

Fundamentals of Embedded Systems Design & Programming

U.C. Irvine Division of Continuing
Education

EECS X497.32

Instructor: Saleem Yamani

**Programming Assignment:
LED ON/OFF program using
Interrupt**

Programming Assignment: LED ON/OFF using Timer Interrupt

1. Use the Xmega timer 0(TCC0) hardware to create delay code by using the timer interrupt. Use the Programming assignment #4 and instead of polling for the overflow bit use the interrupt to change the state of a variable between 0 and 1 when ever the timer expires. In the main code instead of polling for the overflow bit, poll for the state of the variable to be 0, exit the polling when the variable changes from 0 to 1, which will happen when the next interrupt happens.
1. Use the ASFWizard to include the Timer Counter Module.
Search for the “xmega_tc_qs_ovf” for the Over flow use case steps
2. Make sure you make the interrupt variable a volatile
4. When using the interrupt method, you only need to initialize the timer once in the main function.

Programming Assignment: LED ON/OFF using Timer Interrupt

Code example for ISR Callback function

```
//Global data
volatile unsigned char timer_isr_state = 0;

static void my_callback(void)
{
    switch(timer_isr_state)
    {
        case 0:
            timer_isr_state = 1;
            break;

        case 1:
            timer_isr_state = 0;
            break;
    }
}
```

Programming Assignment: LED ON/OFF using Timer Interrupt

Code example for timer init

```
void timer_init(void)
{
    tc_enable(&TCC0);
    tc_set_overflow_interrupt_callback(&TCC0, my_callback);
    tc_set_wgm(&TCC0, TC_WG_NORMAL);
    tc_write_period(&TCC0, 1000);

    tc_set_overflow_interrupt_level(&TCC0, TC_INT_LVL_LO);
    cpu_irq_enable();
    tc_write_clock_source(&TCC0, TC_CLKSEL_DIV1024_gc);
}
```

Add the init functions in main

```
int main(void)
{
    pmic_init();
    sysclk_init();

    timer_init();
    .
    .
}
```