VLADIMIR TRUJILLO ARIAS

Sistemas Digitales

Modulación por ancho de pulso

PWM

(Pulse Width Modulation)

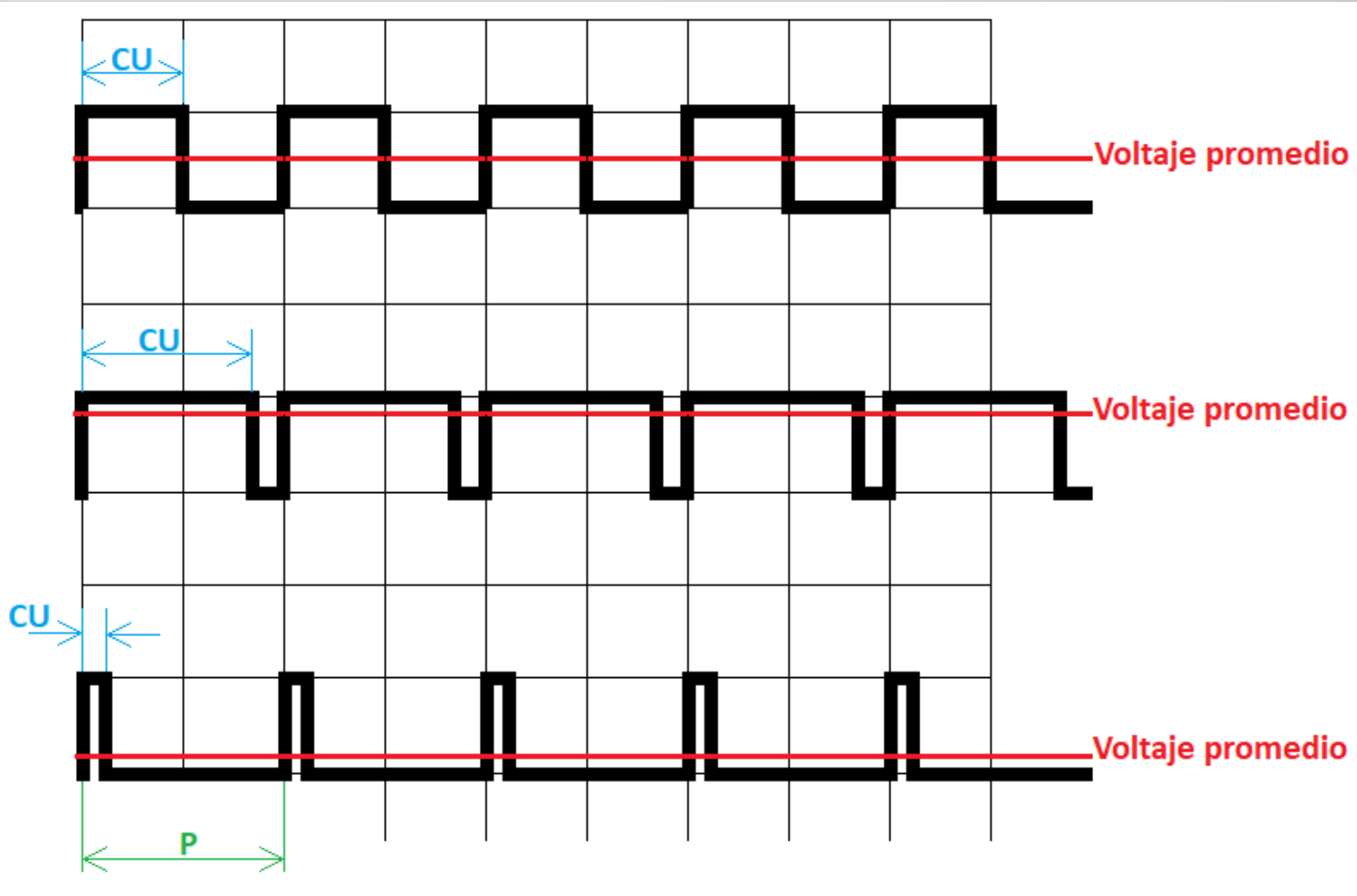
Modulación por Ancho de pulso

PMW

VLADIMIR TRUJILLO ARIAS

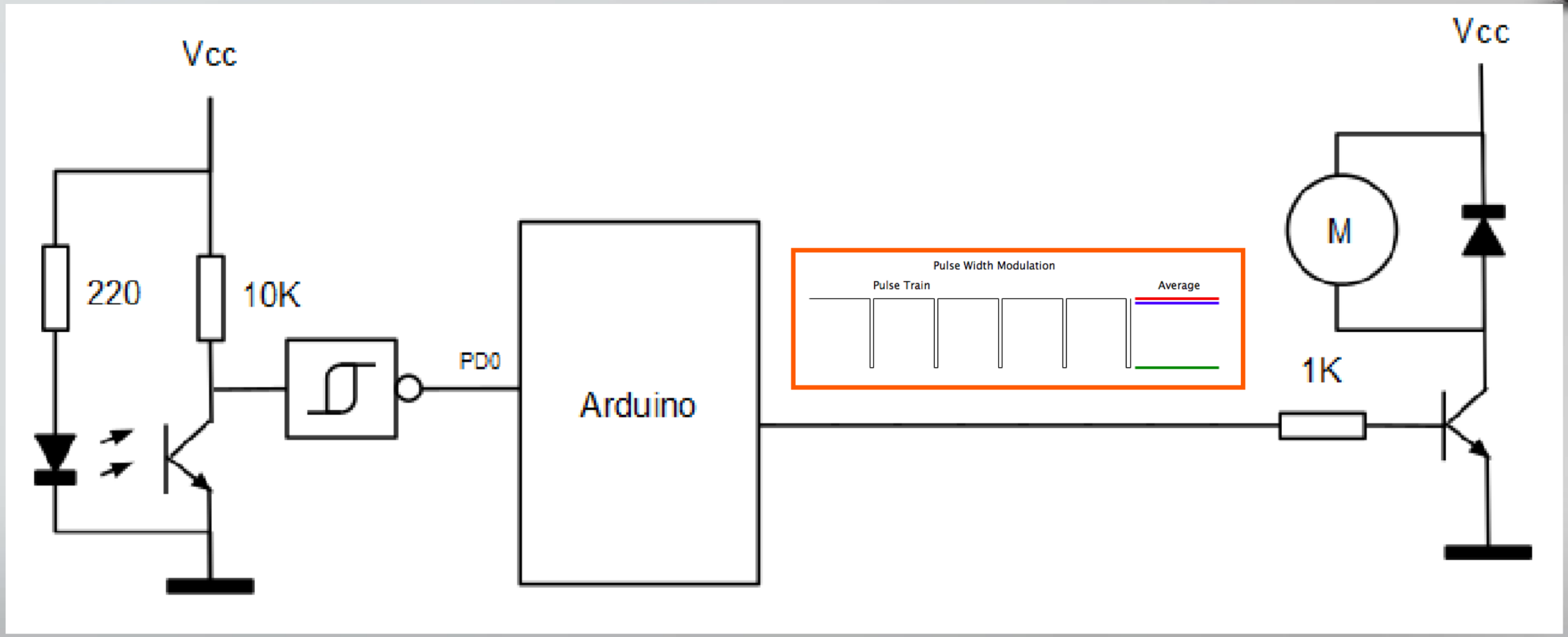
CU = Ciclo útil
P = Periodo
F = frecuencia
F = 1/P

$$Vp = \frac{CU * Vmax}{P}$$



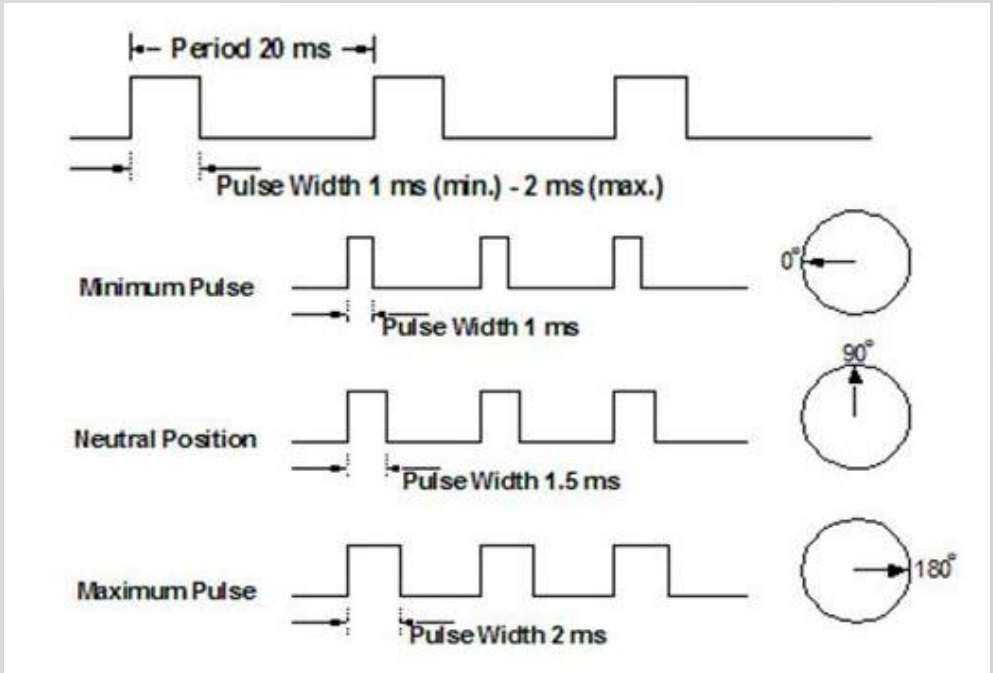
PWM Usos

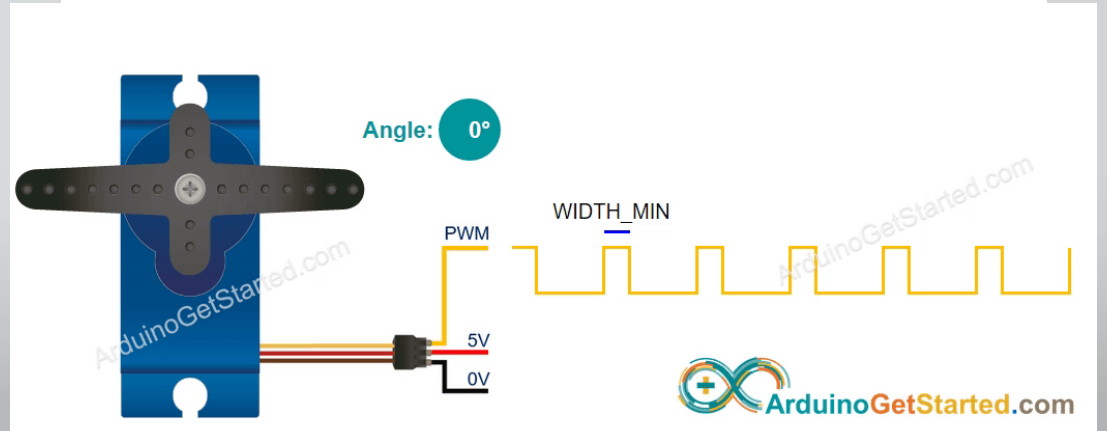
- Control de velocidad de motores DC.

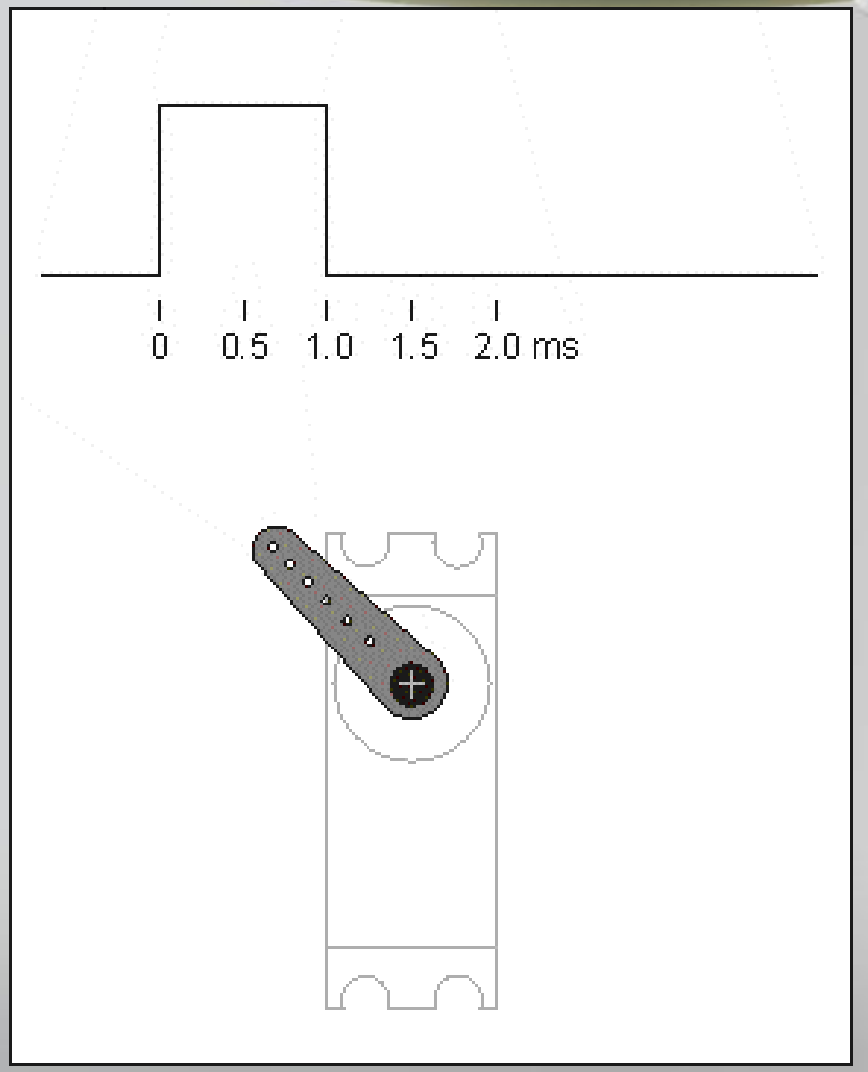


PWM Usos

- Control de posición de servomotores



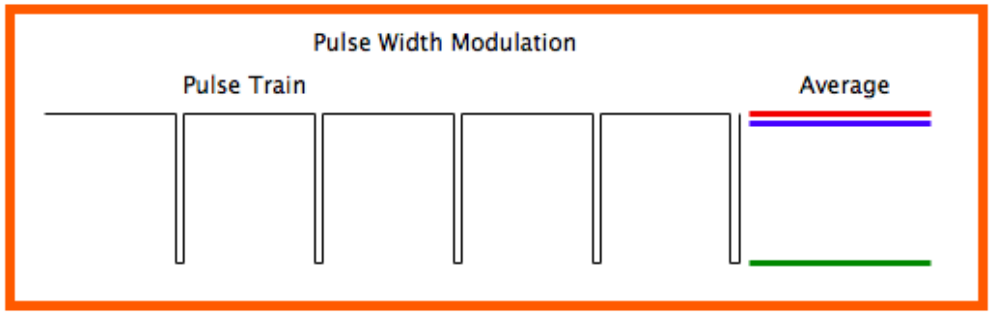


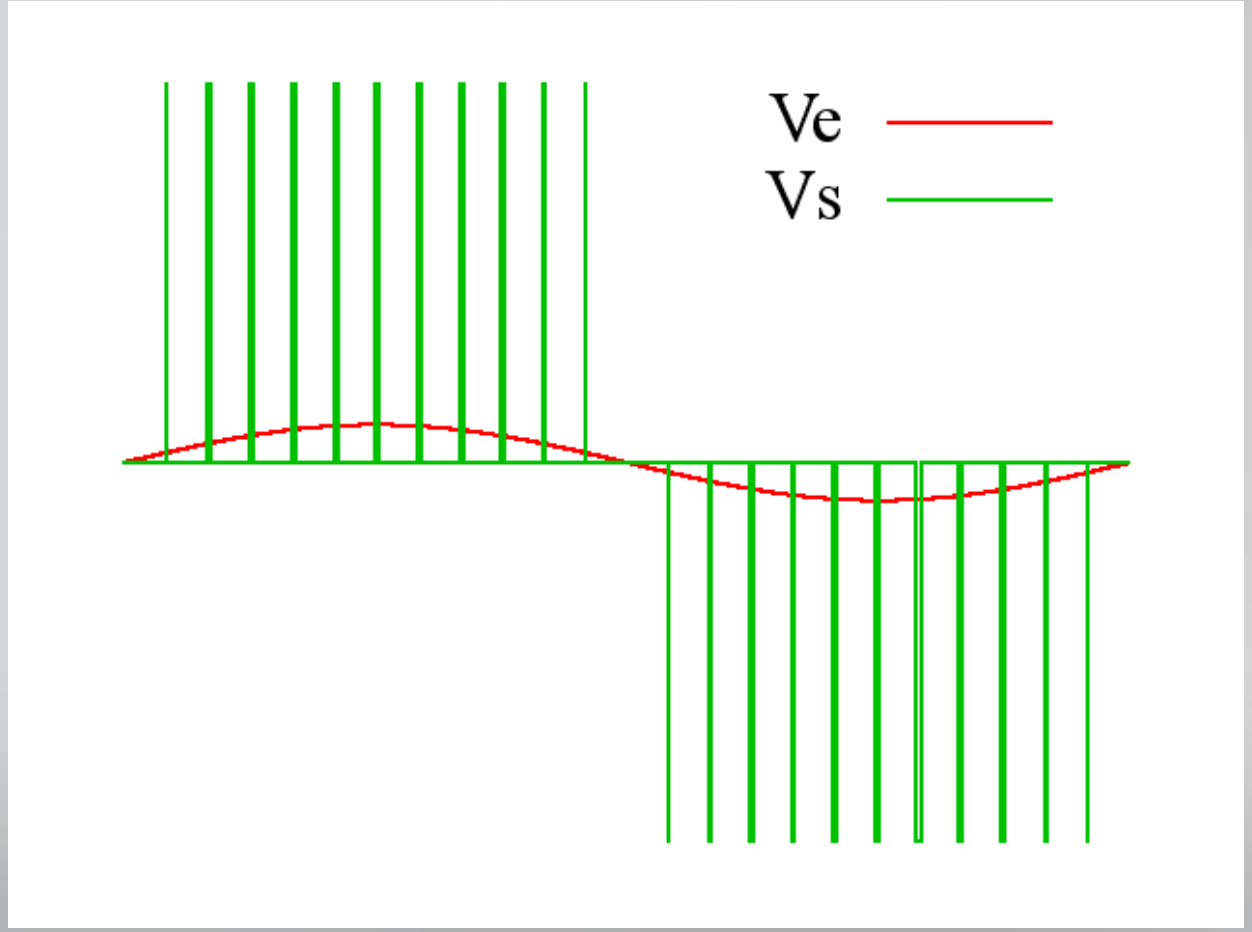


VLADIMIR TRUJILLO ARIAS

PWM Usos

- Voltaje Variable





VLADIMIR TRUJILLO ARIAS

API LEDC para ESP32

- La API LEDC (Low Energy Direct Current) es una interfaz de programación de aplicaciones proporcionada por Espressif Systems para controlar los pines PWM dedicados de baja energía del ESP32.
-
- Estos pines están diseñados específicamente para aplicaciones que requieren un control PWM preciso y de baja energía, como el control de brillo de LED, la generación de señales analógicas y la regulación de velocidad de motores.

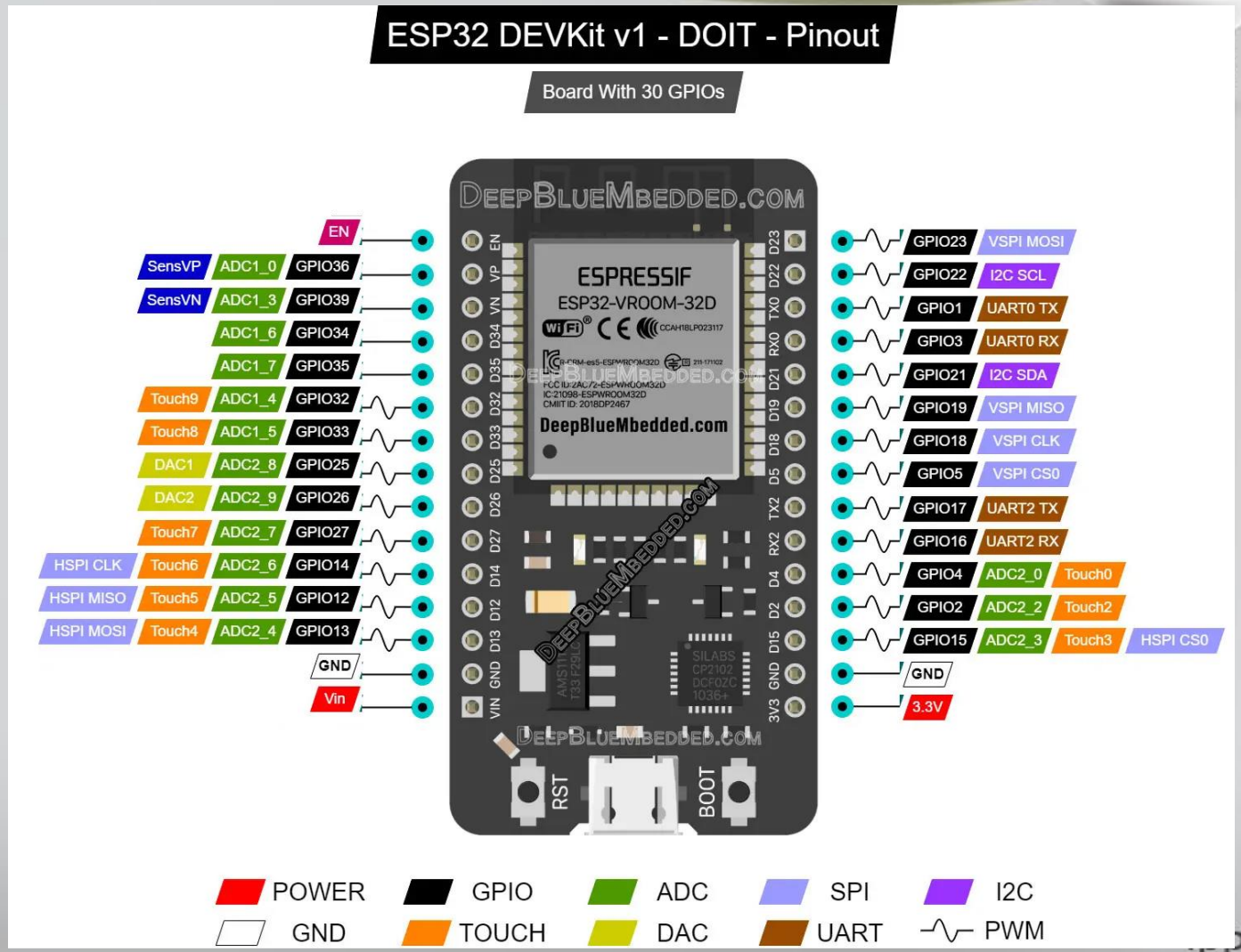
VLADIMIR TRUJILLO ARIAS

API LEDC para ESP32

- ledcSetup(channel, frequency, resolution, pin): Configura un canal LEDC en un pin específico con la frecuencia, la resolución y el pin especificados.
- ledcAttachPin(pin, channel): Asocia un pin GPIO específico con un canal LEDC.
- ledcWrite(channel, duty): Establece el ciclo de trabajo (duty cycle) de la señal PWM en el canal LEDC especificado.

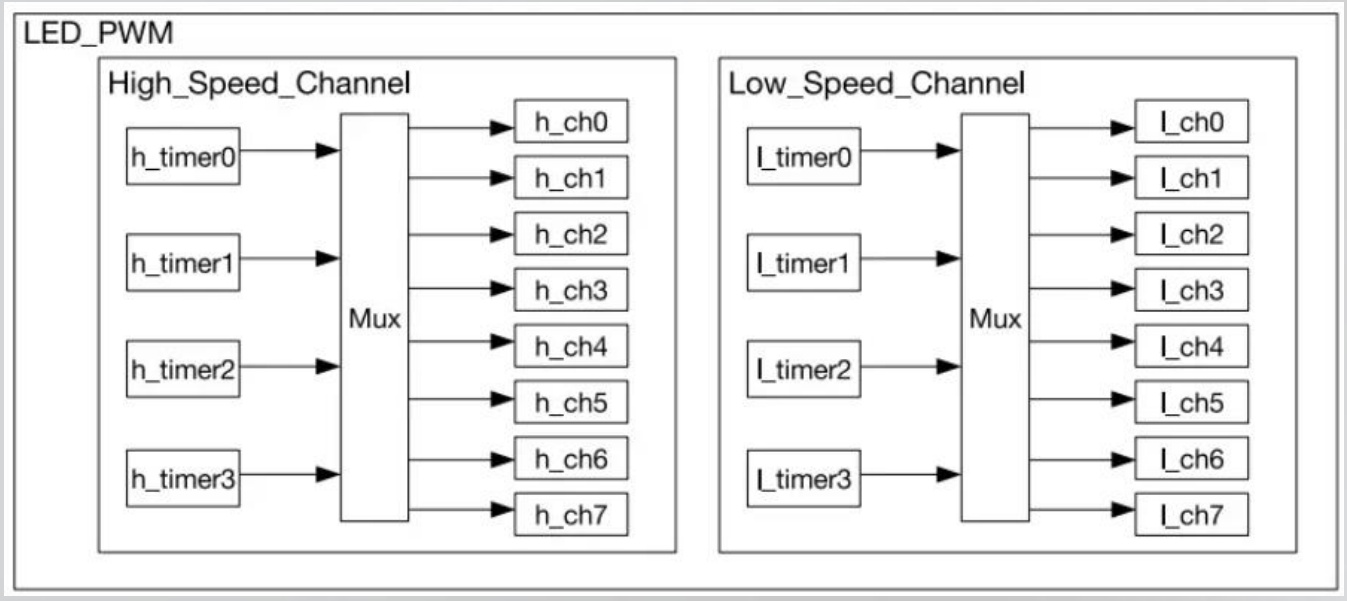
API LEDC para ESP32 (canales)

- En la tarjeta Do it ESP32 DevKit V1, existen 16 canales de PWM de la API
- Se puede asignar cualquiera de estos canales a cualquier pin GPIO que desee. Pero el pin debe poder funcionar como una salida digital
- En las placas de desarrollo ESP32, todos los pines GPIO se pueden configurar para funcionar en modo de salida, excepto 4 pines.



API LEDC para ESP32 (canales)

- El controlador ESP32 PWM tiene 8 canales de alta velocidad y 8 canales de baja velocidad, lo que nos da un total de 16 canales.
- Se dividen en dos grupos dependiendo de la velocidad. Para cada grupo hay 4 temporizadores/8 canales.
- Esto significa que cada dos canales comparten el mismo temporizador. Por lo tanto, no podemos controlar de forma independiente la frecuencia PWM de cada par de canales.
- h_timer → timer de alta velocidad
- l_timer → timer de baja velocidad



VLADIMIR TRUJILLO ARIAS

API LEDC para ESP32 (frecuencia y resolución)

La resolución más alta se calcula cuando el divisor de reloj LEDC_CLK_DIVx es 1 y se redondea hacia abajo. Si la resolución más alta calculada por la fórmula es mayor que el ancho del contador de 20 bits, entonces la resolución más alta debe ser de 20 bits.
La resolución más baja se calcula cuando el divisor de reloj LEDC_CLK_DIVx es 1023 + 255/256 y se redondea hacia arriba. Si la resolución más baja calculada por la fórmula es inferior a 0, entonces la resolución más baja debe ser 1.

$$f_{sig\_outn} = \frac{f_{LEDC\_CLKx}}{LEDC\_CLK\_DIVx \cdot 2^{LEDC\_HSTIMERx\_DUTY\_RES}}$$

$$LEDC\_HSTIMERx\_DUTY\_RES = \log_2\left(\frac{f_{LEDC\_CLKx}}{f_{sig\_outn} \cdot LEDC\_CLK\_DIVx}\right)$$

Table 14-1. Commonly-used Frequencies and Resolutions

LEDC_CLKx	PWM Frequency	Highest Resolution (bit) [1]	Lowest Resolution (bit) [2]
APB_CLK (80 MHz)	1 kHz	16	7
APB_CLK (80 MHz)	5 kHz	13	4
APB_CLK (80 MHz)	10 kHz	12	3
RC_FAST_CLK (8 MHz)	1 kHz	12	3
RC_FAST_CLK (8 MHz)	2 kHz	11	2
REF_TICK (1 MHz)	1 kHz	9	1

API LEDC para ESP32

- ledcWriteTone(channel, frequency): Establece la frecuencia de la señal PWM en el canal LEDC especificado.
- ledcAttachPinTone(pin, channel): Asocia un pin GPIO específico con un canal LEDC y establece su frecuencia de trabajo.
- ledcDetachPin(pin): Desasocia un pin GPIO previamente asociado con un canal LEDC.