CSE121: IoT

PWM

Jose Renau
renau@ucsc.edu

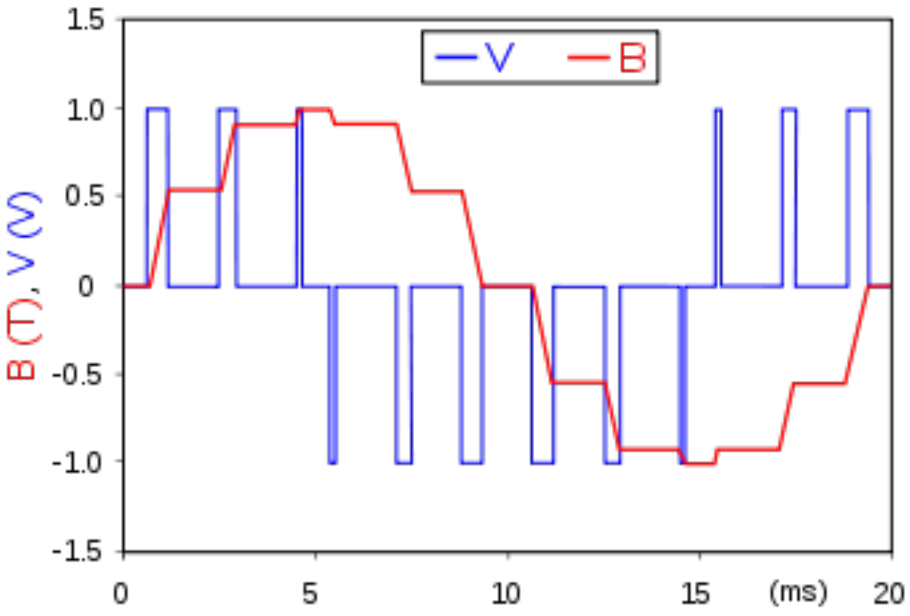
UCSC Baskin School of Engineering
University of California, Santa Cruz

Announcements

- Lab2 due next week

Pulse Width Modulation (PWM)

- Technique to control the average power delivered by a electronic digital system
- Instead of modulating the amplitude, control when the signal is on/off
- Digital in (V)
- “Analog” out (B)
- Inertia (impedance)



PWM Applications

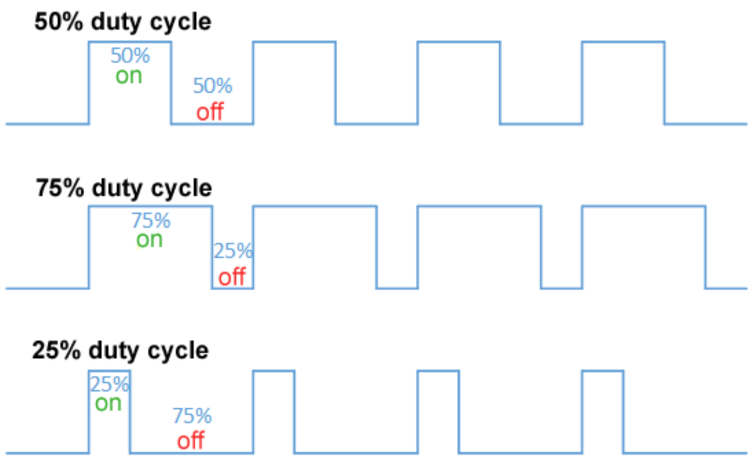
- Control power output (to charge a battery)
- Control electric motor
- Dim brightness of an LED
- Communications
- Duration of a pulse represents value

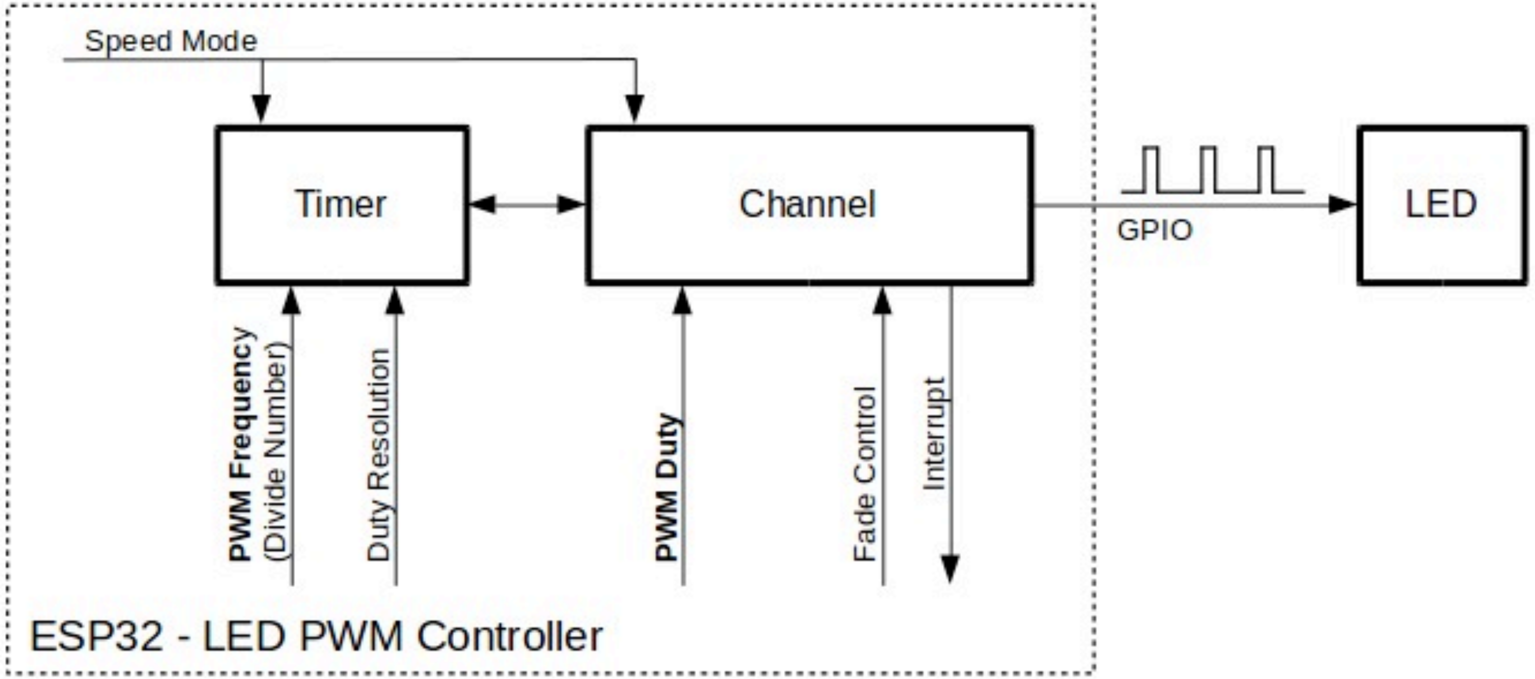
PWM Switching Speed

- Must be higher than the system’s inertia
- Then the output will be smooth
- Example (LED):
  - Human eye: 50-100Hz
  - Old TVs had 24frame/s, regarded acceptable minimum
  - LED that is turned on/off at >100Hz appears flicker free
  - But not for a high speed camera
- At high enough frequency
  - Vout = Vmax * D (duty cycle)

Frequency and Duty Cycle

- Key parameters
  - Frequency
  - Duty





PWM Software Controlled

- Loop

```
while(1) {
      gpio_set_level(LEDC_OUTPUT_IO, 0);
      vTaskDelay(pdMS_TO_TICKS(10));       // 10 ms
      gpio_set_level(LEDC_OUTPUT_IO, 1);
      vTaskDelay(pdMS_TO_TICKS(10));       // 10 ms
}
```

- What is vTaskDelay?
  - FreeRTOS version if “usleep(10*1000)”   // usec sleep not ms

PWM in Software

- Unprecise
  - Other tasks can “break” the loop (more when we do FreeRTOS)

- Power inefficient (CPU runs at 100%)

- Software suffers from non-determinism

PWM in HW

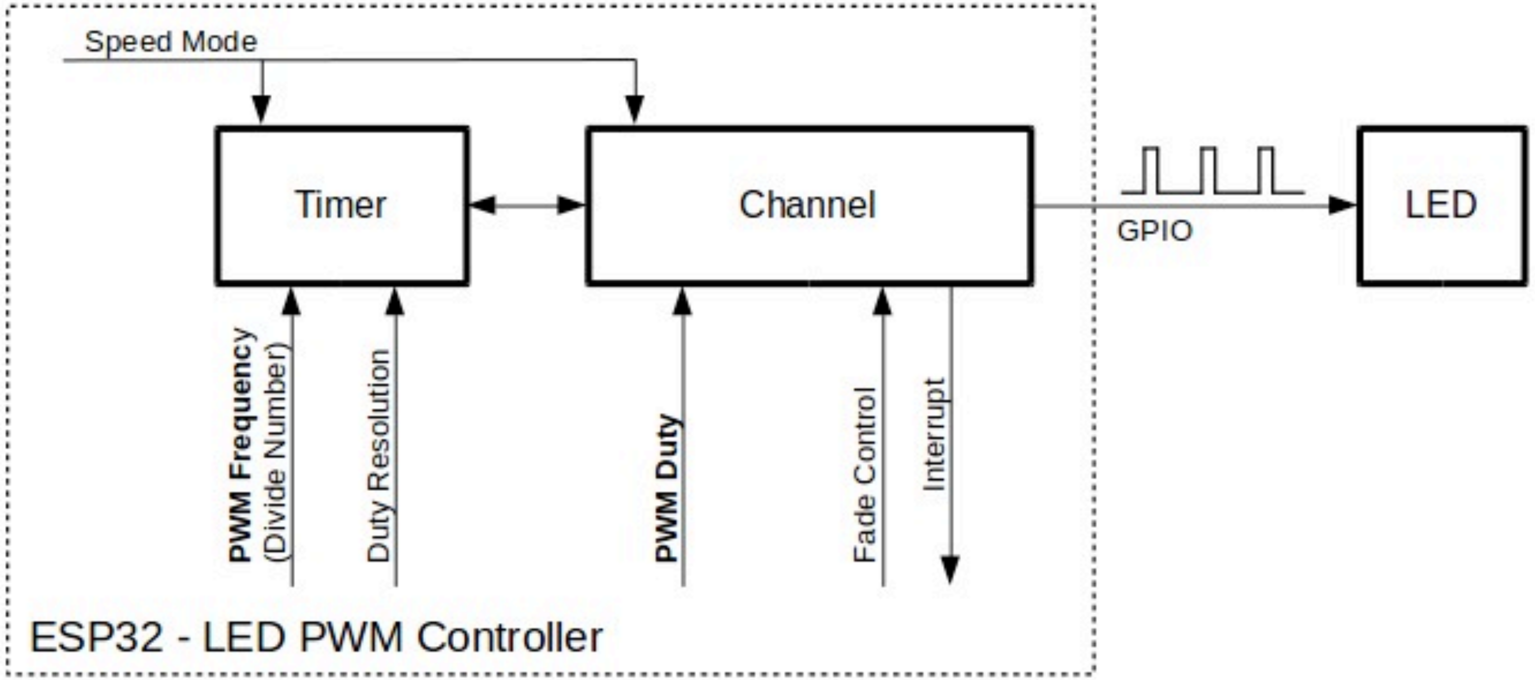
- Most SoC have a PWM controller
  - High precision
  - Higher frequency to control (not so important for LED)
  - Efficient (CPU can go to sleep)

Quiz

- Quiz Time

ledc_ in ESP32

- Configure
  - Timer
  - Channel



- Let’s create an example!

```
ifd.py create-project ppm
cd pwm ; ifd.py set-target esp32c3 ; nvim
```

Next Class

- PWM