Embedded System Laboratory Course

Dr. Fangning Hu

Protected: Lab 4 – Timer/Counter Overflow Interrupt

Introduction:

You may already notice that in the first lab, when we blink the LED, we need to call a subroutine Delay to keep the LED on/off for certain period. In this delay routine, the CPU/ALU is occupied by doing operation nothing more than a simple Counter. In order to save the CPU from working on such a simple counting job, we could use a separate Counter to count the clock cycles. This separate Counter is called Timer in microcontroller.

A Timer can either be an 8-bits counter (register) called **TCNT0** or a 16-bits counter called **TCNT1**. The counter will increase 1 for each clock cycles independently from the CPU and trigger an internal interrupt signal when the counter overflows (becomes 0 again). The interrupt vector is called **TIMER0_OVF** (for TCNT0) or **TIMER1_OVF** (for TCNT1). In order to enable this interrupt, you need to set the corresponding bit to 1 in the Timer Interrupt Mask Register **TIMSK0** (for TCNT0) or **TIMSK1** (for TCNT1).

Let's consider the 8-bit Timer for a start. There are two another important registers you need to take care. The first associated register is Timer/Counter Control Register A (TCCR0A), for now we don't need it, just keep it in the default value which is 0. The second one is Timer/Counter Control Register B (**TCCR0B**), here you need to set the corresponding bits (CS02, CS01, CS00) to
adjust the counting frequency.

By choosing different counting frequency and set an initial value to the counter, you may be able to design a program to have a Timer Overflow Interrupt occurs every 1 second. You can then toggle the PORTD output bits in the Timer Overflow Interrupt Service Routine every 1 second.

Timer1/Counter1 has the similar registers as Timer0/Counter0.

PreLab Tasks:

1.Read the ATmega328 datasheet (**Chapter Timer/Counter0 or Timer/Counter1 – 15.7.1 Normal Mode**) and study the functions of necessary registers (**TIMSK0/TIMSK1, TCCR0B/TCCR1B**) in order to have a Timer Overflow Interrupt. Calculate whether Timer 0 is big enough to have a 1 second overflow. If not, you should change to the 16-bits Timer/Counter1. Please calculate a proper combination of the initial value and a couting frequency to produce 1 second delay.

2. Read the ATmega328 datasheet (Chapter Interrupts) and find out the name of the Timer Overflow Interrupt Vector in the Interrupt Vector Table. Be careful to the different devices in the datasheet, they have different Tables!!

3. In order to program in Assembly language, please read Atmel AVR-8 Instruction Set and find out the usage of the assembly commands: MOV, IN, EOR. In order to flip the bit in PORTD, you first need to load out the status of PORTD to a register, the status of PORTD is stored in a register called PIND. Then flip the status by xor operation and load it back to PORTD.

Lab Assignments:

Make your LED blink by using Timer in C or Assembly language.

Lab Report: The requirements are the same as the previous lab.