Project 1.2 Breathing LED

En este código se va a utilizar la intensidad PWM para dar la sensación de subida y bajada gradual

https://www.youtube.com/embed/VzL1NKdFmjc

- Solución con bloques https://www.steamakersblocks.com/web/project/2520329
- Solución con código https://docs.keyestudio.com/projects/KS5009/en/latest/docs/Python/Python.html#project-1-2-breathing-led

¿Qué son las salidas PWM?

Arduino, ESP32, Micro:bit, PicoW... tienen **entradas** analógicas y digitales. Pero **salidas sólo digitales.**

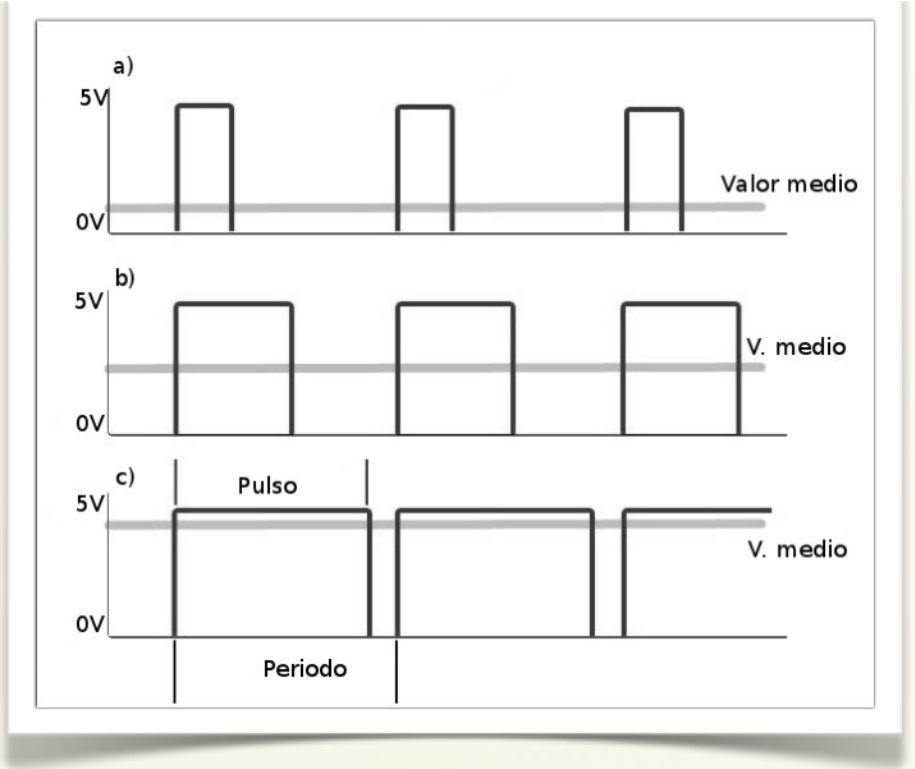
Para **simular** una salida **analógica** entre 0V y 5V se utilizan señales digitales PWM. En Arduino sólo tiene 6 salidas pseudo-analógicas. En los pines digitales 3, 5, 6, 8, 10 y 11 son PWM

¿Qué es eso de PWM? La señal PWM *(Pulse Width Modulation, Modulación de Ancho de Pulso*) es una señal que utiliza el microcontrolador para generar una señal continua sobre el proceso a controlar. Por ejemplo, la variación de la intensidad luminosa de un led, el control de velocidad de un motor de corriente continua,...

Para que un dispositivo digital, microcontrolador de la placa Arduino, genere una señal continua lo que hace es emitir una señal cuadrada con pulsos de frecuencia constante y tensión de 5V. A continuación, variando la duración activa del pulso (ciclo de trabajo) se obtiene a la salida una señal continua variable desde 0V a 5V.

Veamos gráficamente la señal PWM:

Copyright 2026



Los pines digitales de la placa Arduino que se utilizan como salida de señal PWM generan una señal cuadrada de frecuencia constante (490Hz), sobre esta señal periódica por programación podemos variar la duración del pulso como vemos en estos 3 casos:

- La duración del pulso es pequeña y la salida va a tener un valor medio de tensión bajo, próximo a 0V.
- La duración del pulso es casi la mitad del período de la señal, por tanto, la salida va a tener un valor medio de tensión próximo a 2,5V.
- La duración del pulso se aproxima al tiempo del período y el valor medio de tensión de salida se aproxima a 5V.

Ejemplo en código ArduinoIDE y Arduino
Para ejecutar una señal PWM, es simplemente **analogWrite(analogOutPin, outputValor);** donde analogOutPin es el número del Pin PWM, acuérdate que sólo puede ser uno de estos 6 : **3, 5, 6, 8, 10 y 11 y** outpuValor es el valor de la señal PWM pero **ojo desde 0 a 255** es decir si quieres el valor de 0V tienes que poner 0, si quieres el valor de 5V tienes que poner 255 y si quieres poner un valor medio, haz una regla de tres, por ejemplo 2.5V tienes que poner 255/2=127 o 128 da igual
Otro ejemplo en Python con Micro:bit
pin16.write_analog(brillo) donde brillo puede ir de 0 a 255

Copyright 2026

Revision #2
Created 2025-10-22 11:04:25 CEST by Juanjo Mejías
Updated 2025-10-22 11:29:45 CEST by Juanjo Mejías

Copyright 2026