Printout

Monday, February 27, 2017 3:38 PM

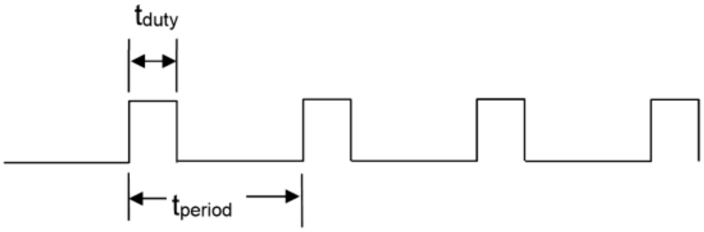
Pulse Width Modulation (PWM) on the ATmega2560

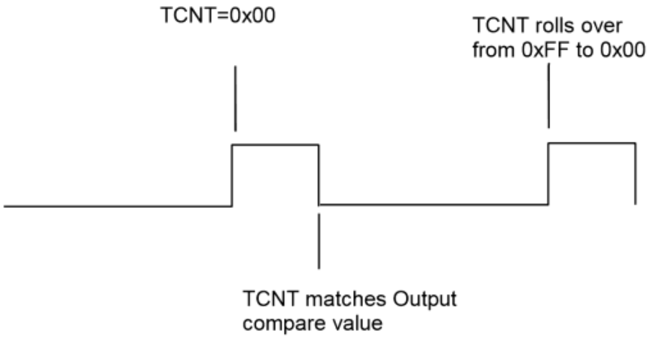
ME/ECE 4370/5370

Review PWM

Pulse Width Modulation (PWM) is a digital representation of an analog event. The representation contains two parameters which can be characterized by different values, for example frequency and duty cycle. As a digital representation, the PWM signal will have a resolution (determined by total number of discrete states over the period). PWM can be used for communication, driving motors at variable speed or torque, dimming lights, sound, etc.

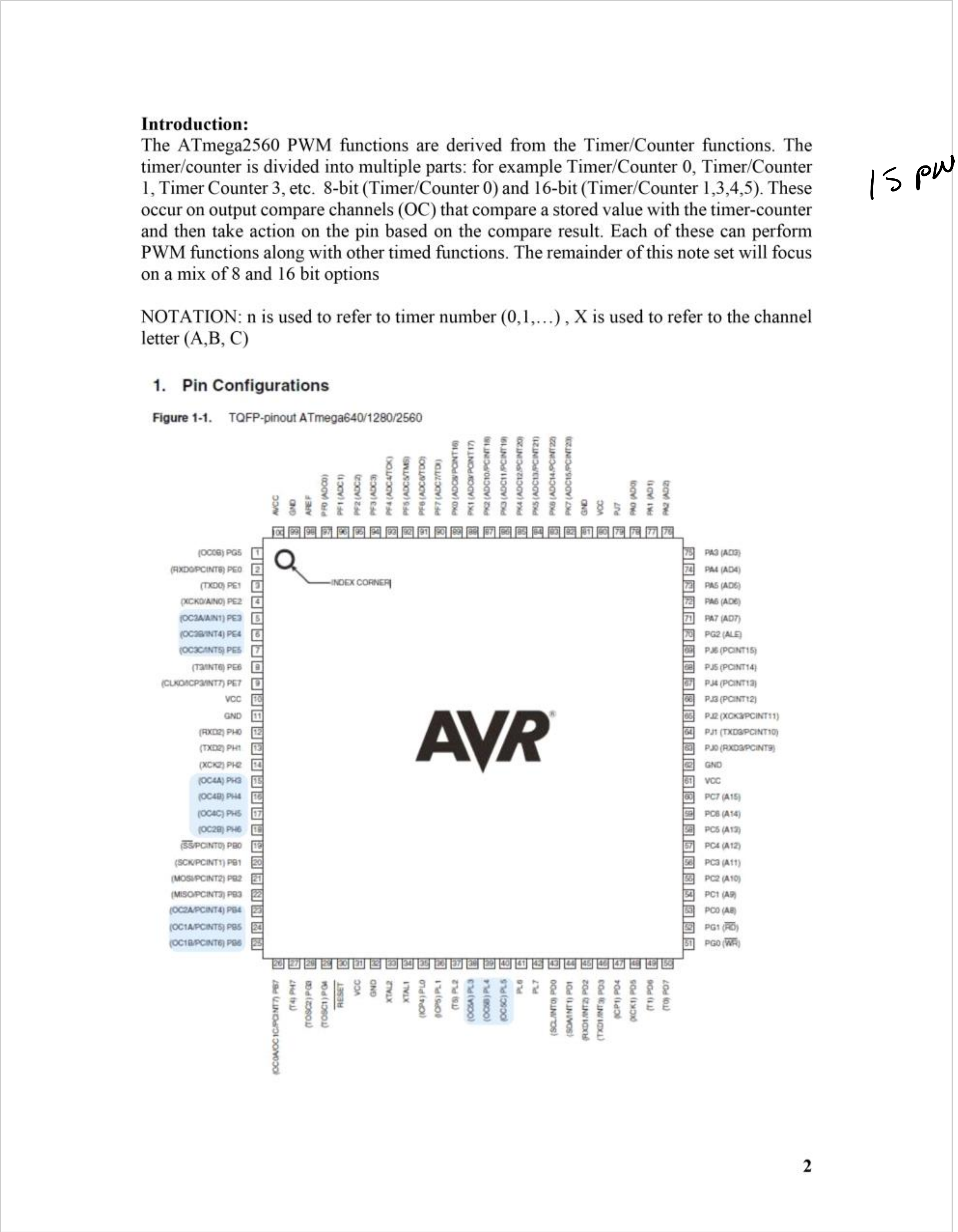
$$DC = \frac{t_{duty}}{t_{period}}.$$




TCNT=0x00
TCNT rolls over from 0xFF to 0x00
TCNT matches Output compare value

Page 2

Monday, February 27, 2017 3:38 PM



Introduction:
The ATmega2560 PWM functions are derived from the Timer/Counter functions. The timer/counter is divided into multiple parts: for example Timer/Counter 0, Timer/Counter 1, Timer Counter 3, etc. 8-bit (Timer/Counter 0) and 16-bit (Timer/Counter 1,3,4,5). These occur on output compare channels (OC) that compare a stored value with the timer-counter and then take action on the pin based on the compare result. Each of these can perform PWM functions along with other timed functions. The remainder of this note set will focus on a mix of 8 and 16 bit options

NOTATION: n is used to refer to timer number (0,1,…) , X is used to refer to the channel letter (A,B, C)

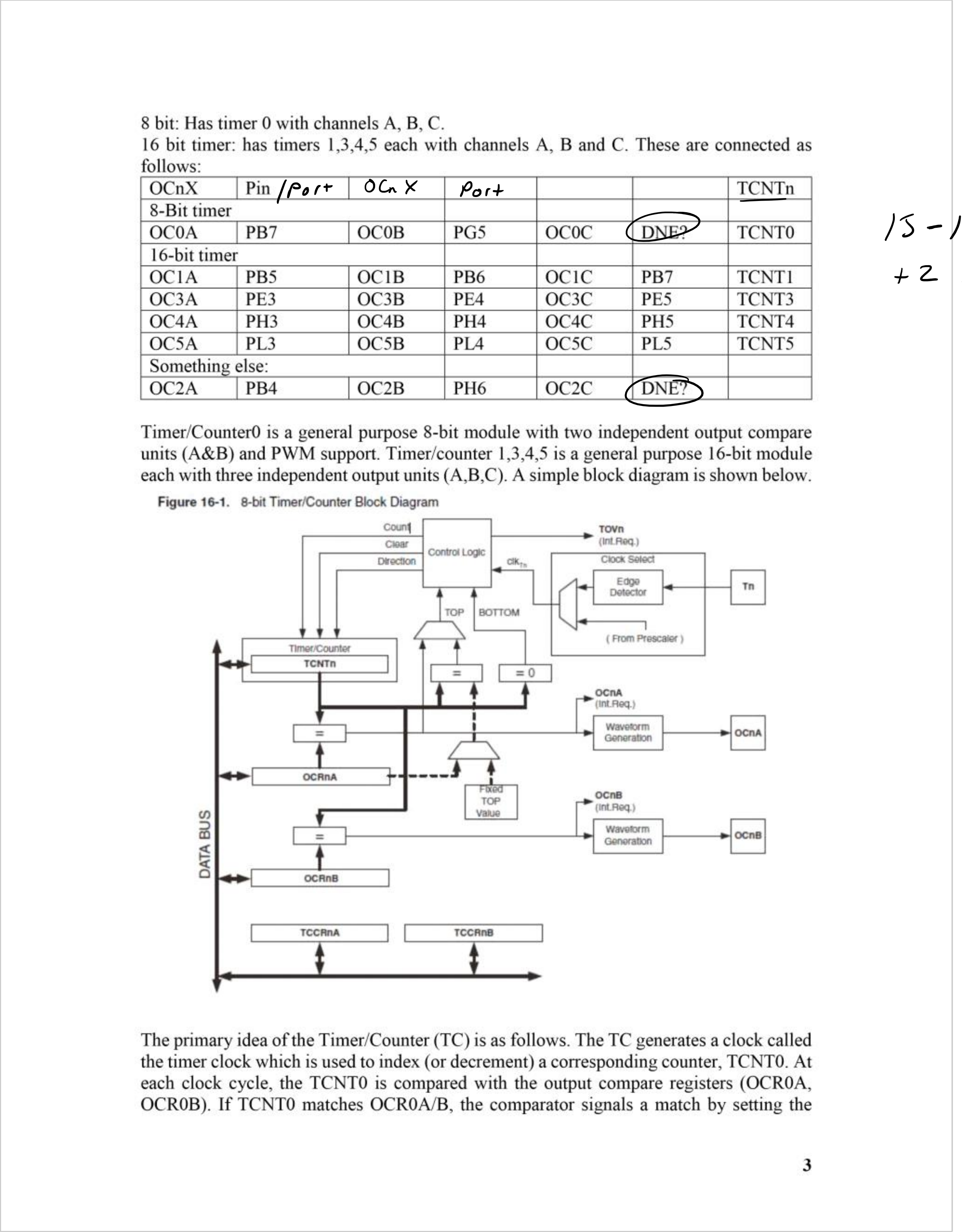
1. Pin Configurations

Figure 1-1. TQFP-pinout ATmega640/1280/2560

15 PWM possible.

Page 3

Monday, February 27, 2017 3:38 PM



8 bit: Has timer 0 with channels A, B, C.
16 bit timer: has timers 1,3,4,5 each with channels A, B and C. These are connected as follows:

OCnX	Pin /Port	OCnX	Port			TCNTn
8-Bit timer						
OC0A	PB7	OC0B	PG5	OC0C	DNE?	TCNT0
16-bit timer						
OC1A	PB5	OC1B	PB6	OC1C	PB7	TCNT1
OC3A	PE3	OC3B	PE4	OC3C	PE5	TCNT3
OC4A	PH3	OC4B	PH4	OC4C	PH5	TCNT4
OC5A	PL3	OC5B	PL4	OC5C	PL5	TCNT5
Something else:						
OC2A	PB4	OC2B	PH6	OC2C	DNE?	

15 - 1
+2 or +1?

Timer/Counter0 is a general purpose 8-bit module with two independent output compare units (A&B) and PWM support. Timer/counter 1,3,4,5 is a general purpose 16-bit module each with three independent output units (A,B,C). A simple block diagram is shown below.

Figure 16-1. 8-bit Timer/Counter Block Diagram

The primary idea of the Timer/Counter (TC) is as follows. The TC generates a clock called the timer clock which is used to index (or decrement) a corresponding counter, TCNT0. At each clock cycle, the TCNT0 is compared with the output compare registers (OCR0A, OCR0B). If TCNT0 matches OCR0A/B, the comparator signals a match by setting the

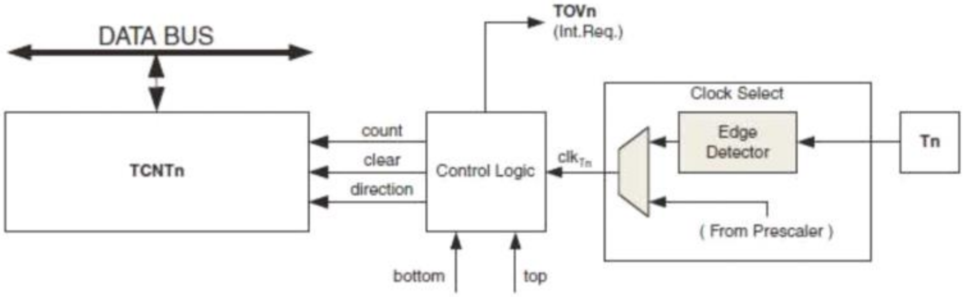
Page 4

Monday, February 27, 2017 3:38 PM

Output Compare Flag (OCF0A or OCF0B) at the next clock cycle. Once this flag is set, a particular activity is executed. The activity could be: Toggle a corresponding output hi or low, generate an interrupt, or generate a PWM signal. The activity is determined by timer setup registers.

The main part of the TC is the counter unit (TCNTn). TCNTn can be incremented, decremented or cleared at each clock cycle. In general we will assume an incrementing TCNTn. The counter is based on the system clock plus a prescaler.

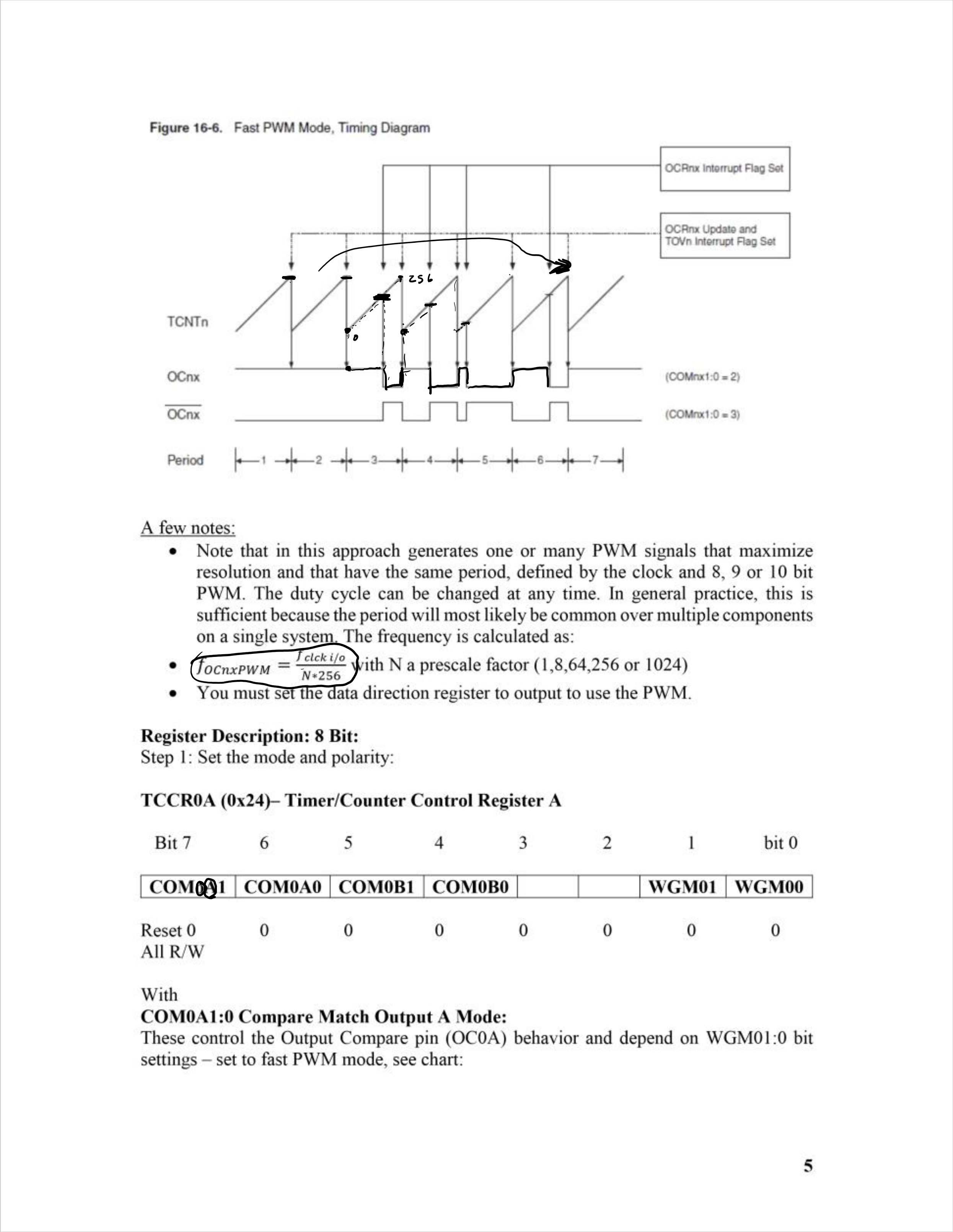
Figure 16-2. Counter Unit Block Diagram



Modes of Operation: The TC has several models of operation. For this set of notes we will focus on **Fast PWM Mode.** In fast PWM mode (WGM02:0=3 or 7 for 8 bit, WGMn3:0 = 10 for 16 bit) the PWM is created on the corresponding OCnX pins. This PWM signal is set high when TCNT0 is reset (0x00) and stays high until TCNT0 equals OCRnX. The PWM signal is then set low and stays low until TCNT0 reaches its maximum value, overflows and is reset to 0x00. The process then repeats. This is called a non-inverted mode. In the inverted mode, the PWM signal starts low at TCNT0=0x00 and goes high on a successful compare with OCRnX. This is shown on the following figure:

Page 5

Monday, February 27, 2017 3:38 PM



0 0 0 0 0 0 0 0

Figure 16-6. Fast PWM Mode, Timing Diagram

OCRnx Interrupt Flag Set
OCRnx Update and TOVn Interrupt Flag Set
256
0
TCNTn
OCnx
(COMnx1:0 = 2)
OCnx
(COMnx1:0 = 3)
Period
1
2
3
4
5
6
7

A few notes:

- Note that in this approach generates one or many PWM signals that maximize resolution and that have the same period, defined by the clock and 8, 9 or 10 bit PWM. The duty cycle can be changed at any time. In general practice, this is sufficient because the period will most likely be common over multiple components on a single system. The frequency is calculated as:
- $f_{OCnxPWM} = \frac{f_{clck\ i/o}}{N*256}$ with N a prescale factor (1,8,64,256 or 1024)
- You must set the data direction register to output to use the PWM.

Register Description: 8 Bit:
Step 1: Set the mode and polarity:

TCCR0A (0x24)– Timer/Counter Control Register A

Bit 7	6	5	4	3	2	1	bit 0
COM0A1	**COM0A0**	**COM0B1**	**COM0B0**			**WGM01**	**WGM00**
Reset 0	0	0	0	0	0	0	0

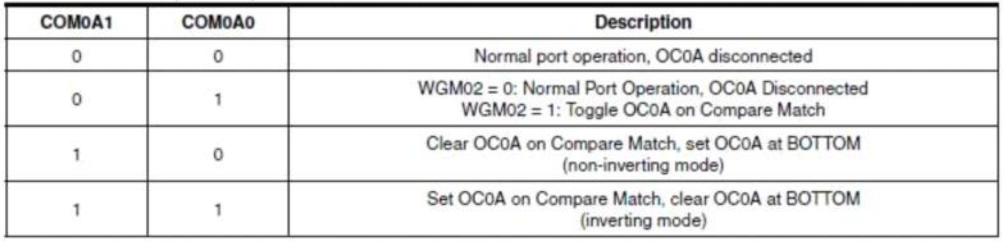
All R/W

With
COM0A1:0 Compare Match Output A Mode:
These control the Output Compare pin (OC0A) behavior and depend on WGM01:0 bit settings – set to fast PWM mode, see chart:

Page 6

Monday, February 27, 2017 3:38 PM

Table 16-3. Compare Output Mode, Fast PWM Mode[1]



	COM0A1	COM0A0	Description
no	0	0	Normal port operation, OC0A disconnected

no	0	1	WGM02 = 0: Normal Port Operation, OC0A Disconnected
			WGM02 = 1: Toggle OC0A on Compare Match

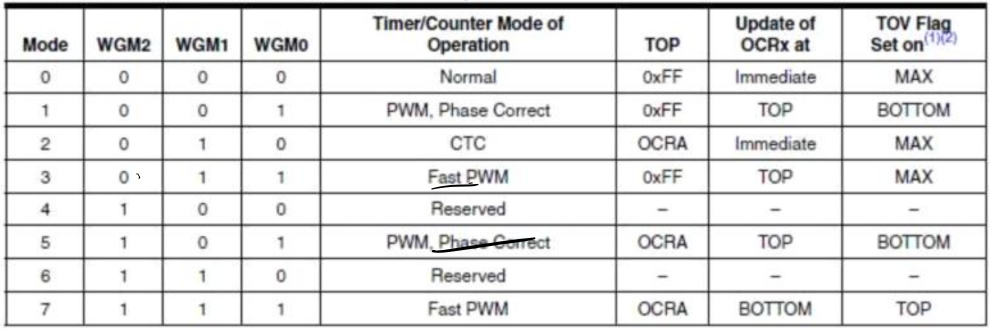
for PWM { Yes	1	0	Clear OC0A on Compare Match, set OC0A at BOTTOM (non-inverting mode)
or Yes.	1	1	Set OC0A on Compare Match, clear OC0A at BOTTOM (inverting mode)

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 121 for more details.

COM0B1:0 Compare Match Output B Mode:
These control the Output Compare pin (OC0B) behavior and depend on WGM01:0 bit settings – set to fast PWM mode the same as chart for OC0A.

WGM01:0 Waveform Generation Mode
These are combined with the WGM02 bit found in the TCCR0B Register, are used to set Fast PWM.

Table 16-8. Waveform Generation Mode Bit Description



	Mode	WGM2	WGM1	WGM0	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	TOP	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

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

Step 2: Set the prescale for the timer clock:

TCCR0B (0x25)– Timer/Counter Control Register B

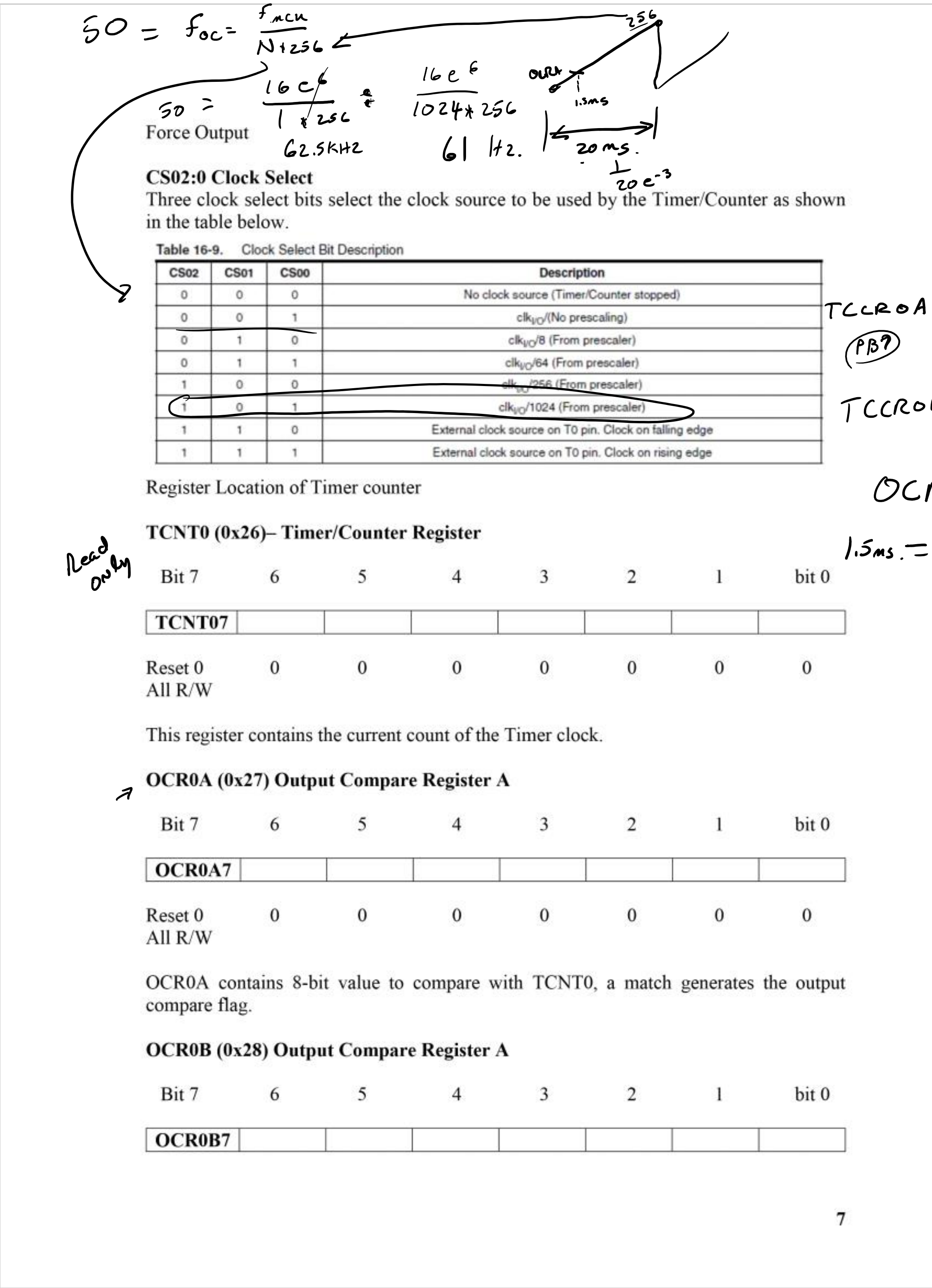
Bit 7	6	5	4	3	2	1	bit 0
FOC0A	**FOC0B**			**WGM02**	**CS02**	**CS01**	**CS00**
Reset 0	0	0	0	0	0	0	0

All R/W

With
FOC0A/B: Force Output Compare A/B
(only active in non-PWM mode, make sure it is set to zero when in PWM mode.

Page 7

Monday, February 27, 2017 3:38 PM



$50 = f_{oc} = \frac{f_{mcu}}{N*256}$

$50 = \frac{16e6}{1*256}$ 62.5KHz

$\frac{16e6}{1024*256}$ 61 Hz.

256
OCRA
1.5ms
20ms.

$\frac{1}{20e^{-3}}$

Force Output

CS02:0 Clock Select

Three clock select bits select the clock source to be used by the Timer/Counter as shown in the table below.

Table 16-9. 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

Register Location of Timer counter

TCNT0 (0x26)– Timer/Counter Register

Read only

Bit 7	6	5	4	3	2	1	bit 0
TCNT07							
Reset 0 All R/W	0	0	0	0	0	0	0

This register contains the current count of the Timer clock.

OCR0A (0x27) Output Compare Register A

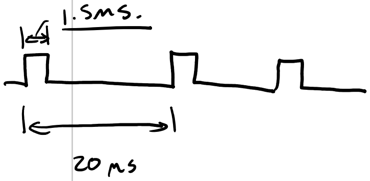
Bit 7	6	5	4	3	2	1	bit 0
OCR0A7							
Reset 0 All R/W	0	0	0	0	0	0	0

OCR0A contains 8-bit value to compare with TCNT0, a match generates the output compare flag.

OCR0B (0x28) Output Compare Register A

Bit 7	6	5	4	3	2	1	bit 0
OCR0B7							

7



use TCNT0

non inverting

TCCR0A { COM0A0:1 → 1 0
WGM 1:0 → 1 1

PB7

TCCR0B { CS2:0 → 101

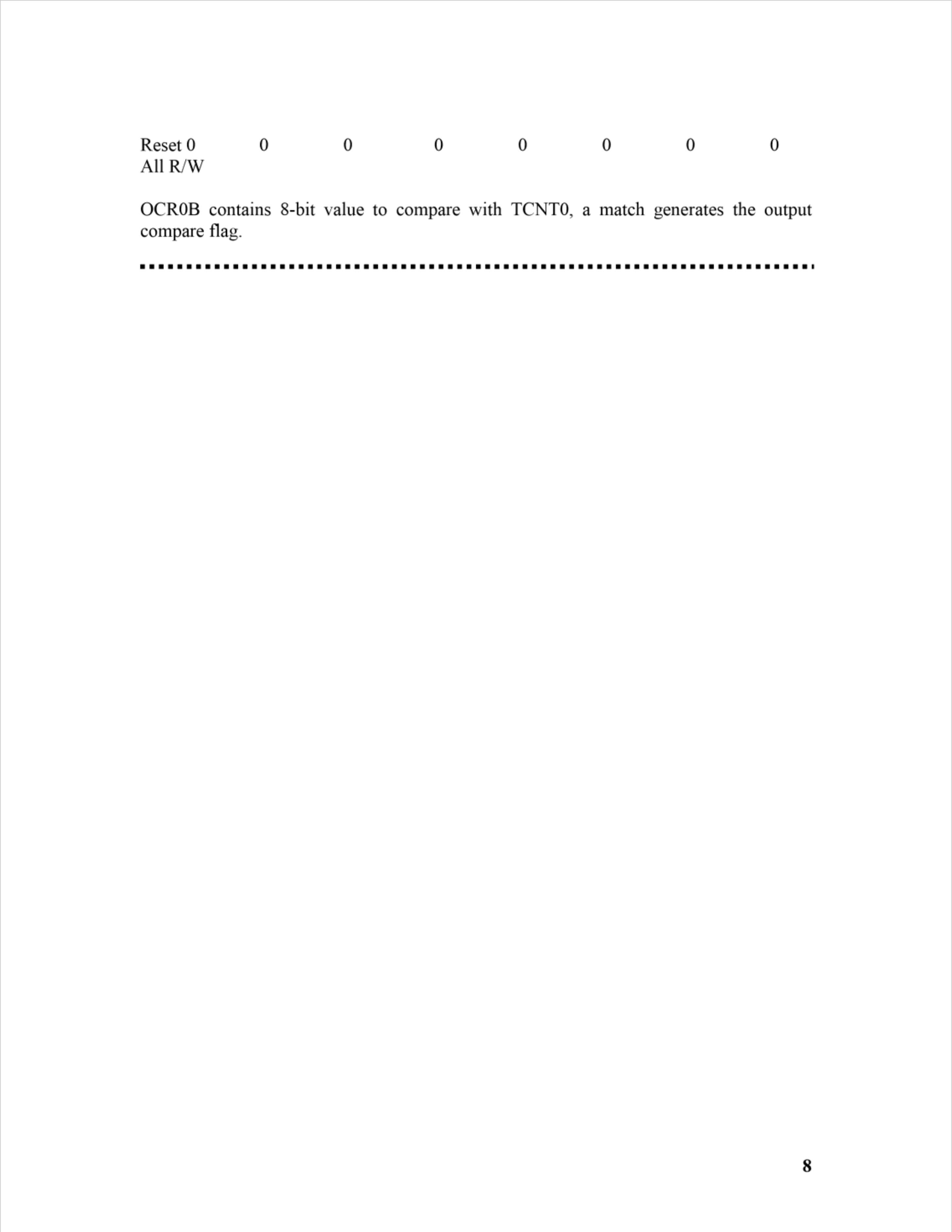
OCR0A → 23.~~43~~

$1.5ms. = \# cts * \frac{1}{16e6} s. * 1024$

$\frac{23 cts}{\left(16e6 \frac{cts}{s} / 1024\right)} = 1.472ms.$

Page 8

Monday, February 27, 2017 3:38 PM



Reset 0 0 0 0 0 0 0 0
All R/W

OCR0B contains 8-bit value to compare with TCNT0, a match generates the output compare flag.

Page 9

Monday, February 27, 2017 3:38 PM

Register Description: 16 bit

Step 1: Set the mode and polarity:

TCCRnA (0x24)– Timer/Counter Control Register A, n=1,3,4,5

Bit 7	6	5	4	3	2	1	bit 0
COMnA1	**COMnA0**	**COMnB1**	**COMnB0**	**COMnC0**	**COMnC0**	**WGMn1**	**WGMn0**
Reset 0	0	0	0	0	0	0	0

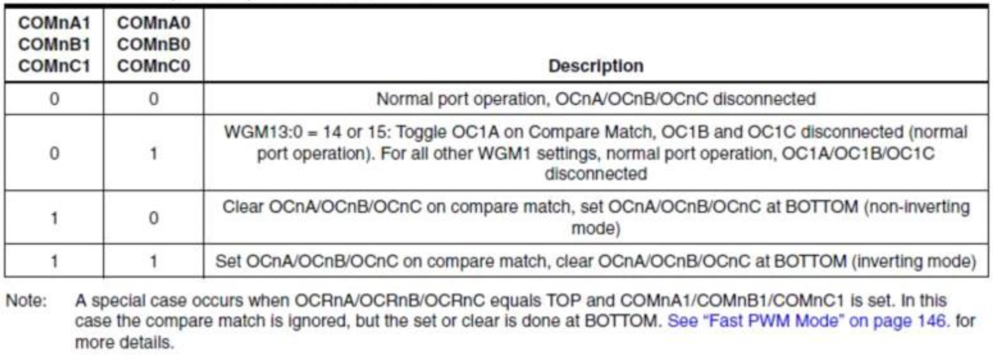
All R/W

With

COMnA1:0 Compare Match Output A Mode:

These control the Output Compare pin (OC0A) behavior and depend on WGM01:0 bit settings – set to fast PWM mode, see chart:

Table 17-4. Compare Output Mode, Fast PWM



COMnA1 COMnB1 COMnC1	COMnA0 COMnB0 COMnC0	Description
0	0	Normal port operation, OCnA/OCnB/OCnC disconnected
0	1	WGM13:0 = 14 or 15: Toggle OC1A on Compare Match, OC1B and OC1C disconnected (normal port operation). For all other WGM1 settings, normal port operation, OC1A/OC1B/OC1C disconnected
1	0	Clear OCnA/OCnB/OCnC on compare match, set OCnA/OCnB/OCnC at BOTTOM (non-inverting mode)
1	1	Set OCnA/OCnB/OCnC on compare match, clear OCnA/OCnB/OCnC at BOTTOM (inverting mode)

Note: A special case occurs when OCRnA/OCRnB/OCRnC equals TOP and COMnA1/COMnB1/COMnC1 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 146. for more details.

COM0B1:0 Compare Match Output B Mode:

These control the Output Compare pin (OC0B) behavior and depend on WGM01:0 bit settings – set to fast PWM mode the same as chart for OC0A.

WGMn1:0 Waveform Generation Mode

These are combined with the WGM02,3 bit found in the TCCR0B Register, are used to set Fast PWM.

Page 10

Monday, February 27, 2017 3:38 PM

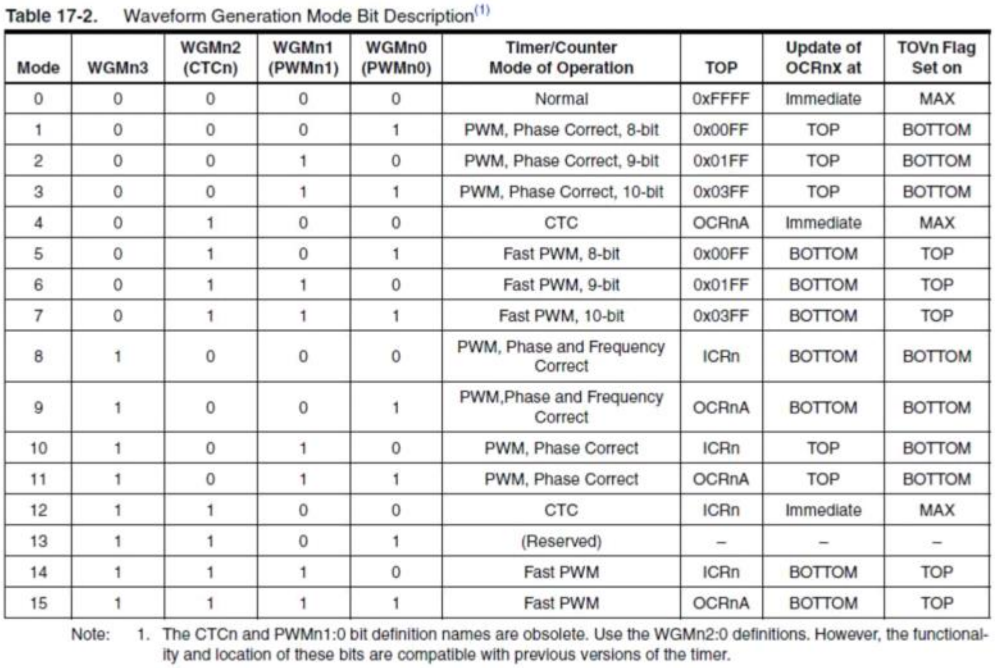


Table 17-2. Waveform Generation Mode Bit Description[1]

Mode	WGMn3	WGMn2 (CTCn)	WGMn1 (PWMn1)	WGMn0 (PWMn0)	Timer/Counter Mode of Operation	TOP	Update of OCRnX at	TOVn Flag Set on
0	0	0	0	0	Normal	0xFFFF	Immediate	MAX
1	0	0	0	1	PWM, Phase Correct, 8-bit	0x00FF	TOP	BOTTOM
2	0	0	1	0	PWM, Phase Correct, 9-bit	0x01FF	TOP	BOTTOM
3	0	0	1	1	PWM, Phase Correct, 10-bit	0x03FF	TOP	BOTTOM
4	0	1	0	0	CTC	OCRnA	Immediate	MAX
5	0	1	0	1	Fast PWM, 8-bit	0x00FF	BOTTOM	TOP ←
6	0	1	1	0	Fast PWM, 9-bit	0x01FF	BOTTOM	TOP ←
7	0	1	1	1	Fast PWM, 10-bit	0x03FF	BOTTOM	TOP ←
8	1	0	0	0	PWM, Phase and Frequency Correct	ICRn	BOTTOM	BOTTOM
9	1	0	0	1	PWM,Phase and Frequency Correct	OCRnA	BOTTOM	BOTTOM
10	1	0	1	0	PWM, Phase Correct	ICRn	TOP	BOTTOM
11	1	0	1	1	PWM, Phase Correct	OCRnA	TOP	BOTTOM
12	1	1	0	0	CTC	ICRn	Immediate	MAX
13	1	1	0	1	(Reserved)	–	–	–
14	1	1	1	0	Fast PWM	ICRn	BOTTOM	TOP
15	1	1	1	1	Fast PWM	OCRnA	BOTTOM	TOP

Note: 1. The CTCn and PWMn1:0 bit definition names are obsolete. Use the WGMn2:0 definitions. However, the functionality and location of these bits are compatible with previous versions of the timer.

Step 2: Set the prescale for the timer clock:

TCCRnB (0x25)– Timer/Counter Control Register B, n=1,3,4,5

Bit 7	6	5	4	3	2	1	bit 0
~~ICNCn~~	**~~ICESn~~**		**WGMn3**	**WGMn2**	**CSn2**	**CSn1**	**CSn0**
Reset 0	0	0	0	0	0	0	0

All R/W

With

ICNCn: Input Capture Noise Canceler

Noise canceler, the input from input capture pin is filtered. It requires four samples of the ICPn pin to charge output and therefore delays input capture by four oscillator cycles.

ICESn: Input Capture Edge select

Selects which edge on the ICPn that is used to trigger an event (zero is falling, one is rising)

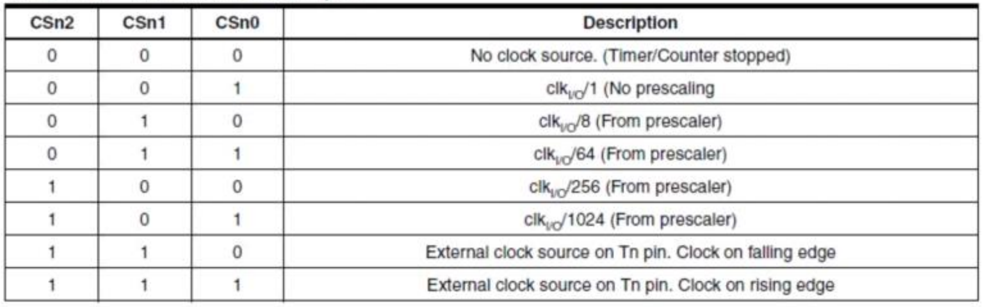
CS02:0 Clock Select

Three clock select bits select the clock source to be used by the Timer/Counter as shown in the table below.

Page 11

Monday, February 27, 2017 3:38 PM

Table 17-6. Clock Select Bit Description



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

Register Location of Timer counter

TCCRnC – Timer/Counter n Control register C (n= 1,3,4,5)
Only used in non-pwm mode.

Read only

TCNTnH,L– Timer/Counter Register H/L

Bit 7	6	5	4	3	2	1	bit 0
TCNT15							
Reset 0	0	0	0	0	0	0	0

All R/W

Bit 7	6	5	4	3	2	1	bit 0
TCNT07							
Reset 0	0	0	0	0	0	0	0

All R/W

This register contains the current count of the 16 bit Timer clock.

OCRnAH/L Output Compare Register A

Bit 7	6	5	4	3	2	1	bit 0
OCRnA15-0							
Reset 0	0	0	0	0	0	0	0

All R/W

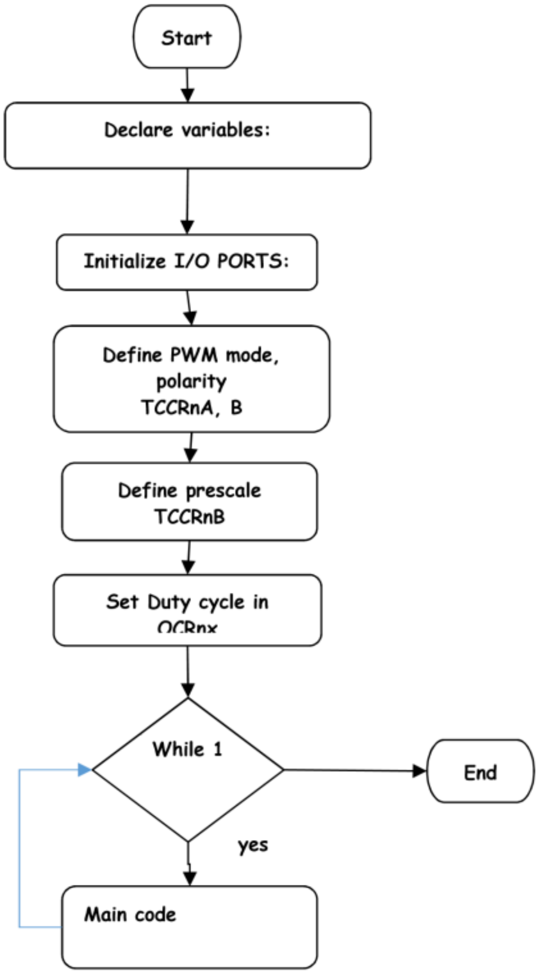
OCRnAHL contains 16-bit value to compare with TCNTnHL, a match generates the output compare flag.

Page 12

Monday, February 27, 2017 3:38 PM

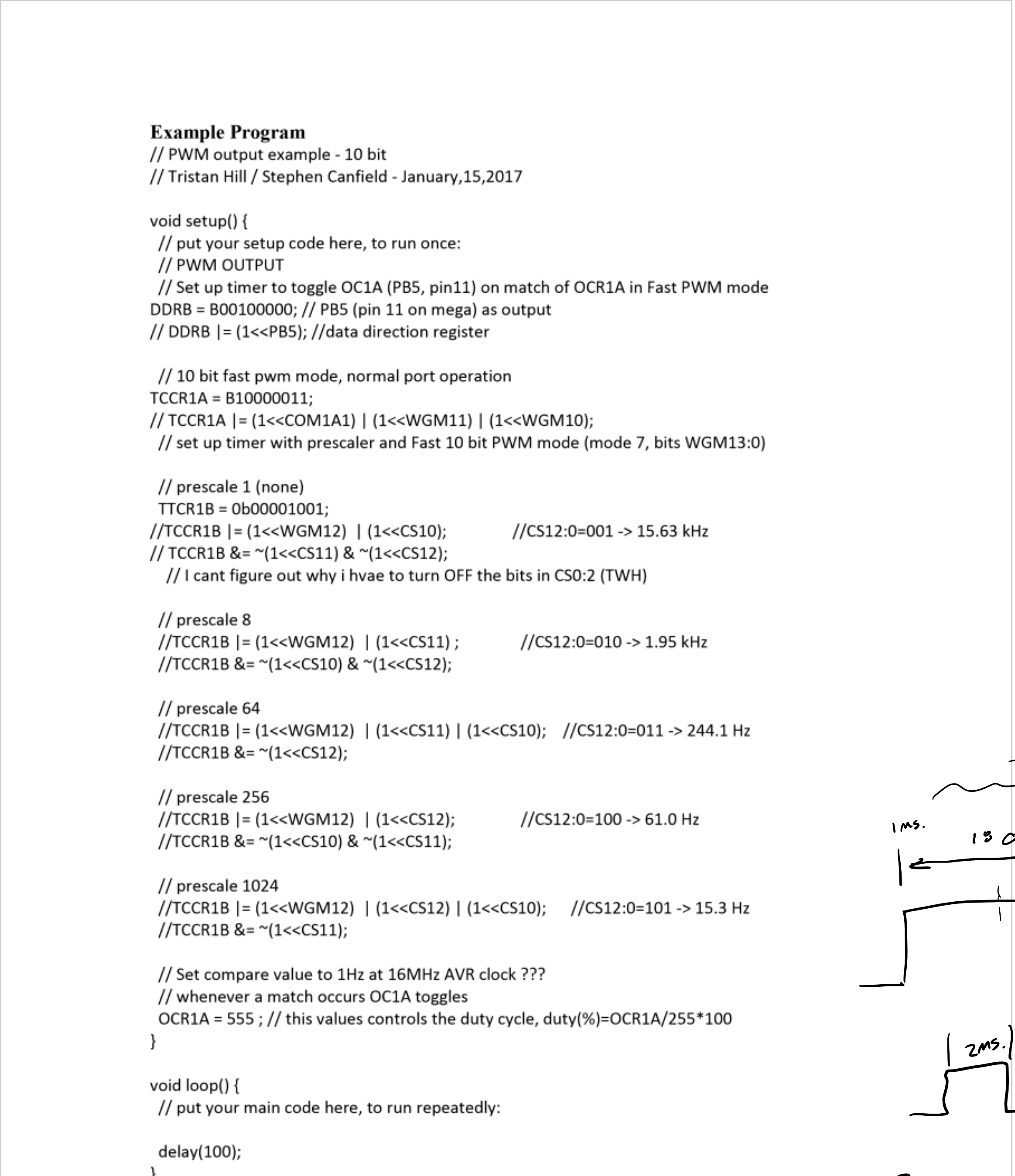
Summary: PWM Programming

1. Choose PWM resolution (8, 9,10 bit)
2. Choose PWM channel(s)
3. Set those channels to outputs (1 in DDRn)
4. Set the TCCRnA (n = 0 for 8 bit, 1,3,4,5 for 16 bit, x = A, B, C) to choose mode, and polarity (mode should be fast PWM)
5. Set the TCCRnB for remainder of mode, timer clock prescale
6. Set output compare value. This starts the PWM.
7. Change output compare value in code as needed.



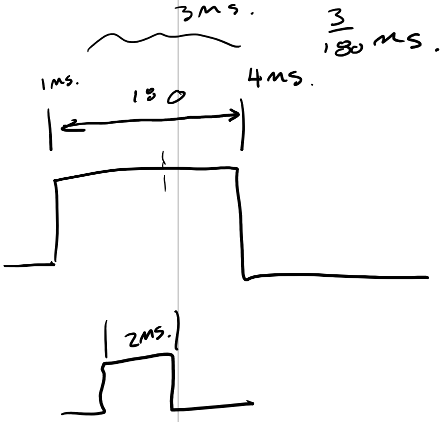
Page 13

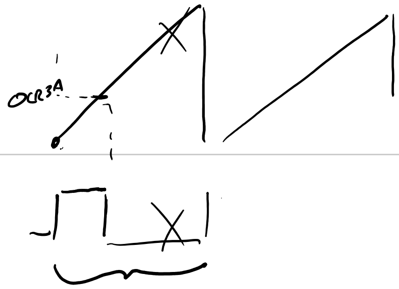
Monday, February 27, 2017 3:38 PM



Example Program

```
// PWM output example - 10 bit
// Tristan Hill / Stephen Canfield - January,15,2017

void setup() {
  // put your setup code here, to run once:
  // PWM OUTPUT
  // Set up timer to toggle OC1A (PB5, pin11) on match of OCR1A in Fast PWM mode
DDRB = B00100000; // PB5 (pin 11 on mega) as output
// DDRB |= (1<<PB5); //data direction register

  // 10 bit fast pwm mode, normal port operation
TCCR1A = B10000011;
// TCCR1A |= (1<<COM1A1) | (1<<WGM11) | (1<<WGM10);
  // set up timer with prescaler and Fast 10 bit PWM mode (mode 7, bits WGM13:0)

  // prescale 1 (none)
  TTCR1B = 0b00001001;
//TCCR1B |= (1<<WGM12)  | (1<<CS10);              //CS12:0=001 -> 15.63 kHz
// TCCR1B &= ~(1<<CS11) & ~(1<<CS12);
   // I cant figure out why i hvae to turn OFF the bits in CS0:2 (TWH)

  // prescale 8
  //TCCR1B |= (1<<WGM12)  | (1<<CS11) ;             //CS12:0=010 -> 1.95 kHz
  //TCCR1B &= ~(1<<CS10) & ~(1<<CS12);

  // prescale 64
  //TCCR1B |= (1<<WGM12)  | (1<<CS11) | (1<<CS10);    //CS12:0=011 -> 244.1 Hz
  //TCCR1B &= ~(1<<CS12);

  // prescale 256
  //TCCR1B |= (1<<WGM12)  | (1<<CS12);              //CS12:0=100 -> 61.0 Hz
  //TCCR1B &= ~(1<<CS10) & ~(1<<CS11);

  // prescale 1024
  //TCCR1B |= (1<<WGM12)  | (1<<CS12) | (1<<CS10);     //CS12:0=101 -> 15.3 Hz
  //TCCR1B &= ~(1<<CS11);

  // Set compare value to 1Hz at 16MHz AVR clock ???
  // whenever a match occurs OC1A toggles
  OCR1A = 555 ; // this values controls the duty cycle, duty(%)=OCR1A/255*100
}

void loop() {
  // put your main code here, to run repeatedly:

  delay(100);
}
```





? = OCR3A PE3?

$= 100 \times \frac{\text{seconds}}{\text{ct.}}$

DDRE = 0xFF

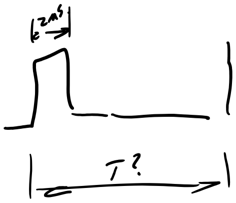
non inverted.

choose CS2:0
101 → ÷ by/1024

$\frac{\text{seconds}}{\text{ct}} = \frac{S * 1024}{16e6 \text{ cts}}$

$\frac{\text{cts}}{s} = \boxed{\frac{1024 \text{ ct.}}{16e6 \text{ s}}}$

$= 100 \cancel{\text{cts}} . \frac{1024 \text{ s}}{16e6 \cancel{\text{cts}}} = 6.4 \text{ ms.}$



Fast 10

$$\underset{\text{cts.}}{1024} * \frac{1024}{16e6} \cong 65\text{ms}$$

or

$$\text{OCR3A} \; \frac{1024}{16e6} = 2\text{ ms.}$$

$$\text{OCR3A} = \frac{2 * 16e6}{1024} = 31$$