**CMPE 443 PRINCIPLES OF EMBEDDED SYSTEMS DESIGN**

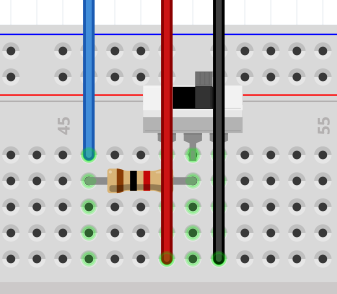
**PRELAB #008 “TIMER”**

1. **Problem Definition**

In this prelab, you will use a switch with RGB LED. According to the switching period, you will change the blinking rate of the LED. When the switching period is short, your LED will blink with high frequency, but if the switching period is long, the LED will blink at low frequency.

You will use Timer Interrupt for output compare and input capture.

1. **Switch**

****

In order to switch between, HIGH and LOW voltage, connect the right side of the switch with GND, left side of the switch with 3.3V, and connect the middle leg of the switch to the related PIN of the board with resistor.

1. **Timer Input Capture**

* Which timer you selected? TIM4
* Which timer channel you selected? Ch1
* Which pin you selected? PD12
* What is the Alternatif Function value for that pin? AF2 🡪 0010
* Configure prescaler to increase the CNT register for every 2 millisecond. (PSC)

TIM4->PSC = 8000 - 1;

* Enable capture interrupt? (DIER)

TIM4->DIER |= (0x01 << 1);

* Configure timer channel for input capture. (CCMRx)

TIM4->CCMR1 &= ~(0x3 << 0);

TIM4->CCMR1 |= (0x1 << 0);

* Enable input capture. (CCER))

TIM4->CCER |= (0x1 << 0);

* Enable IRQ for Timer

NVIC\_ISER1 |= (1 << 15);

* Write IRQ Handles and Clear interrupts (SR)
* **void** **TIM4\_IRQHandler**(**void**)
* {
* **static** **unsigned** **int** ic\_pin = LO;
* **if** (ic\_pin == LO) // HI has come
* {
* ic\_pin = HI;
* edgefirst = TIM4->CCR1;
* }
* **else** // LO has come
* {
* ic\_pin = LO;
* pulsewidth = TIM4->CCR1 - edgefirst;
* **if**(pulsewidth > 1000)
* pulsewidth = 1000;
* **if**(pulsewidth < 50)
* pulsewidth = 50;
* }
* TIM3->CCR1 = TIM3->CNT + pulsewidth/3;
* }

1. **Timer Output Compare**

* Which timer you selected? TIM3
* Which timer channel you selected? Ch1
* Which pin you selected? PA6
* What is the Alternatif Function value for that pin? AF2 🡪 0010
* Configure prescaler to increase the CNT register for every 5 millisecond. (PSC)

TIM3->PSC = 20000 – 1;

* Enable output compare interrupt? (DIER)

TIM3->DIER |= (0x01 << 1);

* Enable main output of the timer. (BDTR) No BDTR
* Configure timer channel for output compare and toggle the output. (CCMRx)

TIM3->CCMR1 &= ~(0x3 << 0);

TIM3->CCMR1 &= ~(0x7 << 4);

TIM3->CCMR1 &= ~(0x1 << 16);

TIM3->CCMR1 |= (0x3 << 4);

* Enable output compare. (CCER))

TIM3->CCER |= (0x1 << 0);

* Enable IRQ for Timer

NVIC\_ISER1 |= (1 << 14);

Write IRQ Handles and Clear interrupts (SR)

**void** **init\_TIM3**()

{

RCC\_APB1ENR1 |= 1 << 1; // enable tim3

// TIM3->EGR |= (0x01); // Generate update interrupt when CNT is reinitialized TIM3

TIM3->PSC = 20000 - 1; // Configure prescaler to increase the CNT register for every 2 millisecond TIM3

// TIM3->CR1 &= ~(1 << 1); // OVF will generate an event

TIM3->SR = 0; // green

TIM3->DIER |= (0x01 << 1); // Enable capture compare interrupt TIM3

// TIM3->CCMR1 &= ~(0x3 << 0);

TIM3->CCMR1 &= ~(0x7 << 4);

TIM3->CCMR1 &= ~(0x1 << 16);

TIM3->CCMR1 |= (0x3 << 4);

TIM3->CCER |= (0x1 << 0);

NVIC\_ISER1 |= (1 << 14);

}

1. **Code**

In this prelab, you need to write code as described at the problem definition. LED turn on/off time can change between 100ms - 2000ms according to switching.

1. **Submission**

You will submit one zip file which contains this document and your project (all the files with the last configuration)

The naming of the zip file should be:

PRELAB<exp num>\_<StudentID>.zip

1. **Related Videos and Links**

Timer Output Compare:

<https://www.youtube.com/watch?v=DQFvV3DJL54>

Timer Input Capture:

<https://www.youtube.com/watch?v=4qjRQL6MVRo>

Timer Input Capture Example Code:

<https://embeddedexpert.io/?p=816>

Timer Output Compare Example Code:

<https://embeddedexpert.io/?p=585>