

# **PROYECTO ARDUINO:**

## **Sistema de monitorización de humos**



**Raúl Jiménez Suárez**

**T2, 49235802R**

**Tecnología de Computadores**

## ÍNDICE

<b>1. Introducción .....</b>	<b>3</b>
<b>2. Solución Propuesta .....</b>	<b>4</b>
<b>2.1. Amplificador Operacional .....</b>	<b>5</b>
<b>2.2. Transistor y Relé .....</b>	<b>6</b>
<b>2.3. Diodos led .....</b>	<b>7</b>
<b>2.4. Código .....</b>	<b>8</b>
<b>3. Conclusiones .....</b>	<b>9</b>
<b>4. Referencias .....</b>	<b>10</b>

# 1.Introducción

El objetivo de este proyecto va consistir en el diseño y montaje de un sistema de monitorización de humos con el uso de Arduino.

Para ello emplearemos un sesnsor de gas MQ2 el cual detecta gas combustible y humo en concentraciones de 300 a 10.000ppm, el cual mediante el uso de un amplificador operacional adaptaremos la señal para proporcionar un rango de 0 – 3,3V.

Una vez obtengamos la señal de nuestro sensor, esta será monitorizada por dos diodos led, uno verde que indica que la calidad del aire es buena(de 0 a 400ppm) y uno rojo que nos indica que es de mala calidad (más de 400ppm). Además una tercera salida activará un relé cuando la calidad del aire sea mala.

Los componentes a emplear serán los siguientes:

- Placa Arduino uno y cable USB para conexión al PC
- Protoboard para montaje y cables de conexión
- 1 Relé SPST
- 1 Transistor BD137
- 1 Diodo 1N4007
- Amplificador operacional TL081
- 1 diodo led verde y 1 diodo led rojo

## 2. Solución Propuesta

Como solución propuesta he dividido el proyecto en dos partes, una primera con su respectivo circuito para adaptar la señal del sensor con el amplificador y una segunda donde se desarrolla la parte del transistor junto al relé.

### 2.1. Amplificador operacional

Usaremos un amplificador operacional (TL081), para adaptar la señal de nuestro sensor que nos da un valor de 1,5 a 3,7V a un valor de 0 a 3,3V.

$$V0 = m * V_{sensor} + b \left\{ \begin{array}{l} 0 = m * 1.5 + b \rightarrow b = -1.5m \rightarrow b = -2,25 \\ 3.3 = m * 3.7 + b \rightarrow 3.3 = 3.7m - 1.5m \rightarrow m = 1,5 \end{array} \right\}$$

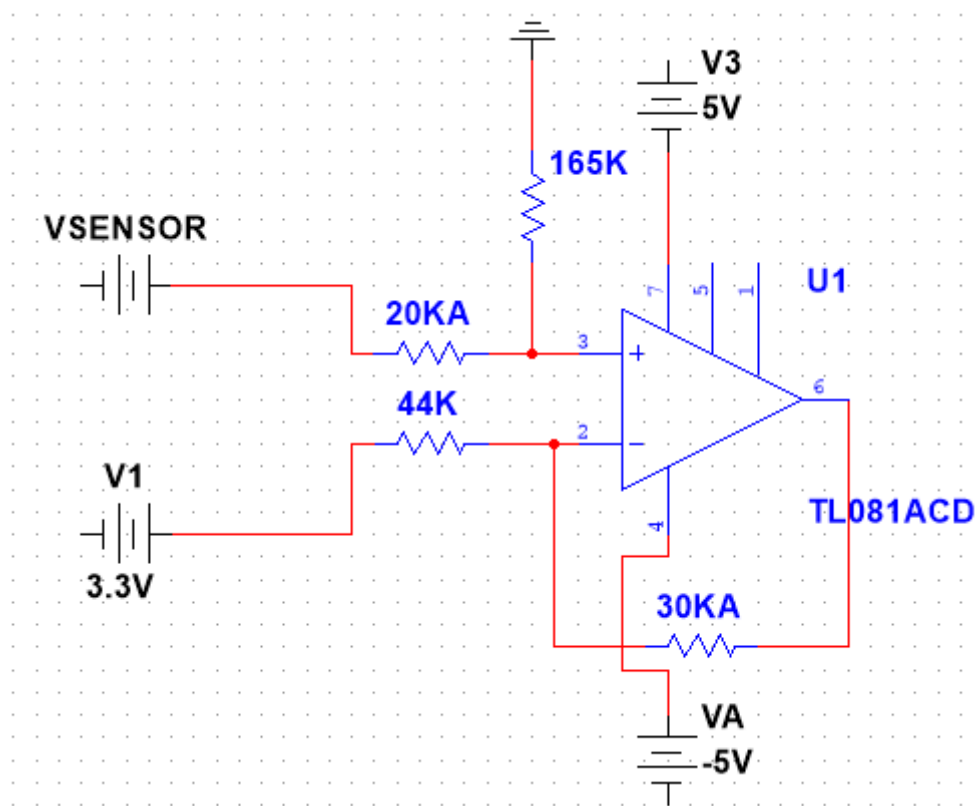
$$V0 = 1,5 * V_{sensor} - 2.25 \text{ (término independiente)}$$

$$V0 = 1,5 * V_{sensor} - \frac{2,25}{3,3} * 3,3$$

$$1 + K_0^- + K_1^- = K_0^+ + K_1^+ \left\{ \begin{array}{l} R0^+ = \frac{30}{0,181} = 165,7 \text{ Ohm} \\ R1^+ = \frac{30}{1,5} = 20 \text{ Ohm} \\ R1^- = \frac{30}{0,681} = 44 \text{ Ohm} \end{array} \right\}$$

[Para  $R_f$  igual a 30 tenemos los valores más estándares.]

$$1 + K_0^- + 0,681 = K_0^+ + 1,5 \{ K0^+ = 0,181 \}$$



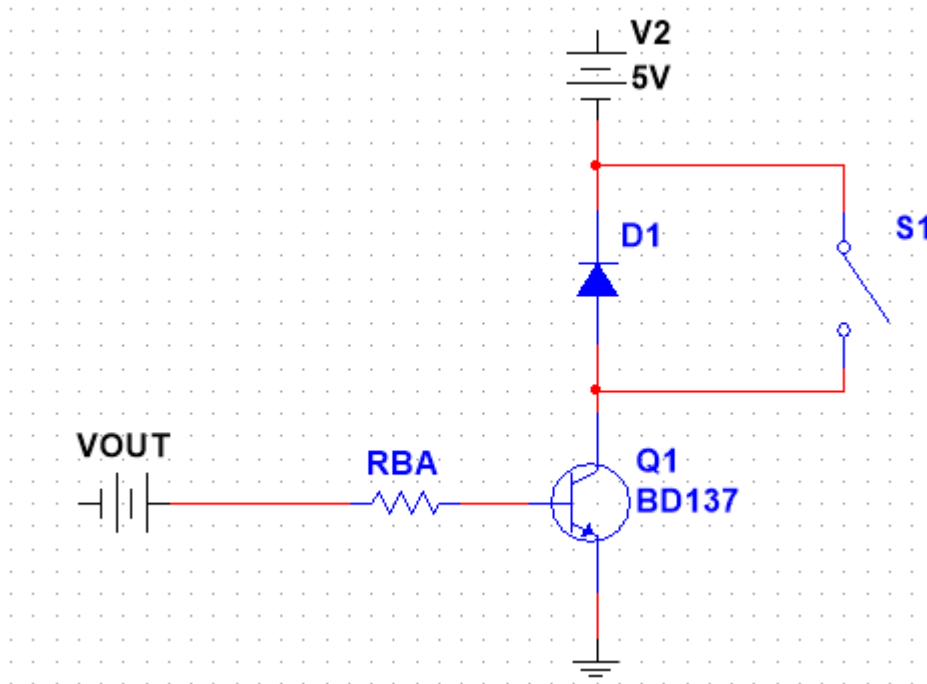
## 2.2. Transistor y Relé

En esta parte vamos a ver como conectar el relé, para el cual usaremos un transistor (BD137) de tipo nPn. Contamos que nuestro relé cuenta con una resistencia limitadora de 125Ohm y va en paralelo a un diodo (1N4007) el cual se encarga de regular la caída de corriente de la bobina para que se reduzca exponencialmente y no bruscamente para así no destruir el transistor cuando esté en corte, ambos en el colector del transistor. La base la conectaremos a uno de nuestros pines de Arduino y el emisor lo conectamos a tierra.

$$I_c = \frac{5-0,2}{125} = 38,4mA$$

$$I_c < Hfe * I_b \rightarrow 38,4 < 40 * \frac{2,4-0,8}{R_b}$$

$$R_b < \frac{40*1,6}{38,4} < 1,66kOhm \rightarrow 1,5kOhm$$



## 2.3. Diodos led

Para la monitorización del sensor usaremos dos diodos led los cuales conectaremos a las entradas digitales 11 y 13 de nuestra placa Arduino. Se activará el rojo cuando la calidad sea mala (valores mayores a 400ppm) y el led verde cuando no supere el umbral.

Seguidamente activaremos una alarma (con un diodo led naranja) que se activará cuando la calidad del aire sea mala, pero en este caso con el relé

## 2.4. Código

En este apartado final se mostrará el código implementado en el proyecto

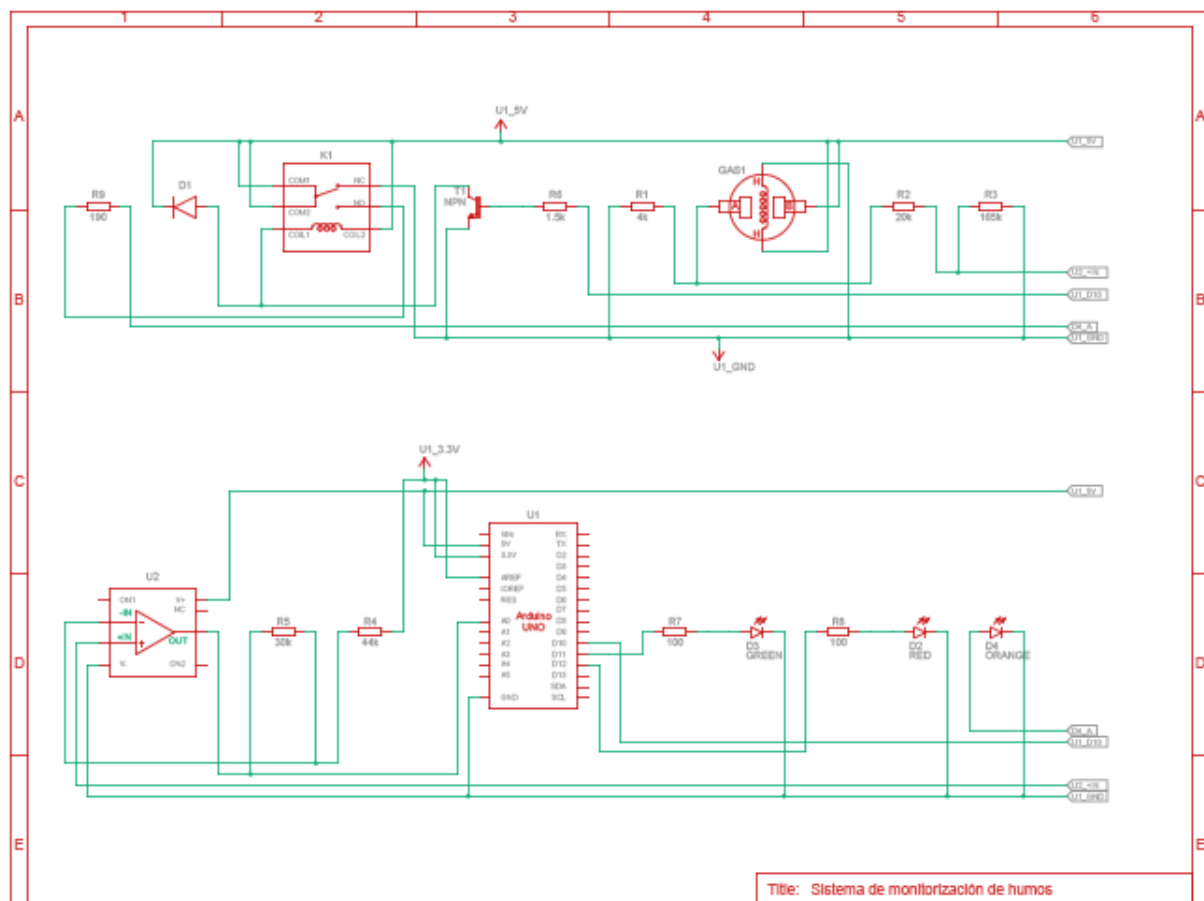
```
1. //Declaramos pines de los leds indicadores
2. const int LED_VERDE=11;
3. const int LED_ROJO=12;
4. volatile int UMBRAL=400;
5. //Declaramos el pin del relé
6. const int RELE=10;
7.
8. //Definimos pin del sensor MQ2 (analógico)
9. const int SENSOR=A0;
10.
11.     const int T=5000; //Tiempo entre cada linea del monitor
serie
12.     int nivel=0;
13.
14.     unsigned long t_inicial=0;
15.
16.     void setup(){
17.         Serial.begin(9600);
18.         pinMode(11, OUTPUT);
19.         pinMode(12, OUTPUT);
20.         pinMode(13, OUTPUT);
21.         analogReference(EXTERNAL);
22.     }
23.
24.     void loop(){
25.         int t_inicial=millis();
26.         do{
27.             nivel=analogRead(SENSOR);
28.             Serial.println(nivel);
29.             if(nivel>=1 && nivel <=UMBRAL){
30.                 UMBRAL=400;
31.                 digitalWrite(12, LOW);
32.                 digitalWrite(11, HIGH);
33.                 digitalWrite(10, LOW);
34.             }
35.             if(nivel>=UMBRAL){
36.                 UMBRAL=UMBRAL-20;
37.                 digitalWrite(11, LOW);
38.                 digitalWrite(12, HIGH);
39.                 digitalWrite(10, HIGH);
40.             }
41.             delay(5000);
42.         }while(millis()<t_inicial+T);
43.     }
```



### 3. Conclusiones

Como hemos podido observar anteriormente mediante la división en circuitos y el cálculo de todas nuestras resistencias hemos podido desarrollar nuestro proyecto con los cálculos más precisos y el resultado ha sido totalmente funcional.

A continuación se adjunta un esquema electrónico final:



## 4. Referencias

Documentación y archivos aportados por la Universidad de Huelva