

# “MANEJO DE SENSORES Y CONVERSIÓN A/D EN ARDUINO”

Informe N°7

Laboratorio de Sistemas Embebidos

Melanny Dávila

*Ingeniería en Telecomunicaciones*  
*Facultad de Eléctrica y Electrónica*  
Quito, Ecuador  
melanny.davila@epn.edu.ec

2<sup>nd</sup> Jonathan Álvarez

*Ingeniería en Telecomunicaciones*  
*Facultad de Eléctrica y Electrónica*  
Quito, Ecuador  
jonathan.alvarez@epn.edu.ec

**Abstract**—En el presente informe se tratará la importancia que tienen los sensores en usos cotidianos al igual que los modos de trabajo. Luego se verán las características y principio de funcionamiento de los sensores de calidad MQ junto con sus curvas de comportamiento para finalmente presentar una implementación en Tinkercad de un sistema de control de calidad de aire.

**Index Terms**—Arduino, barrido, teclado, display.

## I. INTRODUCCIÓN

Es importante aprender sobre la conversión AD en Arduino ya que no todas las señales serán de origen digital, por lo que será necesario realizar una conversión en ellas. Este proceso empieza midiendo la señal analógica, posteriormente se debe muestrear la señal y que esta tenga  $n$  componentes. Luego se asignan valores discretos a las muestras, obteniéndose un error de cuantización. Manejar sensores en la placa Arduino permite crear diferentes aplicaciones con propósitos cada vez más aplicados a la realidad por ejemplo, a la crisis sanitaria que se está viviendo actualmente, gracias al uso de un sensor de temperatura y otros en complemento es posible generar un dispensador de gel automático.

## II. OBJETIVOS

- Relacionar al estudiante con el uso y manejo de sensores en Arduino.
- Relacionar al estudiante con el uso y manejo del conversor Analógico Digital de la Placa Arduino Uno.
- Utilizar diferentes sensores para resolver problemas cotidianos mediante sistemas de control [2].

## III. CUESTIONARIO

A. Describir la importancia del uso de sensores en los dispositivos cotidianos, mencionar tres ejemplos y describir detalladamente su funcionamiento.

En la vida cotidiana se pueden encontrar varias situaciones en las que el uso de un sensor podría evitar incidentes, mejorar la calidad de un servicio o evitar desperdicio de recursos. El uso de sensores presenta una mejora en eficiencia

y ahorro en cualquier implementación. Además, si se trabaja con microcontroladores como Arduino se encontrará en el mercado una gran gama de sensores de bajo costo; los mismos que son relevantes para prácticamente todos los aspectos de la vida, incluida la seguridad, la vigilancia, el monitoreo.

A continuación, se describen tres ejemplos del uso de sensores:

- **Control del clima inalámbrico:** Mediante el uso del sensor DHT11, se envían datos de forma inalámbrica a los transceptores NRF24L01; para de esta manera llevar un control del cambio de temperatura exterior

Elementos:

- Sensor DHT11
- Módulo NRF24L01
- LCD 16x2
- Arduino UNO
- Arduino Nano
- Módulo de reloj DS3231

- **Localizador de sol:** En base al uso de las fotoresistencias, este circuito es capaz de detectar la posición del sol para que los servomotores moverán al panel solar, para que este pueda captar la luz solar.

Elementos:

- 4 resistencias de  $220[\Omega]$
- 4 fotoresistores
- 2 Servomotores
- 1 panel solar
- Arduino UNO

- **Recepción de cambios de temperatura en dispositivos externos mediante Bluetooth:** El termistor que se utilizará, presentará la temperatura actual mediante la variación del valor de la resistencia que se obtiene al implementar un divisor de voltaje. Así, se debe incorporar un módulo Bluetooth, el cual debe ser configurado para enviar la información capturada a través del mismo.

Elementos:

- Arduino Nano R3
- Termistor NTC

- Resistencias de 1[KΩ] y 2.2 [KΩ]
- Módulo Bluetooth HC 06

*B. Consultar los modos de trabajo del ADC single conversion mode y free running mode.*

La principal diferencia entre estos modos de trabajo consiste en que en modo single conversion el ADC hace una sola conversión y para, pero en modo free running el ADC está continuamente convirtiendo, es decir, hace una conversión y luego comienza con la siguiente.

- ADC single conversion mode: Activa la conversión y se realiza una vez. Cuando finaliza la conversión, tiene un resultado y el ADC no está haciendo nada más (no sin un nuevo activador de inicio de conversión). Dado que el dispositivo solo realiza una conversión cuando se lo indica. Esto significa que si desea una muestra, simplemente se debe solicitar una. Si desean diez muestras, se deben pedir una muestra diez veces.
- ADC free running mode: Significa que el ADC comienza automáticamente una nueva conversión cada vez que finaliza la última. No es necesario volver a activar el ADC en este modo.

*C. Consultar las características más relevantes y el principio de funcionamiento de los sensores de calidad de aire de la familia MQ y presentar sus curvas de comportamiento.*

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 autómatas 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. Pese a sus limitaciones, los sensores de gases tipo MQ son muy usados en proyectos de electrónica casera con Arduino.

**TABLA I**

Principales características de la familia MQ

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
MQ-811	Dióxido de carbono	6V
AQ-104	Calidad del aire *	
AQ-2	Gases inflamables, humo	
AQ-3	Alcohol, Benceno	
AQ-7	Monóxido de carbono	

Las curvas de comportamiento se adjunta en el ANEXO I.

*D. Utilizando el sensor de humo de Tinkercad simular un sistema de control de calidad de aire, el cual al detectar la presencia de humo activará una alarma sonora y un motor DC que simulará el sistema de ventilación, adicionalmente se presentará un led RGB el cual cambiará su color de verde a rojo dependiendo de la presencia o ausencia de humo en el ambiente.*

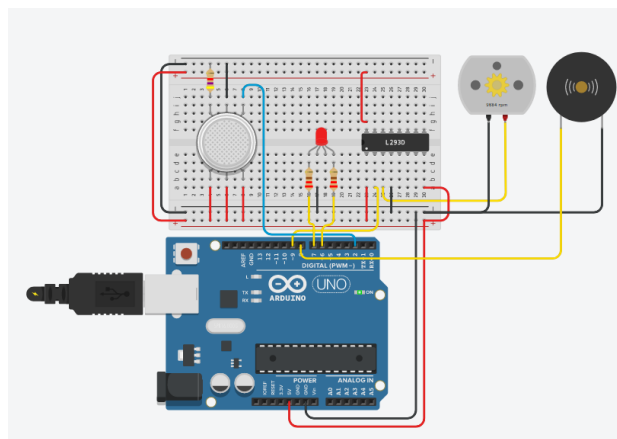


Fig. 1. Esquemático sistema de detección de humo implementado en Arduino

```

1 // Informe Practica 7
2 //Sistema detector de humo, al detectar enciende una
  alarma, motor
3 // y led RGB a color rojo
4
5 //Definicion de pines
6 int sensorHumo = 2; //Recibe la senial del sensor
7 int verde = 6;
8 int rojo = 7;
9 int alarma = 8;
10 int motor = 9;
11
12 int i;
13 void setup()
14 {
15   //Inicializacion de pines
16   for(i = 6; i<10; i++){
17     pinMode(i,OUTPUT);
18   }
19   pinMode(sensorHumo, INPUT);
20   //Serial.begin(9600);
21 }
22
23 void loop(){
24   //Se controla el estado del sensor de estar
    activado enciende
25   //las alertas
26   if(digitalRead(sensorHumo)){
27     analogWrite(motor, 255);
28     tone(alarma, 200);
29     digitalWrite(verde, LOW);
30     digitalWrite(rojo, HIGH);
31     //Asi sea por un momento que detecta humo las
      seniales de
32     //alerta estaran encendidas un tiempo
33     delay(3000);
34   }else{
35     noTone(alarma);
36     digitalWrite(rojo, LOW);
37     digitalWrite(verde, HIGH);
38     analogWrite(motor, 0);
39   }
40 }
41

```

Script 1. Código fuente para la detección de humo mediante sensor implementado en Arduino

*E. Realizar una comparación de los sensores empleados en una Weather Station implementada con un kit Arduino y una Weather Station de tipo comercial de la marca VAISALA. Enfatizar rangos de operación, precisión en la medición, mecanismos de comunicación y consumo energético.*

Para las comparaciones presentadas se hablara del modelo WXT536 de VAISALA, tiene un consumo de 42mW y utiliza protocolos como son: SDI-12 v1.3, ASCII, NMEA 0183 v3.0

1) *Medidor de Presión:* Para sensor de Arduino y VAISALA

**TABLA II**

Datos de los sensores BMP180 de presión para Arduino y VAISALA WXT536

	BMP180	VAISALA WXT536
Rango	950 a 1050 hPa	600 a 1100 hPa
Precisión de la medida	+0.12 hPa	0.1 hPa
Resolución	0.01 hPa	0.1 hPa
Consumo energético	80uW	42mW
Mecanismo de comunicación	I2C	SDI-12, NMEA

2) *Temperatura y humedad:*

**TABLA III**

Datos de los sensores DHT11 Humedad y Temperatura para Arduino y VAISALA WXT536

	DHT11		VAISALA WXT536	
	Temperatura	Humedad	Temperatura	Humedad
Rango	0 a 50	20 a 90% RH	-52 °C a 60 °C	0-100 % RH
Precisión de la medida	+2 °C	+5% RH	+ 0.3 °C	+ 3 %RH 0-90 % RH + 5% RH 90-100 % RH
Resolución	1 °C	1 % RH	0.1 °C	0.1 %RH
Consumo energético	1mW		42 mW	
Mecanismo de comunicación	Transmisión de datos serial de 40 bits		SDI-12, ASCII, NMEA 0183	

3) *Pluviómetro:*

**TABLA IV**

Principales características del pluviómetro

	Pluviómetro Arduino	VAISALA WXT536
Área de recolección	22 cm <sup>2</sup>	60 cm <sup>2</sup>
Resolución de salida	0.1 mm	0.01 mm
Consumo Energético	75 mW	42 mW

*F. Conclusiones:*

Jonathan Álvarez

- Los sensores son muy útiles para interactuar con el ambiente, su uso solia estar limitado al industrial pero con plataformas con Arduino se puede implementar circuitos de control a nivel del hogar
- Los sensores vienen con distintos rangos de sensibilidad y de operación es importante que estos se encuentren dentro del rango que Arduino puede manejar para las entradas analógicas y convertirlos para su apropiada manipulación
- Los sensores a nivel profesional tienen mejores prestaciones que los que se pueden manipular en Arduino, y ya vienen compuestos con diferentes sensores y muchas veces en espacios mas pequeños

- Es importante realizar el proceso de conversión señales analógicas a digitales, para que el microcontrolador Arduino trabaje con los datos adecuados y de esta manera los resultados obtenidos carezcan de errores.
- La implementación de funciones y de interrupciones es importante en aplicaciones que utilizan sensores es de gran importancia; dado que se asemejan cada vez más aplicaciones reales y por eso se debe garantizar eficiencia y eficacia.
- Se llegó a apreciar la importancia del proceso de muestreo, cuantización en una señal analógica con el fin de digitalizarla.

#### G. Recomendaciones:

##### Jonathan Álvarez

- Utilizar la conexión apropiada con el circuito integrado para que el motor gire.
- Convertir apropiadamente los datos provenientes de los sensores que vienen en escala de voltaje a las unidades apropiadas ya que será más intuitivo trabajar de esta forma,

##### Melanny Dávila

- Garantizar que la construcción del circuito sea realizada de una manera ordenada con el fin de poder detectar y corregir errores en el caso de que sucedan.
- Revisar previamente el uso de librerías y sus diferentes componentes y aplicaciones. Todo esto con el fin de desarrollar un código mucho más ordenado y sin errores.
- Considerar que las limitantes que presenta el uso de un sensor en específico ya que si la aplicación requiere una resolución bastante alta, es posible que el sensor no se comporte de manera adecuada.

#### REFERENCES

- [1] "Conversor Analógico Digital – Aprendiendo Arduino", Aprendiendo Arduino, 2020. [Online]. Disponible en: <https://aprendiendoarduino.wordpress.com/tag/conversor-analogico-digital/>. [Acced: 26- Enero- 2021].
- [2] E. Espinosa, E. Tatayo, "MANEJO DE PERIFÉRICOS DE E/S (Teclado Matricial y barrido de displays)". C.P. SISTEMAS EMBEBIDOS, Accedido: ene. 26, 2020. [En línea].
- [3]
- [4] "ADC", Aprendiendo Arduino. <https://aprendiendoarduino.wordpress.com/tag/adc/> (accedido ene. 27, 2021).
- [5] "STM32 ADC single channel updated method", ControllersTech, mar. 25, 2019. <https://controllerstech.com/stm32-adc-multi-channels/> (accedido ene. 27, 2021).
- [6] L. Llamas "Detector de gases con Arduino y la familia de sensores MQ", <https://www.luisllamas.es/arduino-detector-gas-mq/> (accedido ene. 27, 2021).

## ANEXO I

Las curvas de  $R_s/R_o$  vs PPM se tomaron a una temperatura de 20 grados centígrados, una humedad del 65% y con una concentración del 21%.

#### • Curvas características del sensor MQ-2

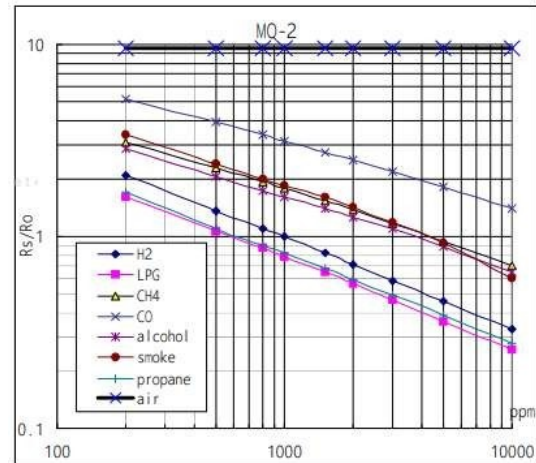


Fig. 2. Gráfica de  $R_s/R_o$  vs PPM medido con una carga de  $5[k\Omega]$

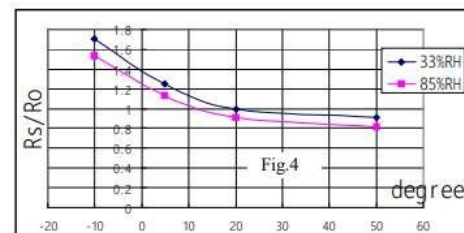


Fig. 3. Dependencia típica de la temperatura y la humedad

#### • Curvas características del sensor MQ-3

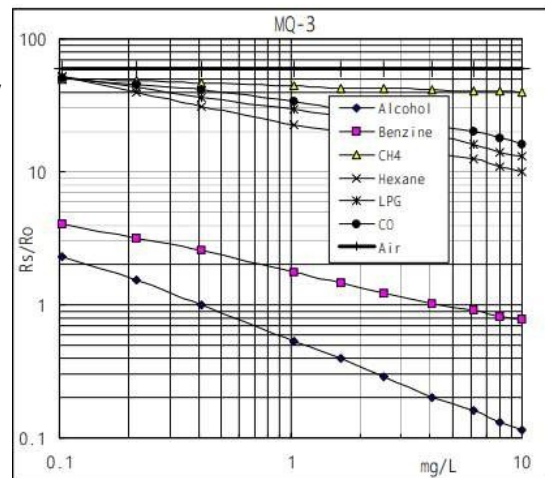


Fig. 4. Gráfica de  $R_s/R_o$  vs PPM medido con una carga de  $200[k\Omega]$

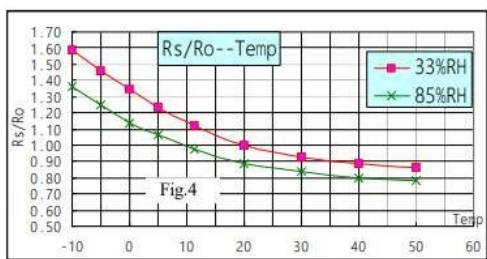


Fig. 5. Dependencia típica de la temperatura y la humedad

### • Curvas características del sensor MQ-4

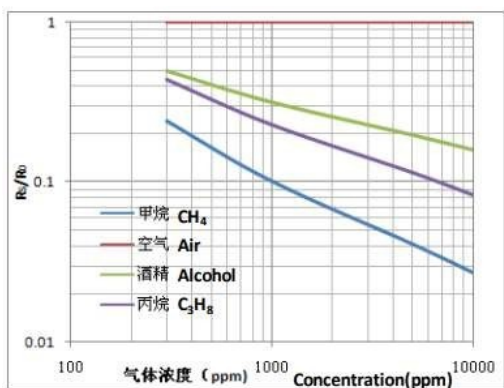


Fig. 6. Gráfica de Rs/Ro vs PPM medido con una carga de 4.7[kΩ]

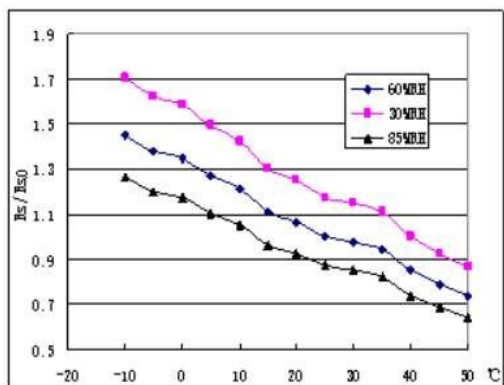


Fig. 7. Dependencia típica de la temperatura y la humedad

### • Curvas características del sensor MQ-5

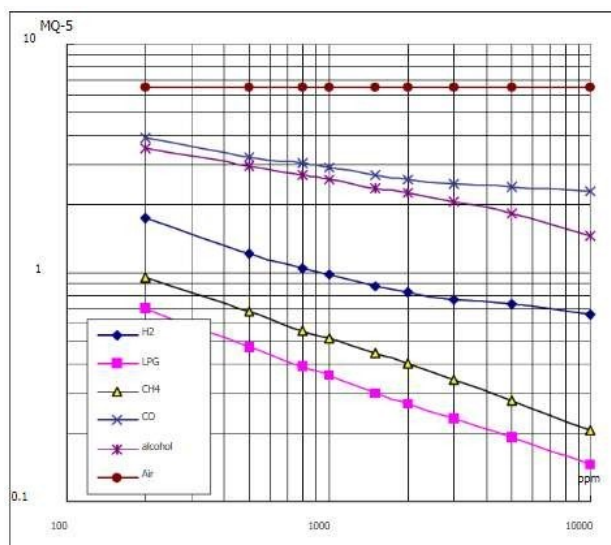


Fig. 8. Gráfica de Rs/Ro vs PPM medido con una carga de 20[kΩ]

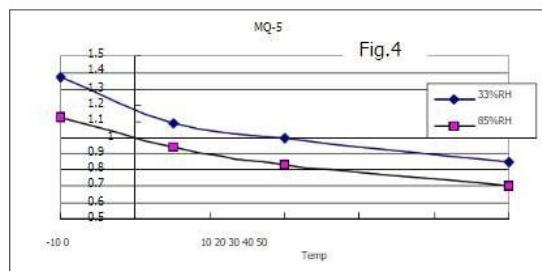


Fig. 9. Dependencia típica de la temperatura y la humedad

### • Curvas características del sensor MQ-6

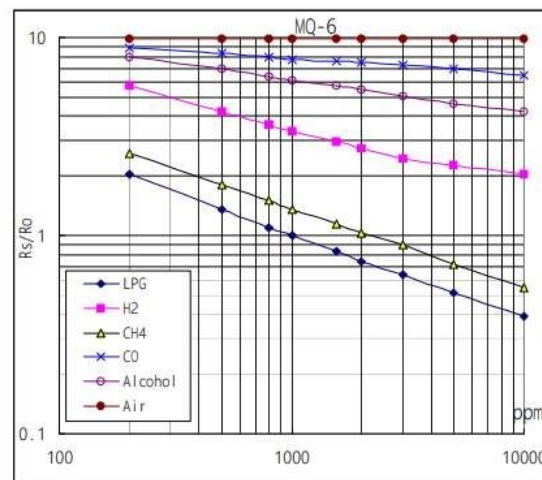


Fig. 10. Gráfica de Rs/Ro vs PPM medido con una carga de 20[kΩ]



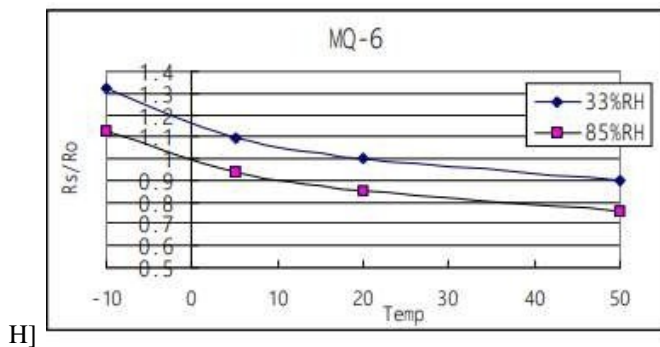


Fig. 11. Dependencia típica de la temperatura y la humedad

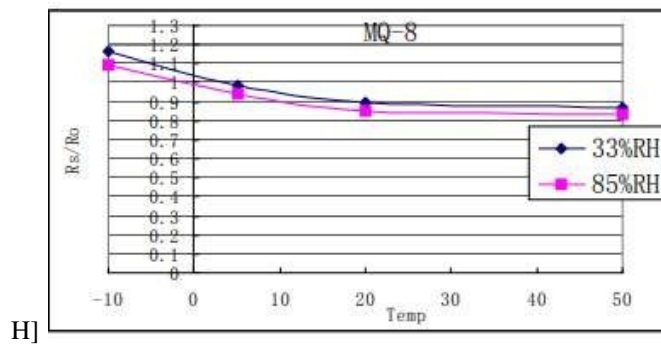


Fig. 15. Dependencia típica de la temperatura y la humedad

### • Curvas características del sensor MQ-7

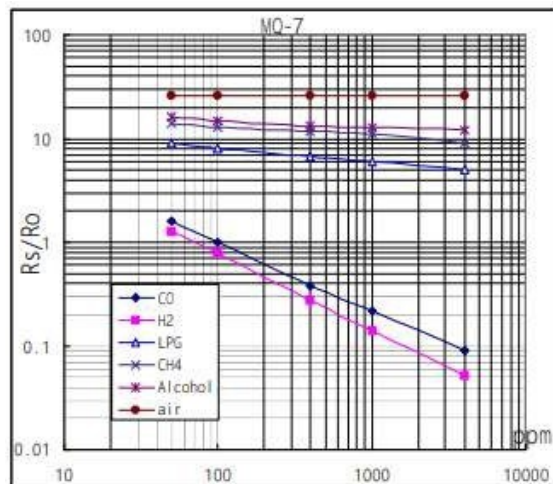


Fig. 12. Gráfica de  $R_s/R_o$  vs PPM medido con una carga de 10[kΩ]

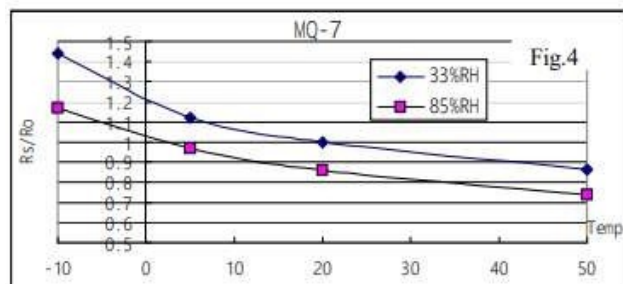


Fig. 13. Dependencia típica de la temperatura y la humedad

### • Curvas características del sensor MQ-8

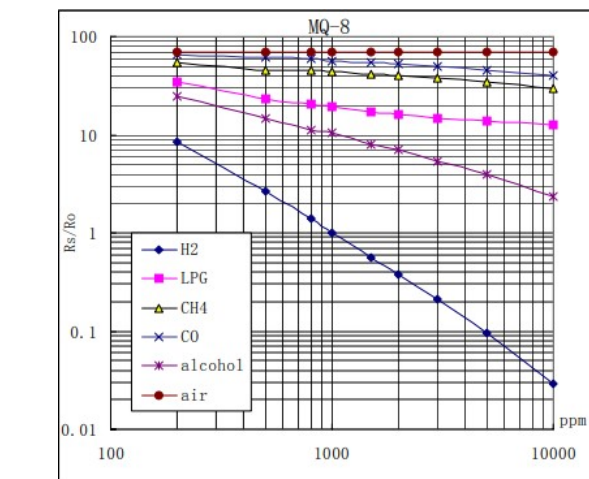


Fig. 14. Gráfica de  $R_s/R_o$  vs PPM medido con una carga de 10[kΩ]

### • Curvas características del sensor MQ-9

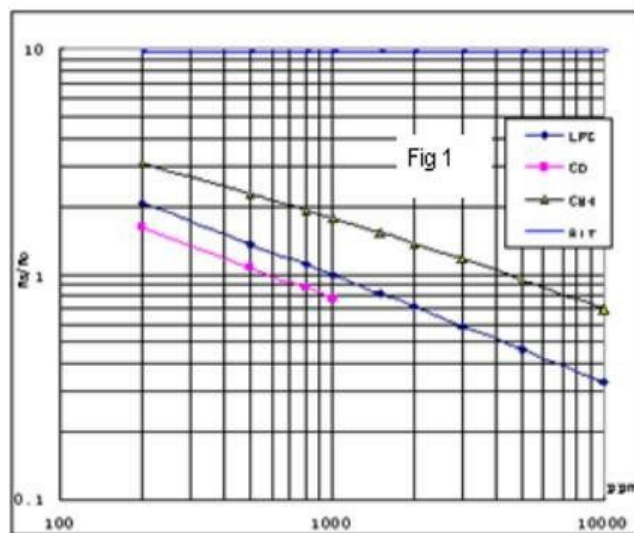


Fig. 16. Gráfica de  $R_s/R_o$  vs PPM medido con una carga de 5[kΩ]

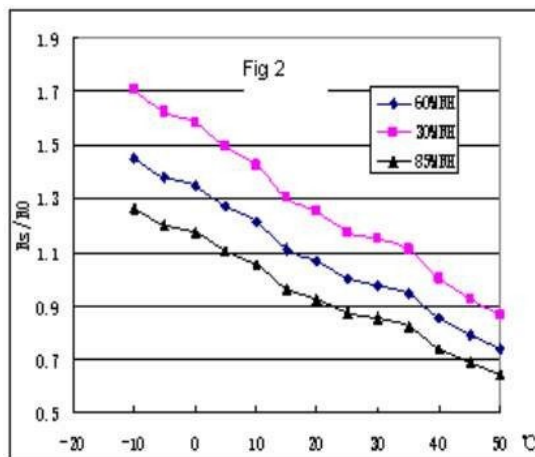


Fig. 17. Dependencia típica de la temperatura y la humedad

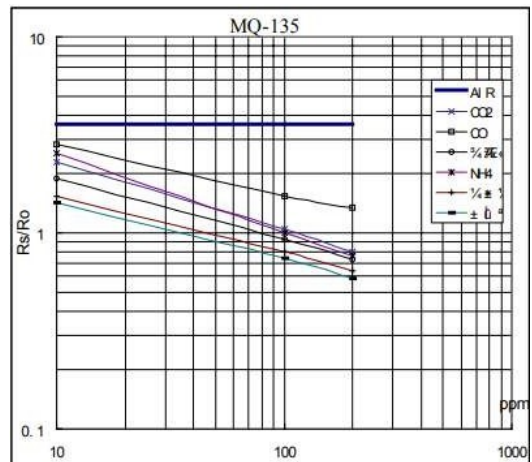


Fig. 20. Gráfica de Rs/Ro vs PPM medido con una carga de 20[kΩ]

### • Curvas características del sensor MQ-131

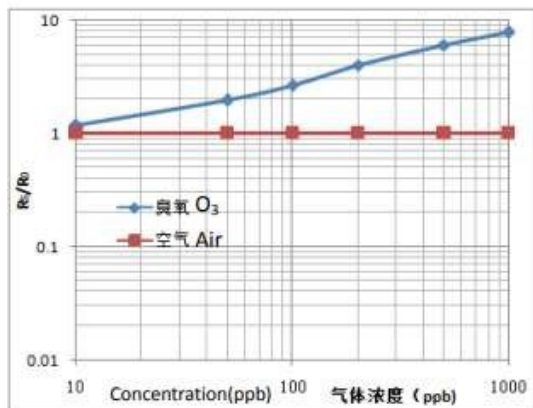


Fig. 18. Gráfica de Rs/Ro vs PPM medido con una carga de 10[kΩ]

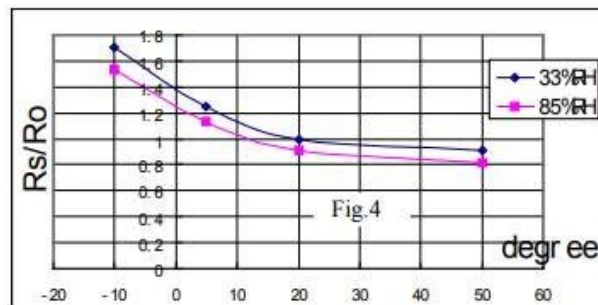


Fig. 21. Dependencia típica de la temperatura y la humedad

### • Curvas características del sensor MQ-136

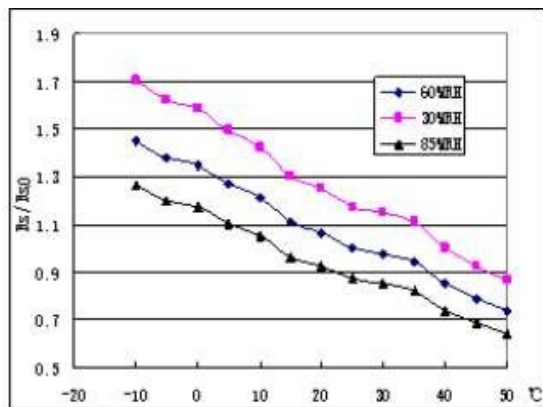


Fig. 19. Dependencia típica de la temperatura y la humedad

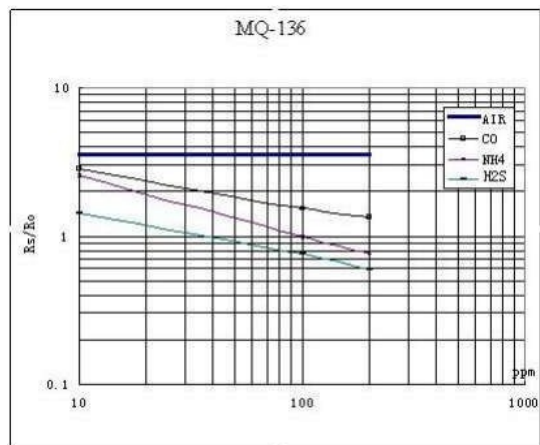


Fig. 22. Gráfica de Rs/Ro vs PPM medido con una carga de 20[kΩ]

### • Curvas características del sensor MQ-135

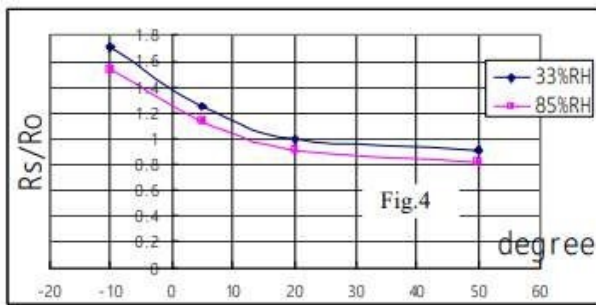


Fig. 23. Dependencia típica de la temperatura y la humedad

### • Curvas características del sensor MQ-137

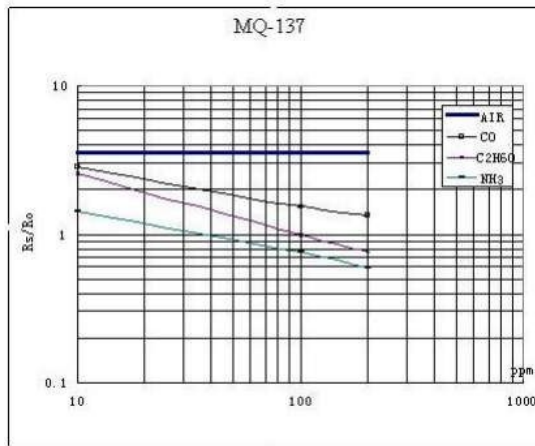


Fig. 24. Gráfica de Rs/Ro vs PPM medido con una carga de 47[kΩ]

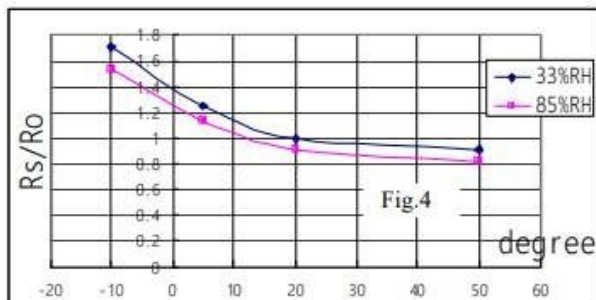


Fig. 25. Dependencia típica de la temperatura y la humedad

### • Curvas características del sensor MQ-138

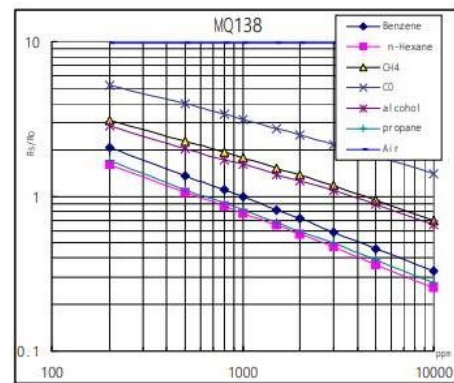


Fig. 26. Gráfica de Rs/Ro vs PPM medido con una carga de 20[kΩ]

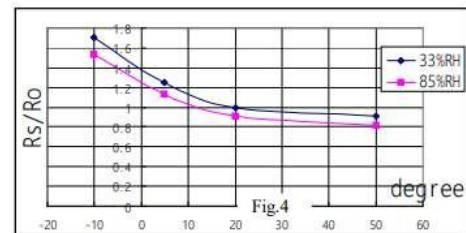


Fig. 27. Dependencia típica de la temperatura y la humedad

### • Curvas características del sensor MQ-214

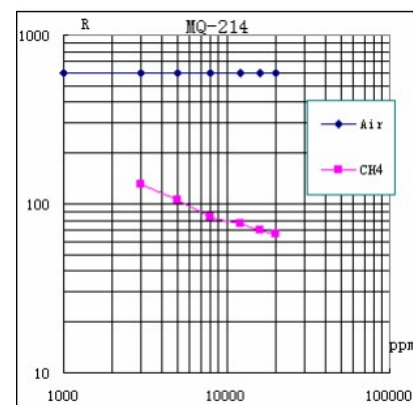


Fig. 28. Gráfica de Rs/Ro vs PPM medido con una carga de 50[kΩ]

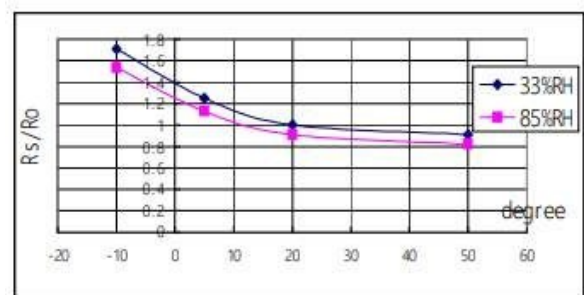


Fig. 29. Dependencia típica de la temperatura y la humedad



- Curvas características del sensor MQ-216

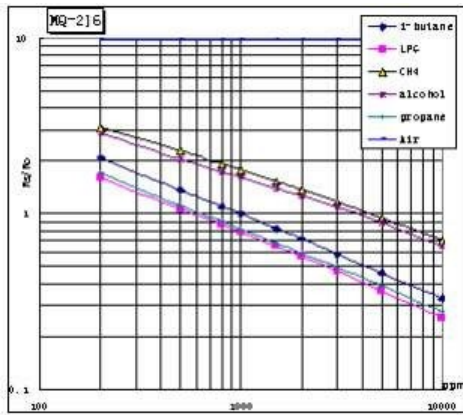


Fig. 30. Gráfica de  $R_s/R_o$  vs PPM medido con una carga de 50[Ω]

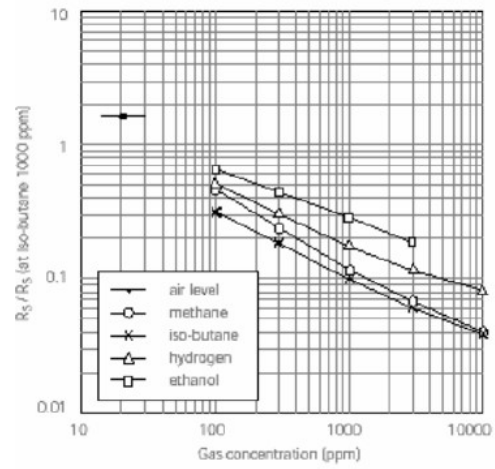


Fig. 33. Gráfica de  $R_s/R_o$  vs PPM

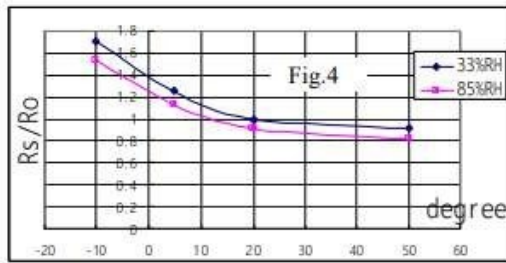


Fig. 31. Dependencia típica de la temperatura y la humedad

- Curvas características del sensor MQ-303A

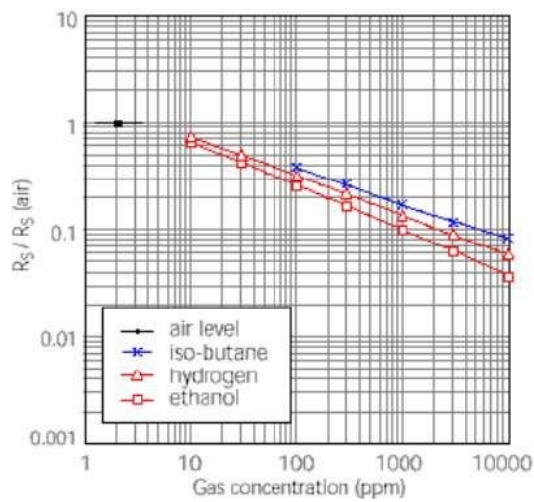


Fig. 32. Gráfica de  $R_s/R_o$  vs PPM

- Curvas características del sensor MQ-306A

- Curvas características del sensor MQ-307A

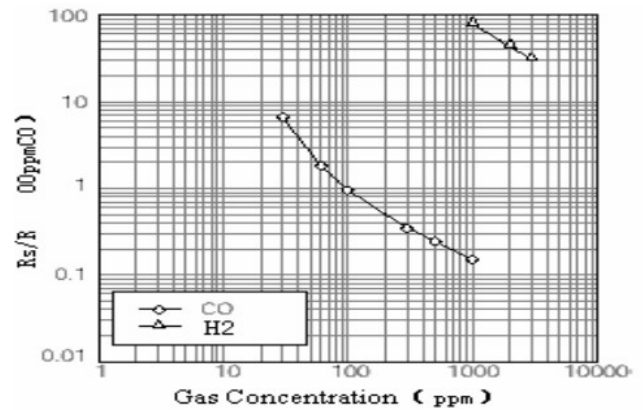


Fig. 34. Gráfica de  $R_s/R_o$  vs PPM

- Curvas características del sensor MQ-309A

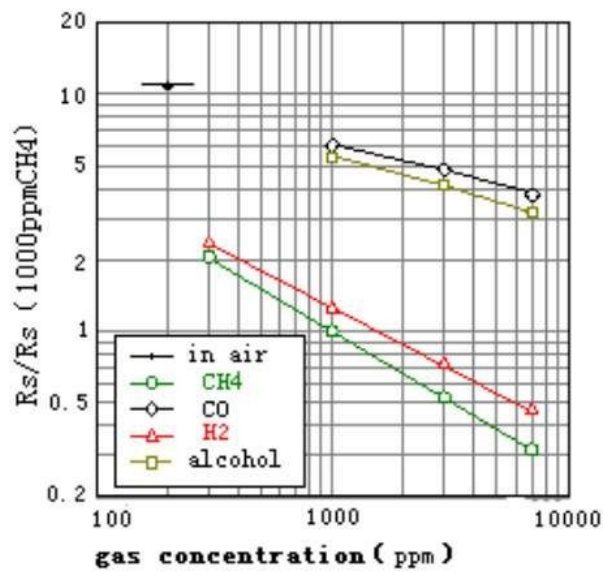


Fig. 35. Sensibilidad a alta señal para Metano

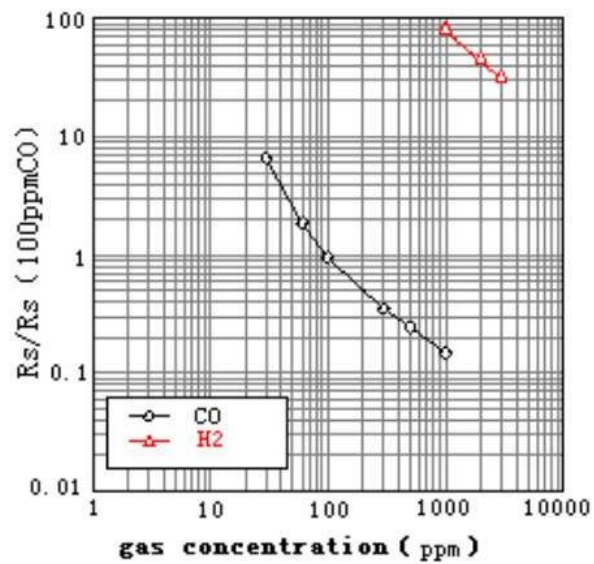


Fig. 36. Sensibilidad a baja señal para CO