

Warmup

Write (or re-run) code to blink an LED with your ESP32

How fast can you blink the LED before it doesn't appear to your eyes to be blinking anymore?

Challenge: figure out a way to measure (or at least estimate) the fastest you can possibly blink an LED with the ESP32.

EN 1: Engineering in the Kitchen

Steven Bell

16 October 2023



Project 3

nolop.org/haunted-house/

Build an interactive pumpkin exhibit

Project 3 schedule

Today: Controlling outputs

Wednesday: Making sound

Monday: Microwave demo day

Wednesday 10/25: Pumpkin distribution / project assembly

Monday 10/30: Final tweaks, setup for haunted house at 4pm

How fast can we blink?

Let's try!

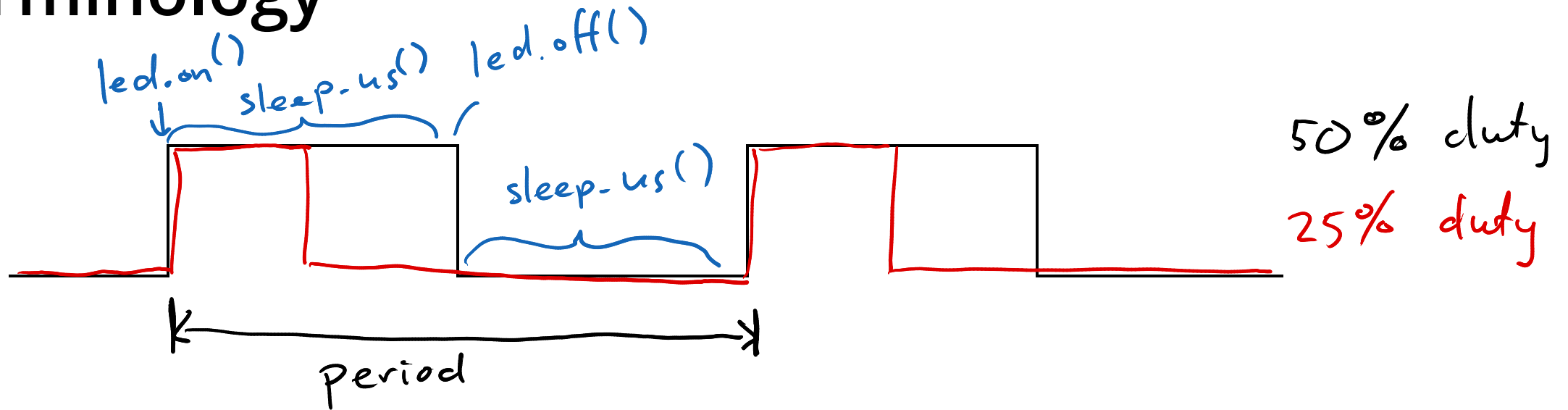
Use `sleep_us()` for microsecond sleeps.

Controlling brightness

What if we varied the amount of time that the LED was on?

Challenge: make your LED "breathe", slowly fading on and off.
Note that human perception is approximately logarithmic!

Terminology



period - seconds

$$\text{frequency} = \frac{1}{\text{period}} \quad \text{cycles/second} \quad \frac{1}{\text{sec}} = \text{Hz}$$

duty cycle - fraction of time high

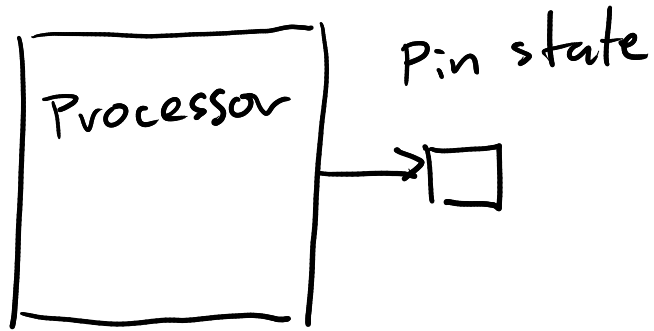
Pulse-width modulation (**PWM**): controlling something by varying the duty cycle (i.e, the width of the pulses)

Ok, but...

What if you wanted to blink 2 LEDs? (at different rates?)

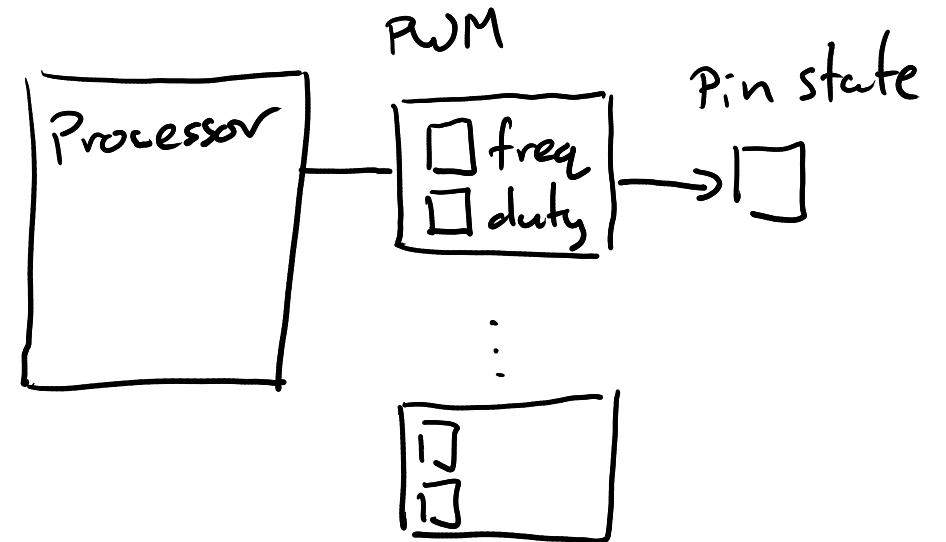
Hardware to the rescue!

What we've done so far:



~ 100% usage

Using PWM hardware:



(Take ES 4 if you wanna learn how this works!)

Using PWM hardware

Built-in hardware to turn pins on and off

```
from machine import Pin, PWM
led = PWM(Pin(13)) # Any output pin
led.freq(1000) # Hz, ranges from 1Hz to 40MHz
led.duty(512) # Fraction, ranges from 0-1023  $2^{10} - 1$ 
```

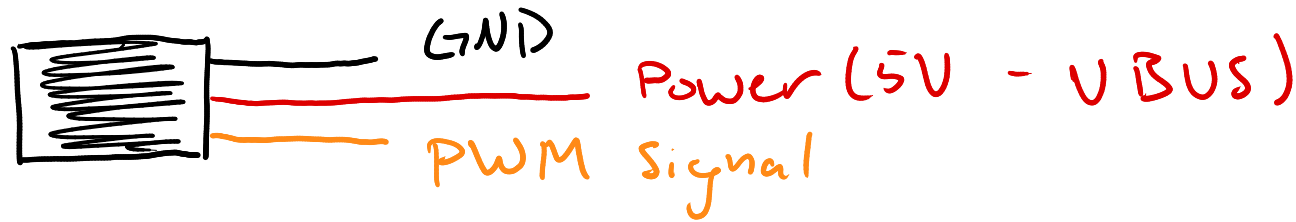
Set up one or more LEDs to blink/fade using PWM!

Challenge: Is there a limit to how many pins you can control this way? Try making multiple LEDs breathe at different rates.

Servo motors

Have internal circuitry which uses proportional feedback to control the position of the shaft.

Take 3 wires: power (5V), ground, and "signal"



But how can we tell it the position?

Driving a servo with ESP32

```
from machine import Pin, PWM
servo = PWM(Pin(2), freq=50) # Must be 50Hz (or close)
servo.duty(30) # Vary to set position
```

30 - 150 duty cycle

Sound with PWM?

```
from machine import Pin, PWM
servo = PWM(Pin(2), freq=440) # Some audible frequency
servo.duty(512) # Experiment with this!
```

Sound with PWM?

```
from machine import Pin, PWM
servo = PWM(Pin(2), freq=440) # Some audible frequency
servo.duty(512) # Experiment with this!
```

Play a sound with your speaker

Challenge: Experiment with the duty cycle. Can you hear it?
Try playing a song!

Microwave project work time