

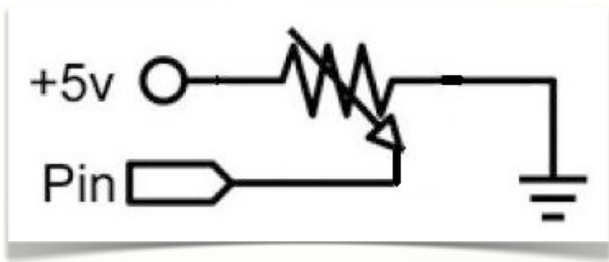
# Intensidad LED

## Montaje 5: Control de la intensidad de iluminación de un LED

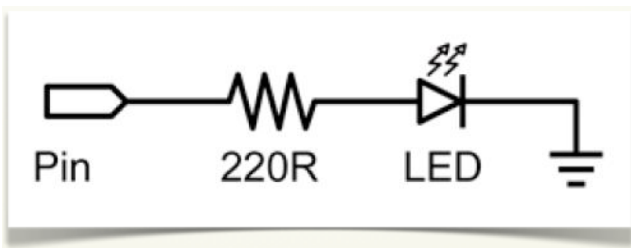
Como ejemplo práctico de la señal PWM vamos a realizar un control de iluminación sobre un diodo led.

### Montaje 5 Sin EDUBASICA

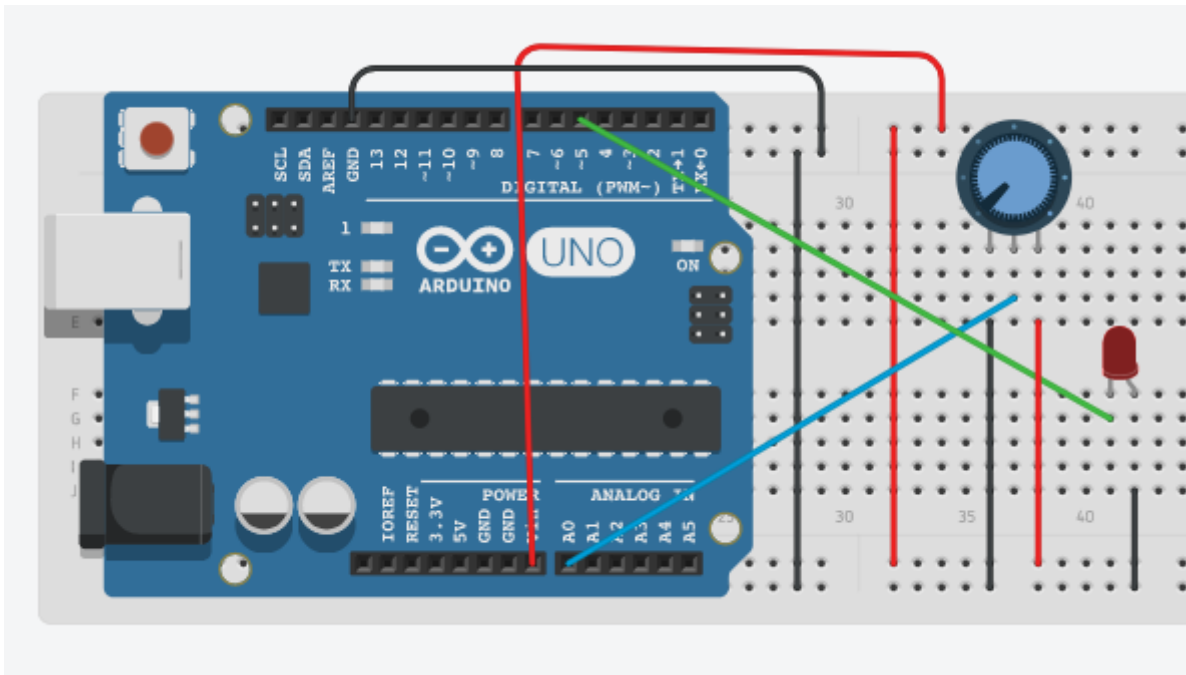
En una protoboard montamos el circuito formado por el **potenciómetro** conectado a la entrada analógica A0.



y también montamos el circuito conectado al pin digital D3, al D4 o al D5 como quieras, utilizado como salida PWM, de esta manera nos va a permitir variar la luminosidad del LED.

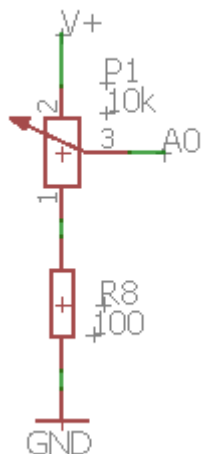


Aquí en la ilustración puedes ver las conexiones del potenciómetro al A0 y el LED al D5

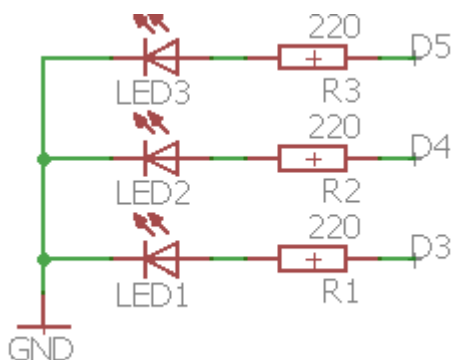


## Montaje 5 CON EDUBÁSICA

En este caso ya tiene integrado un potenciómetro conectado a la entrada A0.



Y vamos a utilizar de salida el diodo verde conectado a D3 y a una resistencia ya integrado en EDUBÁSICA (podemos usar también D4 y D5) :



No tienes que montar nada, la Edubasica lo tiene ya montado.

## **Un vídeo de cómo queda.**

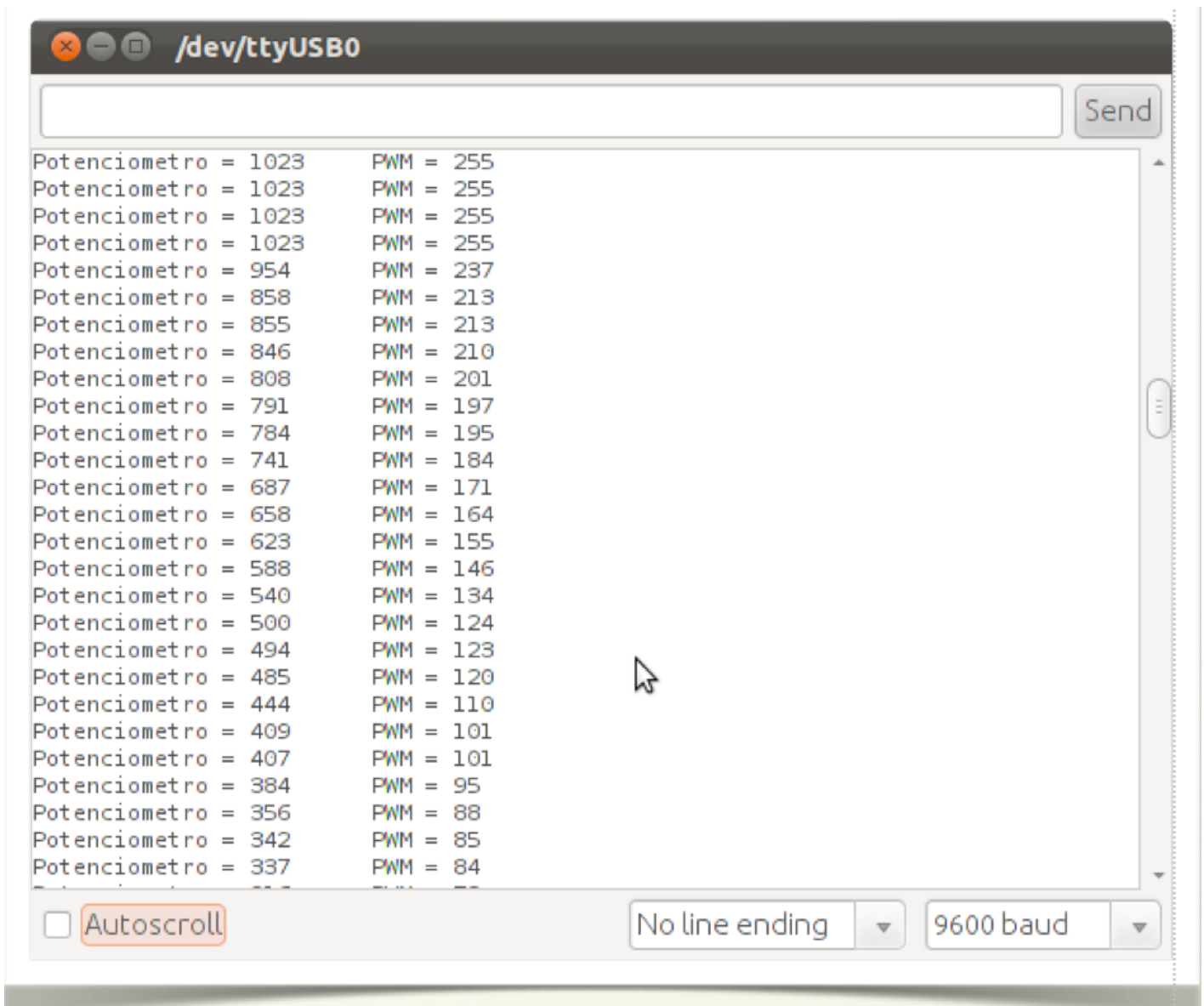
Vamos a ver una pequeña demostración:

[https://www.youtube.com/embed/KHeNjn\\_4Pcg](https://www.youtube.com/embed/KHeNjn_4Pcg)

Cargamos el programa ejemplo, en la placa Arduino y teniendo acoplada la placa EduBásica o montados los circuitos en una placa protoboard, podemos ver su funcionamiento.

Para ver su funcionamiento activamos el monitor serie del IDE de programación y variamos el potenciómetro. El resultado es una variación de luminosidad y variación de valores en el monitor serie.

El siguiente gráfico es una pantalla del monitor serie con los valores leídos y el valor aplicado a la señal PWM para variar la luminosidad.



## Montaje 5 Programa. Instrucciones `analogRead` `analogWrite` y `map`

El valor del potenciómetro lo leerá la instrucción **`analogRead(pin)`** que lee valores analógicos cada  $100\mu\text{s}$  y los convierte por el conversor analógico digital en valores de 10 bits por lo tanto de 0 a 1024. Los pines para leer sólo pueden ser los A0 al A5. ([ver+](#))

Utilizaremos la instrucción **`analogWrite(pin,value)`** para crear la onda PWM en el pin correspondiente, que sólo puede ser en el Arduino Uno el 3, 5, 6, 9, 10, 11. El valor `value` es un número entre 0 y 255 y es simulado con una onda cuadrada de 490Hz o 980Hz (depende del pin, [ver+](#)).

Como uno lee 0 a 1024 y el otro necesita valores de 0 a 255 necesitamos un traductor o mapeo con la instrucción **`map(value, fromLow, fromHigh, toLow, toHigh)`** esta instrucción la colocaremos, como suele estar los intérpretes en medio, o sea, entre la instrucción `analogRead` y la `analogWrite`

<https://create.arduino.cc/editor/javierquintana/8bee8469-df52-4b88-a406-9f54fe3e20f8/preview>

<https://create.arduino.cc/editor/javierquintana/8bee8469-df52-4b88-a406-9f54fe3e20f8/preview?embed>

La regulación del potenciómetro provocará una variación de voltaje en el pin de entrada analógico 0 de Arduino. Se realizará una conversión analógica-digital en el que los valores de tensión analógicos entre 0 y 5 V se transforma a un rango discreto de valores de 0 a 1023. Para modificar la intensidad del led D3 rojo le se enviará una señal pseudoanalógica PWM utilizando la salida 5 digital de Arduino. Para ello se enviará un valor de 0 a 255 que marcará el ciclo de trabajo de la onda cuadrada PWM. Previamente habrá que realizar un mapeo (instrucción map) para asignar valores desde el intervalo [0, 1023] al [0, 255].

En este vídeo se ha modificado el mensaje que sale por el puerto serie a

```
Serial.print("valor analógico leído=");  
Serial.println(analogRead(A0));
```

<https://www.youtube.com/embed/y4OAnbQkR-c>

---

Revision #3

Created 1 February 2022 11:20:25 by Equipo CATEDU

Updated 23 December 2023 10:17:44 by Javier Quintana