TUTORIAL: PWM

Timer Output Compare

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

* Configure registers of Timers (TIMx)
* Generate PWM signals

### 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 \_CCMRy | TIMx capture/compare mode register for yth channel |
|  | TIMx \_CCRy | TIMx capture/compare register for yth channel |
|  | TIMx \_CCER | TIMx capture/compare output enable register |

Schematic

텍스트, 도표, 스크린샷, 평면도이(가) 표시된 사진

자동 생성된 설명

## B. Register Setting

**System Clock setting**

1. RCC setting (PLL)

**GPIO Pin setting**

1. Set RCC for GPIO

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

**Timer setting:**

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

2. Set Timer clock Pre-scaler value (**TIMx🡪PSC : PSC[15:0]**)

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

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

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

**PWM Out setting:**

1. Set PWM Output mode (**TIMx🡪CCMR : OCyM**)

2. Set CompareCapture value (**TIMx🡪CCRy : CCR**)

3. Select Output Polarity (**TIMx🡪CCER : CCyP**)

4. Enable CompareCaptureOutput (**TIMx🡪CCER : CCyE**)

# 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 setting) | TIM2 counting direction: DIR0 | TIM2->CR1 & = ~ (1≪0) |
|  | Set Timer Clock Pre-scaler value  84MHz To 100kHz | TIM2->PSC = 839 |
|  | Set Auto-reload value:  With 100kHz, counting of 1kHz | TIM2->ARR = 99 |
|  | Enable Timer DMA/Interrupt | TIM2->DIER |= 1<<0 |
|  | Enable Counter | TIM2->CR1 |= 1<<0 |
| GPIOA | Set GPIOA pin 5 as Alternate Function mode: 10 | GPIOA->MODER &=~(1<<5\*2)  GPIOA->MODER |= 2 << (5\*2) |
|  | GPIOA Alternate Function Selection: AF1(TIM2/TMI5)  AFRL\_5[3:0]=0001 | GPIOA->AFR[0] = 1<<(4\*5) |
|  | AF Output as No-PUPD, Push-Pull,Very High Speed. | // write your HAL API  GPIO\_pupd(GPIOA, 5, EC\_NONE)  GPIO\_otype(GPIOA, 5,EC\_PUSH\_PULL)  GPIO\_ospeed (GPIOA,5, EC\_HIGH) |
| TIM2  (PWM setting) | Output compare mode as PWM mode1 | TIM2->CCMR1 |= 6 << 5 |
|  | Enable preload on CCR1 | TIM2->CCMR1 |= 1 << 4 |
|  | Set CCR value for 50% duty ratio: (ARR+1)/2 | TIM2->CCR1 = 99/2 |
|  | Output as active high | TIM2->CCER |= 0 << 1 |
|  | Enable Compare and Capture output | TIM2->CCER|= 1 << 0 |

## B. Programming

**Procedure**

* Create a new folder ‘**EC/Tutorial/TU\_Timer\_PWM/**’
* Open the program ‘Keil uVision5’ and create a new project.
* Name the project as ‘**TU\_Timer\_PWM’**.
* Create a new item called ‘**TU\_Timer\_PWM.c**’
* Use the given source code of ‘**TU\_Timer\_PWM\_student.c**’ [Click to download](https://github.com/ykkimhgu/EC-student/tree/main/tutorial/tutorial-student)
* Fill in the empty spaces in the code.
* Run the program and check your result.
* Your tutorial report must be submitted to LMS

**Exercise**

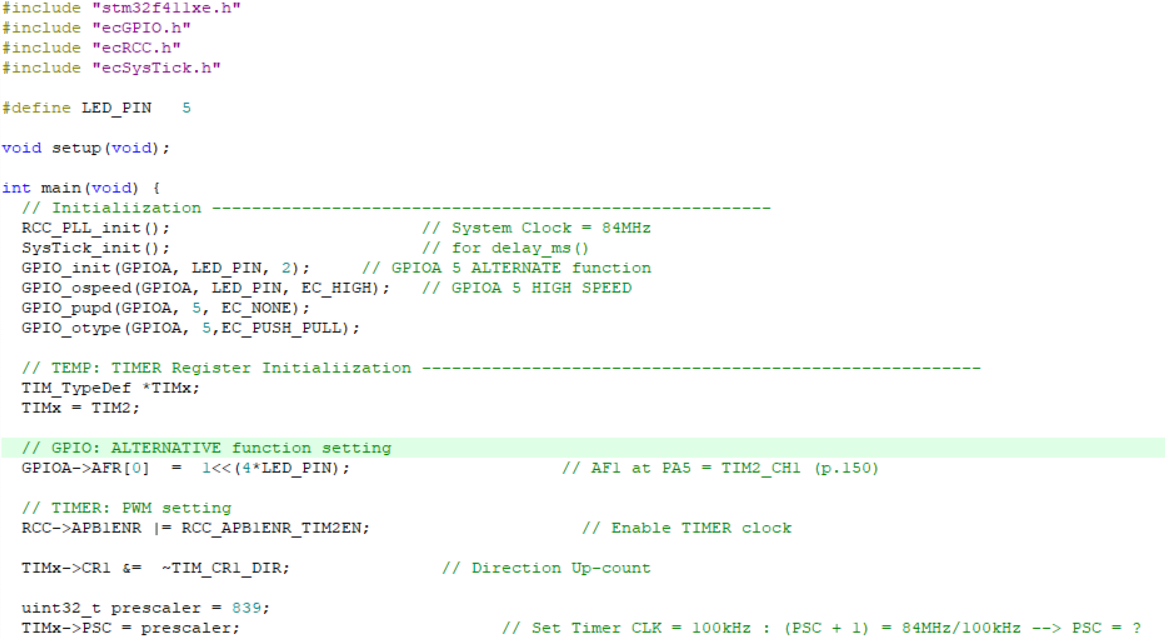
Output PWM on TIM2\_CH1 (PA5): 1kHz with Dutry-ratio of 0 / 50% / 100%

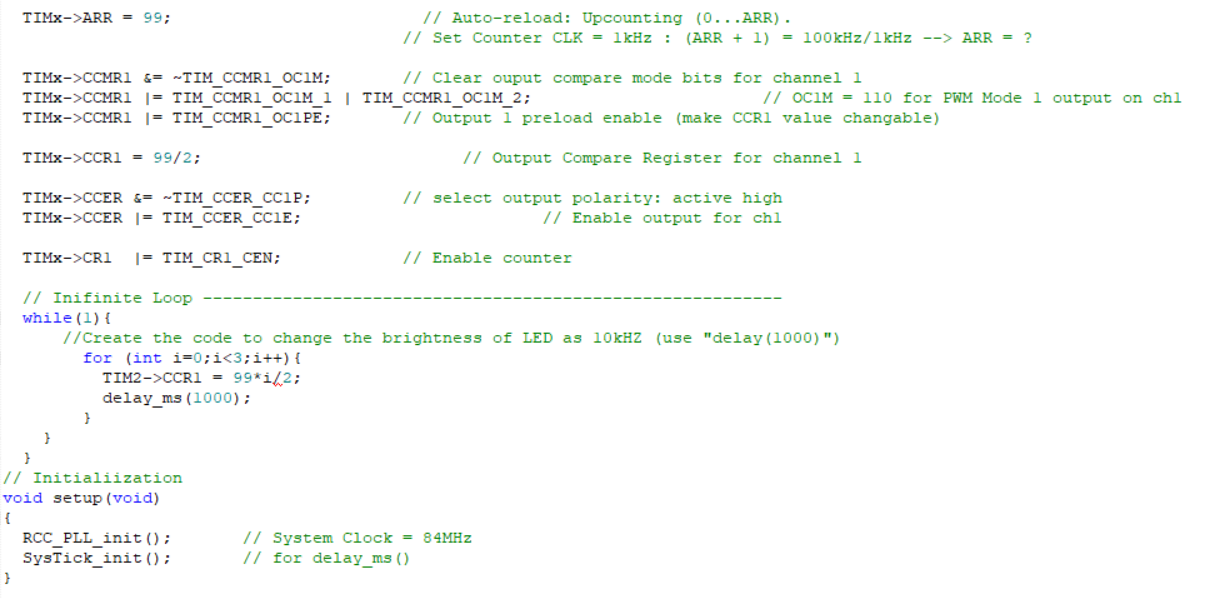
* System CLK is PLL 84MHz for STM32F411RE
* TIM2: Up-counting, Timer2\_CLK = 100 kHz, COUNT\_CLK=1 kHz
* PWM Mode 1 / Period 1kHz / Duty ratio 50%.
* PWM Out on PA\_5 (TIM2\_CH1)
* There are several pins for TIM2\_CH1. We will use GPIO Pin5 (LD2) for tutorial
* Set the GPIO pin as Alternate function (AF) for TIM2

: No pull-up & No pull-down / High speed / Push-Pull

: AF on PA\_5 (TIM2\_CH1)

* Give PWM output of 0%, 50%, 100% of 1kHz PWM
* Check the brightness of LD2 (PA5) for each different Duty ratio

****

****

## Appendix

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

텍스트, 번호, 도표, 스크린샷이(가) 표시된 사진

자동 생성된 설명

텍스트, 도표, 평행, 평면도이(가) 표시된 사진

자동 생성된 설명

* TIM\_TypeDef : <stm32f411xe.h>

텍스트, 스크린샷, 메뉴, 폰트이(가) 표시된 사진

자동 생성된 설명

텍스트, 폰트, 스크린샷, 화이트이(가) 표시된 사진

자동 생성된 설명

텍스트, 스크린샷, 폰트, 라인이(가) 표시된 사진

자동 생성된 설명