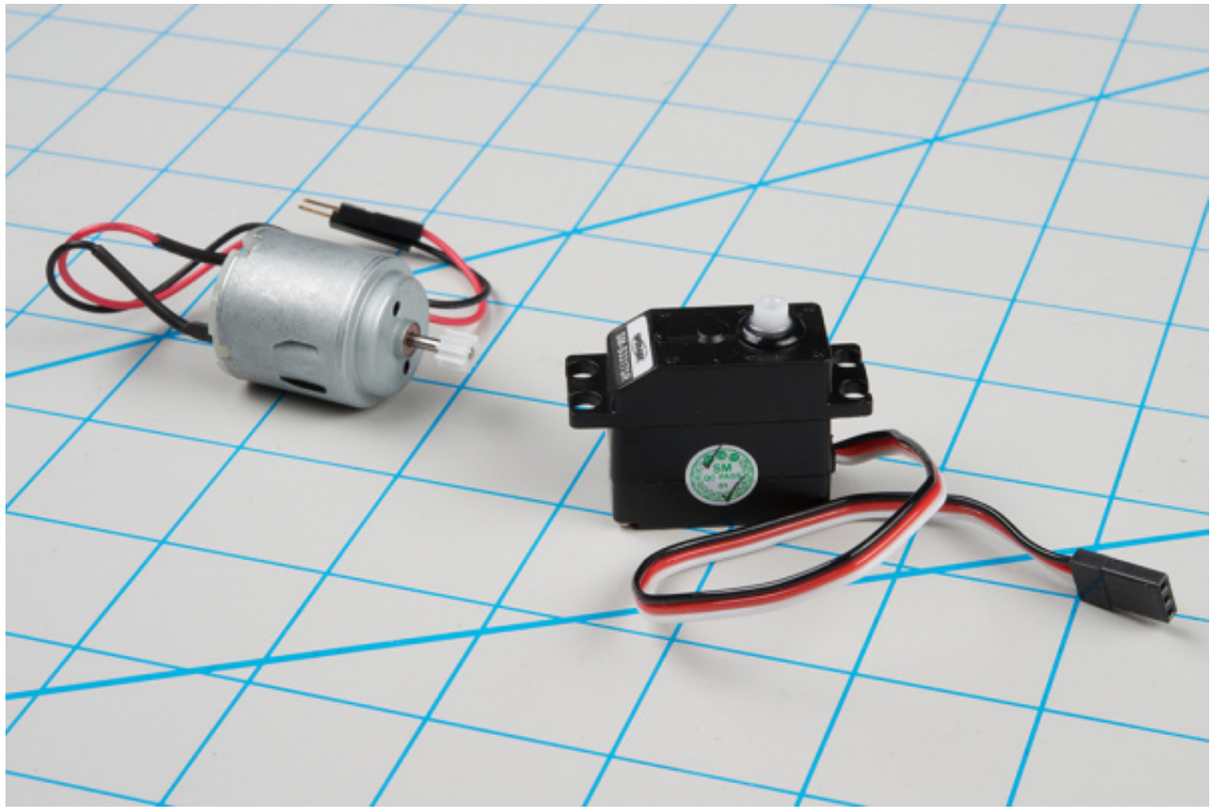
Servo Motor and PWM (Pulse Width Modulation)

Motors



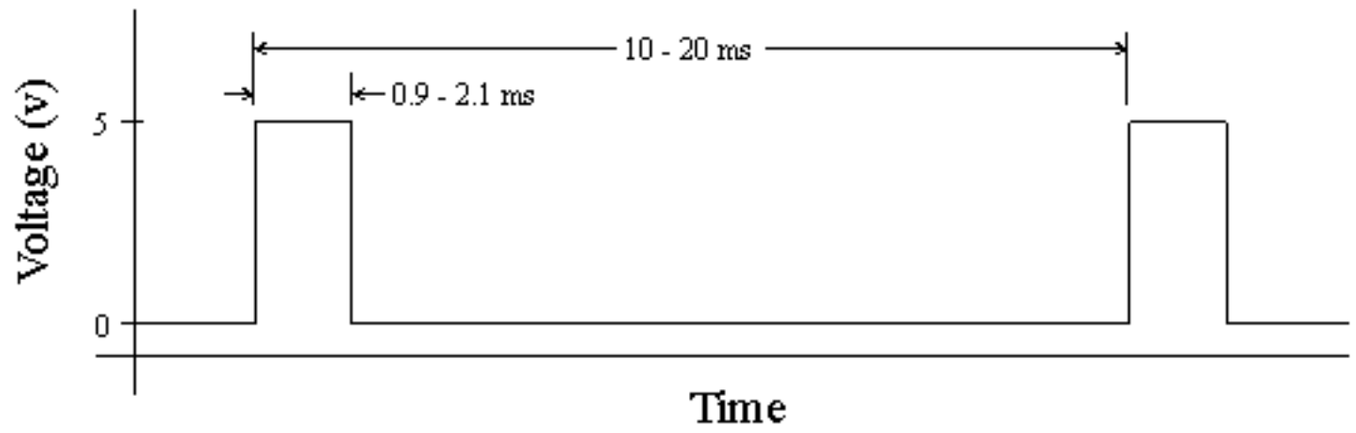
DC Motor

- Two wires (Power , Gnd)
- Turn continuously
- Vary input voltage to change speed

Servo Motor

- Three wires (Power, Gnd, Control)
- Turn to a specific position
- Vary duty cycle to turn
- Internally consists of DC motor, potentiometer and control circuitry

Servo Motor



- The control signal is pulse width modulated
- duration of voltage pulses determine the angle of the output shaft
- The pulses can be from 0.9 ms to 2.1 ms long, 1.5 ms being the center position
- 0.9 ms and 2.1ms will cause the arm to go all the way clockwise or counter clock wise
- the frequency of the pulses is typically 50Hz (sometimes up to 100Hz)

Introduction - PWM

- **PWM** = Pulse Width Modulation
- PWM is a **Square wave signal** with **HIGH**s and **LOW**s states in each signal period
- General purpose of PWM to control analog devices via digital signal
- High powered devices can be controlled by PWM signals
- The voltage seen by the load is directly proportional to the source voltage
- Another use of PWM is to encode information for transmission

PWM Signal

Basic Characteristics of a PWM signal:

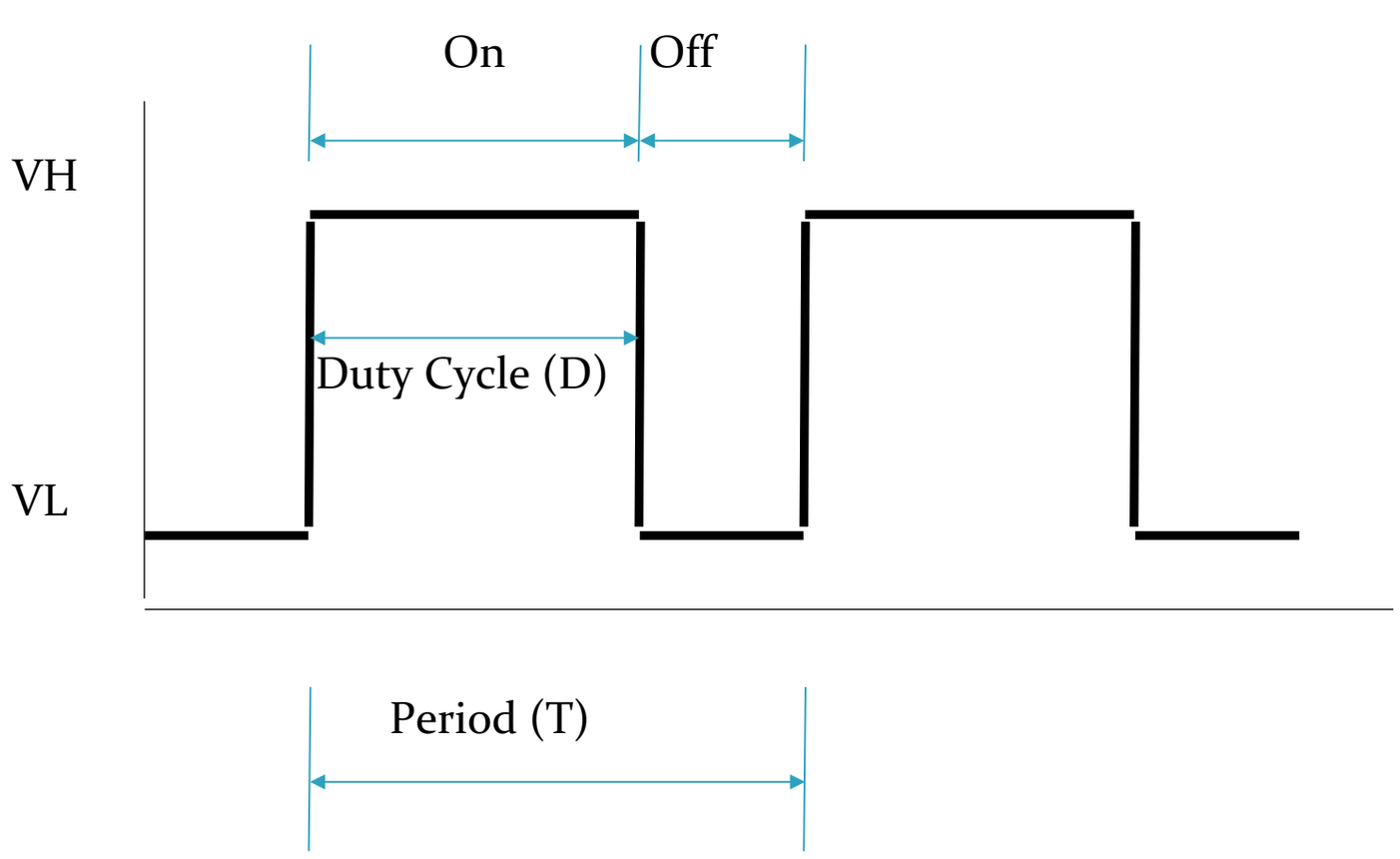
- **Frequency** (F) or **Period** (T) of the Square Wave signal
  - *Frequency limited by your clock and desired resolution*
  - *Resolution is defined by clock speed and frequency of the PWM*
  - *The faster you run the PWM, the fewer clock ticks occur in the period considered —> lower duty cycle resolution*

- **Duty Cycle** of On-Off states

Duty cycle adjustments represent the transmitted information or cause the effect that we expect at the load -analog devices to be controlled

PWM Signal

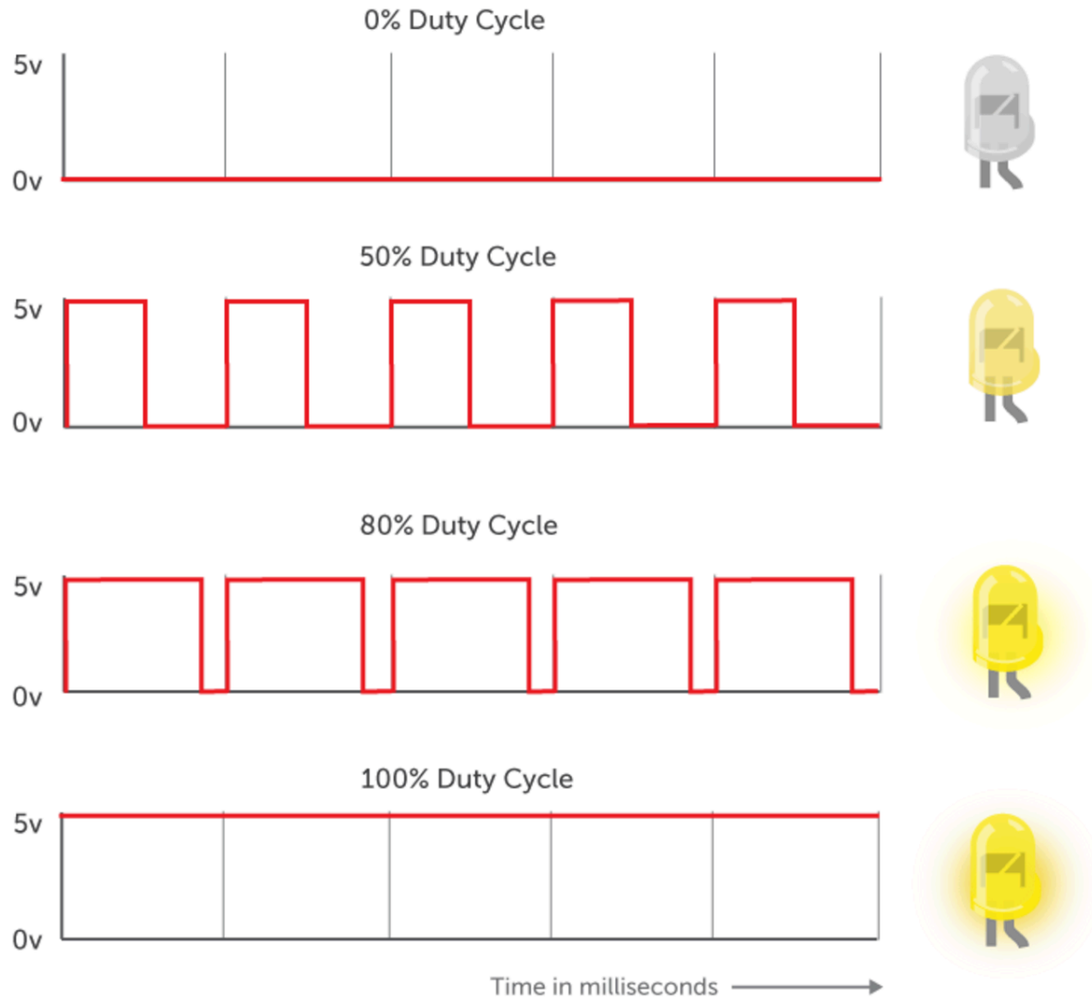
- **Definition:** The Duty Cycle is a measure of the time the modulated signal is in its “high” state.
- It is generally recorded as the percentage of the signal period where the signal is considered on.



PWM Signal

- The percentage duty cycle specifically describes the percentage of time a digital signal is on over an interval or period of time

- Example:



PWM Applications

- In the past, **motors** were controlled at intermediate speed by using resistors to lower delivered power. PWM changed that!

- **Robotics:** PWM is suitable to control the angle of a servo motor attached to something mechanical like a robot arm

- **Dimmers:** Lamp, LED lights

- **Voltage regulation:** convert 12 volts to 5 volts by having a 41.7% duty cycle

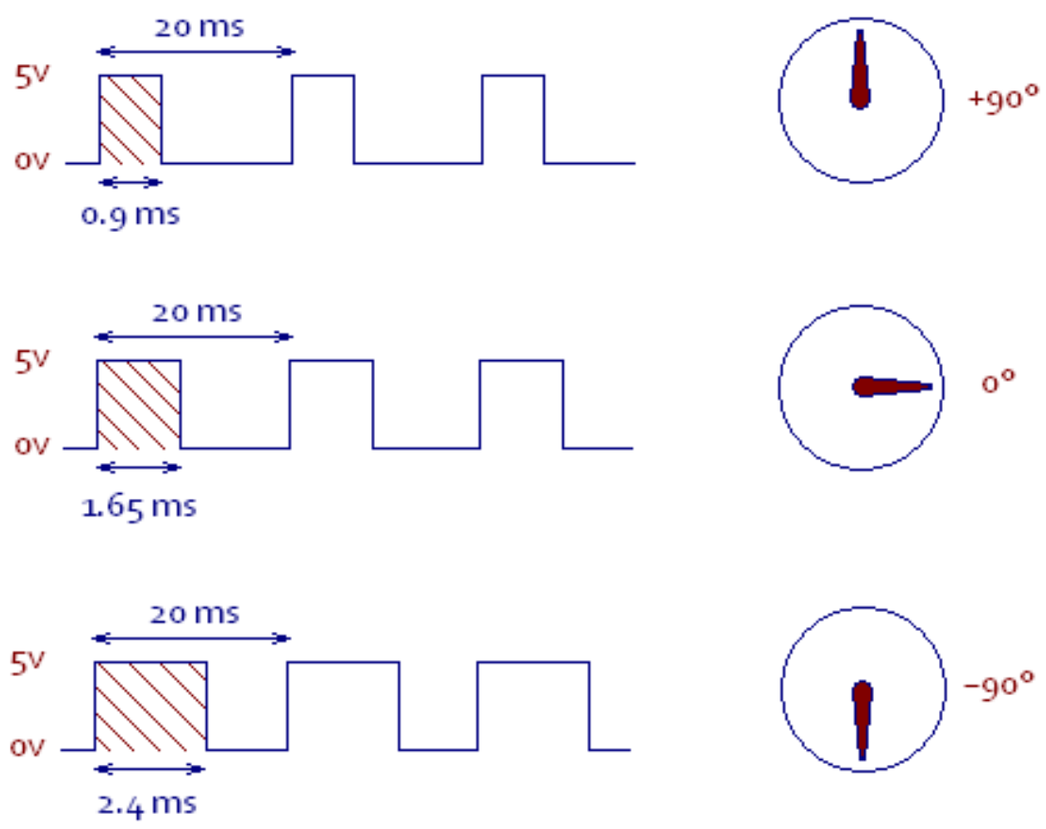
- **Audio signals:** can be easily produced from a PWM signal adjusting frequency and duty cycle variations and using a RC filter.

- **Power transfer:** PWM used to reduce the total power given to a load without relying on resistive losses

PWM Frequency Selection

- PWM controlled applications may use different frequency

- Dimming an LED: 100Hz - 20% (duty cycle)



- Controlling a Servo: 50Hz - 20% (duty cycle)

How I can generate PWM Signals

*Using an 8bit general purpose micro-controller such as a **PIC***

- The CCP(Capture/Compare/PWM) module is a peripheral which allows the user to time and control different events.

"**How to**" Steps:

- Set CCP1 in **PWM mode**
- Choose the associated **timer**
- Set PWM **Frequency**
- Start **timer**
- Set PWM Duty Cycle

How to Generate PWM Signals on PIC

1. Use Delay functions

example: ***__delay_ms(1000);***

2. Using an 8bit general purpose micro-controller such as a ***PIC***

- PIC PWM Resolution: up to 10 bits

- PWM Control Registers using CCPx module:

  - **PR2** registers (timer)

  - **T2CON** registers (timer)

  - **CCPRxL** registers (duty cycle)

  - **CCPxCON** registers (control)