Lab 8: Timer

Submit timers.asm by 5:00pm on March 9, 2018

I. Introduction

In lab 3, we wrote a program to make a LED light blink. The LED light was turned on for a while, then turned off for a while. The effect of "for a while" was implemented by writing a three nested loops doing nothing "nop". There is a more elegant way to do it – using a timer. The ATmega 2560 microcontroller includes 6 timers (two 8-bit and four 16-bit timer/counters):

- Four 16-bit timers (main focus of the lab)
  - Timer/Counter1, 3, 4, 5 are 16-bit Timer/Counter unit allows accurate program execution timing (event management), wave generation, and signal timing measurement. (P133 of the datasheet). [1]
- Two 8-bit timers
  - Timer/Counter0 is a general purpose 8-bit Timer/Counter module. It allows accurate program execution timing (event management) and wave generation. [1]
  - Timer/Counter2 is a general purpose, single channel, 8-bit Timer/Counter module.

We are going to use Timer3 in this lab. The 16-bit timer counts up or down between 0 and 65,535, independent of the execution of a program. There are 65,536 (2^{16}) steps. Actually, the timer can be configured to count between 0 to a predetermined value, let's call it TOP, depends on how much time the user set. The counter is incremented by a user selectable clock source, in this lab, we use the same clock signal that is supplied to the CPU, 16MHz. The CPU clock is very fast, to slow down the timer (note timer and counter are used interchangeably), the clock can be prescaled (divided) by selected powers of two. For example, if we want to have the LED light blink every 0.5 second, what should the TOP value be if the pre-scaler is set at 1024?

Since the highest number for a 16-bit timer/counter is 65,535, we need to check if **TOP ≤ 65,535**

II. Configure Timer/Counter3

The timers can be set to operate in different modes so that a right timer can be used in an application. The timer of the AVR can be specified to monitor several events. Some of the modes the timers can operate are: a) Timer Overflow mode (normal mode); b) Compare Match and c) Input capture. **For the purpose of this lab, we are going to setup the timer to operate only in the Compare Match mode**. Below is the summary of registers associated with Timer/Counter3:

- The 16-bit timer counter value is loaded at: TCNT3H:TCNT3L. The counter will start counting up from the initial value loaded here and count up to the value set in TOP. In the next clock pulse, the 16-bit counter will overflow and reset to 0x0000 and causes the OCF3A bit set in the TIFR3 register.
- The timer mode (CTC Clear Timber on Compare Match mode) is set by writing "0100" to WGM3(3:0) bits in TCCR3A and TCCR3B control registers.

- The timer can use an internal clock signal or an external signal as the source. In this lab, the timer is setup to use an internal clock. Note that the CPU is operating at 16Mhz. We will use this clock but reduce the counting speed by setting up a pre-scaler of 1024. That is the clock signal is divided by 1024 to reduce the counting speed. This is done using the TCCR3B control register.

III. Timer/Counter3 in AVR ATmega 2560

In this lab, the 16-bit Timer/Counter3 is used. The full names and addresses of the registers are:
.equ OCR3A = 0x98 ;Output Compare Register of Timer/Counter3, channel A **(stores TOP)**
.equ TCCR3A = 0x90 ;Timer/Counter3 Control Register
.equ TCCR3B = 0x91 ;Timer/Counter3 Control Register
.equ TCCR3C = 0x92 ;Timer/Counter3 Control Register C **(not used in this lab)**
.equ TCNT3H = 0x95 ; high byte of the Timer/Counter3
.equ TCNT3L = 0x94 ; low byte of the Timer/Counter3
.equ TIFR3 = 0x18(0x38) ;Timer/Counter3 Interrupt Flag Register
.equ TIMSK3 = 0x71 ;Timer/Counter3 Interrupt Mask Register **(not used in this lab)**

OCR3A - Output Compare Register of Timer/Counter3, channel A

Name	Address	Value	Bits
ICR3	na (0x96)		
OCR3A	na (0x98)		
OCR3AH	na (0x99)		
OCR3AL	na (0x98)		

The 16-bit register stores the TOP value that the timer is going to reach. The TOP value is between 0 and 0xffff (highest value for a 16 unsigned integer). Since the data bus is 8 bits, Special procedures must be followed when read/write to the 16-bit register. There is a single 8-bit register for temporary storing of the high byte. Accessing the low byte triggers the 16-bit "read" or "write" operation. When the CPU writes to the 16-bit register, it must write the high byte first. For a 16-bit read, the low byte must be read first. In this lab, OCR3A is set at 7813, which means 0.5 second delay.

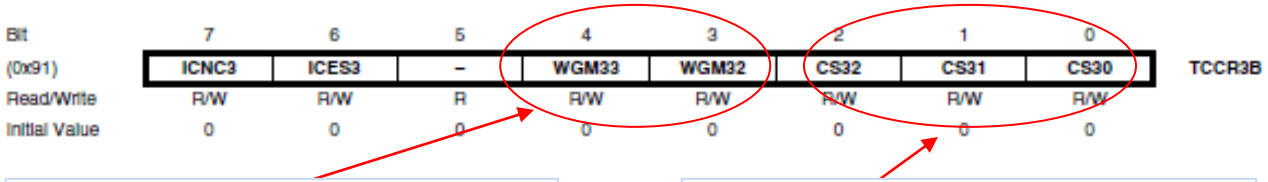
TCCR3A - Timer/Counter 3 Control Register A

Bit	7	6	5	4	3	2	1	0	
(0x90)	COM3A1	COM3A0	COM3B1	COM3B0	COM3C1	COM3C0	WGM31	WGM30	TCCR3A
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Set at 0b00. Means normal port operation, OC3A disconnected.

Combine with WGM_33: WGM_32 in TCCR3B, set timer3 to CTC mode (non-PWM) mode.

TCCR3B - Timer/Counter3 Control Register B



Bit	7	6	5	4	3	2	1	0	
(0x91)	ICNC3	ICES3	–	WGM33	WGM32	CS32	CS31	CS30	TCCR3B
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Combine with two bits in TCCR3A, $WGM_3$3:0 = 0b0100 set the timer mode of operation to CTC (Clear Timer on Compare Match mode)

The three clock select bits select the clock source to be used by timer3.

WGM stands for Waveform Generation Mode.

In this lab, the $CS_3$2:0 = 0b101, which means the CPU clock is used as source and the prescaler is set to 1024.

- **Bit 2:0 – CSn2:0: Clock Select**

The three clock select bits select the clock source to be used by the Timer/Counter, see Figure 17-10 and Figure 17-11 on page 152.

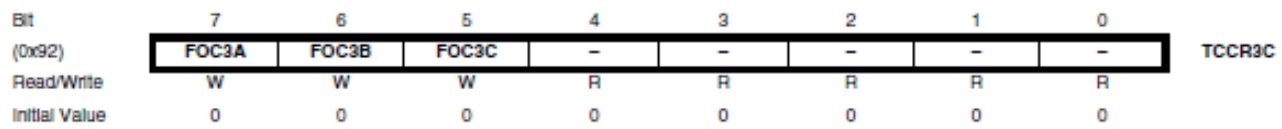
Table 17-6. Clock Select Bit Description

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

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

The least significant three bits of TCCR3B are used to slow down the timer in our example. Instead of counting 1 per clock cycle, it counts 1 every 1024 clock cycles in the example.

TCCR3C - Timer/Counter3 Control Register C



Bit	7	6	5	4	3	2	1	0	
(0x92)	FOC3A	FOC3B	FOC3C	–	–	–	–	–	TCCR3C
Read/Write	W	W	W	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

Not used in this example.

TIMSK3 – Timer/Counter3 Interrupt Mask Register

Bit	7	6	5	4	3	2	1	0	
(0x71)	–	–	ICIE3	–	OCIE3C	OCIE3B	OCIE3A	TOIE3	TIMSK3
Read/Write	R	R	R/W	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Not used in this example.

TIFR3 – Timer/Counter3 Interrupt Flag Register

Bit	7	6	5	4	3	2	1	0	
0x18 (0x38)	–	–	ICF3	–	OCF3C	OCF3B	OCF3A	TOV3	TIFR3
Read/Write	R	R	R/W	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Output Compare A Match Flag. It is set after the timer3 value matches the TOP (TCNT3 == OCR3A). In this lab, when OCF3A is 1, toggle LED, then clear bit OCF3A of TIFR3.

IV. Exercises: download timers.asm, implement delay by timer3 such that LED3 blinks every 0.5 second.

Optional, add timer4 and 5 such that the LED4 blinks every 1.5 seconds and LED5 blinks every 3 seconds.

Submit timers.asm by 5:00pm on March 9, 2018

V. References:

1. Section 17 (page 133) of the ATmega2560-datasheet.pdf at http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-2549-8-bit-AVR-Microcontroller-ATmega640-1280-1281-2560-2561_datasheet.pdf
2. Some Assembly Requited By Timothy S. Margush CRC Press, 2012 (page 288-297)