

# CSSE2010/CSSE7201

## Lecture 16

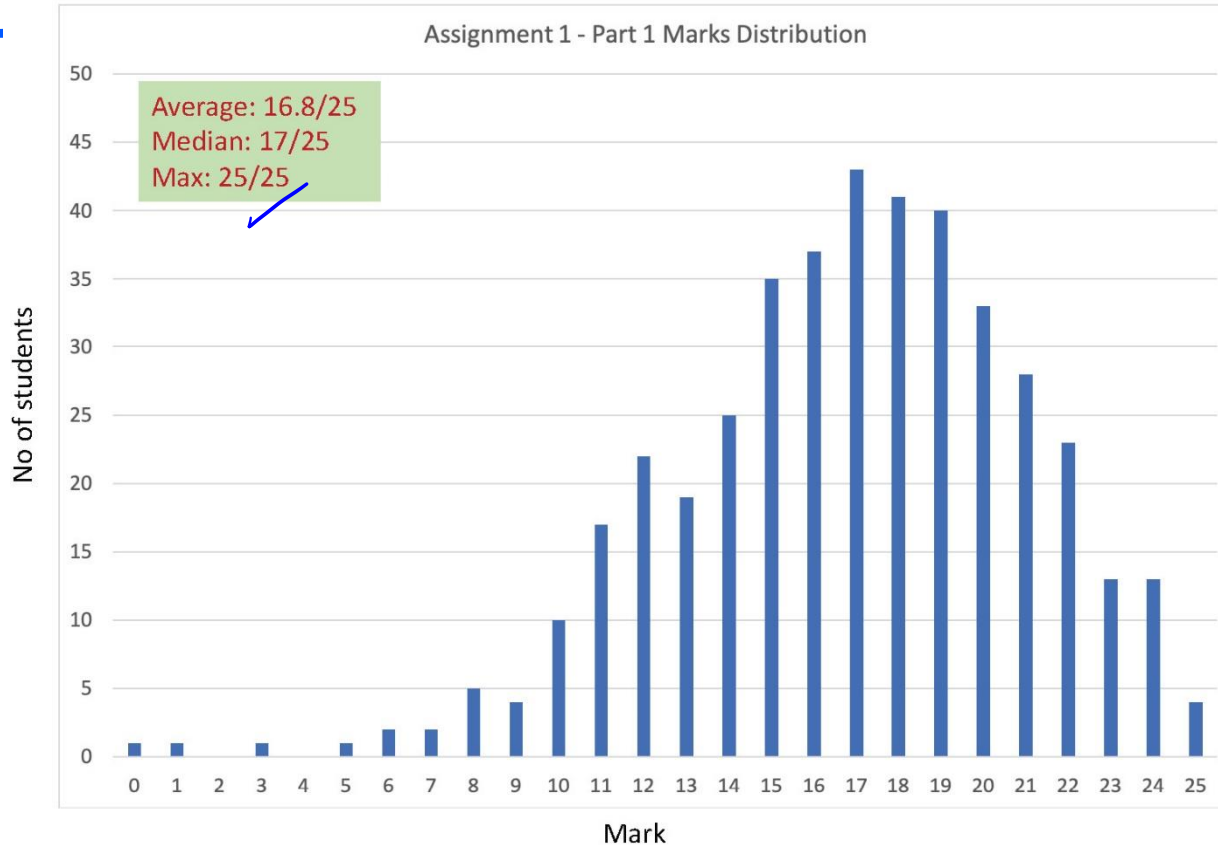
# Pulse Width Modulation (PWM)

School of Information Technology and Electrical Engineering  
The University of Queensland

# Outline

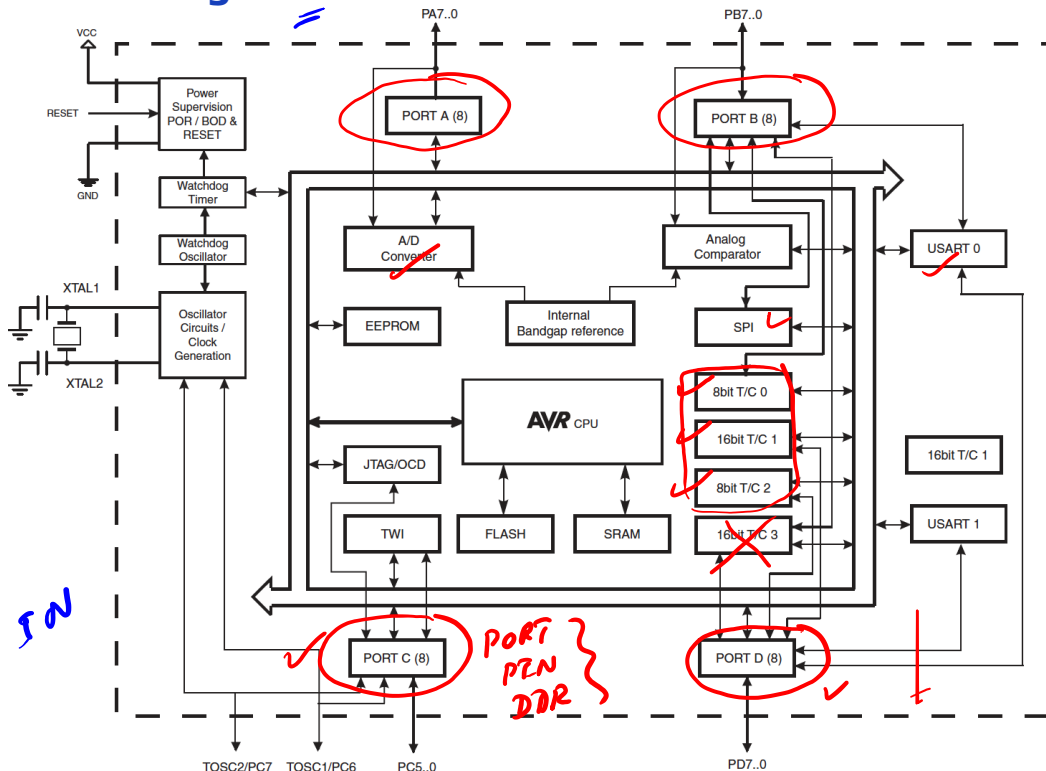
- Pulse Width Modulation – Application of timers
  - ✓ ■ Needed for Lab 14 (week 9)
- It is probably best if you revisit this lecture after having done lab 13 this week (AVR Timers)
- Assignment 1 – part 1 marks are released. Part 2 is still being marked.
- Quiz 7 is due this week (week 9) Friday (24/9/21) 4pm

# Assignment 1 – Part 1 Marks

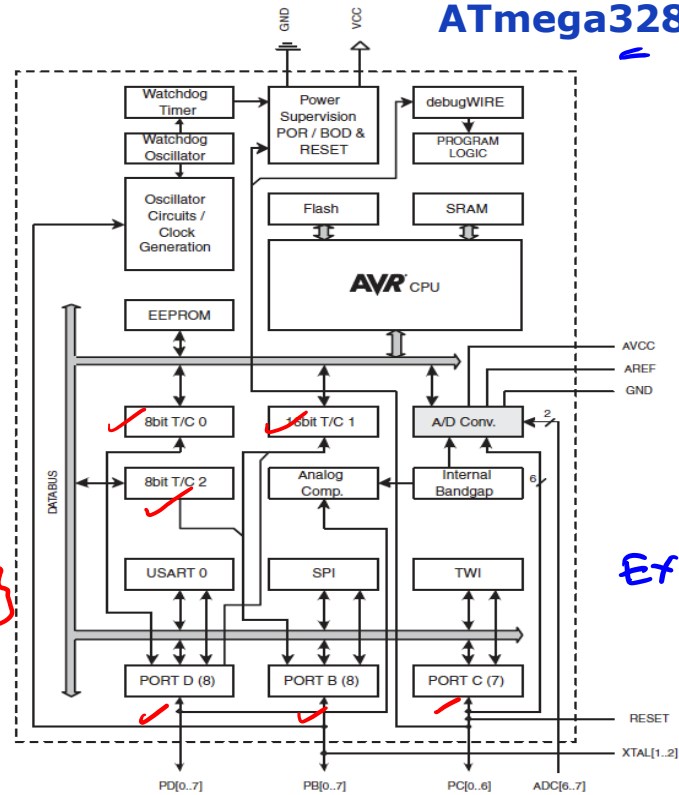


# AVR Timers/Counters (last lecture)

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

(PCINT8/XCK0/T0) PB0	1	40	PA0 (ADC0/PCINT0)
(PCINT9/CLKO/T1) PB1	2	39	PA1 (ADC1/PCINT1)
(PCINT10/INT2/AIN0) PB2	3	38	PA2 (ADC2/PCINT2)
(PCINT11/OC0A/AIN1) PB3	4	37	PA3 (ADC3/PCINT3)
(PCINT12/OC0B/SS) PB4	5	36	PA4 (ADC4/PCINT4)
(PCINT13/ICP3/MOSI) PB5	6	35	PA5 (ADC5/PCINT5)
(PCINT14/OC3A/MISO) PB6	7	34	PA6 (ADC6/PCINT6)
(PCINT15/OC3B/SCK) PB7	8	33	PA7 (ADC7/PCINT7)
RESET	9	32	AREF
VCC	10	31	GND
GND	11	30	AVCC
XTAL2	12	29	PC7 (TOSC2/PCINT23)
XTAL1	13	28	PC6 (TOSC1/PCINT22)
(PCINT24/RXD0/T3*) PD0	14	27	PC5 (TDI/PCINT21)
(PCINT25/TXD0) PD1	15	26	PC4 (TDO/PCINT20)
(PCINT26/RXD1/INT0) PD2	16	25	PC3 (TMS/PCINT19)
(PCINT27/TXD1/INT1) PD3	17	24	PC2 (TCK/PCINT18)
(PCINT28/XCK1/OC1B) PD4	18	23	PC1 (SDA/PCINT17)
(PCINT29/OC1A) PD5	19	22	PC0 (SCL/PCINT16)
(PCINT30/OC2B/ICP) PD6	20	21	PD7 (OC2A/PCINT31)

## ATmega328P

(PCINT14/RESET) PC6	1	28	PC5 (ADC5/SCL/PCINT13)
(PCINT16/RXD) PD0	2	27	PC4 (ADC4/SDA/PCINT12)
(PCINT17/TXD) PD1	3	26	PC3 (ADC3/PCINT11)
(PCINT18/INT0) PD2	4	25	PC2 (ADC2/PCINT10)
(PCINT19/OC2B/INT1) PD3	5	24	PC1 (ADC1/PCINT9)
(PCINT20/XCK/T0) PD4	6	23	PC0 (ADC0/PCINT8)
VCC	7	22	GND
GND	8	21	AREF
(PCINT6/XTAL1/TOSC1) PB6	9	20	AVCC
(PCINT7/XTAL2/TOSC2) PB7	10	19	PB5 (SCK/PCINT5)
(PCINT21/OC0B/T1) PD5	11	18	PB4 (MISO/PCINT4)
(PCINT22/OC0A/AIN0) PD6	12	17	PB3 (MOSI/OC2A/PCINT3)
(PCINT23/AIN1) PD7	13	16	PB2 (SS/OC1B/PCINT2)
(PCINT0/CLKO/ICP1) PB0	14	15	PB1 (OC1A/PCINT1)

✓ *IN* *(GPIO)* / T0, 1, 2

✓ *EX*

# AVR Timers/Counters Summary

## - ATmega324A

$$F = 8 \text{ MHz}$$

### 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 prescaler: 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 prescaler: 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 prescaler: 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 prescaler: **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 prescaler: **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 prescaler: **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)

$$V_{avg} = \frac{1}{T} \int_0^T V(t) dt$$

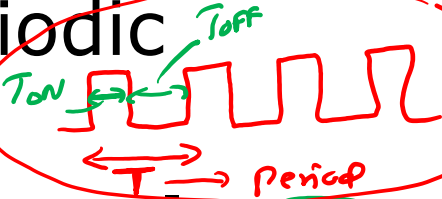
$$V_{avg} = \frac{T_{ON} \cdot 5}{T}$$

- Varying the duty cycle of a periodic pulse

## Duty cycle

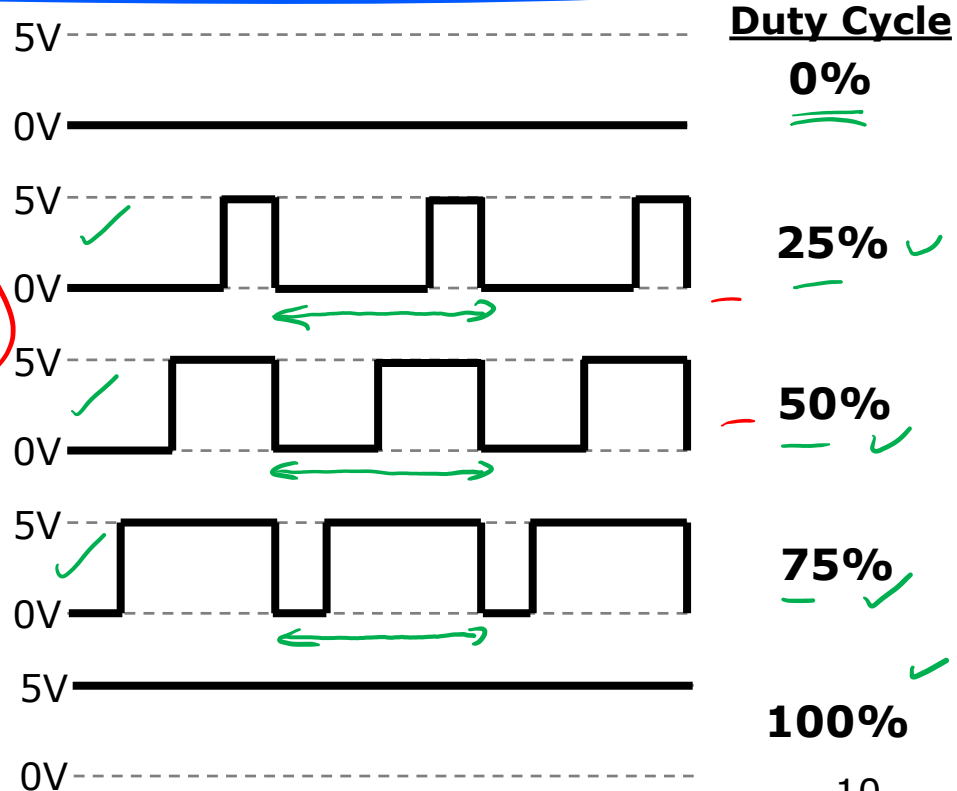
= % of time that signal is on (high)

Vary the Duty cycle



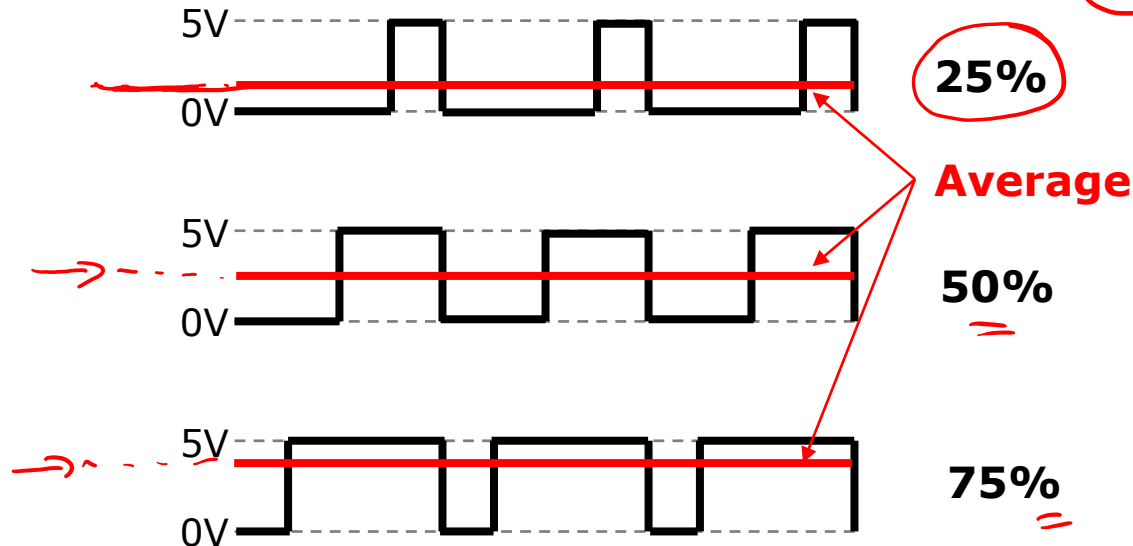
$$F = \frac{1}{T}$$
 frequency.

$$\frac{T_{ON}}{T_{ON} + T_{OFF}} \times 100\%$$



# PWM – Average Value

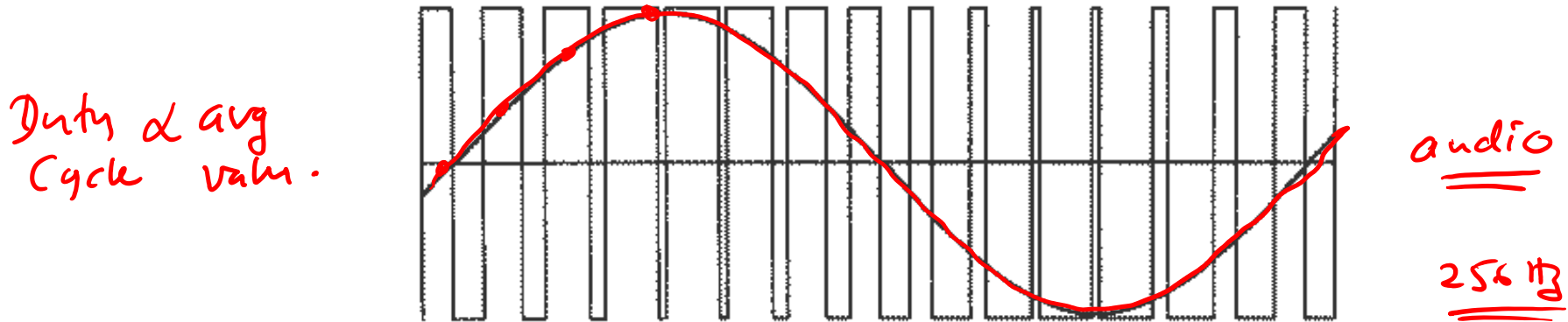
- If periodic pulse is “fast enough” – signal looks like the average value



**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

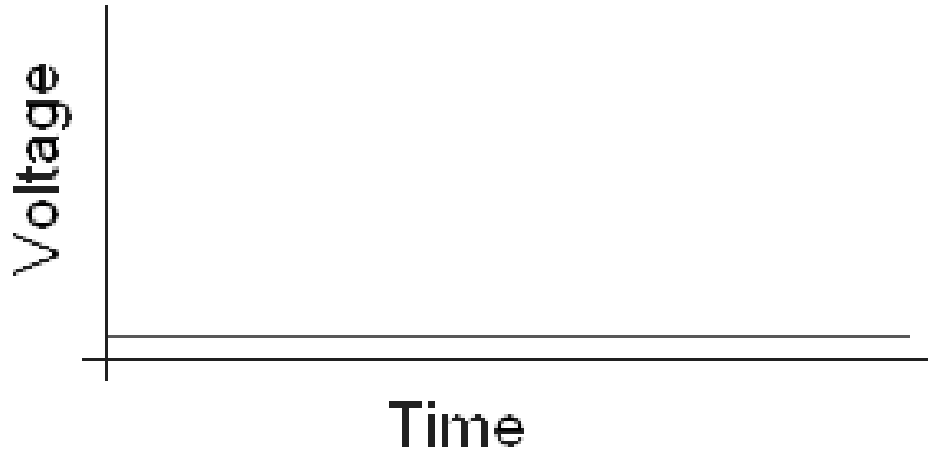


[electronics.stackexchange.com]

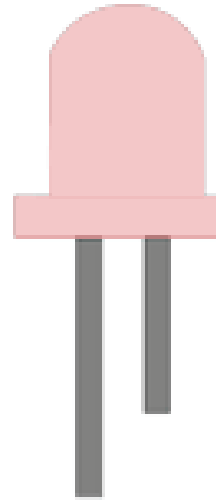
# Uses of PWM

- 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%



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

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

①  
2 PWM signals possible

$$\text{WGM}[2:0] = \mathbf{011} (0x3)$$

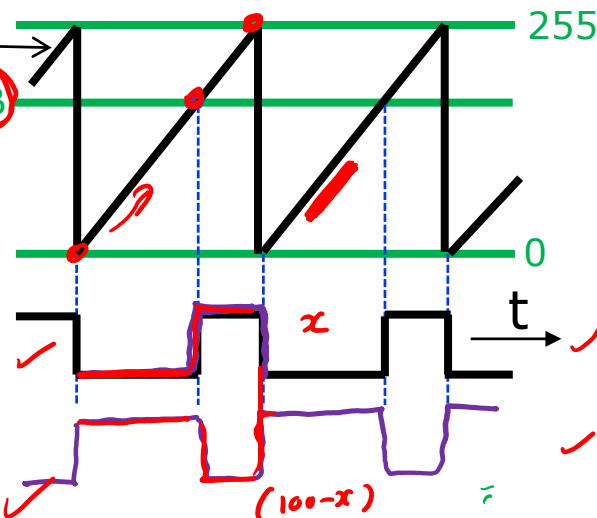
Timer value

OCR0A or OCR0B

Output Compare Register values. Can have two separate PWMs on one timer

Inverting PWM

Non-inverting PWM

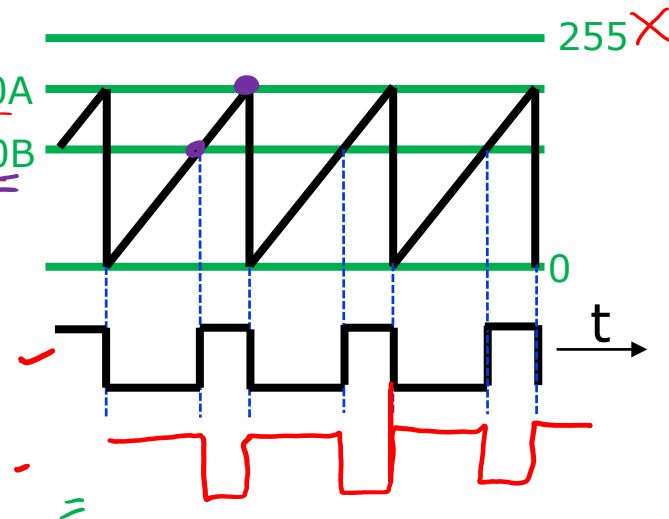


②

only 1 PWM o/p is possible

$$\text{WGM}[2:0] = \mathbf{111} (0x7)$$

OCR0A  
OCR0B



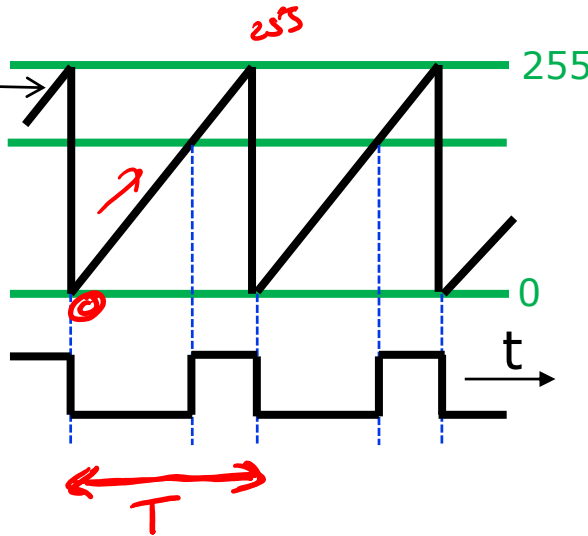
# Fast PWM Mode

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

Timer value

OCR0A or OCR0B

Output Compare Register values. Can have two separate PWMs on one timer



$$F = \frac{1}{T}$$

→ 2 PWM signals possible

$$\text{Time taken per one tick} = \frac{1}{(\text{CLK}/\text{PRE})}$$

256 ticks

$$\therefore T = \frac{256}{\text{CLK}/\text{PRE}}$$

$$\therefore F = \frac{\text{CLK}}{\text{PRE} \cdot 256}$$

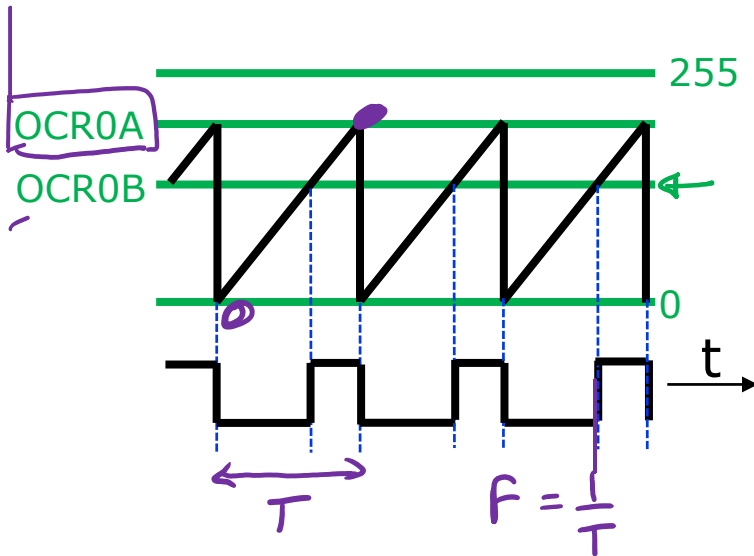
Duty cycle is controlled by the value written to OCR0A/B

(Ex)  
8 MHz  
16 MHz



# PWM on the AVR (cont.)

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



→ only 1 PWM ocp is possible

$$\therefore T = \underbrace{(1 + \text{OCR0A})}_{\text{how many ticks}} \cdot \underbrace{\frac{1}{(\text{CLK}/\text{PRE})}}_{\text{time per one tick}}$$

$$F = \frac{\text{CLK}}{\text{PRE} (1 + \text{OCR0A})}$$

Duty cycle is controlled by the value written to OCR0B

# 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

Duty Cycle

