LIBRARY · REFERENCE GUIDE

# PWM: A FAKE ANALOG OUTPUT

*How switching a digital pin on and off fast, with a controllable duty cycle, averages to an in-between level. The way one pin dims an LED or sets a motor's speed.*

ONE THOUSAND DRONES ENGINEERING TEAM · VERIFIED 2026-07

A microcontroller pin is digital: it can only be fully on or fully off. But switch it on and off fast, with a controllable on-fraction, and it averages to a level in between. That is PWM, for pulse-width modulation, and it is how one digital pin dims an LED or sets a motor's speed without any analog output hardware.

## DUTY CYCLE: THE ON-FRACTION

The duty cycle is the fraction of each cycle the pin spends high. At 25 percent duty the output averages a quarter of the supply; at 75 percent, three-quarters. Sweep the duty cycle and the average voltage follows it smoothly, which is the whole trick.

$$V_{avg} = D \times V_{supply}$$

## FREQUENCY: FAST ENOUGH TO BLUR

The switching frequency has to be fast enough that whatever you are driving cannot follow the individual pulses. Your eye blurs a fast-blinking LED into a steady brightness; a motor's inertia smooths the pulses into a steady pull. Too slow, and the LED visibly flickers or the motor whines at the switching rate. Fast enough, and only the average shows.

## WHAT PWM DRIVES

PWM sets LED brightness, motor speed, a servo's position (through a specific pulse width), a buzzer's tone, and, once you smooth it with an RC filter, a rough analog voltage. On the ESP32 the `LEDC` peripheral generates PWM in hardware on any output pin, so your code just sets a duty cycle and the hardware toggles the pin at the frequency you chose, with no CPU effort per pulse.


- Espressif. ESP-IDF Programming Guide: LED Control (LEDC) PWM peripheral. `docs.espressif.com`
- SparkFun. Pulse Width Modulation tutorial (duty cycle and average level). `learn.sparkfun.com`

**DEEP DIVE · DUTY-CYCLE RESOLUTION, AND ITS TRADE WITH FREQUENCY**

The duty cycle is not truly continuous: it is set as a count out of two-to-the-bits, an ADC in reverse. A higher PWM resolution gives finer brightness or speed steps, but on the ESP32 the resolution and the frequency both come from dividing one source clock, so they trade against each other. Push the frequency very high and you are left with fewer usable duty-cycle bits; drop it and you get more. For LED dimming a few kilohertz with 10 or more bits is plenty; a fast switching converter would flip the balance the other way.

THREE DUTY CYCLES AND THE AVERAGE EACH ONE PRODUCES: THE ON-FRACTION SETS THE LEVEL.

The fundamentals guide on reactive parts and filtering shows the RC low-pass that averages a PWM signal into a real, steady voltage, if you want the analog output rather than just the blinking.

---

## CHECKPOINT

**1. What sets a PWM output's average level?**

**a. Its duty cycle, the on-fraction of each period**

b. The height of each pulse above the supply

c. The switching frequency alone

ANSWER · A

*The average tracks the duty cycle. The pulses only ever reach the supply voltage, and the frequency just has to be fast enough that the load sees the average.*

**2. Why must the PWM frequency be fast enough?**

a. To save battery

b. To use fewer pins

**c. So the load cannot follow the individual pulses and instead sees the average**

ANSWER · C

*If the load can follow the pulses, you get flicker or whine instead of a smooth in-between level.*

**3. A hobby servo's position is set by what kind of PWM signal?**

a. A specific frequency

**b. A specific pulse width**

c. A specific voltage

ANSWER · B

*Servos read the width of the high pulse, roughly 1 to 2 ms. The frequency and voltage stay fixed; the pulse width is what carries the position.*

- Prerequisite: reactive parts and filtering
- Next: boot and strapping pins