TUTORIAL: Timer Input Capture

Timer Input Capture Setting

Name: 김은찬 ID: 21801017

# I. Introduction

In this lab, you will learn how to set up MCU Timers for Timer Interrupt and PWM output.

Objectives of this lab are learning how to

* Read and configure registers of General-purpose timers(TIMx) for Input Capture Mode
* Program firmwares to control frequency and interrupt events of a timer
* Create your own functions for timer input capture

### Hardware

NUCLEO -F411RE

### Software

Keil uVision IDE, CMSIS, EC\_HAL

### Documentation

[STM32 Reference Manual](https://ykkim.gitbook.io/ec/stm32-m4-programming/hardware/nucleo-f411re#manual-documentation)

# II. Basics of Timer Interrupt

## A. Register List

TIMx registers

|  |  |  |
| --- | --- | --- |
| Type | Register Name | Description |
| TIMx | TIMx\_ CR1 | TIMx control register 1 |
|  | TIMx\_ PSC | TIMx prescaler register |
|  | TIMx\_ARR | TIMx auto-reload register |
|  | TIMx\_DIER | TIMx DMA Interrupt Enable register |
|  | TIMx\_BDTR | TIM1(only) break and dead-time register |

PWM output registers

|  |  |  |
| --- | --- | --- |
| Type | Register Name | Description |
| TIMx | TIMx\_CCMRn | Capture/Compare Mode |
|  | TIMx\_CCRn | Capture/Compare |
|  | TIMx\_CCER | Capture/Compare Enable |
|  | TIMx\_SR | Status |

Schematic

* Block Diagram for General Purpose Timer

텍스트, 도표, 스크린샷, 라인이(가) 표시된 사진

자동 생성된 설명

* Block Diagram for General Purpose Timer

텍스트, 도표, 평면도, 개략도이(가) 표시된 사진

자동 생성된 설명

* Block Diagram for Input Capture (before IC)

**텍스트, 도표, 평면도, 라인이(가) 표시된 사진

자동 생성된 설명**

* Block Diagram for Input Capture (after IC)

**텍스트, 도표, 스크린샷, 평면도이(가) 표시된 사진

자동 생성된 설명**

## B. Register Setting

**System Clock setting**

1. RCC setting (PLL)

**[GPIO Pin setting]** (See PWM)

1. Set RCC

2. AF(TIMx) mode selection for Pin\_y in GPIOx

**[Timer setting]** (See PWM)

1. Enable Timer peripheral Clock (**RCC 🡪 APB1ENR**)

2. Set Counting direction (**TIMx 🡪 CR1: DIR**)

3. Set Timer clock Pre-scaler value (**TIMx 🡪 PSC: PSC**)

4. Set Auto-reload value (**TIMx 🡪 ARR: ARR**)

5. Enable counter. (**TIMx 🡪 CR1: CEN**)

**[Input Capture Configuration]**

1. Select Capture Input: **(TIMx 🡪 CCMRn: CCyS)**,
2. Set Filter duration **(TIMx 🡪 CCMRn: ICyF)**
3. Set Capture Input Prescaler **(TIMx 🡪 CCMRn: ICyPSC[1:0])**
4. Set Activation edge **(TIMx** 🡪 **CCER: CCyNP, CCyP)**
5. Enable Capture **(TIMx** 🡪 **CCER: CCyE)**
6. Enable Counter. (**TIMx 🡪 CR1: CEN**)
7. Enable CC, Update Interrupt **(TIMx 🡪 DIER: CCyIE, UIE)**

**[NVIC setting]**

1. Set the priority of TIMx\_IRQn

2. Enable NVIC interrupt.

3. TIMx\_IRQHandler()

# III. Tutorial

## A. Register Configuration

Fill in the table

|  |  |  |
| --- | --- | --- |
| **Register** | **Description** | **Register setting** |
| RCC | PLL Initialization | RCC\_PLL\_init();  EC\_SYS\_CLK = EC\_PLL = 84,000,000 |
| Enable Timer Peripheral Clock: TIM2 | RCC->APB1ENR |=1≪0 |
| TIM2  (Timer Initilization) | TIM2 of 100kHz CounterCLK & Set ARR=0xFFFF | TIM\_init(TIM2);  TIM2->ARR=0xFFFF; |
| TIM2  (Input Capture  setting) | Select Capture Input, IC1=TI1 | TIM2->CCMR1 |= 1 <<0 |
| Filter Duration N=4 | TIM2->CCMR1 |= 2 << 4 |
| IC1(CC1)= Rising edge | TIM2->CCER |= 0 << 1 |
| Capture Enable | TIM2->CCER |= 1 << 0 |
| Capture/Compare interrupt enable  Update Interrupt enable | TIM2->DIER |= 1 << 0 |

## B. Programming

**Procedure**

* Create a new folder ‘**EC/Tutorial/TU\_Timer\_InputCapture/**’
* Open the program ‘Keil uVision5’ and create a new project.
* Name the project as ‘**TU\_Timer\_InputCapture’**.
* Create a new item called ‘**TU\_Timer\_InputCapture.c**’
* Use the given source code of ‘**TU\_Timer\_InputCapture\_student2023.c**’

[Click to download](https://github.com/ykkimhgu/EC-student/blob/main/tutorial/tutorial-student/TU_Timer_InputCapture_student_2023.c)

* Download Input Capture Code of ‘**ecICAP\_student.c**’, ‘**ecICAP\_student.h**’

[Click to download](https://github.com/ykkimhgu/EC-student/tree/main/include/lib-student)

* Fill in the empty spaces in header code.
* Run the program and check your result.
* Your tutorial report must be submitted to LMS

**Exercise**

Input Capture with External Signal.

* System CLK is PLL 84MHz for STM32F411RE
* TIM2: Up-counting Timer2\_CLK = 100 kHz
* IC: TIM2 IC1 / PA\_0
* External Signal: Connect PA\_0 to Function Generator / Square Pulse at 0.5 Hz(or Using PWM).
* Set the GPIO pin as Alternate function (AF) for TIM2 (PA\_0)
* Give PWM output of 0%, 50%, 100% of 1kHz PWM
* Check the brightness of LD2 (PA5) for each different Duty ratio

Generate a slow pulse signals (0.5 Hz) from a function generator. Capture the input pulses and measure the pulse period in msec.

* Since there can be several overflow events during two consecutive capture events, you need to track the overflow countings in TIM2\_IRQHandler( ) and use it in measuring input pulse duration

HINT: If (TIM2->SR & TIM\_SR\_UIF) { // add logic to count overflow events }

* Print out the calculated period([ms]) and pulse count numbers of input signals and show the results on serial window of TeraTerm
* Print the result about every second.

## Appendix

[See here for MCU resources](https://ykkim.gitbook.io/ec/resource/nucleo-f411re)

텍스트, 번호, 도표, 스크린샷이(가) 표시된 사진

자동 생성된 설명