CSSE2010

CSSE2010/CSSE7201
Lecture 16

Pulse Width Modulation

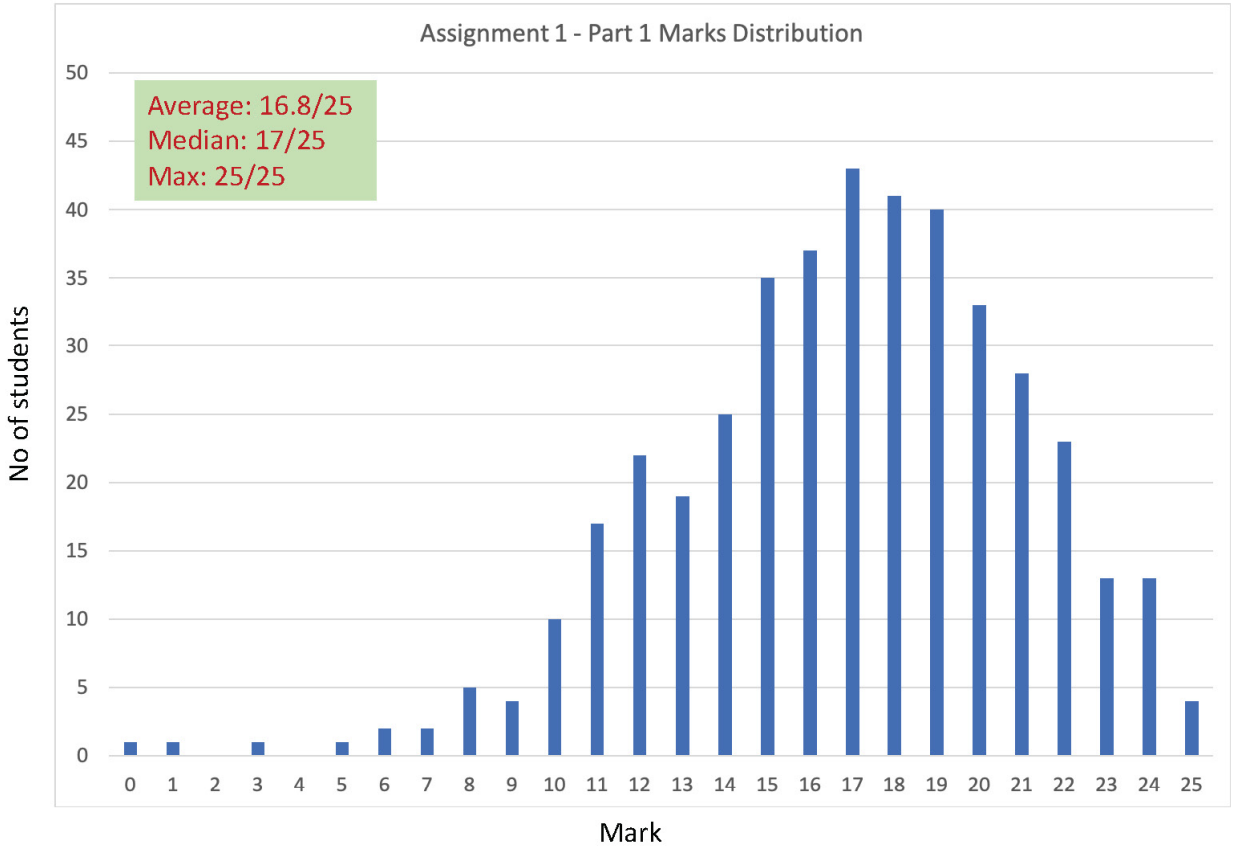
School of Information Technology and Electrical Engineering
The University of Queensland

Outline

- Pulse Width Modulation – Application of timers
  - Needed for Lab 14 (week 9)

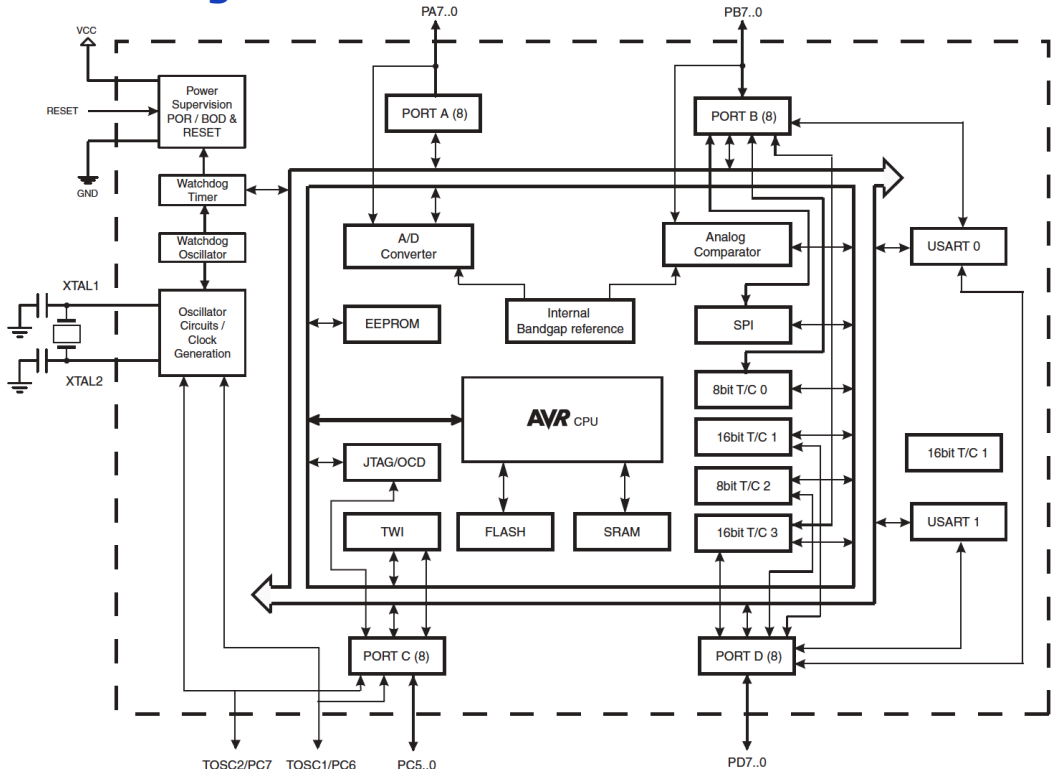
- It is probably best if you revisit this lecture after having done lab 13 this week (AVR Timers)
- Assignment 1 – part 1 marks are released. Part 2 is still being marked.
- Quiz 7 is due this week (week 9) Friday (24/9/21) 4pm

Assignment 1 – Part 1 Marks

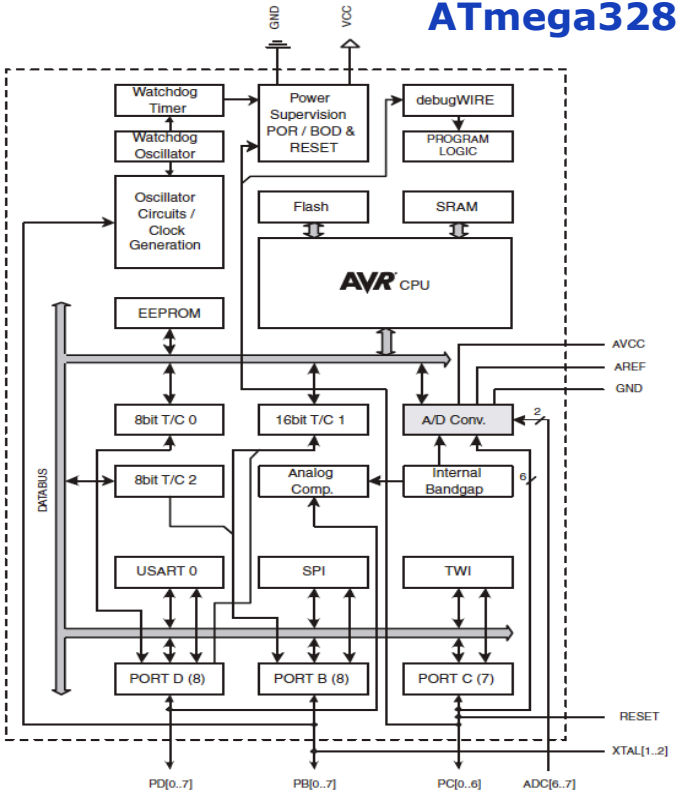


AVR Timers/Counters (last lecture)

ATmega324A



ATmega328

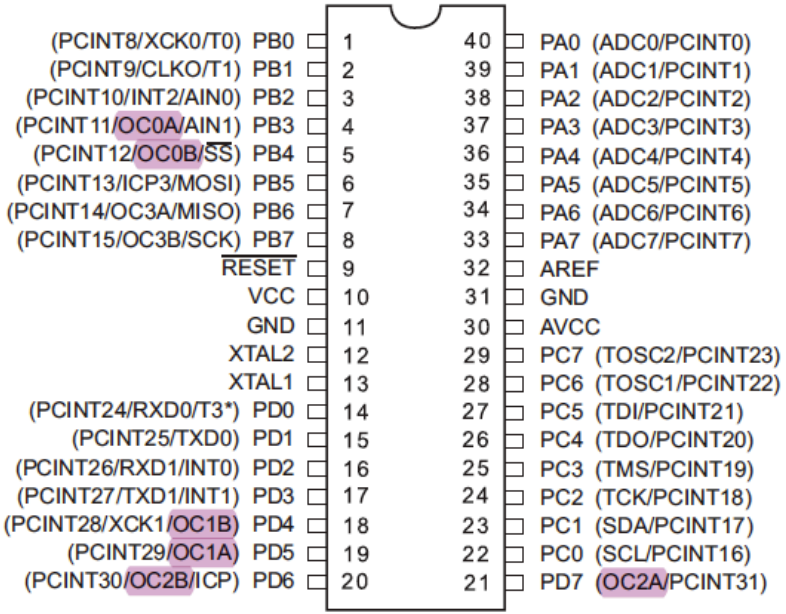


AVR Timers/Counters (last lecture)

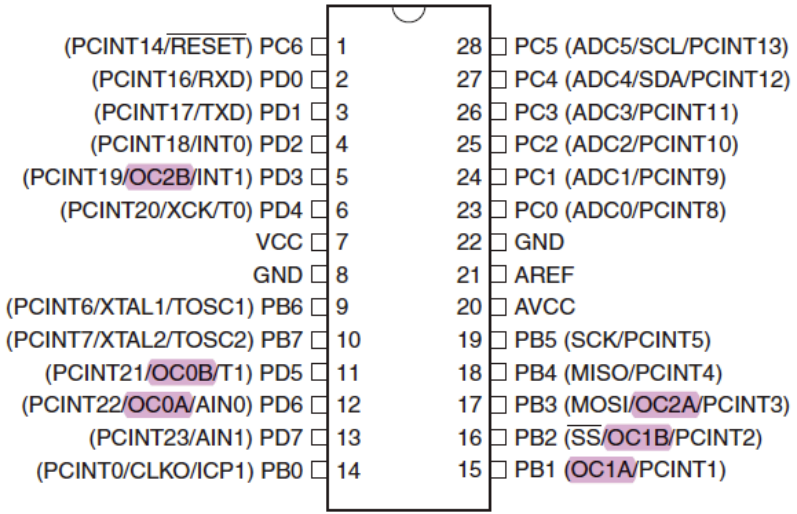
- Can be used to perform actions with pre-defined timing: e.g. do some action in every 100ms
- Can be used to **generate PWM** (today's topic)
- Can be used to **count** pulses (logic transitions on a given pin)
- Separate hardware to CPU
- Each timer/counter unit is tied to particular pins on the microcontroller. By configuring the timer/counter these pins are changed from their default GPIO operation to timer/counter operation.
- Configuration includes reading/writing to I/O registers:
  - Timer/Counter register
  - Control registers
  - Output compare registers
  - Interrupt registers

AVR Timers/Counters (last lecture)

ATmega324A



ATmega328P



AVR Timers/Counters Summary – ATmega324A

Timer/Counter 0

Pins: PB3 (OC0A) & PB4 (OC0B)

8-bit timer/counter

Supports PWM

Modes of operation:

- Normal, CTC, Fast-PWM and phase correct PWM

Clock prescalar: No clock, F, F/8, F/64, F/256, F/1024

I/O Registers:

- TCNT0
- TCCR0A, TCCR0B
- OCR0A, OCR0B
- TIMSK0, TIFR0

Timer/Counter 1

Pins: PD5 (OC1A) & PD4 (OC1B)

16-bit timer/counter

Supports PWM

Modes of operation: Normal, CTC, Fast-PWM, PC-PWM and PFC-PWM

Clock prescalar: No clock, F, F/8, F/64, F/256, F/1024

I/O Registers:

- TCNT1H, TCNT1L
- TCCR1A, TCCR1B, TCCR1C
- OCR1AH, OCR1AL
- OCR1BH, OCR1BL
- TIMSK1, TIFR1

Timer/Counter 2

Pins: PD7 (OC2A) & PD6 (OC2B)

8-bit timer/counter

Supports PWM

Modes of operation:

- Normal, CTC, Fast-PWM and phase correct PWM

Clock prescalar: No clock, F, F/8, F/32, F/64, F/128, F/256, F/1024

I/O Registers:

- TCNT2
- TCCR2A, TCCR2B
- OCR2A, OCR2B
- TIMSK2, TIFR2

AVR Timers/Counters Summary – ATmega328P

Timer/Counter 0

Pins: PD6 (OC0A) & PD5 (OC0B)

8-bit timer/counter

Supports PWM

Modes of operation:

- Normal, CTC, Fast-PWM and phase correct PWM

Clock prescalar: No clock, F, F/8, F/64, F/256, F/1024

I/O Registers:

- TCNT0
- TCCR0A, TCCR0B
- OCR0A, OCR0B
- TIMSK0, TIFR0

Timer/Counter 1

Pins: PB1 (OC1A) & PB2 (OC1B)

16-bit timer/counter

Supports PWM

Modes of operation: Normal, CTC, Fast-PWM, PC-PWM and PFC-PWM

Clock prescalar: No clock, F, F/8, F/64, F/256, F/1024

I/O Registers:

- TCNT1H, TCNT1L
- TCCR1A, TCCR1B, TCCR1C
- OCR1AH, OCR1AL
- OCR1BH, OCR1BL
- TIMSK1, TIFR1

Timer/Counter 2

Pins: PB3 (OC2A) & PD3 (OC2B)

8-bit timer/counter

Supports PWM

Modes of operation:

- Normal, CTC, Fast-PWM and phase correct PWM

Clock prescalar: No clock, F, F/8, F/32, F/64, F/128, F/256, F/1024

I/O Registers:

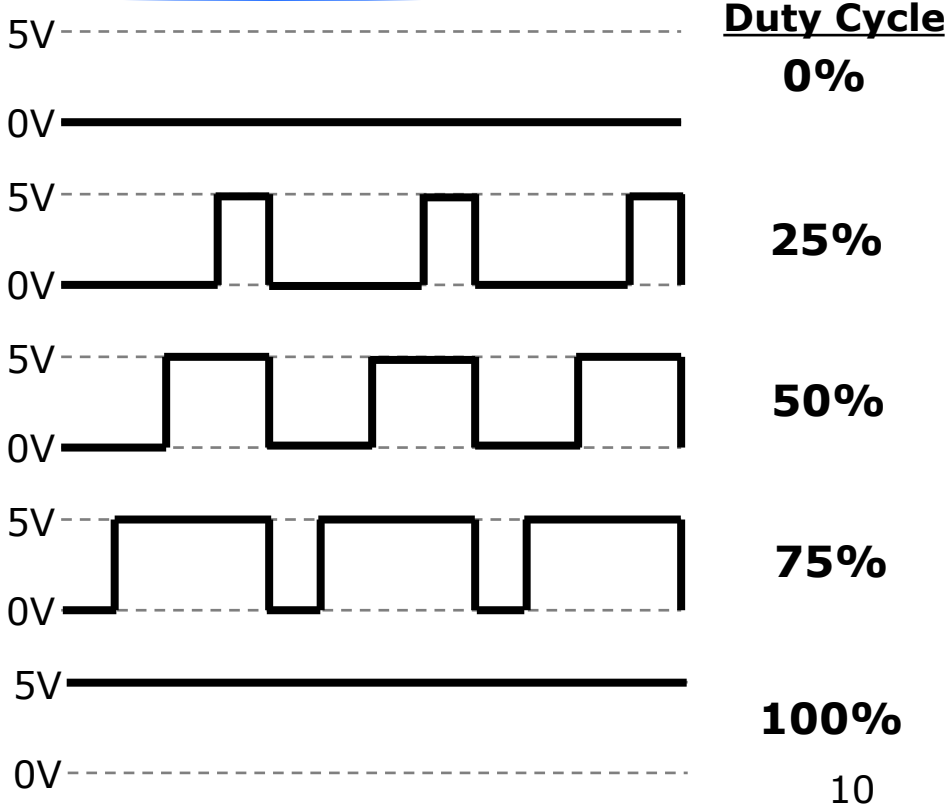
- TCNT2
- TCCR2A, TCCR2B
- OCR2A, OCR2B
- TIMSK2, TIFR2

[Recall from Lab 13 – AVR Timers] Output Compare Registers

- Each timer/counter has **output compare** registers (these are I/O registers)
  - These are for matching timer/counter values
- Actions can be taken when the value is reached, e.g.
  - Set output-compare match bit in register
  - Clear timer (reset to 0)
  - Toggle / set / clear external pin

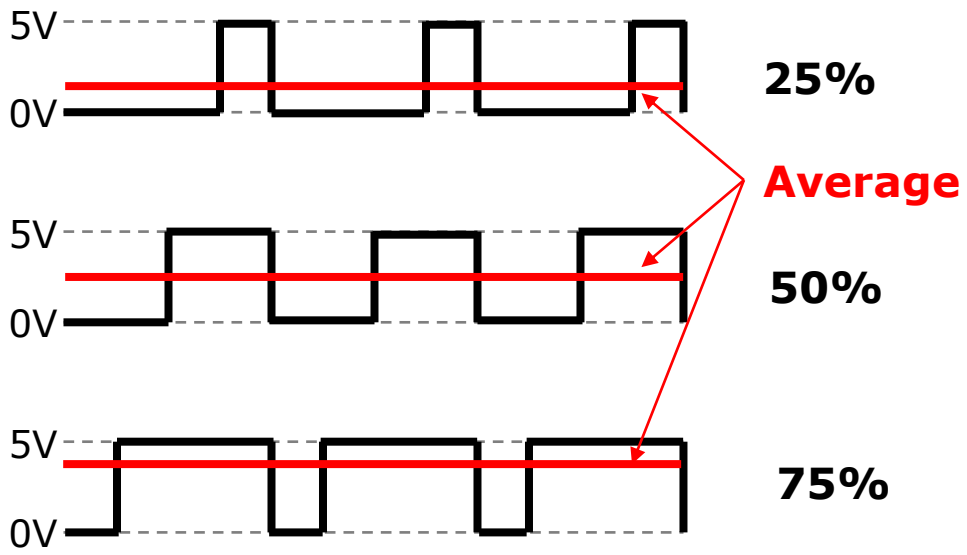
Pulse Width Modulation (PWM)

- Varying the duty cycle of a periodic pulse
- **Duty cycle** = % of time that signal is on (high)



PWM – Average Value

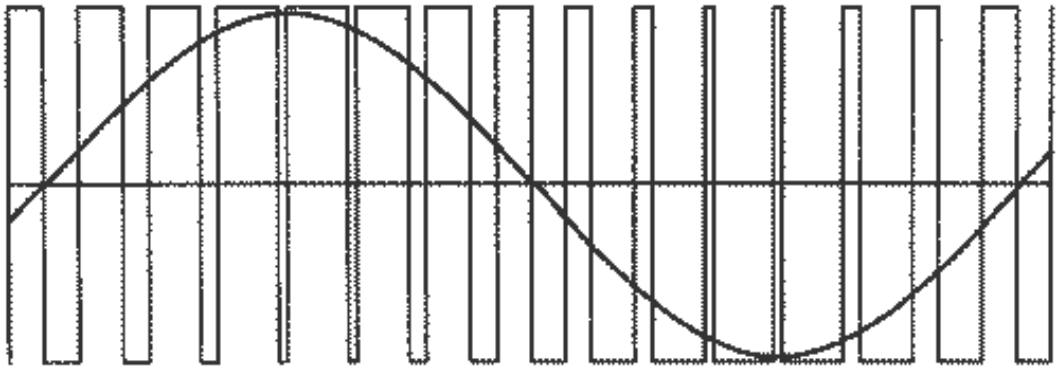
- If periodic pulse is "fast enough" – signal looks like the average value



Implies that PWM can be used to control the average voltage delivered to a load → controlling the speed of a DC motor

PWM – Average Value (cont.)

- Pulse width can be varied so that the "average value" changes over time

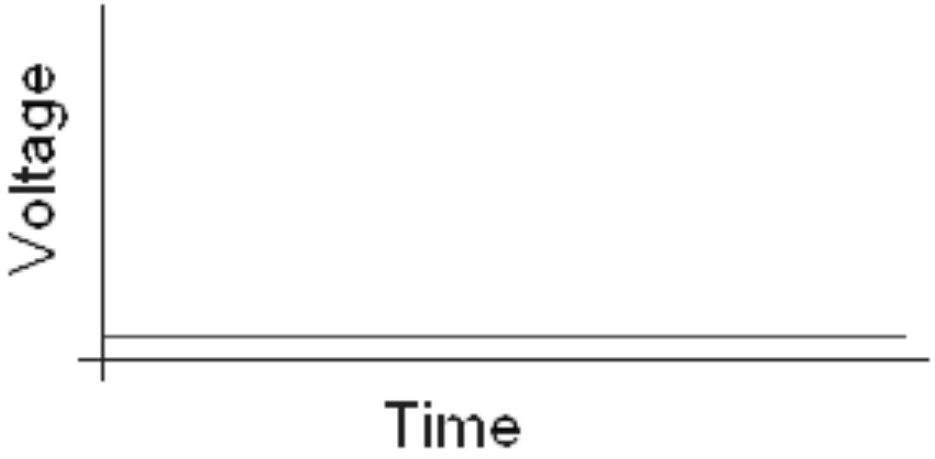


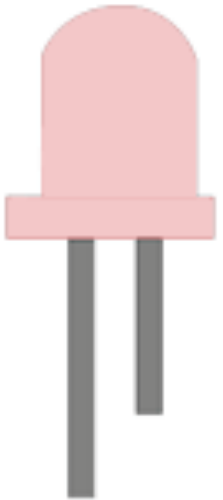
[electronics.stackexchange.com]

Uses of PWM

- PWM is useful for generating signals that look analog from a purely digital source
- Can be used for
  - Varying brightness of lights/LEDs (e.g. fading)
  - Motor control (variable speed)
  - Generating audio (with appropriate filtering)

LED Brightness Fading





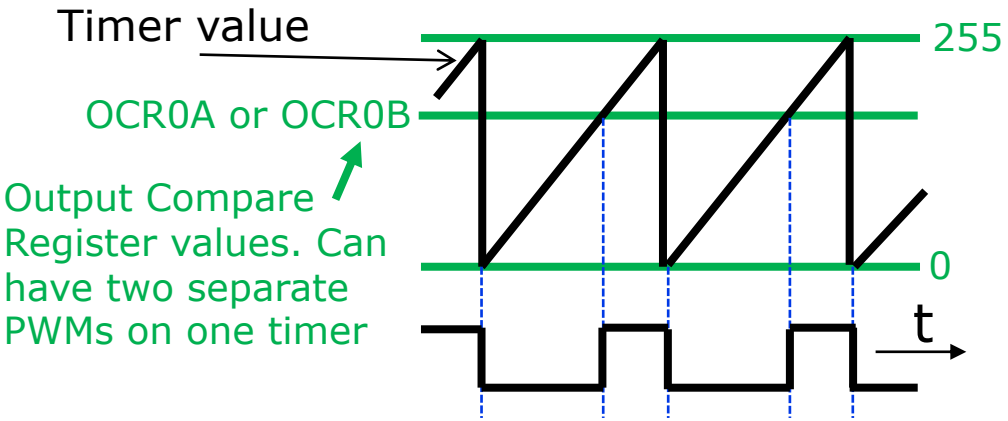
Duty Cycle: 0%

[From www.pyroelectro.com]

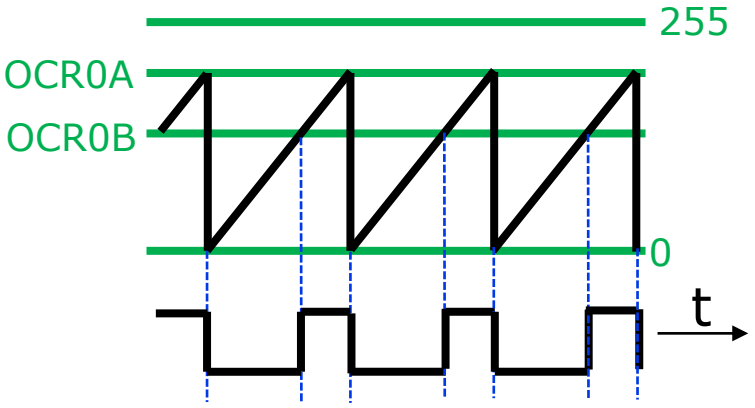
PWM on the AVR: Fast PWM Mode (Example with Timer/Counter 0)

- Two options – Waveform Generation Mode bits (recall other modes from lab 13)

WGM[2:0] = **011** (0x3)

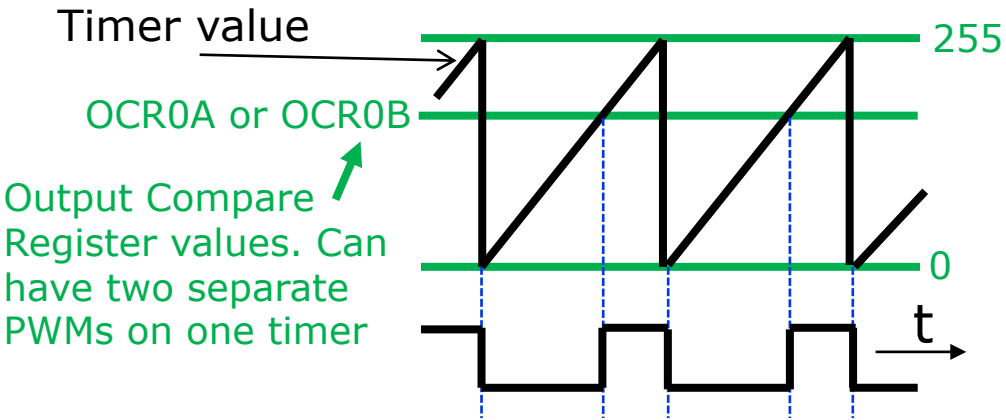


WGM[2:0]= **111** (0x7)



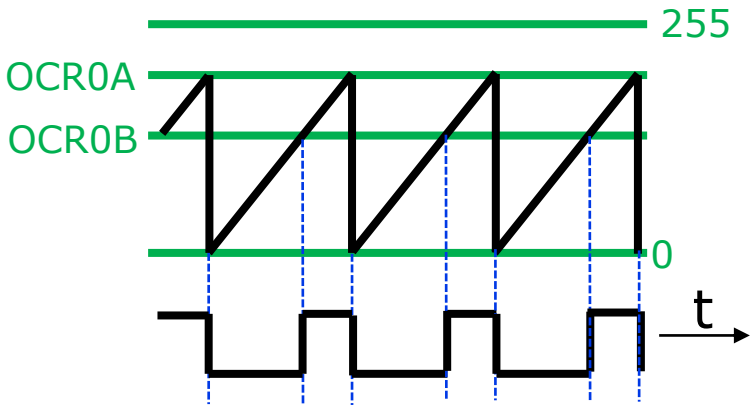
Fast PWM Mode

WGM[2:0] = 011 (0x3)



PWM on the AVR (cont.)

WGM[2:0]= 111 (0x7)



Phase Correct PWM

- Centres of pulses remain at same point in period
- Better for motor speed control
- Implemented by having timer count up and then down
- We won't use this – just FAST PWM mode