



INSTITUTO POLITÉCNICO NACIONAL.  
Escuela Superior de Física y Matemáticas.



PROYECTO DE PRÁCTICA:  
Sensor de Alcohol.

Laboratorio de Física III.  
Grupo sección de Laboratorio: 3FM1-A.

Profesor: Hernani Tiago Yee Madeira.  
Alumno: Flores Rodríguez Jaziel David.

5 de Diciembre de 2017.

# Índice

<b>1. Resumen.</b>	<b>1</b>
<b>2. Objetivos.</b>	<b>1</b>
<b>3. Metodología.</b>	<b>2</b>
<b>4. Instrumentación.</b>	<b>3</b>
<b>5. Datos y Resultados.</b>	<b>4</b>
<b>6. Conclusiones.</b>	<b>5</b>
<b>7. Bibliografía.</b>	<b>5</b>

## **1. Resumen.**

En esta práctica se encuentra la existencia de una relación entre la resistencia de un material y su reacción sometido a los cambios de ambiente como lo es en este caso un gas (alcohol), se da una medida a esa relación y se explica de manera teórica como es que funciona. Se observa que hacer aproximaciones sin los instrumentos adecuados nos lleva a errores bastante grandes. Y se menciona la importancia de la calibración.

## **2. Objetivos.**

Estudiar la relación entre la resistencia variable de un material con respecto a un tipo de gas o varios y darle una medida y así diseñar un instrumento de medición sólo con principios básicos de programación en C y circuitos.

### 3. Metodología.

Los dispositivos eléctricos sensores (térmicos, de humedad, o de humo) se aprovechan de los cambios en las propiedades eléctricas de los materiales, como la resistencia. El sensor MQ-2 utiliza como material sensible  $\text{SnO}_2$  el cual reduce su conductividad en aire limpio. El sensor MQ-2 tiene alta sensibilidad al gas L.P., alcohol, al propano y al hidrógeno; también puede ser utilizado con metano y con humo. La resistencia del sensor  $R_s$  va de 2 k $\Omega$  a 20 k $\Omega$  y el rango del sensor va desde 300 a 10 000 ppm.

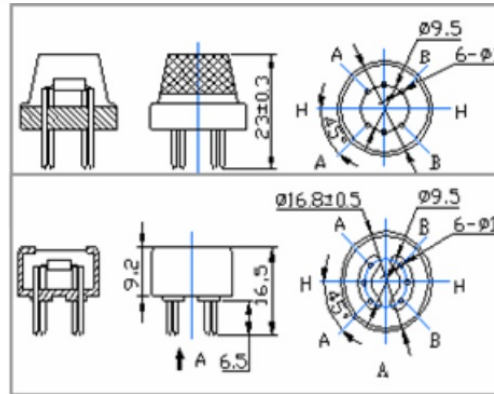


Figura 1: Configuración del sensor.

Un **microcontrolador** (abreviado  $\mu\text{C}$ , UC o MCU) es un circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria. Está compuesto de varios bloques funcionales, los cuales cumplen una tarea específica. Un microcontrolador incluye en su interior las tres principales unidades funcionales de una computadora: unidad central de procesamiento, memoria y periféricos de entrada/salida. Los PIC son una familia de microcontroladores tipo RISC fabricados por Microchip Technology Inc. y derivados del PIC1650, originalmente desarrollado por la división de microelectrónica de General Instrument. El nombre actual no es un acrónimo. En realidad, el nombre completo es PICmicro, aunque generalmente se utiliza como Peripheral Interface Controller (controlador de interfaz periférico). El PIC original se diseñó para ser usado con la nueva CPU de 16 bits CP16000. Siendo en general una buena CPU, ésta tenía malas prestaciones de entrada y salida, y el PIC de 8 bits se desarrolló en 1975 para mejorar el rendimiento del sistema quitando peso de entrada/salida a la CPU. El PIC utilizaba microcódigo simple almacenado en ROM para realizar estas tareas; y aunque el término no se usaba por aquel entonces, se trata de un diseño RISC que ejecuta una instrucción cada 4 ciclos del oscilador. En este caso Utilizaremos este tipo de familias de Microcontrolador, en específico el siguiente:

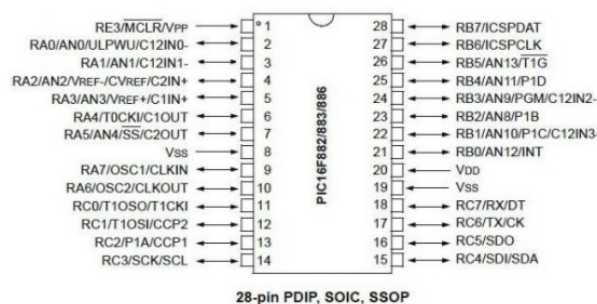


Figura 2: Microcontrolador pic 16f886

Debido a que el modulo tiene una cámara de calentamiento a donde tiene que ingresar o salir el gas, el tiempo de respuesta es lento, el sensor seguirá detectando los residuos de gas que se quedan dentro de la cámara de calentamiento hasta que estos desaparezcan. Tener en cuenta que todos los módulos son

sensibles a más de un gas, claro que en diferente proporción; pero si se trabaja en ambientes en donde hay diferentes tipos de gases no podríamos diferenciar entre ellos y podríamos tener una referencia equivocada si solo necesitamos leer un gas. Si en nuestra aplicación que estamos implementando necesitamos los valores en unidades correspondientes a la medición del gas, necesitamos escalar el valor leído, el problema de esto es que la relación entre la lectura analógica y el valor real no es lineal. Por lo que necesitamos estimar la curva que nos da el datasheet. Los siguientes pasos los trabajamos para el MQ-2, pero se aplica analogamente para los otros MQ. Por Ejemplo para el sensor MQ-2 según el datasheet la curva es la siguiente:

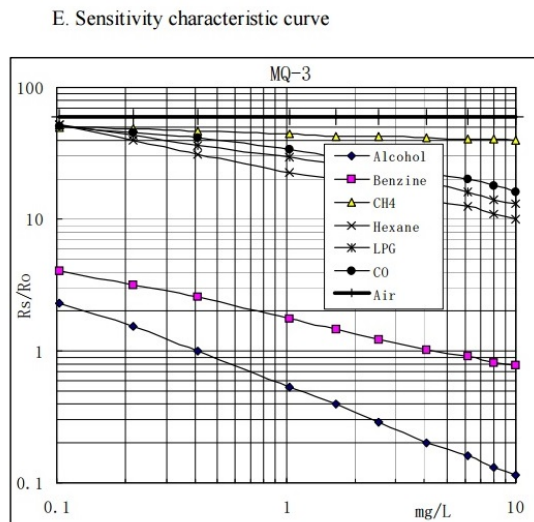


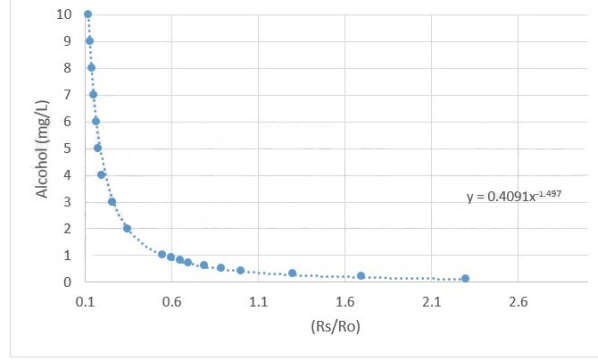
Fig. 3 shows the typical sensitivity characteristics of the MQ-3 for several gases. in their: Temp: 20°C, Humidity: 65%, O<sub>2</sub> concentration 21% RL=200k  $\Omega$   
 Ro: sensor resistance at 0.4mg/L of Alcohol in the clean air.  
 Rs: sensor resistance at various concentrations of gases.

## 4. Instrumentación.

- 1).- Fuente regulada de 5 V.
- 2).- Microcontrolador 16f886A.
- 3).- Potenciómetro.
- 4).- Resistencias de varios valores.
- 5).- Sensor de temperatura MQ-2

## 5. Datos y Resultados.

Debido a que nos da la curva y no la ecuación es necesario estimar y por regresión hallar la ecuación, en nuestro caso usaremos Excel, para eso ingresamos datos de la curva de Alcohol, la mayor cantidad de puntos que podamos, y graficamos en Excel. Agregamos línea de tendencia y escogemos ecuación potencial



**Ajuste de datos:** La ecuación que obtenemos es.

$$OH(R_s/R_o) = 4,091 \times 10^{-1} (R_s/R_o)^{-1,497}$$

Donde OH es el valor de concentración de alcohol,  $R_o$  es una constante que equivale al valor de la resistencia del sensor cuando se lo expone a una concentración de 0.4mg/L y  $R_s$  es la resistencia del sensor, el cual leemos desde el Microcontrolador. Nosotros en el ejemplo anterior calculamos el voltaje leído del sensor, para calcular el valor de  $R_s$  despejamos la ecuación del divisor de voltaje que forma el sensor con la resistencia de carga  $R_L$  que en la mayoría de módulos es de 1K.

$$V_{leido} = 5 \frac{1000}{R_s + 1000}$$

$$Entonces \quad R_s = 1000 \frac{5 - V}{V}$$

El valor de  $R_o$  se calcula en laboratorio,  $R_o$  es el valor de  $R_s$  cuando se usa una muestra de aire con 0.4mg/L. Y hay que hacer esto para cada sensor puesto que  $R_o$  es diferente en cada sensor. Otra forma de calibrar es usando concentraciones conocidas, para esto es necesario en el caso del sensor de alcohol usar un alcoholímetro para calibrarlo. Para esto no es necesario hacer la regresión de la curva que nos da el datasheet la cual es válida siempre y cuando se mida en las condiciones indicadas. De dicha curva solo tomaremos la forma que tiene e incluso  $R_o$  lo consideramos como constante. La ecuación sería de la siguiente forma.

$$OH = a(R_s)^b$$

**CUIDADO:** Para calcular las constantes a y b solo necesitamos tomar dos muestras, con nuestro sensor medimos el  $R_s$  de dichas muestras y con el alcoholímetro calculamos el valor correspondiente de concentración de alcohol para dichas muestras. Con esos dos puntos ingresamos a la ecuación y calculamos las dos constantes. Las dos formas anteriores son la forma correcta de calibrar nuestro sensor, hallando  $R_o$  con una muestra 0.4mg/L o usando un alcoholímetro para tomar muestras de referencia.

Pero para no dejar inconcluso nuestra práctica abusaremos de otro método para conseguir de forma aproximada el valor de  $R_o$  y así tener la ecuación resuelta. En el gráfico de la ecuación potencial observamos que el valor de  $R_s/R_o$  se acerca a 0.1 para valores superiores a la máxima concentración de alcohol que el sensor puede censar, en los puntos cercanos a este punto, la variación de  $R_s/R_o$  es mínima por lo que los errores que podamos tener acá para calcular  $R_o$  son pequeños. Entonces asumiendo que el sensor se satura con 10mg/L al cual según la gráfica le corresponde un  $R_s/R_o$  de 0.12, los puntos cercanos por mayor variación de alcohol que exista el  $R_s/R_o$  tendrá variaciones

mínimas. Para estar en este punto generamos un ambiente con bastante alcohol y si medimos con nuestro sensor obtendremos un voltaje de 3.02, que equivale a un  $R_s=655\Omega$ , entonces

$$R_s/R_o = 0,12 \quad \text{si} \quad R_s = 655\Omega \quad \text{entonces} \quad R_o = 5463\Omega$$

Con  $R_o$  calculado, ya tenemos una solución de la ecuación y con esto podemos obtener los valores aproximados de concentración de alcohol. Cualquiera sea el método de calcular la ecuación deberán de ingresarlo y modificar la ecuación en el programa, luego obtenemos la siguiente tabla de datos:

n	Rs	OH mg/L
1	820.28	6.99
2	813.83	7.07
3	807.42	7.16
4	804.23	7.16
5	801.06	7.2
6	801.06	7.24
7	801.06	7.24
8	801.06	7.24
8	801.06	7.24
8	797.89	7.29
8	797.89	7.29

Luego tenemos la siguiente gráfico:

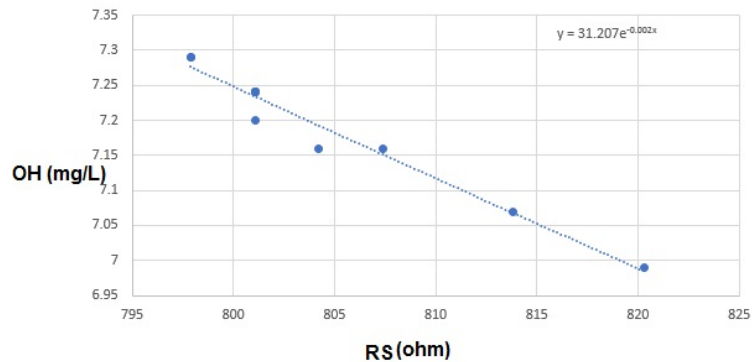


Figura 3: Gráfico con linea de tendencia.

**Discusión** Mencionar nuevamente que todos los sensores MQ como los MQ-2, MQ-3, MQ-7, MQ-135, etc tienen su propia curva y las cuales lo pueden encontrar en sus datasheet correspondientes. Y la forma correcta de calibrar nuestros MQ es usar un instrumento maestro para tomar muestras y con estos puntos calcular las constantes de la curva y no exactamente lo que se hizo aquí tomar valores aproximados porque puede que haya errores muy grandes.

## 6. Conclusiones.

Se comprendió de manera teórica los resultados y vimos que era imprescindible un aparato exterior para poder adquirir constantes de nuestro sensor, ya que en este caso no encontramos una ecuación muy aproximada como lo marcaba la hoja de datos. Se comprendió que calibrar los aparatos de medición requiere bastante sutileza y comprensión de conocimientos básicos de física, además claro de condiciones iniciales que se requieren.

## 7. Bibliografía.

- 1.-[https://es.wikipedia.org/wiki/Microcontrolador\\_PIC](https://es.wikipedia.org/wiki/Microcontrolador_PIC).
2. - <http://wiki.seeed.cc/Grove-Gasensor-MQ2/>

3. – <https://sites.google.com/site/labenriquesalgadoruiz/home/politecnico-1/fisica-iii>.
4. – Resnick/Halliday/Krane. *Fundamentos de Física. Volumen 2. Edición 6, extendida. CESA*

## ANEXO:

/\*

### INSTITUTO POLITÉCNICO NACIONAL ESCUELA SUPERIOR DE FÍSICA Y MATEMÁTICAS

//

Sensor digital de gases, alcohol, con MQ-2 y Display LCD de 16x2

MCU: PIC16F886 Osc: Interno a 4 MHz

Lenguaje de Programación: MikroC (dialecto de C orientado a microcontroladores)

Complejidad computacional del algoritmo:  $O = n$

Hardware

- PIC16F886
- LCD 16x2 con controlador Hitachi
- Fuente de CC de 5V
- Capacitor electrolítico de 100uF (filtrado de señales)
- Protoboard
- Sensor LM35
- Potenciómetro de 5KOhms

Software

- MikroC IDE
- PicKit2

Alumno: Flores Rodríguez Jaziel David

\*/

// Conexiones de la LCD

sbit LCD\_RS at RC4\_bit;

sbit LCD\_EN at RC5\_bit;

sbit LCD\_D4 at RC0\_bit;

sbit LCD\_D5 at RC1\_bit;

sbit LCD\_D6 at RC2\_bit;

sbit LCD\_D7 at RC3\_bit;

//Direccionamiento de los puertos (Entrada, Salida o Alta Impedancia)

sbit LCD\_RS\_Direction at TRISC4\_bit;

sbit LCD\_EN\_Direction at TRISC5\_bit;

sbit LCD\_D4\_Direction at TRISC0\_bit;

sbit LCD\_D5\_Direction at TRISC1\_bit;

sbit LCD\_D6\_Direction at TRISC2\_bit;

sbit LCD\_D7\_Direction at TRISC3\_bit;

//Declaración de variables globales

float temp = 0; // Valor obtenido del Analog/Digital Converter (ADC)

unsigned int datoADC = 0; // Variable que almacena lo obtenido en el ADC

char tempString[15]; // Almacena temperatura como cadena de caracteres

char txt1[] = "GAS: Alcohol";

char txt2[] = "Sensor de gas";

char txt3[] = "Flores Rodriguez Jaziel David.";

char txt4[] = "Hernani Tiago Yee Madeira";

//Declaración de funciones

void printStrings(); //Función tipo void (no regresa ningún dato)

void getTemp();

void main(){

// Configuración de bits y puertos

ANSEL = 0x01; // Establece bit RA0 como analógico



```

ANSELH = 0x00;      // Configura otros bits analógicos como digitales
TRISA0_bit = 1;     // Establece bit RA0 como entrada
C1ON_bit = 0;       // Deshabilita comparador 1
C2ON_bit = 0;       // Deshabilita comparador 2

//Iniciación de funciones externas
ADC_Init();         // Inicializa el Modulo ADC
Delay_us(100);      // Retardo para estabilización del ADC
Lcd_Init();         // Inicializa el LCD

//Llamado a función printStrings()
printStrings();

while(1){
    getTemp();
    Delay_ms(50);
}

void printStrings(){
    int i;
    Lcd_Cmd(_LCD_CLEAR);    // Comando para limpiar el LCD
    Lcd_Cmd(_LCD_CURSOR_OFF); // Comando para quitar el cursor
    Lcd_Out(1,3,"IPN - ESFM");//Imprime texto personalizado
    delay_ms(700);
    Lcd_Cmd(_LCD_CLEAR);
    Lcd_Out(1,2,"Alumno:");
    Lcd_Out(2,1, txt3);
    delay_ms(1000);
    Lcd_Cmd(_LCD_CLEAR);
    Lcd_Out(1,4,"Profesor:");//Imprime texto personalizado
    Lcd_Out(2,1, txt4);
    delay_ms(1000);
    Lcd_Cmd(_LCD_CLEAR);
    Lcd_Out(1,1,txt2);
    delay_ms(400);
    //Bucle de desplazamiento de texto en la LCD (para más de 16 caracteres)
    for(i = 3; i > 0; i--){
        Lcd_Cmd(_LCD_SHIFT_LEFT);
        delay_ms(500);
    }
    Lcd_Cmd(_LCD_CLEAR);
    Lcd_Out(1,1, txt1);    // Se imprime la cadena text LCD
    Lcd_Chr(2,9,223);      // Imprime el caracter "°" con código ASCII 223
    Lcd_Chr(2,14,"ppm");   // Imprime ppm
    Delay_ms(100);
}

void getTemp(){
    datoADC = ADC_Get_Sample(0);    // Obtenemos lectura del AN0
    delay_ms(600);                  // Periodo de actualización de datos
    temp = (float)(datoADC *(5.0/1023.0));
    FloatToStr(temp, tempString); //Se convierte de float a string para la LCD
    Lcd_Out(2,1, tempString);    //Se imprime el resultado
}

```