Software PWM

Adapted from Martin Fox’s ECE266 Slides
Introduction

• Add intensity to bar graph
• Timer 0
• PWM: hardware vs. software
• Interrupt driven process
Overview

- Initialize
  - Mainloop
    - ADConvert
      - AN1/RP1
      - AN0/RP2
    - LoopTime = .01 s

- ISR
  - Check State
    - Ton
    - Toff
Avoid AD 0/1 interaction

• Be sure ADCs do not interact: After turn on wait 25 us for capacitor to charge, THEN set ADGO=1 and do conversion.

• After conversion, turn AD off by setting ADON =0

• Store RP1 and RP2 pot values in LEDValue and LEDIntensity variables respectively
PWM

From 2816A-FPLSI Application Note ©Atmel
Timer 0 Block diagram

FIGURE 5-1: BLOCK DIAGRAM OF THE TIMER0/WDT PRESCALER

Note: T0CS, T0SE, PSA, PS2:PS0 are (OPTION_REG<5:0>).

Source: PIC data sheet
## Timer 0 Registers

### TABLE 5-1:  REGISTERS ASSOCIATED WITH TIMER0

<table>
<thead>
<tr>
<th>Address</th>
<th>Name</th>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
<th>Bit 2</th>
<th>Bit 1</th>
<th>Bit 0</th>
<th>Value on: POR, BOR</th>
<th>Value on all other resets</th>
</tr>
</thead>
<tbody>
<tr>
<td>01h,101h</td>
<td>TMR0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>yyyy yyyy yyyy yyyy</td>
<td>yyyy yyyy yyyy yyyy yyyy</td>
</tr>
<tr>
<td>0Bh,8Bh, 10Bh,18Bh</td>
<td>INTCON</td>
<td>GIE</td>
<td>PEIE</td>
<td>T0IE</td>
<td>INTE</td>
<td>RBIE</td>
<td>T0IF</td>
<td>INTF</td>
<td>RBIF</td>
<td>0000 000x</td>
<td>0000 000u</td>
</tr>
<tr>
<td>81h,181h</td>
<td>OPTION_REG</td>
<td>RBPU</td>
<td>INTEDG</td>
<td>T0CS</td>
<td>T0SE</td>
<td>PSA</td>
<td>PS2</td>
<td>PS1</td>
<td>PS0</td>
<td>1111 1111</td>
<td>1111 1111</td>
</tr>
</tbody>
</table>

Legend:  
- x = unknown,  
- u = unchanged,  
- - = unimplemented locations read as '0'. Shaded cells are not used by Timer0.

Source: PIC data sheet, Microchip
Option Register

### REGISTER 5-1: OPTION_REG REGISTER

<table>
<thead>
<tr>
<th>R/W-1</th>
<th>R/W-1</th>
<th>R/W-1</th>
<th>R/W-1</th>
<th>R/W-1</th>
<th>R/W-1</th>
<th>R/W-1</th>
<th>R/W-1</th>
<th>R/W-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>RBPU</td>
<td>INTEDG</td>
<td>T0CS</td>
<td>T0SE</td>
<td>PSA</td>
<td>PS2</td>
<td>PS1</td>
<td>PS0</td>
<td></td>
</tr>
</tbody>
</table>

- **R** = Readable bit
- **W** = Writable bit
- **U** = Unimplemented bit, read as '0'
- **n** = Value at POR reset

#### Bit Descriptions:

**bit 7:** RBPU

**bit 6:** INTEDG

**bit 5:** T0CS: TMR0 Clock Source Select bit
- 1 = Transition on T0CKI pin
- 0 = Internal instruction cycle clock (CLKOUT)

**bit 4:** T0SE: TMR0 Source Edge Select bit
- 1 = Increment on high-to-low transition on T0CKI pin
- 0 = Increment on low-to-high transition on T0CKI pin

**bit 3:** PSA: Prescaler Assignment bit
- 1 = Prescaler is assigned to the WDT
- 0 = Prescaler is assigned to the Timer0 module

**bit 2-0:** PS2:PS0: Prescaler Rate Select bits

<table>
<thead>
<tr>
<th>Bit Value</th>
<th>TMR0 Rate</th>
<th>WDT Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>000</td>
<td>1:2</td>
<td>1:1</td>
</tr>
<tr>
<td>001</td>
<td>1:4</td>
<td>1:2</td>
</tr>
<tr>
<td>010</td>
<td>1:8</td>
<td>1:4</td>
</tr>
<tr>
<td>011</td>
<td>1:16</td>
<td>1:8</td>
</tr>
<tr>
<td>100</td>
<td>1:32</td>
<td>1:16</td>
</tr>
<tr>
<td>101</td>
<td>1:64</td>
<td>1:32</td>
</tr>
<tr>
<td>110</td>
<td>1:128</td>
<td>1:64</td>
</tr>
<tr>
<td>111</td>
<td>1:256</td>
<td>1:128</td>
</tr>
</tbody>
</table>
Timer 0

- 8 bit timer/counter-8 bit programmable prescaler.
- 3 prescaler bits in OPTION_REG; set to 100 ->1:32

\[32 \times 256 = 4446 \text{ us turnover, } 4.446 \text{ ms; } 225 \text{ Hz}\]
- TMR0 register can be preset to N, meaning 256 –N counts before turnover from 255->0 or 256-N meaning N counts before turnover.
Timers - Timer 2

7.37 MHz/4

Prescaler

TMR2

Postscaler

PR2 = 5

TMR2IF = 0
Timers - Timer 0

7.37 MHz/4

Prescaler → TMR0 = 0 → TMR0IF = 0
PWM Timing

4.446 ms

256-LEDIntensity

LEDIntensity

Interrupts
Intensity ISR

- ISR: “Intensity”, triggered on TMR0 Overflow
- Two states: PORTD=0 or PORTD=LEDValue
  - State 0 for 256-LEDIntensity counts
  - State 1 for LEDIntensity counts
- When you get an interrupt, you need to check which state you were in and set TMR0 accordingly
  - If PORTD=0, you were in state 1
  - If LEDIntensity is 0, stay in state 0
Summary

• PWM is a key tool. There is a Hardware PWM, but software PWM provides even more control.

• Could use to turn devices on in different combinations and control intensity by analog inputs.

• Hand in code by Tuesday of next week