



Timer in AVR Micro-Controller

Required for Homework-4

Also Refer Chapter 14 and 15

Timers in AVR Micro-Controller

- Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
- One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
- In an 8-bit timer, the register used is 8-bit wide whereas in 16-bit timer, the register width is of 16 bits.
- This means that the 8-bit timer is capable of counting $2^8=256$ steps from 0 to 255
- What happens once they reach their MAX? Does the program stop executing?
 - ❖ It returns to its initial value of zero. We say that the timer/counter **overflows**.

Timer Concept

- Basic Concept of Timer:

$$\text{Time Period} = \frac{1}{\text{Frequency}}$$

$$\text{Timer Count} = \frac{\text{Required Delay}}{\text{Clock Time Period}} - 1$$

- Required Delay = 10 ms and Clock Time Period = 0.00025 ms (4MHz), and you get **Timer Count = 39999**.
- Assuming $F_{\text{CPU}} = 4 \text{ MHz}$ and a 16-bit timer (MAX = 65535), and substituting in the above formula, we can get a maximum delay of 16.384 ms.
- Now what if we need a greater delay, say 20 ms?

- Frequency Division is called as Pre-Scaling
- The actual F_{CPU} remains the same (at 4 MHz in this case). So basically, we *derive* a frequency from it to run the timer.
- **There is a trade-off between resolution and duration.**
- Overall duration measurement \uparrow , Resolution \uparrow thus accuracy is \downarrow
- Always choose prescaler which gives the counter value within the feasible limit (255 or 65535) and the counter value should always be an integer.

Problem Statement

- The simplest one being the LED flasher. Let's say, we need to flash an LED every 6 ms and we are have a CPU clock frequency of 32 kHz.
- For a delay of 6 ms, we need a timer count of 191. This can easily be achieved with an 8-bit counter (MAX = 255).
- We need help of following Registers:
 - ❖ TCNT0 Register
 - ❖ TCCR0 Register

Problem Statement

■ Timer/Counter Register

Bit	7	6	5	4	3	2	1	0	
	TCNT0[7:0]								TCNT0
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	

■ Timer/Counter Control Register

Bit	7	6	5	4	3	2	1	0	
	FOC0	WGM00	COM01	COM00	WGM01	CS02	CS01	CS00	TCCR0
Read/Write	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	

TCCR0 Register

- **Clock Select Bits, CS02:00**, we set the timer up by choosing proper prescaler.
- FOC0A: Force Output Compare A
- Bit 6, 3 – WGM01:0: Waveform Generation Mode

Problem Statement - Code

```
1 | #include <avr/io.h>
2 |
3 | void timer0_init()
4 | {
5 |     // set up timer with no prescaling
6 |     TCCR0 |= (1 << CS00);
7 |
8 |     // initialize counter
9 |     TCNT0 = 0;
10 | }
11 |
12 | int main(void)
13 | {
14 |     // connect led to pin PC0
15 |     DDRC |= (1 << 0);
16 |
17 |     // initialize timer
18 |     timer0_init();
19 |
20 |     // loop forever
21 |     while(1)
22 |     {
23 |         // check if the timer count reaches 191
24 |         if (TCNT0 >= 191)
25 |         {
26 |             PORTC ^= (1 << 0);    // toggles the led
27 |             TCNT0 = 0;           // reset counter
28 |         }
29 |     }
30 | }
```