ECE375
Timer/Counter

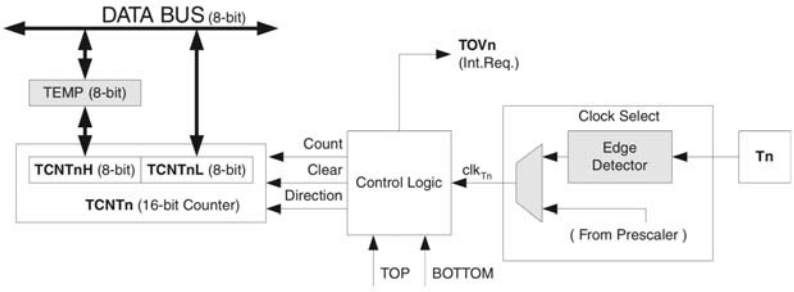
TA: Dongjun Lee
School of Electrical Engineering and Computer Science
Oregon State University

Timer/Counters

- Understand the 8-bit Timer/Counters to generate Pulse-Width Modulation (PWM)
- Control the motor speed of BumpBot using PWM signal
- Read Atmega128 Datasheet
  - 73p (Alternate Functions of Port B)
  - 92p - 110p (Timer/Counter)

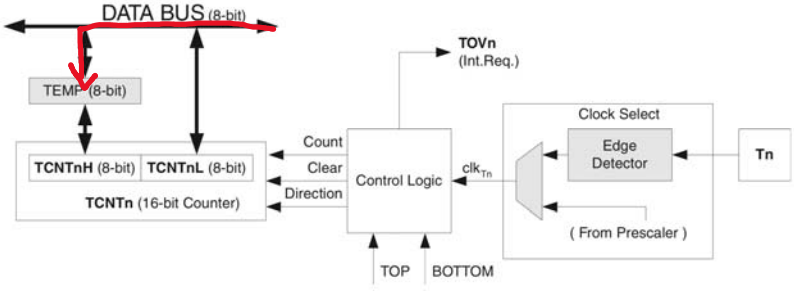
Why write higher byte first, read lower byte first?

Figure 47. Counter Unit Block Diagram



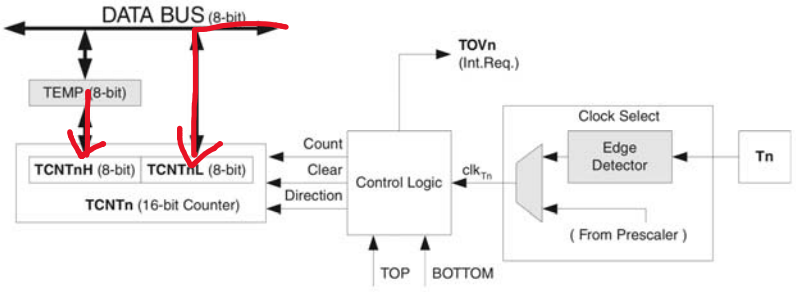
Why write higher byte first, read lower byte first?

Figure 47. Counter Unit Block Diagram



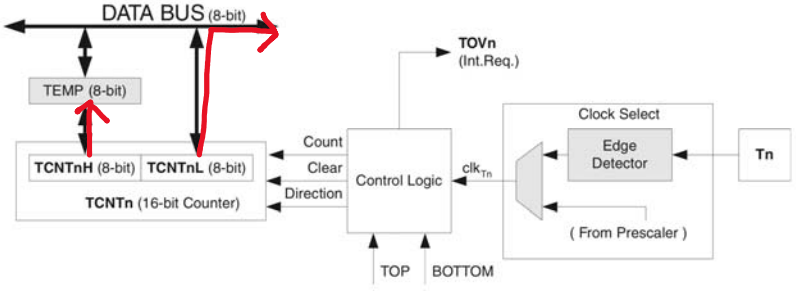
Why write higher byte first, read lower byte first?

Figure 47. Counter Unit Block Diagram



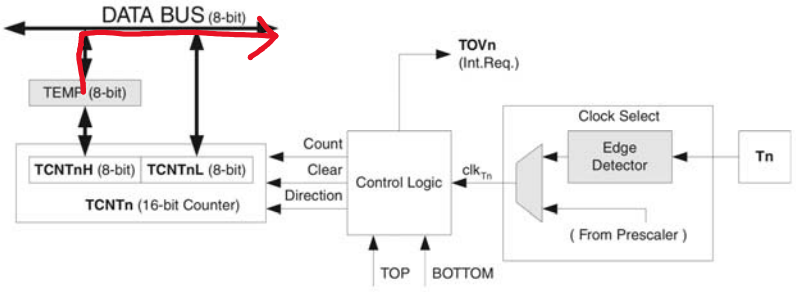
Why write higher byte first, read lower byte first?

Figure 47. Counter Unit Block Diagram



Why write higher byte first, read lower byte first?

Figure 47. Counter Unit Block Diagram



Read/Write 16bit Register

- Write 16 bit-register
  - out TCNT1H, r17 ; write to high byte first
  - out TCNT1L, r16 ; write to low byte second
- Read 16 bit-register
  - in r16, TCNT1L ; read from low byte first
  - in r17, TCNT1H ; read from high byte second
- 111p-120p

PWM Output

Alternate Functions of Port B

Port Pin	Alternate Functions
PB7	OC2/OC1C[1] (Output Compare and PWM Output for Timer/Counter2 or Output Compare and PWM Output C for Timer/Counter1)
PB6	OC1B (Output Compare and PWM Output B for Timer/Counter1)
PB5	OC1A (Output Compare and PWM Output A for Timer/Counter1)
PB4	OC0 (Output Compare and PWM Output for Timer/Counter0)
PB3	MISO (SPI Bus Master Input/Slave Output)
PB2	MOSI (SPI Bus Master Output/Slave Input)
PB1	SCK (SPI Bus Serial Clock)
PB0	$\overline{SS}$ (SPI Slave Select input)

Duty Cycle

- Change Duty Cycle to control speed
  - 100% duty cycle – Halt

  - 50% duty cycle - Half Speed

  - 0% duty cycle - Full Speed

- Use Output Compare Register (OCR)

Timer/Counter Control Register

7	6	5	4	3	2	1	0
FOC 0	WGM 00	COM 01	COM 00	WGM 01	CS 02	CS 01	CS 00

TCCR0

Wave Generation Mode (WGM)

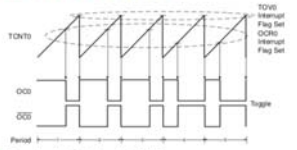
Compare Output Mode (COM)

Clock Selection (CS)

Wave Generation Mode (WGM)

Mode	WGM01 (1) (CTC0)	WGM00 (1) (PWM0)	Timer/Counter Mode of Operation	TOP	Update of OCR0 at	TOV0 Flag Set on
0	0	0	Normal	0xFF	Immediate	MAX
1	0	1	PWM, Phase Correct	0xFF	TOP	BOTTOM
2	1	0	CTC	OCR0	Immediate	MAX
3	1	1	Fast PWM	0xFF	BOTTOM	MAX

Fast PWM Mode



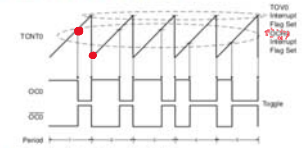
- Uses both OCF0 and TOV0
  - Clear OC0 on output compare match, set OC0 at TOP
- Can generate waveform with varying duty cycle and fixed frequency $f_{PWM} = \frac{clk_{I/O}}{prescale \cdot 256}$

Compare Output Mode (COM)

Table 54. Compare Output Mode, Fast PWM Mode[1]

COM01	COM00	Description
0	0	Normal port operation, OC0 disconnected.
0	1	Reserved
1	0	Clear OC0 on compare match, set OC0 at BOTTOM, (non-inverting mode)
1	1	Set OC0 on compare match, clear OC0 at BOTTOM, (inverting mode)

Fast PWM Mode



- Uses both OCF0 and TOV0
  - Clear OC0 on output compare match, set OC0 at TOP
- Can generate waveform with varying duty cycle and fixed frequency

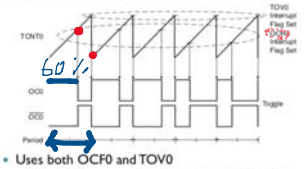
$$f_{PWM} = \frac{clk_{I/O}}{prescale \cdot 256}$$

Compare Output Mode (COM)

Table 54. Compare Output Mode, Fast PWM Mode[1]

COM01	COM00	Description
0	0	Normal port operation, OC0 disconnected.
0	1	Reserved
1	0	Clear OC0 on compare match, set OC0 at BOTTOM, (non-inverting mode)
1	1	Set OC0 on compare match, clear OC0 at BOTTOM, (inverting mode)

Fast PWM Mode



60%

- Uses both OCF0 and TOV0
  - Clear OC0 on output compare match, set OC0 at TOP
- Can generate waveform with varying duty cycle and fixed frequency

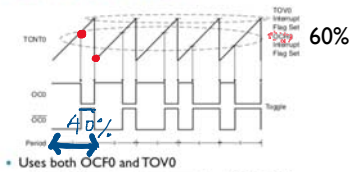
$$f_{PWM} = \frac{clk_{I/O}}{prescale \cdot 256}$$

Compare Output Mode (COM)

Table 54. Compare Output Mode, Fast PWM Mode[1]

COM01	COM00	Description
0	0	Normal port operation, OC0 disconnected.
0	1	Reserved
1	0	Clear OC0 on compare match, set OC0 at BOTTOM, (non-inverting mode)
1	1	Set OC0 on compare match, clear OC0 at BOTTOM, (inverting mode)

Fast PWM Mode



- Uses both OCF0 and TOV0
  - Clear OC0 on output compare match, set OC0 at TOP
- Can generate waveform with varying duty cycle and fixed frequency

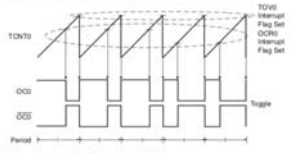
$$f_{PWM} = \frac{clk_{I/O}}{prescale \cdot 256}$$

Clock Selection (CS)

Table 56. Clock Select Bit Description

CS02	CS01	CS00	Description
0	0	0	No clock source (Timer/Counter stopped)
0	0	1	clk_{T0S}/(No prescaling)
0	1	0	clk_{T0S}/8 (From prescaler)
0	1	1	clk_{T0S}/32 (From prescaler)
1	0	0	clk_{T0S}/64 (From prescaler)
1	0	1	clk_{T0S}/128 (From prescaler)
1	1	0	clk_{T0S}/256 (From prescaler)
1	1	1	clk_{T0S}/1024 (From prescaler)

Fast PWM Mode



- Uses both OCF0 and TOV0
  - Clear OC0 on output compare match, set OC0 at TOP
- Can generate waveform with varying duty cycle and fixed frequency

$$f_{PWM} = \frac{clk_{I/O}}{prescale \cdot 256}$$

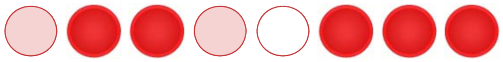
Demo Check

- 16 speed levels
- PORTB 0-3 indicate current speed level
- PORTB 4,7 brightness change
- 4 Functions for Control Speed
  - SPEED_DOWN
  - SPEED_UP
  - SPEED_MIN
  - SPEED_MAX
- Speed levels bound max and min
- Single button press results single action

Speed control

- 100% duty cycle – Halt

- 50% duty cycle - Half Speed



- 0% duty cycle - Full Speed

Checklists for Lab 7

- Demo Checklist
  - All four speed changes work correctly
  - Smooth transitions (1 press, 1 change)
  - No Speed Level overflow or underflow
  - MovFwd signals never overwritten
  - Motor enable signals correctly active low
  - Actually using PWM, no manual toggling
- Challenge Checklist
  - Time updates every 1 sec, no leading 0s
  - Buttons still work and reset count on LCD

Questions?