleada en el módulo para realizar la lectura del sensor MQ.

Por ejemplo, la siguiente imagen muestra las curvas de concentración de cada gas medido en un sensor MQ-2



Tendrás que tomar como referencia las curvas de concentración del Datasheet del sensor que estéis usando.

Las gráficas se disponen en escala logarítmica para ambos ejes y, en general, son aproximadamente rectas bajo estas escalas. Por lo que la concentración resultará,

$$\text{Concentracion} = 10^{(A \cdot \log(R_s/R) + B)}$$

Para determinar la concentración necesitaremos la recta que la aproxima, para lo cuál debemos coger dos puntos cuales quiera de las gráficas $P_0 = \{X_0, Y_0\}$ y $P_1 = \{X_1, Y_1\}$, resultando la ecuación de la recta

$$Y = A \cdot x + B$$

Siendo

$$A = \frac{Y_1 - Y_0}{X_1 - X_0}$$

$$B = Y_0 - A \cdot X_0$$

```

const int MQ_PIN = A0;    // Pin del sensor
const int RL_VALUE = 5;  // Resistencia RL del modulo en Kilo ohms
const int R0 = 10;      // Resistencia R0 del sensor en Kilo ohms

// Datos para lectura multiple
const int READ_SAMPLE_INTERVAL = 100; // Tiempo entre muestras
const int READ_SAMPLE_TIMES = 5;     // Numero muestras

// Ajustar estos valores para vuestro sensor según el Datasheet
// (opcionalmente, según la calibración que hayáis realizado)
const float X0 = 200;
const float Y0 = 1.7;
const float X1 = 10000;
const float Y1 = 0.28;

// Puntos de la curva de concentración {X, Y}
const float punto0[] = { log10(X0), log10(Y0) };
const float punto1[] = { log10(X1), log10(Y1) };

// Calcular pendiente y coordenada abscisas
const float scope = (punto1[1] - punto0[1]) / (punto1[0] - punto0[0]);
const float coord = punto0[1] - punto0[0] * scope;

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

void loop()
{
  float rs_med = readMQ(MQ_PIN);    // Obtener la Rs promedio
  float concentration = getConcentration(rs_med/R0); // Obtener la concentración

  // Mostrar el valor de la concentración por serial
  Serial.println("Concentración: ");
  Serial.println(concentration);
}

// Obtener la resistencia promedio en N muestras
float readMQ(int mq_pin)
{
  float rs = 0;
  for (int i = 0; i < READ_SAMPLE_TIMES; i++) {
    rs += getMQResistance(analogRead(mq_pin));
    delay(READ_SAMPLE_INTERVAL);
  }
  return rs / READ_SAMPLE_TIMES;
}

// Obtener resistencia a partir de la lectura analogica
float getMQResistance(int raw_adc)
{
  return (((float)RL_VALUE / 1000.0 * (1023 - raw_adc) / raw_adc));
}

// Obtener concentracion 10^(coord + scope * log (rs/r0))
float getConcentration(float rs_ro_ratio)
{
  return pow(10, coord + scope * log(rs_ro_ratio));
}

```