

PRÁCTICA 7. Control de la luminosidad de un LED (función analogRead)

Materiales. 1 led rojo, 1 potenciómetro 10 k Ω , 1 resistencia de 220 Ω (rojo-rojo-marrón).

Explicación. Se pretende controlar la luminosidad de un led mediante un potenciómetro usando Arduino. Si bien esto se puede hacer sin Arduino, al incorporarlo vamos a poder saber qué voltaje le llega al led en cada momento. La idea es que vamos a leer el voltaje de la pata central del potenciómetro, incluso podemos sacar por pantalla dicho voltaje, y vamos a darle ese voltaje al led.

Como la luminosidad del led va a ser variable, conectaremos su ánodo a un pin digital tipo PWM~, por ejemplo el pin 11. Recordemos que si queremos escribir un voltaje en el pin PWM~ 11, primero lo declaramos como pin de salida en el setup() y después escribiremos:

```
analogWrite(11,brillo);
```

donde la variable entera brillo puede valer entre 0 y 255.

Ya sabemos que los pines analógicos son A0, A1, ..., A5 y que todos los pines analógicos son de lectura, por lo que no hace falta declararlos en el setup(). Lo que vamos a aprender ahora es cómo leer el voltaje de un pin analógico. Para ello usaremos la función analogRead. La sentencia:

```
analogRead(A0);
```

lee el voltaje del pin analógico A0 con la siguiente salvedad; el valor de `analogRead(A0)` es 0 si A0 está a 0 V y es 1023 si A0 está a 5 V. Si A0 está a 2,5 voltios, entonces valdrá 512,5. Por tanto, `analogRead(A0)`; nos va a dar un valor entre 0 y 1023.

Así, si queremos pasar el voltaje del pin analógico (de lectura) A0 al pin digital PWM~ (de escritura) 11 de manera sencilla, escribiremos:

```
brillo=analogRead(A0)/4;
```

```
analogWrite(11,brillo);
```

pues 255 es muy parecido a 1023/4.

Como la luminosidad del led se controla mediante el potenciómetro, deberemos saber el voltaje de la pata central del potenciómetro. Por tanto, uno de los extremos del potenciómetro irá a 5 V, el otro extremo irá a 0 V (da igual cual) y la pata central irá conectada a un pin analógico, por ejemplo el pin A0, para poder leer su voltaje.

Para poder usar la pantalla del serial, en el void setup() debemos escribir la sentencia:

```
Serial.begin(9600);
```

Aquí, el 9600 significa que la velocidad de transferencia es de 9600 baudios, esto es, 9600 bit/s, que es la velocidad adecuada entre Arduino y el serial o monitor serie.

Para ver por el serial el valor de la variable brillo seguido de un salto de línea escribiremos:

```
Serial.println(brillo);
```

Si no queremos el salto de línea escribiremos:

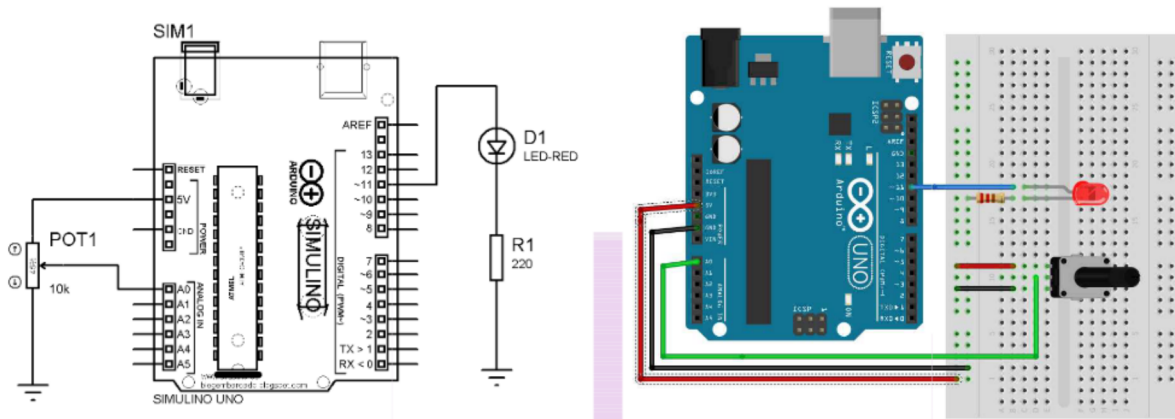
```
Serial.print(brillo);
```

Si queremos que en el serial ponga: "El valor de la variable brillo es: " seguido del valor de la variable brillo y luego un salto de línea, escribiremos:

```
Serial.print("El valor de la variable brillo es: ");
```

```
Serial.println(brillo);
```

Esquemas eléctricos



Sketch

```

const int led = 11; //el ánodo del led al pin 11
const int pot = A0; //el terminal medio del potenciómetro al pin A0
int brillo; // nos dará el brillo del led (entre 0 y 255)

void setup() {
  Serial.begin(9600); //esta sentencia es necesaria para usar la pantalla
  pinMode(11, OUTPUT); //el pin al que conecto el ánodo del led es de salida
  //no es necesario declarar que el pin A0 es de entrada...
  //pues los pines analógicos son de entrada automáticamente
}

void loop() {
  brillo=analogRead(pot)/4; //convertimos el valor de lectura del potenc...
                          //(entre 0 y 1023) en el valor de escritura analógica...
                          //en el led (entre 0 y 255) y lo guardamos en brillo
  analogWrite(led,brillo); //encendemos el led con la luminosidad brillo
  Serial.println(brillo); //sacamos por la pantalla el valor de brillo...
  delay(100); //cada 0,1 s
}

```

Ampliación. ¿Cómo harías para que el led no tenga luminosidad variable sino que en el primer tercio del rango del potenciómetro el led esté apagado, en el segundo encendido y en el tercero otra vez apagado?