Microcontrollers

Class 4: Timer/Counters

March 28, 2011

Outline

Timer/Counter Introduction

Timers as a Timebase

Timers for PWM

Outline

Timer/Counter Introduction

Timers as a Timebase

Timers for PWM

Outline

Timer/Counter Introduction

Timers as a Timebase

Timers for PWM

Review

The story so far...

- Bits and shifting
- Analog and Digital input and output
- Serial stuffies
- Interrupts
- Now we get to give our programs a sense of timing

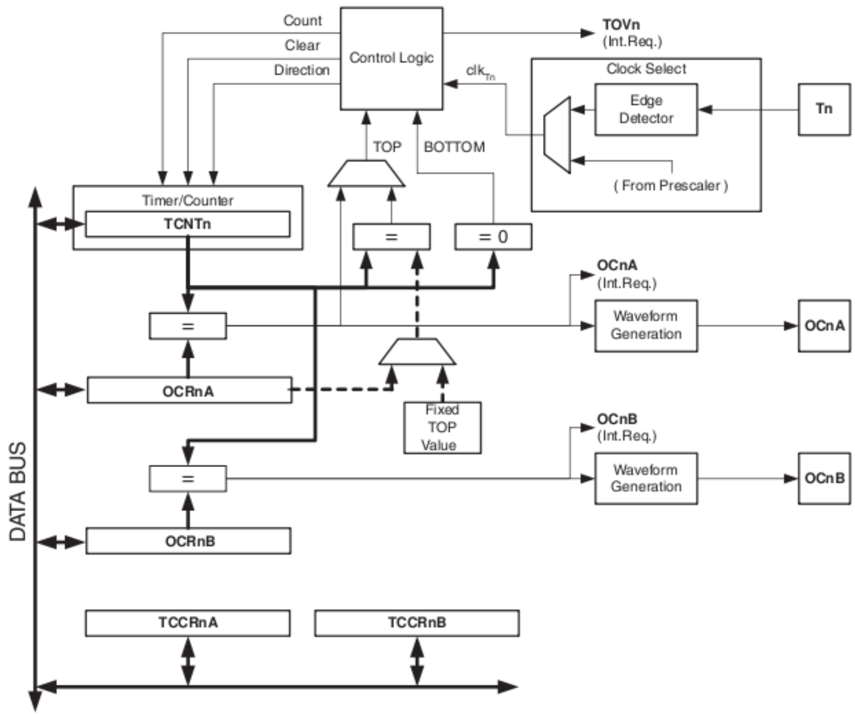
Timer/Counter Overview

The big picture

- We've been using lots of software counters:
  `for(i=0; i<8; i++)...`
- The Mega88 chip has 3 built-in hardware counters that we can use instead
- Why? They're more efficient, reliable, and they free up our software to do other things
- Why timer/counter? Well, internally they're really counters but if you hook them up to a clock, counting clock pulses, you've got a timer.
- They can also count hardware events (pin toggling, etc) but I won't run any examples of that here. Bother me by e-mail if you're interested.

Timer/Counter Hardware Block Diagram

Figure 14-1. 8-bit Timer/Counter Block Diagram



Timers

Setup

- Timers are my least-favorite thing to configure on the AVR, but it's not so bad once you've got the basics down.
- First need a source: what you're counting
  we'll use built-in clocks, and will need to set a prescaler
- Hardware compares the contents of two special memory registers (OCR0A and OCR0B) to the current counter value
- When the counter equals TOP, three things can happen:
  reset the timer
  fire an OC interrupt
  set, clear, or toggle a dedicated output pin
- What happens depends on which mode you're in
- In addition to the OCRs, there is also an interrupt available for when the timer overflows (wraps back around to 0)

Modes

What can we do with these timers?

- Normal mode: just counts up from 0 to 255. Boring, but useful for a quick-and-dirty timebase using the overflow interrupt.
- Clear Timer on Compare (CTC) mode: Count up to value in OCR0A (not necessarily 255), then go back to zero. Provides different timing cycle durations.
- Fast PWM mode: Counts 0..255, does things on output pins when counter hits the OCR0A/B values
- Phase-correct PWM mode: first counts up, then counts down

Modes

Figure 14-5. CTC Mode, Timing Diagram

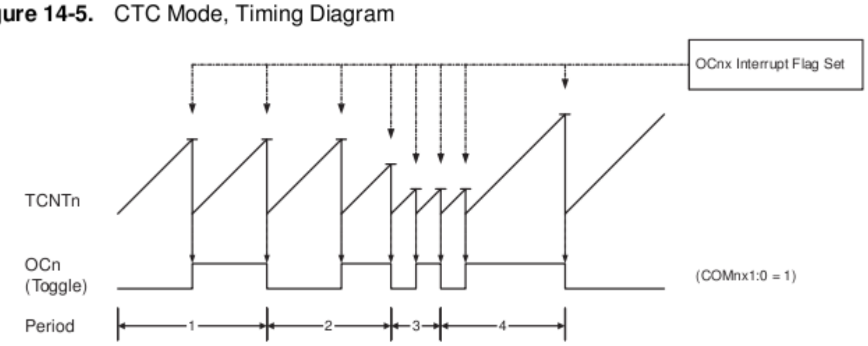
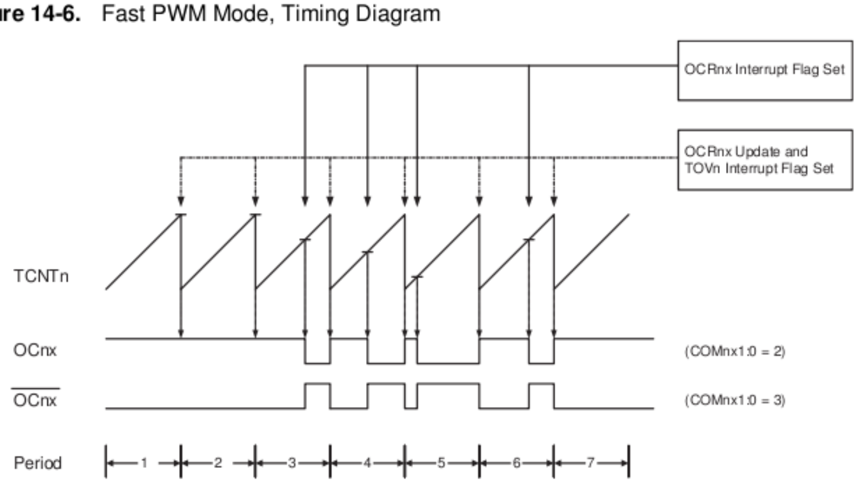


Figure 14-6. Fast PWM Mode, Timing Diagram



Configuration

Putting the right bits in the right registers

- Control registers: TCCR0A, TCCR0B
  selects clock source, counting mode, and pin-output
- Interrupt mask register: TIMSK0
  selects which interrupts to fire (overflow, compare)
- Output compare registers: OCR0A, OCR0B
  put your compare-values in here,
  changes duty-cycle (both, in some modes) or frequency (OCR0A only)
- Examples for details!

Waveform Select Modes (p. 108)

Table 14-8. Waveform Generation Mode Bit Description

Mode	WGM02	WGM01	WGM00	Timer/Counter Mode of Operation	TOP	Update of OCRx at	TOV Flag Set on[(1)(2)]
0	0	0	0	Normal	0xFF	Immediate	MAX
1	0	0	1	PWM, Phase Correct	0xFF	TOP	BOTTOM
2	0	1	0	CTC	OCRA	Immediate	MAX
3	0	1	1	Fast PWM	0xFF	BOTTOM	MAX
4	1	0	0	Reserved	–	–	–
5	1	0	1	PWM, Phase Correct	OCRA	TOP	BOTTOM
6	1	1	0	Reserved	–	–	–
7	1	1	1	Fast PWM	OCRA	BOTTOM	TOP

Notes: 1. MAX = 0xFF
2. BOTTOM = 0x00

Clock Select (p. 110)

Table 14-9. Clock Select Bit Description

CS02	CS01	CS00	Description
0	0	0	No clock source (Timer/Counter stopped)
0	0	1	$clk_{I/O}$/(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 T0 pin. Clock on falling edge.
1	1	1	External clock source on T0 pin. Clock on rising edge.

If external pin modes are used for the Timer/Counter0, transitions on the T0 pin will clock the counter even if the pin is configured as an output. This feature allows software control of the counting.

Output Mode (p. 106)

Table 14-2. Compare Output Mode, non-PWM Mode

COM0A1	COM0A0	Description
0	0	Normal port operation, OC0A disconnected.
0	1	Toggle OC0A on Compare Match
1	0	Clear OC0A on Compare Match
1	1	Set OC0A on Compare Match

Table 14-3 shows the COM0A1:0 bit functionality when the WGM01:0 bits are set to fast PWM mode.

Table 14-3. Compare Output Mode, Fast PWM Mode[1]

COM0A1	COM0A0	Description
0	0	Normal port operation, OC0A disconnected.
0	1	WGM02 = 0: Normal Port Operation, OC0A Disconnected. WGM02 = 1: Toggle OC0A on Compare Match.
1	0	Clear OC0A on Compare Match, set OC0A at BOTTOM, (non-inverting mode).
1	1	Set OC0A on Compare Match, clear OC0A at BOTTOM, (inverting mode).

Note: 1. A special case occurs when OCR0A equals TOP and COM0A1 is set. In this case, the Compare Match is ignored, but the set or clear is done at BOTTOM. See "Fast PWM Mode" on page 101 for more details.

Outline

Timer/Counter Introduction

Timers as a Timebase

Timers for PWM

Simple Clock

Tick, tick, tick

- I'm surprised by the number of times I see people buy clock modules for their hardware projects
- Add in a crystal (next class) and you'll have a pretty-darn accurate timebase
- The basics:
  Set up global time-keeping variables
  Set up a CTC timer to interrupt once every (howeverlong)
  In the interrupt, increment your various counters (lightweight!)
  In the mainloop, use the time variables
- Initialization for the timer: mode, clock prescale, interrupt enables
- See the example: counterClock.c

Outline

Timer/Counter Introduction

Timers as a Timebase

Timers for PWM

Timer-based PWM

Dimming LEDs in hardware

- Pins labelled OCxxx are directly connected to the output compare logic
- Result: you can set OCR0A, OCR0B and get PWM done for you automatically
- Timer1 is a 16-bit timer: has enough resolution for easy servo driving
- Setup:
  Select PWM mode (we'll use fast)
  Set up the output pin modes
- See example: counterPWM.c

Next Class

Odds and Ends

- Special requests??
- PROGMEM, EEPROM, and funny memory types
- Moving past the classboard: how to wire up your own circuits from the ground up
- Turn your classboard into a bare-chip programmer! (Optional)

The End

◂ Outline