CSSE2010

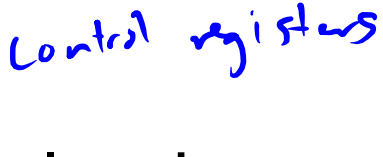
CSSE2010/CSSE7201
Learning Lab 13

AVR Timers

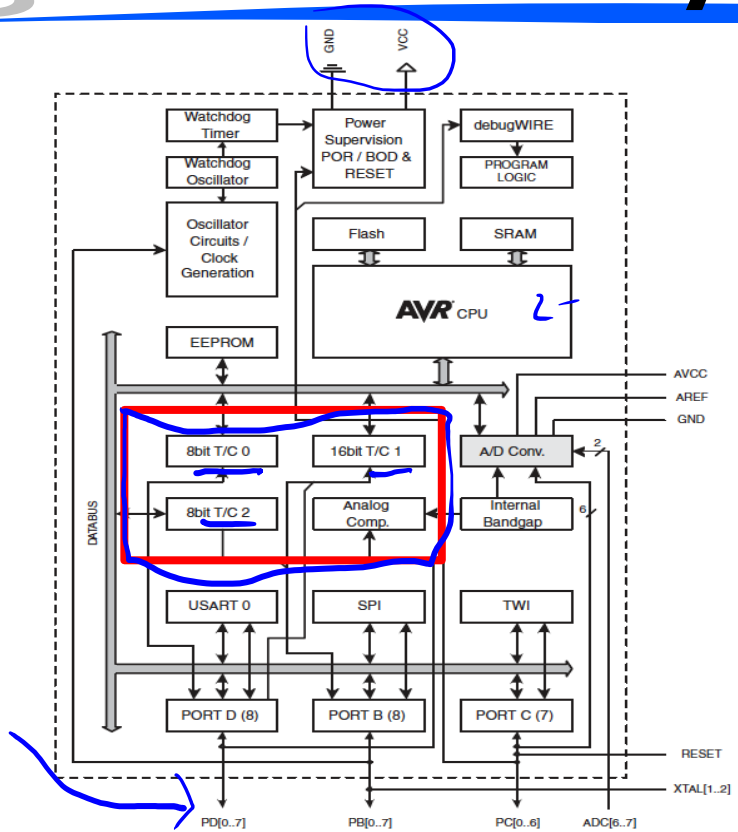
School of Information Technology and Electrical Engineering
The University of Queensland

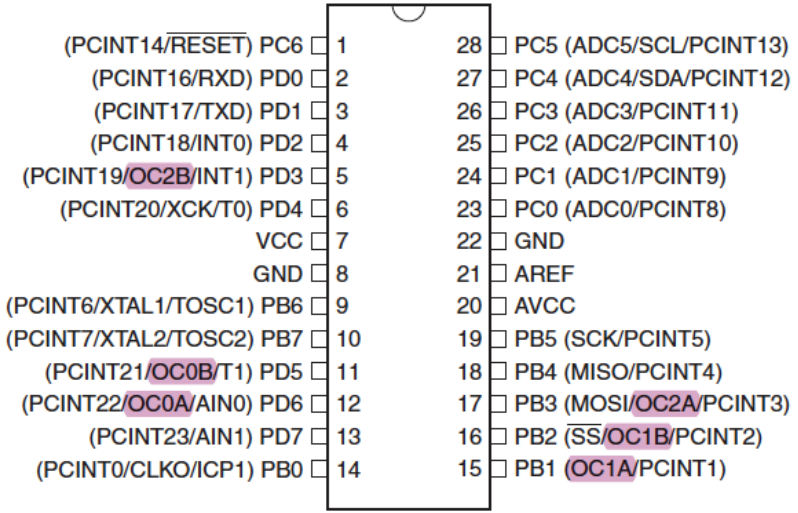
Today

- AVR Timers
  - I/O registers involved
  - How to set up a timer
- Exercises



ATmega328P – Timer/Counter Summary – From Lectures





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

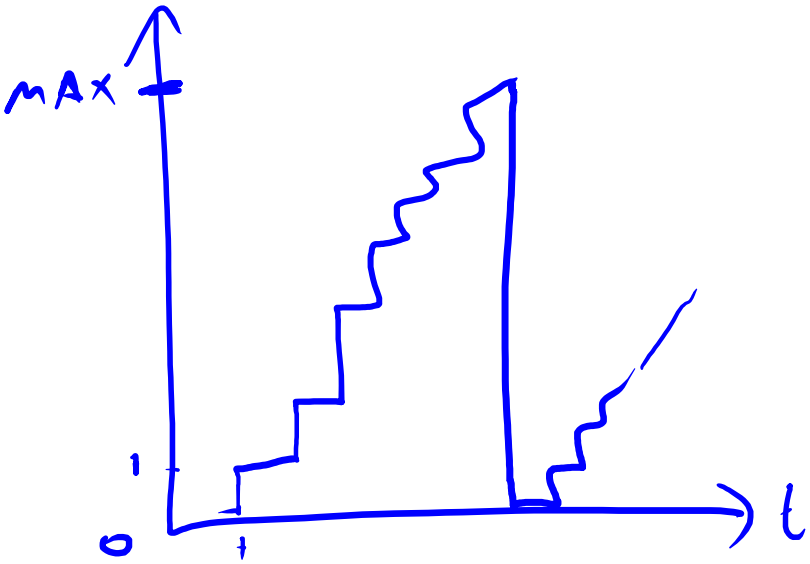
Modes of operation:

- 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

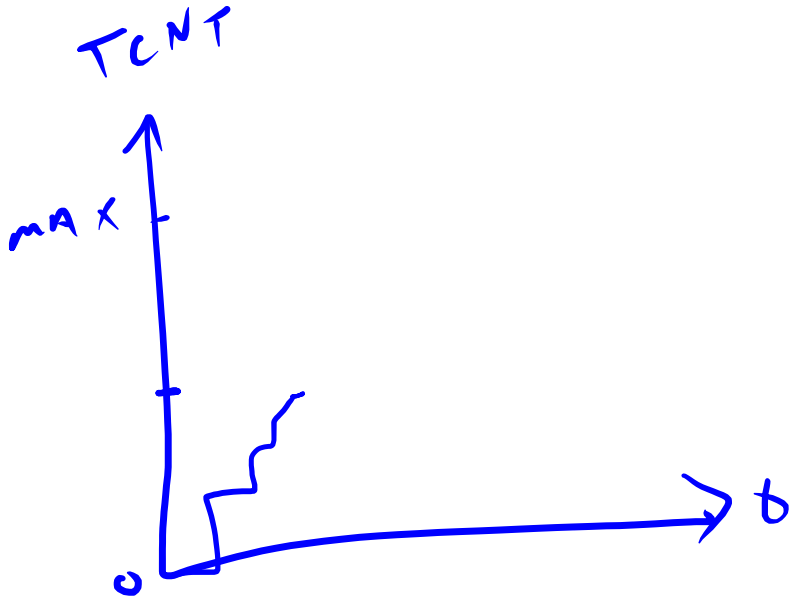
I/O Registers:

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


MAX
1
0
t

Recall from Lecture 15: ATmega324A – Timer/Counter Clock sources

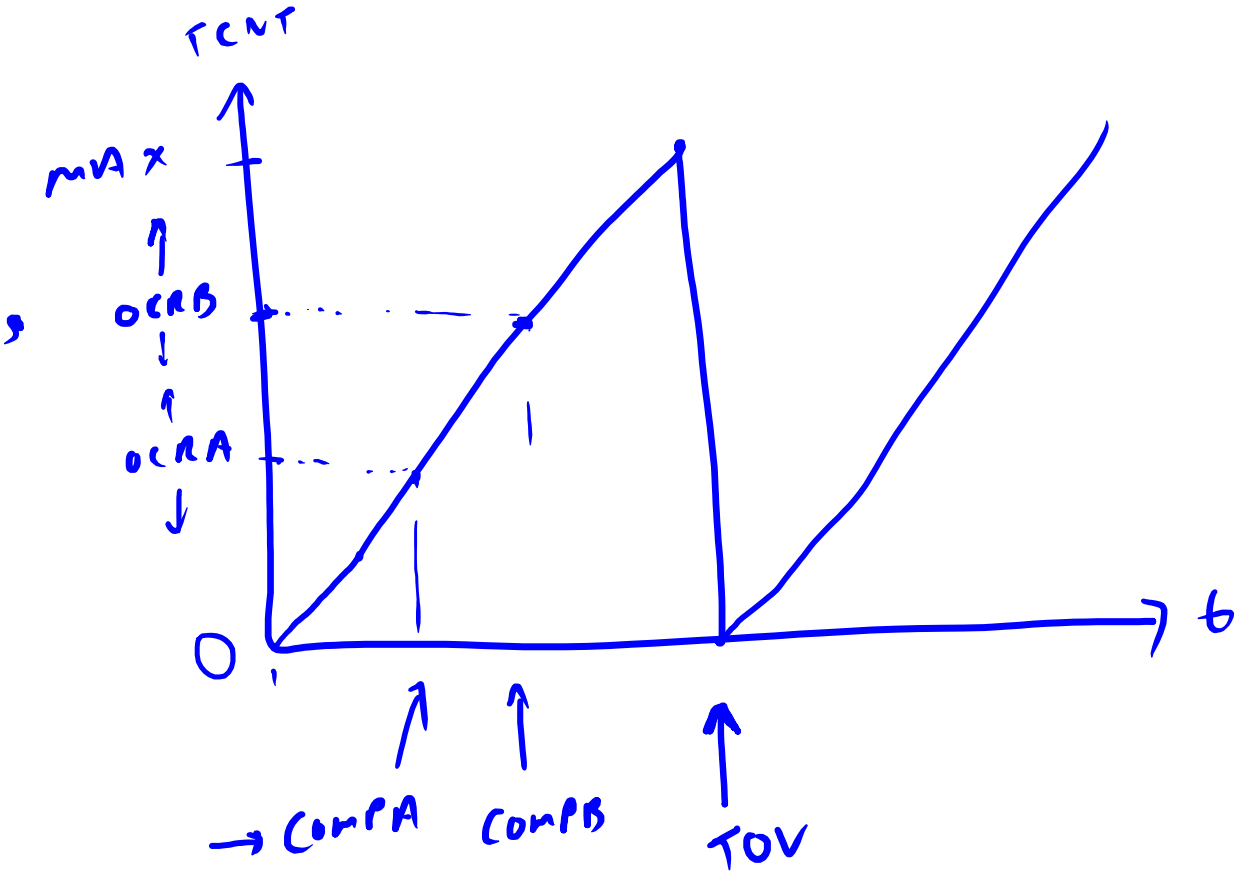
- 3 timer/counters
  - **0**: 8 bit (0 to 255)
  - **1**: 16 bit (0 to 65535)
    - Clock sources: STOPPED, CLK, CLK/8, CLK/64, CLK/256, CLK/1024, external pin rising edge, external pin falling edge
      - CLK = system clock
  - **2**: 8 bit (0 to 255)
    - Clock sources: STOPPED, CLK, CLK/8, CLK/32, CLK/64, CLK/128, CLK/256, CLK/1024
      - CLK = system clock or external oscillator


TCNT
MAX
0
t

Timer/Counter Registers
TCNT – count values

- TCNT0
  - Memory address 0x46
  - IO register 0x26
- TCNT1 (16 bits)
  - TCNT1L – memory address 0x84
  - TCNT1H – memory address 0x85
- TCNT2
  - Memory address 0xB2

OCRB = 5



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

Output Compare Registers

- Timer 0 (8 bit)
  - **OCR0A**, **OCR0B**
- Timer 1 (16 bit)
  - **OCR1AH**,**OCR1AL** (in C, access as 16-bit "variable" OCR1A)
  - **OCR1BH**,**OCR1BL** (in C: OCR1B)
- Timer 2 (8 bit)
  - **OCR2A**, **OCR2B**

Accessing 16-bit I/O registers

Example - set output compare register 1B (16-bits) to 4321:

- In assembly language
  - ```
    ldi r16, high(4321)
    sts OCR1BH, r16
    ldi r16, low(4321)
    sts OCR1BL, r16
    ```
- Note that processor requires high byte to be written first
- In C
  - ```
    OCR1B = 4321;
    ```

TCC → config registers

Setting up a 16-bit timer

- 3 control registers, e.g. for timer/counter 1 (n=1)

"n" replaced by 1, e.g. COM1A1 | COM1A0 | COM1B1 | COM1B0

TCCR1A

Bit	7	6	5	4	3	2	1	0
	COMnA1	COMnA0	COMnB1	COMnB0	–	–	WGMn1	WGMn0
Read/Write	R/W	R/W	R/W	R/W	R	R	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

TCCR1B

Bit	7	6	5	4	3	2	1	0
	ICNCn	ICESn	–	WGMn3	WGMn2	CSn2	CSn1	CSn0
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

TCCR1C

Bit	7	6	5	4	3	2	1	0
	FOCnA	FOCnB	–	–	–	–	–	–
Read/Write	R/W	R/W	R	R	R	R	R	R
Initial Value	0	0	0	0	0	0	0	0

setting up a timer

- mode of operation (WGM) ✓
- clock source (CS / PRE) ✓
- link actions to events

WGM (Waveform Generation Modes)

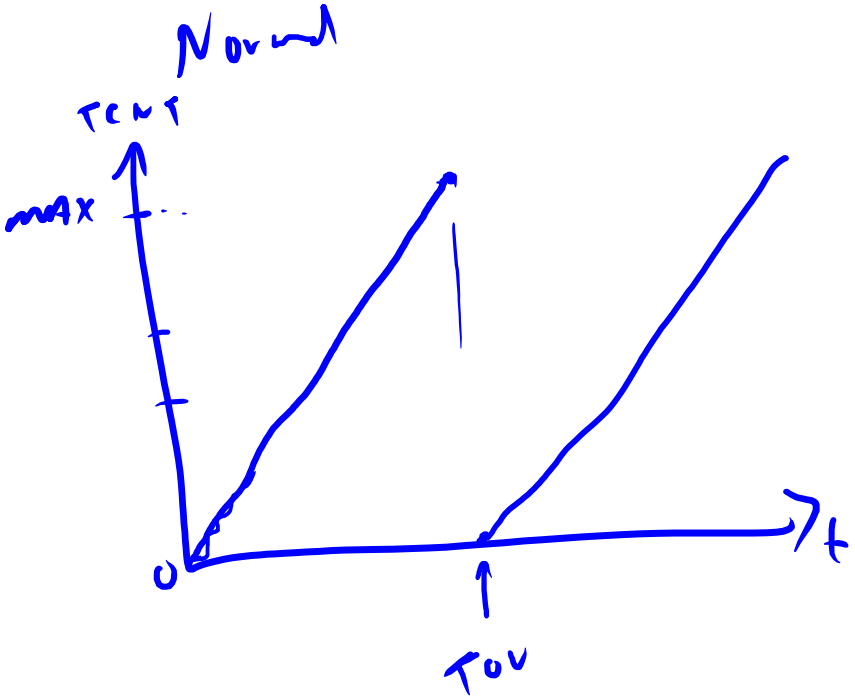
TCCR1A

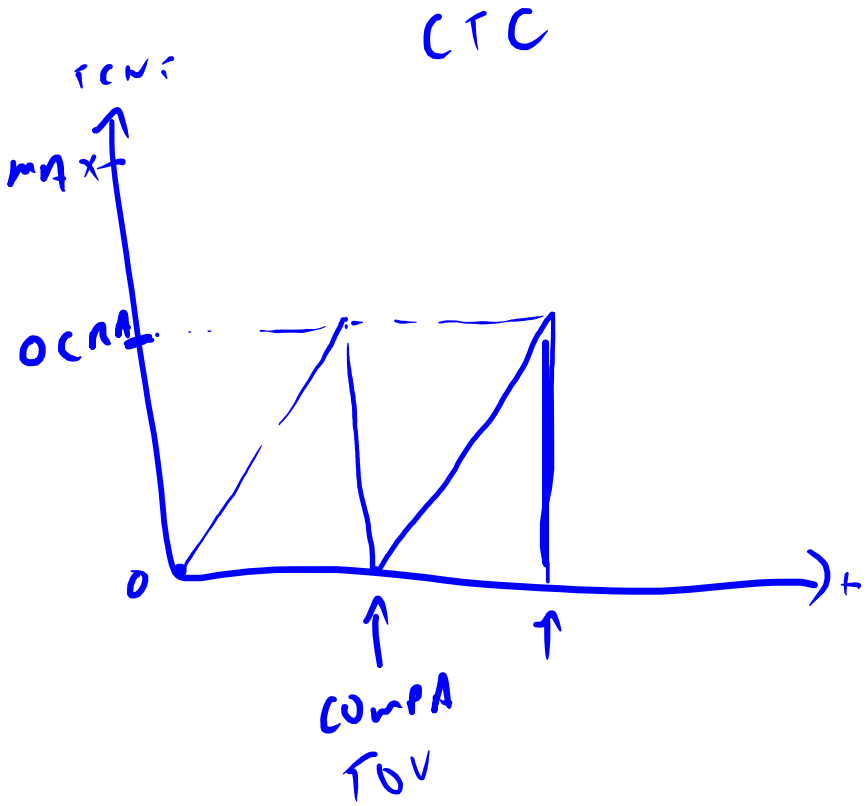
Bit	7	6	5	4	3	2	1	0
	COM1A1	COM1A0	COM1B1	COM1B0	–	–	WGM11	WGM10
Access	R/W	R/W	R/W	R/W			R/W	R/W
Reset	0	0	0	0			0	0

TCCR1B

Bit	7	6	5	4	3	2	1	0
	ICNC1	ICES1	–	WGM13	WGM12	CS12	CS11	CS10
Access	R/W	R/W		R/W	R/W	R/W	R/W	R/W
Reset	0	0		0	0	0	0	0

- Table 16-5 on page 138 of datasheet describes modes
- Two of interest:
  - 0: 0000 – normal: 0 -> 1 -> … -> 65535 -> 0 -> …
  - 4: 0100 – CTC – **C**lear **T**imer on **C**ompare match
    - Counter resets to 0 when reaches value in OCR1A register


Normal
TCNT
MAX
0
t
TOV


CTC
TCNT
MAX
OCRA
0
t
COMPA
TOV

Clock selection

Bit	7	6	5	4	3	2	1	0
TCCR1B	ICNC1	ICES1		WGM13	WGM12	CS12	CS11	CS10
Access	R/W	R/W		R/W	R/W	R/W	R/W	R/W
Reset	0	0		0	0	0	0	0

$f_{tick} = \frac{1}{t}$

- From Table 16-6 on page 139 of datasheet: (note n=1)

CSn2	CSn1	CSn0	Description
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	$clk_{I/O}$/1 (No prescaling)
0	1	0	$clk_{I/O}$/8 (From prescaler)
0	1	1	$clk_{I/O}$/64 (From prescaler)
1	0	0	$clk_{I/O}$/256 (From prescaler)
1	0	1	$clk_{I/O}$/1024 (From prescaler)
1	1	0	External clock source on Tn pin. Clock on falling edge.
1	1	1	External clock source on Tn pin. Clock on rising edge.

$F_{sys} = 16\ MHz$

$f_{tick} = \frac{f_{sys}}{PRE}$

see p11 of datasheet – T1 pin is port B, pin 1

Clock selection – code example

Bit	7	6	5	4	3	2	1	0
TCCR1B	ICNC1	ICES1		WGM13	WGM12	CS12	CS11	CS10
Access	R/W	R/W		R/W	R/W	R/W	R/W	R/W
Reset	0	0		0	0	0	0	0

CS12	CS11	CS10	Description
0	1	1	$clk_{I/O}$/64 (From prescaler)

- Bit labels can be used in code expressions, e.g.

```
TCCR1B = (1<<CS11) | (1<<CS10);
```

Output Compare Actions - can toggle/clear/set pin

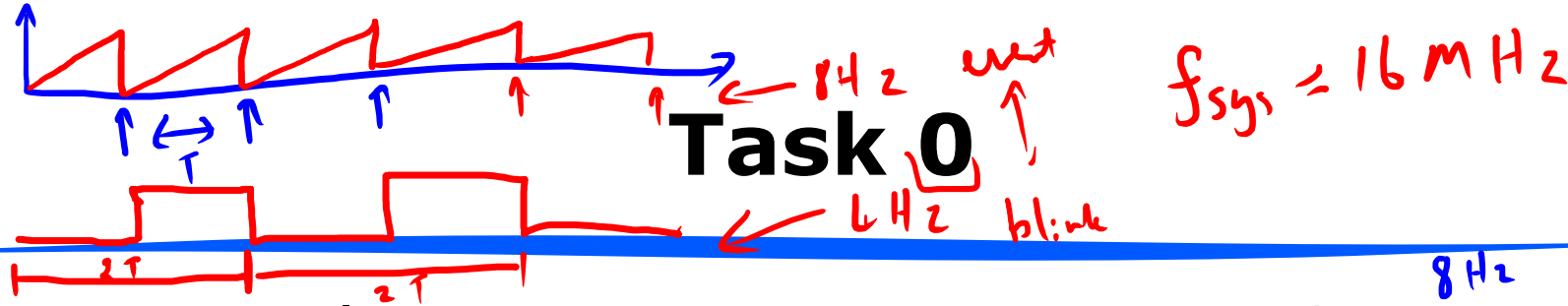
TCCR1A

Bit	7	6	5	4	3	2	1	0
	COM1A1	COM1A0	COM1B1	COM1B0			WGM11	WGM10
Access	R/W	R/W	R/W	R/W			R/W	R/W
Reset	0	0	0	0			0	0

e.g. OC1A pin – see p11 of datasheet

- From table 16-2 on page 136 of datasheet: (n=1)

COMnA1/COMnB1	COMnA0/COMnB0	Description
0	0	Normal port operation, OCnA/OCnB disconnected.
0	1	Toggle OCnA/OCnB on Compare Match.
1	0	Clear OCnA/OCnB on Compare Match (Set output to low level).
1	1	Set OCnA/OCnB on Compare Match (Set output to high level).



Task 0

- To toggle an output compare pin 8 times per second (4Hz period), what
  - Clock prescale PRE or N
  - Output compare OCR

  values do we need?

1) all values should be integers.
2) values $\leq$ MAX

$$f_{blink} = \frac{f_{sys}}{2 \cdot N \cdot (1 + OCRA)}$$

$$f_{event} = \frac{f_{sys}}{N \cdot (1 + OCRA)}$$

$$OCRA = \frac{f_{sys}}{f_{uart}} \cdot \frac{1}{N} - 1$$

$$= \frac{16 \times 10^6 \text{ Hz}}{8 \text{ Hz}} \cdot \frac{1}{N} - 1 = \frac{2 \times 10^6}{N} - 1$$

guess PRE = 1

$$OCRA = 2 \times 10^6 - 1$$

$$= 1,999,999$$

max $OCROA/2A = 255$

$OCR1A = 65535$

guess PRE = 1024

$$OCRA = \frac{2 \times 10^6}{1024} - 1$$

$$= 1952.125$$

guess PRE = 64

$$OCRA = \frac{2 \times 10^6}{64} - 1$$

$$= 31249 \checkmark$$

Task 1 - Output Compare Based Timer

- Consider lab13-1.c (on Blackboard)
- Code is to toggle OC1A pin (what pin is this?) 8 times per second (i.e. 4Hz period)
  - This pin is connected to an LED (any is fine)
- Fill in the blanks to make the code work
  - Set up the hardware to do the work, software does nothing after that
- Build and test your code on the AVR board
- Look at the generated list (lss) file for assembly language equivalent
  - Project directory -> Debug -> <project name>.lss

Task 2 – Count push button presses & display count on SSD with display multiplexing

- Consider lab13-2.c (on Blackboard)
- 7 segment display connected to port C (lower 4 bits to C0-C3) and port D (upper 4-bits to D0-D3), with 2 CC pins (digit select) now connected to port B, pin 0 and 2.
  - Consider bit masking and bit shift operators for SSD split across two ports.
- Push button connected to pin T0 (work out which pin this is)
  - Count number of rising edges on this pin
    - Use timer/counter 0
- Display tens place on left digit for 1ms, then ones place on right digit for 1ms
  - This is ***display multiplexing***
    - Alternate fast enough – it will appear that both digits are on (though brightness will be reduced)
- Fill in blanks in code, build & test
- Slow down the multiplexing rate so you can see the digits changing (e.g. 4 times per second)