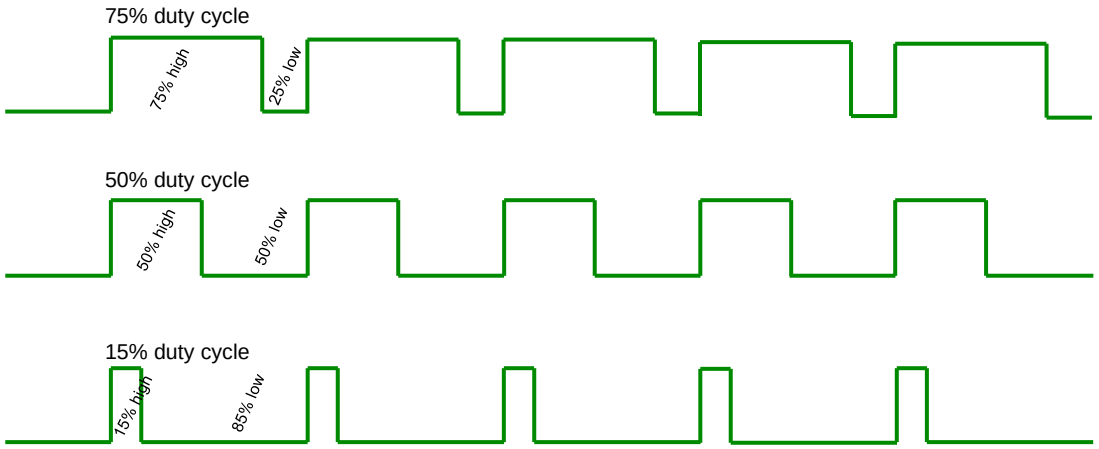
Pulse width modulation

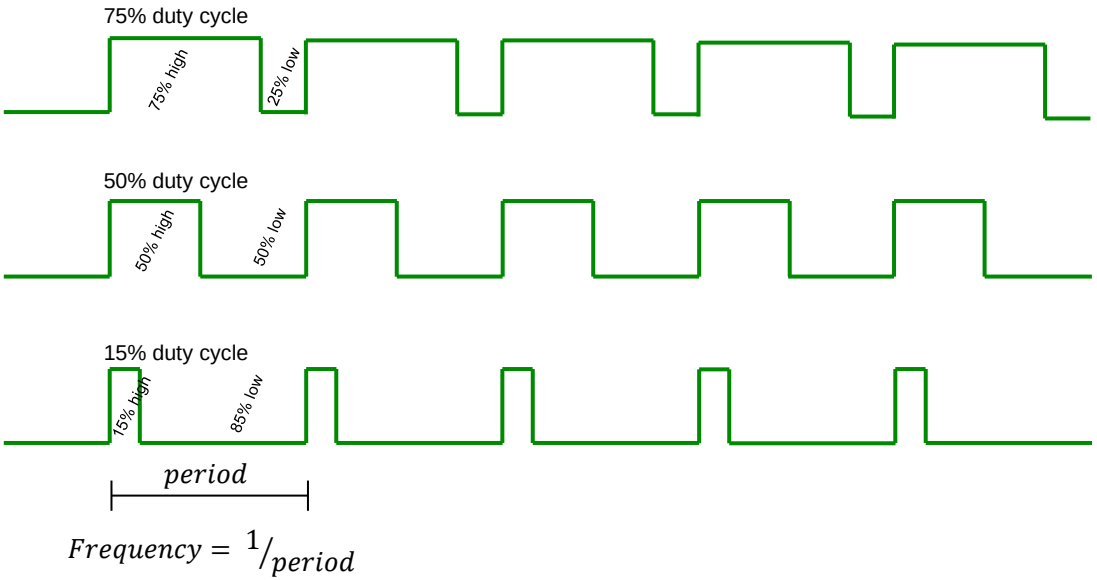
- Pulse width modulation (PWM) is a scheme where a digital signal provides a smoothly varying average voltage. This is achieved with positive pulses of some controlled width, at regular intervals. The fraction of time spent high is known as the duty cycle. This may be used to approximate an analog output, or control switchmode power electronics.

RP2040 Datasheet p.524



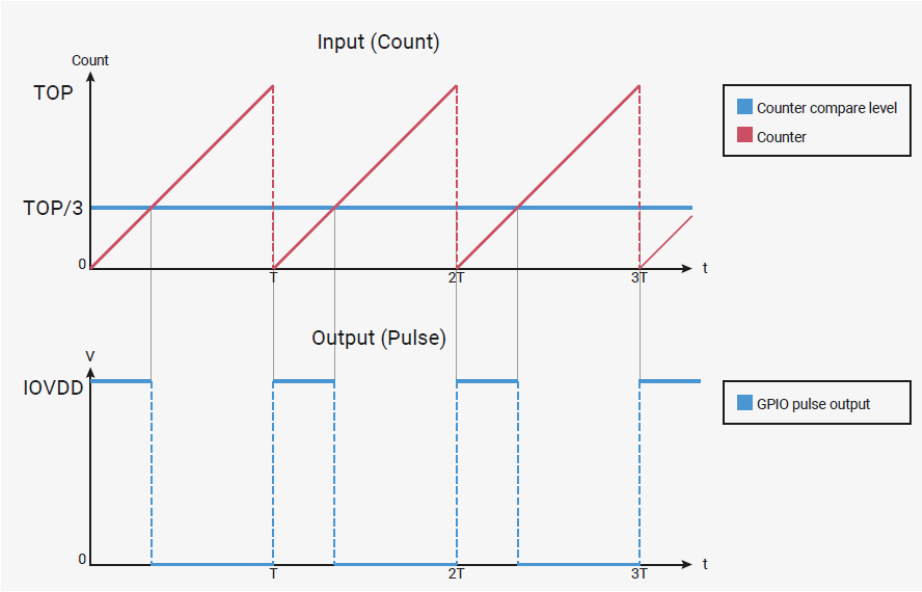
PWM frequency

- When duty cycle is greater than 0% and less than 100% the pulses appear at fixed frequency that does not depend on the pulse width
- The period is measured from one edge (e.g. rising edge) to the next edge of same type
- The frequency needs to be high enough to provide ”smooth” average voltage. For example when dimming an LED with PWM using a too low frequency causes LED to flicker



PWM implementation

- Typically PWM hardware consists of counter with a programmable **high limit** (top, wrap) and a **compare value**
- The counter counts up at the PWM input frequency and when the counter reaches the high limit it wraps to zero
  - Note that PWM input frequency is not the same as PWM frequency
- When counter value is less than compare value the output is high. When counter is greater than or equal to compare value the output is low
- By adjusting the compare value we can set the duty cycle of the output signal
- The high limit determines the precision of the duty cycle adjustment. If high limit is 1000 there are 1000 steps between 0 – 100% duty cycle (0,1% precision)

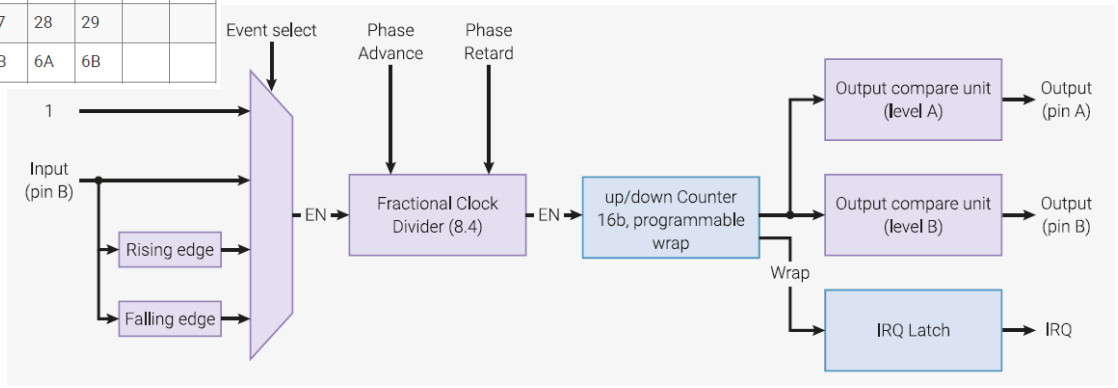


RP2040 Datasheet p.525

RP2040 PWM

- 8 PWM slices
  - 16 bit counter with programmable wrap (TOP)
  - Two independent channels – both channels have their own compare value (CC)
  - Total of 16 PWM channels
- All 30 GPIO pins can be used for PWM
  - Since there are less PWM channels than pins there are some limitations on use of PWM

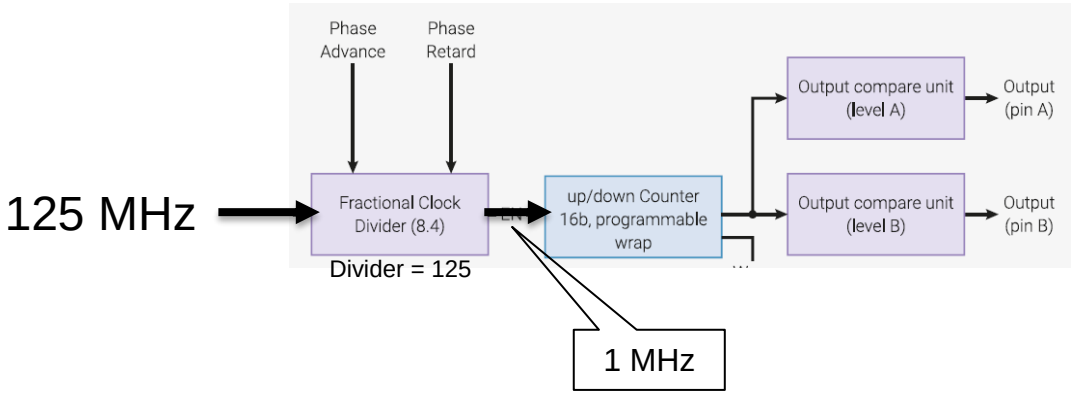
GPIO	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
PWM Channel	0A	0B	1A	1B	2A	2B	3A	3B	4A	4B	5A	5B	6A	6B	7A	7B
GPIO	16	17	18	19	20	21	22	23	24	25	26	27	28	29		
PWM Channel	0A	0B	1A	1B	2A	2B	3A	3B	4A	4B	5A	5B	6A	6B		



RP2040 Datasheet p.524

RP2040 PWM frequency

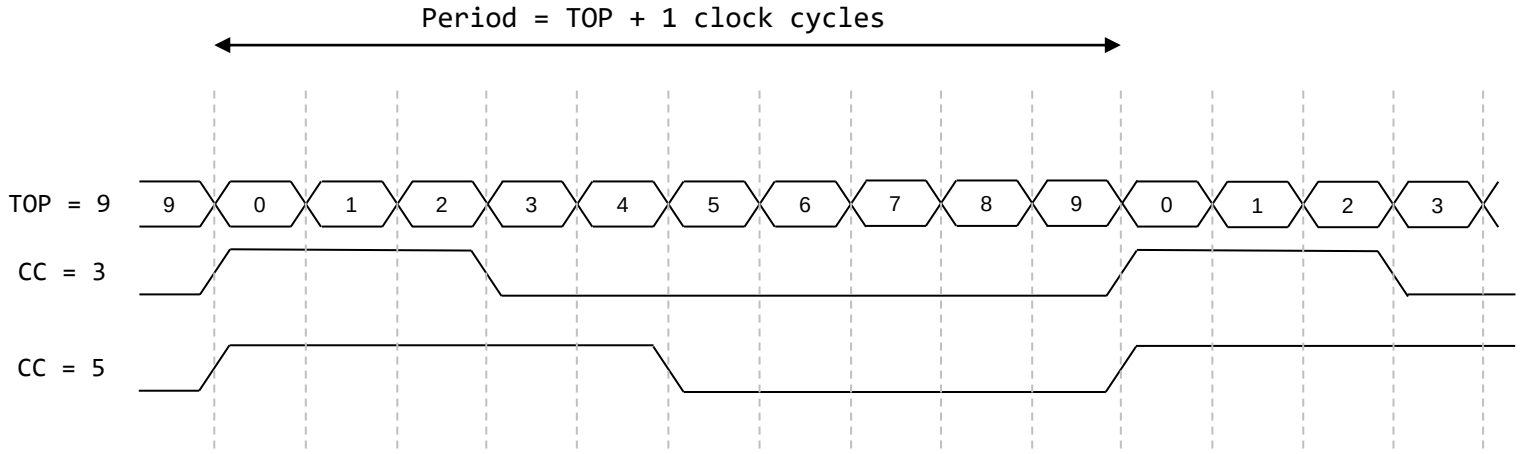
- PWM frequency of a slice is the frequency at which the counter wraps back to zero
  - Wrapping is determined by the high limit (TOP). Wrapping takes place every TOP+1 clock cycles
- For example:
  - PWM input frequency is 1 MHz
    - Pico standard clock frequency is 125 MHz
    - Clock divider is set to 125 → divider outputs 1MHz clock signal
  - TOP is set to 999 → wraps to zero every 1000 clock cycles
    - At 1 MHz the length of clock cycle is 1 µs → 1000 x 1 µs = 1 ms



RP2040 PWM duty cycle

- The slice counts repeatedly from 0 to TOP
- Output is high when counter value is less than CC

- In the example below the duty cycle can be adjusted in 10% increments
  - CC = 0 → 0%
  - CC = 3 → 30%
  - CC = 10 → 100%



How to configure RP2040 PWM

- Call pwm_gpio_to_slice_num and pwm_gpio_to_channel to get slice and channel of your GPIO pin
- Call pwm_set_enabled to stop PWM
- Call pwm_get_default_config to default PWM configuration
- Call pwm_config_set_clkdiv_int to set clock divider
- Call pwm_config_set_wrap on the default config to set wrap (TOP)
- Call pwm_init with your slice number and the config you created with start set to false
- Call pwm_set_chan_level with your slice and channel numbers to set level (CC) → duty cycle
- Call gpio_set_function to select PWM mode for your pin
- Call pwm_set_enabled to start PWM

- Adjust duty cycle as needed by calling pwm_set_chan_level

https://www.raspberrypi.com/documentation/pico-sdk/hardware.html#hardware_pwm