ECE375
Timer/Counter

TA: Han Jang
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

DATA BUS (8-bit)

TEMP (8-bit)

TCNTnH (8-bit)  TCNTnL (8-bit)

TCNTn (16-bit Counter)

Control Logic

Count  Clear  Direction

clk_{Tn}

TOP  BOTTOM

TOVn (Int.Req.)

Edge Detector

Clock Select

( From Prescaler )

Tn
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

<table>
<thead>
<tr>
<th>Port Pin</th>
<th>Alternate Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>PB7</td>
<td>OC2/OC1C(^{(1)}) (Output Compare and PWM Output for Timer/Counter2 or Output Compare and PWM Output C for Timer/Counter1)</td>
</tr>
<tr>
<td>PB6</td>
<td>OC1B (Output Compare and PWM Output B for Timer/Counter1)</td>
</tr>
<tr>
<td>PB5</td>
<td>OC1A (Output Compare and PWM Output A for Timer/Counter1)</td>
</tr>
<tr>
<td>PB4</td>
<td>OC0 (Output Compare and PWM Output for Timer/Counter0)</td>
</tr>
<tr>
<td>PB3</td>
<td>MISO (SPI Bus Master Input/Slave Output)</td>
</tr>
<tr>
<td>PB2</td>
<td>MOSI (SPI Bus Master Output/Slave Input)</td>
</tr>
<tr>
<td>PB1</td>
<td>SCK (SPI Bus Serial Clock)</td>
</tr>
<tr>
<td>PB0</td>
<td>SS (SPI Slave Select input)</td>
</tr>
</tbody>
</table>
Duty Cycle

- Change Duty Cycle to control speed
  - 100% duty cycle – Halt
    ![100% duty cycle](image)
  - 50% duty cycle - Half Speed
    ![50% duty cycle](image)
  - 0% duty cycle - Full Speed
    ![0% duty cycle](image)

- Use Output Compare Register (OCR)
Timer/Counter Control Register

Wave Generation Mode (WGM)
Compare Output Mode (COM)
Clock Selection (CS)
Wave Generation Mode (WGM)

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

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
  \[ f_{PWM} = \frac{clk_{CPU}}{\text{prescale} \times 256} \]
# Compare Output Mode (COM)

## Table 54. Compare Output Mode, Fast PWM Mode

<table>
<thead>
<tr>
<th>COM01</th>
<th>COM00</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Normal port operation, OC0 disconnected.</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Reserved</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Clear OC0 on compare match, set OC0 at BOTTOM, (non-inverting mode)</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Set OC0 on compare match, clear OC0 at BOTTOM, (inverting mode)</td>
</tr>
</tbody>
</table>

## 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
  \[
  f_{PWM} = \frac{clk_{16}}{\text{prescale} \times 256}
  \]
Compare Output Mode (COM)

Table 54. Compare Output Mode, Fast PWM Mode

<table>
<thead>
<tr>
<th>COM01</th>
<th>COM00</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Normal port operation, OC0 disconnected.</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Reserved</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Clear OC0 on compare match, set OC0 at BOTTOM, (non-inverting mode)</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Set OC0 on compare match, clear OC0 at BOTTOM, (inverting mode)</td>
</tr>
</tbody>
</table>

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
  \[ f_{PWM} = \frac{clk_{I/O}}{\text{prescale} \times 256} \]
Compare Output Mode (COM)

Table 54. Compare Output Mode, Fast PWM Mode(1)

<table>
<thead>
<tr>
<th>COM01</th>
<th>COM00</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Normal port operation, OC0 disconnected.</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>Reserved</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>Clear OC0 on compare match, set OC0 at BOTTOM,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(non-inverting mode)</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Set OC0 on compare match, clear OC0 at BOTTOM,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(inverting mode)</td>
</tr>
</tbody>
</table>

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
  \[ f_{PWM} = \frac{clk_{1/0}}{prescale \cdot 256} \]
Clock Selection (CS)

<table>
<thead>
<tr>
<th>Table 56. Clock Select Bit Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS02</td>
</tr>
<tr>
<td>------</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>0</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>1</td>
</tr>
</tbody>
</table>

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
  \[ f_{PWM} = \frac{clk_{10}}{\text{prescale} \cdot 256} \]
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
  ![100% duty cycle](image)
- 50% duty cycle - Half Speed
  ![50% duty cycle](image)
- 0% duty cycle - Full Speed
  ![0% duty cycle](image)
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?