Pulse-Width Modulation

ECE 362
https://engineering.purdue.edu/ece362/

Reading Assignment

- Textbook, Chapter 15.3, Timers: PWM Output, pp. 384 – 395.

Future Reading Assignment

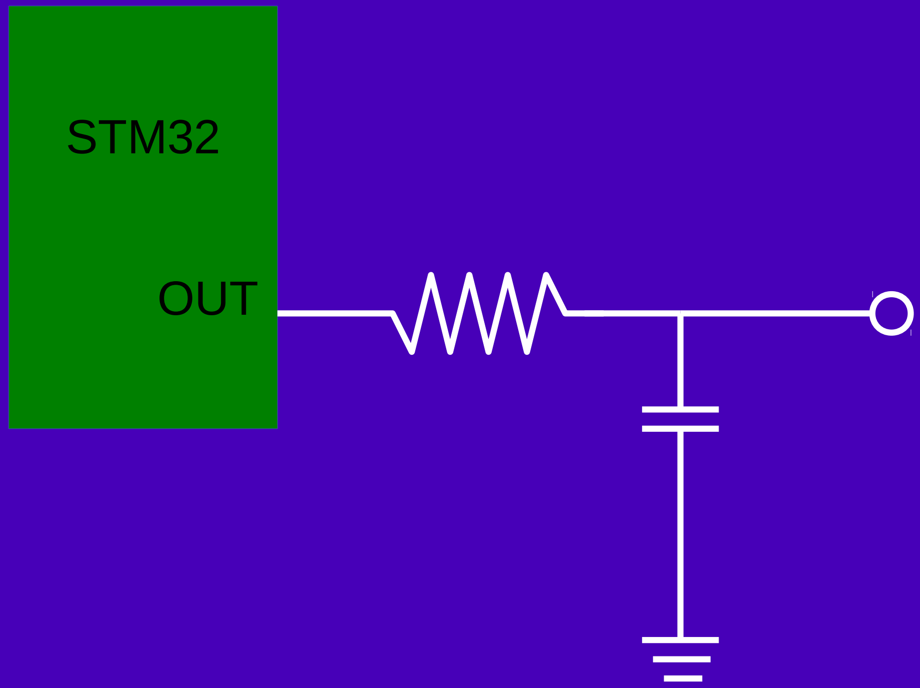
- Textbook, Chapter 22, Serial Communication Protocols, pp. 527 – 598
  - It’s a long chapter.
  - Let’s first look at Section 22.3, SPI, pp. 568–577.
  - Don’t worry so much about the USB section.
    - Read that only if you’re curious.
    - Your development board has no USB interface.
      - There are some USB provisions in the STM32F091, but would take much work.
    - Other books are better for understanding USB.

Using GPIO as a DAC

- We'd like to be able to have a DAC with a strong drive strength like we have with GPIO.
- We don't just want on/off states. We want multiple voltage levels.
- Consider a square wave with a varying duty cycle:
  - We average the area under the curve.
  - The duty cycle of the wave is then the analog level.
  - We can average using a low-pass filter.
  - As long as the wave frequency is much higher than the signal we want to model, no one will notice the averaging.

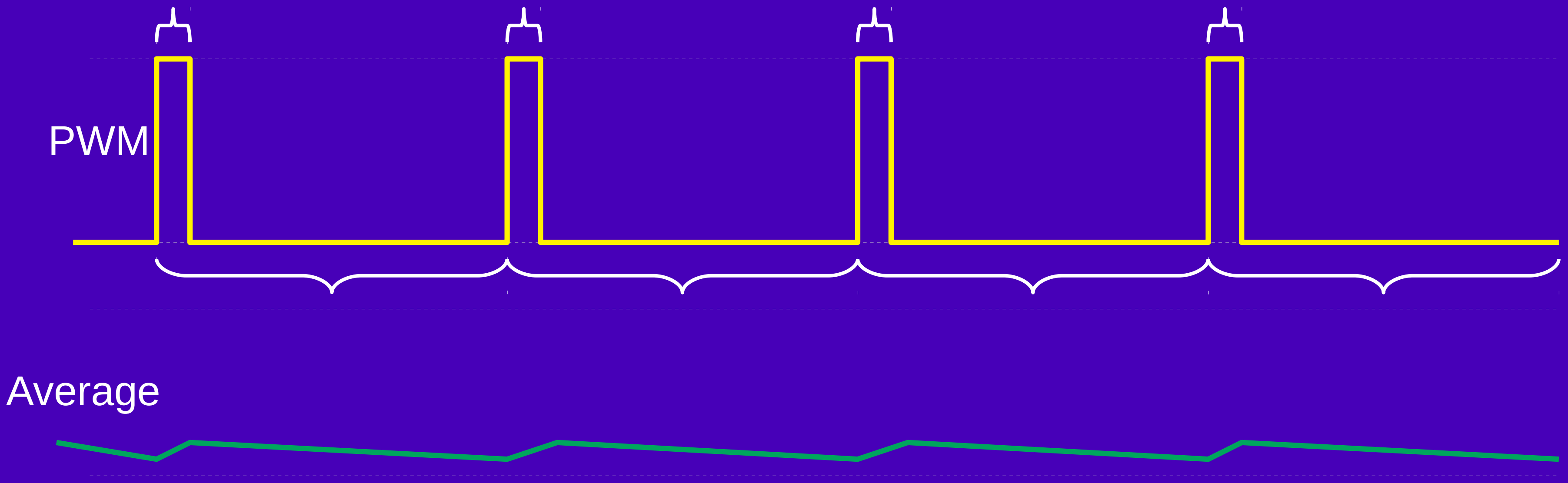
Averaging output

- Various ways to do this using low-pass filters.



PWM: Pulse-Width Modulation

- Each timer has a PWM mode where the output starts high, and goes low when CNT == CCRx



Determining PWM characteristics

- Textbook, 15.3:

  PWM freq = CK_INT / (PSC + 1) / (ARR + 1)

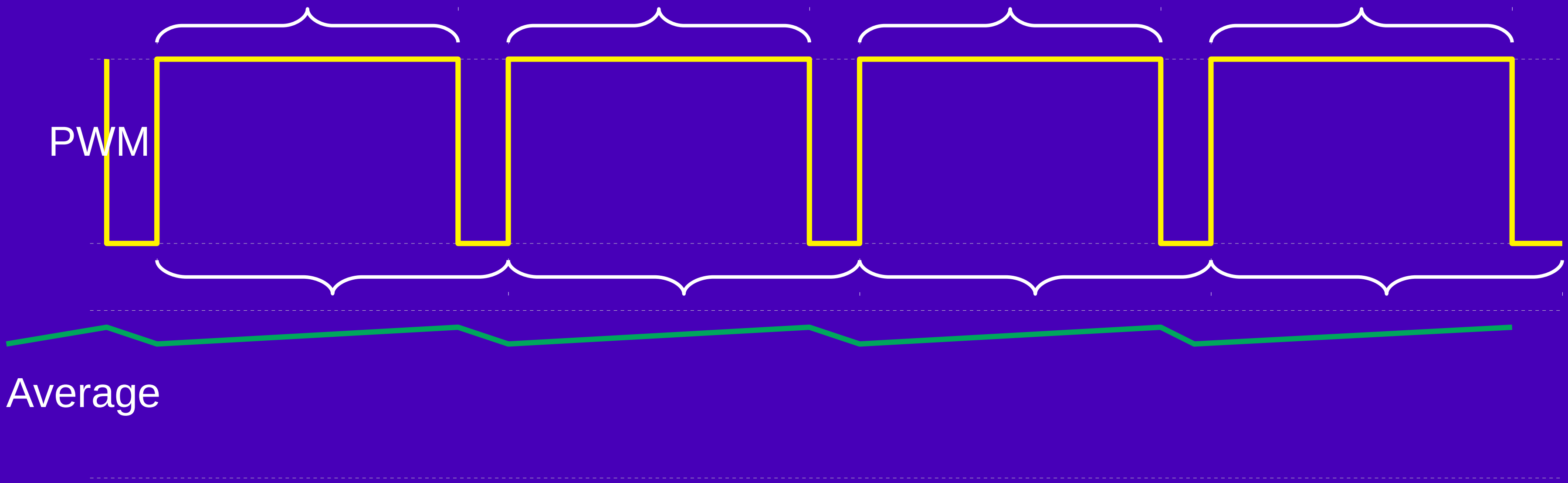
  - e.g., CK_INT == 48MHz

  PWM duty cycle = CCRx / (ARR + 1)

- Note: CCRx == 0: Never on.
- Note: CCRx >= ARR + 1: Always on.

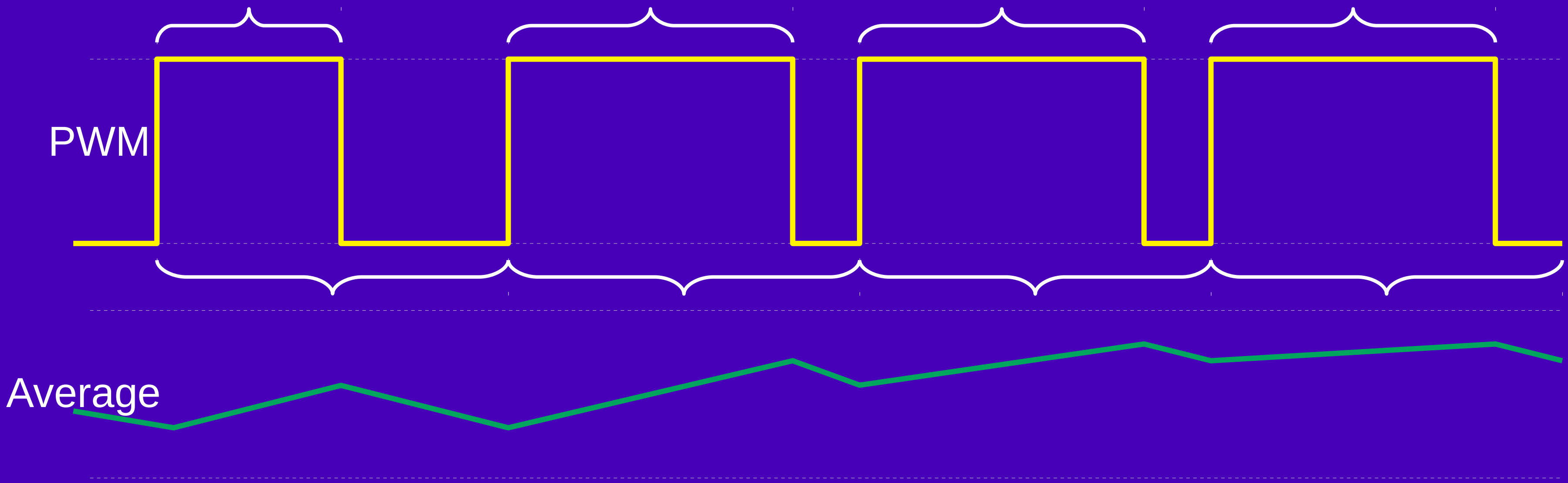
Higher duty-cycle, higher average

- Over the long term, the average output will be proportional to the duty cycle.



Dynamically changing duty-cycle

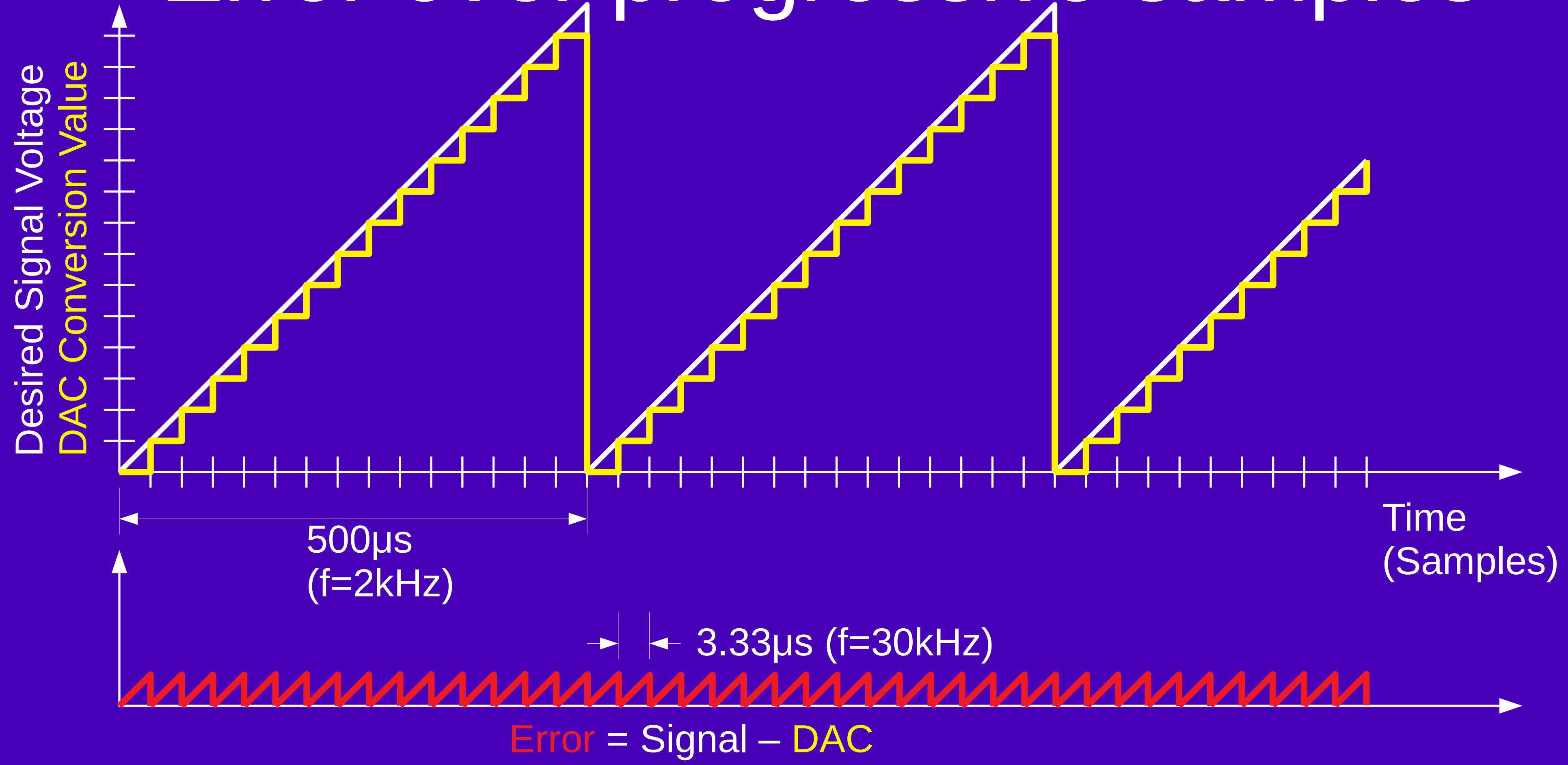
- The timer system has a convenient way of changing the duty cycle of an output wave.



Must use high pulse frequency

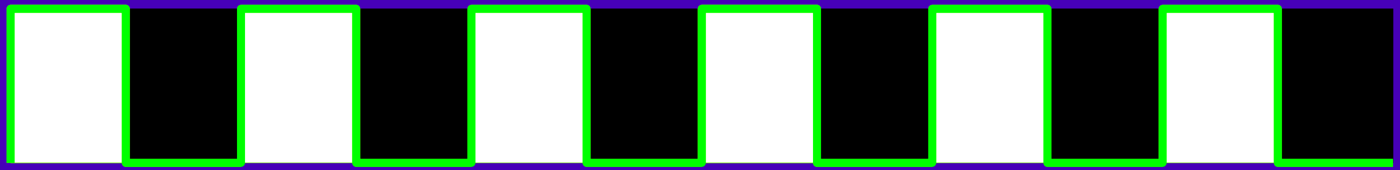
- If the pulse frequency is much higher than that of the desired signal, the "noise" will be imperceptible.

Error over progressive samples

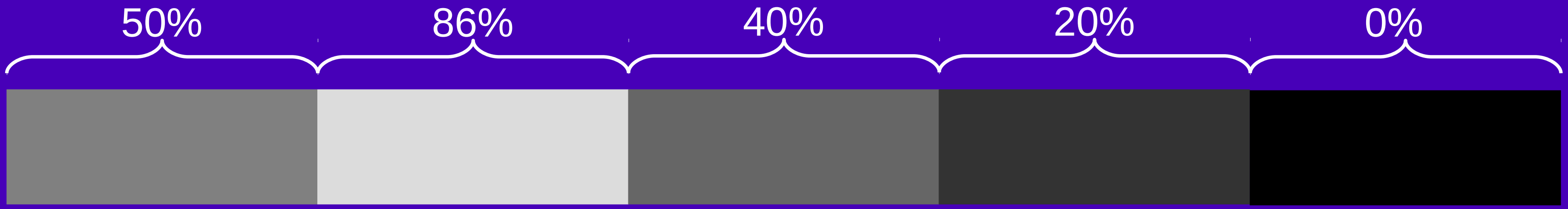


PWM noise is similar

- Color the "high" part of the wave white and the "low" part black. What would you see?

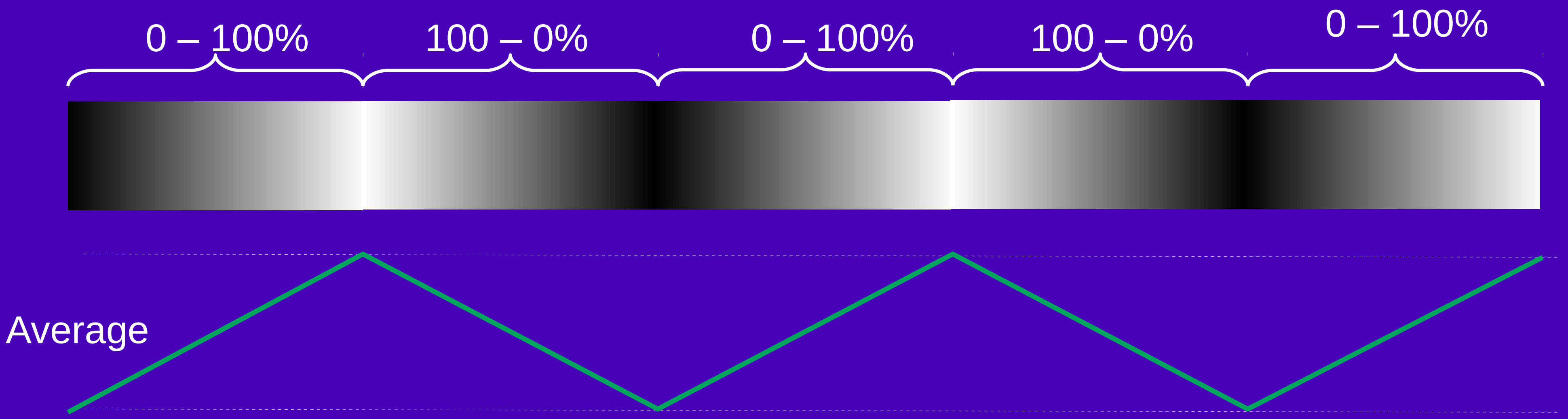


  – If we turn off the green waveform, and if the alternation (frequency) is dense enough, you see only a shade of gray.



Continuous variation

- If you can vary the duty cycle after every period, you can have continuous variation.



Not hard to imagine a sine wave

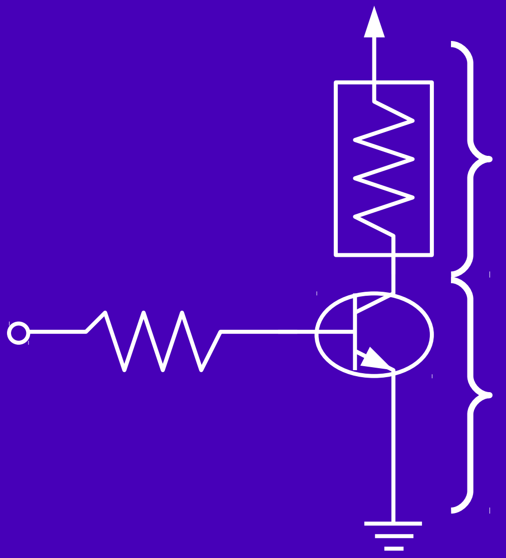
- If we had a way to update the duty cycle at the end of every clock period… Hmm….
- Could use the update interrupt, which happens whenever the counter reaches the maximum (when up-counting).
  - ISR can read a value from a wave table, and store it into the CCRx register.
- But the interrupt will occur when the next clock period is already in progress… Hmm...

Set the OCxPE bit

- If OCxPE is set in the CCMRx register, it will turn on "preload enable".
  - An update to the CCRx value will actually not be loaded until the next counter "update".
  - The new comparison value will be deferred until after the current comparison value is definitely used.
- Similar flag in TIMx_CR1 called ARPE: "Auto-reload preload enable"
  - If you want to change the PWM period, but not until the next cycle begins, set ARPE.

Power-inefficiency of analog control

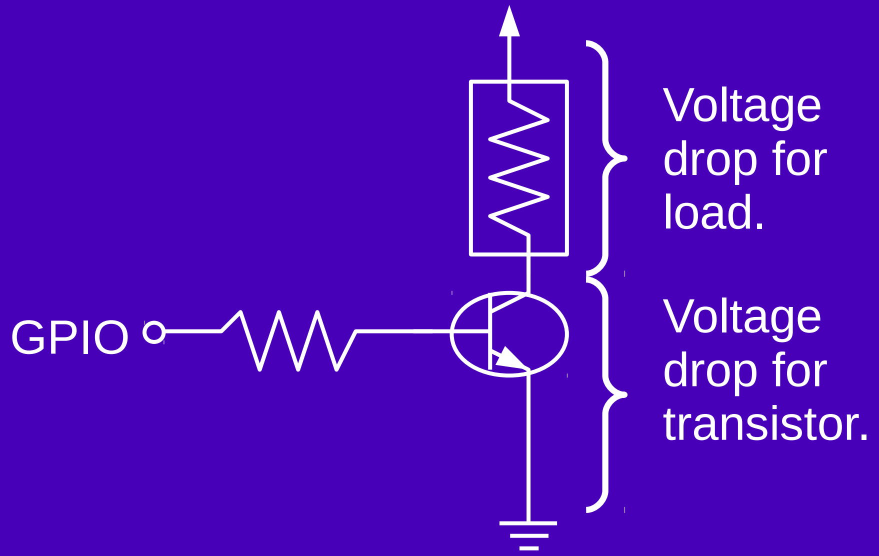
- A DAC can change voltage incrementally, but you wouldn’t use it to dim a light bulb or heavy load.



  - Voltage of DAC determines I_{BE} current, which controls I_{CE} current, which controls the load current.
  - Transistor driving the bulb would be in the "active mode" with current flowing through a voltage drop.
    - Watts of power dissipated as heat by transistor.

On-off switching is efficient

- When switching on or off (as with PWM), a transistor is either:
  - conducting fully with no voltage drop, or
  - not conducting at all with no current.
- Either way, no power loss by the transistor.



Sometimes, you don’t need a physical low-pass filter.

- If signal is going to be *perceived* by a human:
  - Your eyes cannot register changes faster than 50 Hz.
  - Your ears cannot register changes faster than 16 kHz.
- If signal is damped by something else:
  - A motor (inertia of rotor is a low-pass filter).

Other perception problems

- What happens if you use PWM with a frequency in the range of 100 – 1000 Hz to drive a DC motor?
  - You’ll hear it whine.
  - Sometimes motors resonate at certain PWM frequencies.
  - Sometimes you can use PWM above audible frequencies (>16 kHz)
  - Sometimes the speed of the motor limits the usable PWM frequencies.
  - Usually need to do some experimentation.

How about ADC?

- PWM, done fast enough, is equivalent to DAC
- Is there a discrete equivalent to the ADC?
  - Sort of
- Early personal computers had crude analog joystick ports
  - Each axis of the joystick slowly charged a capacitor ($1-e^{-t/RC}$).
  - Once the capacitor reached a threshold voltage, it set a flip-flop.
  - Resetting the flip-flop restarted the charging.
  - The time between reset and set indicated the voltage being read.
  - Some systems used an op-amp *integrator* to create a linearly-increasing voltage that was compared to sample voltage. A free-running counter registered the *time* of equality.
- If you could measure *time* you could measure *voltage*.

ADC still takes time

- Most ways of converting an analog signal to a digital value involve *waiting* and *comparing*
- Is there a way to make an instant ADC?
  - You could parallelize 2^N comparitors and route their outputs to a priority encoder to find the highest one
  - This is an N-bit *flash converter*
  - High-speed oscilloscopes use things like this