TUTORIAL: SysTick Interrupt

Timer and Time Delay

Name: 김은찬 ID: 21801017

# I. Introduction

In this tutorial, we will learn how to use SysTick interrupt. We will create functions to count up numbers at a constant rate using SysTick.

The objectives of this tutorial are how to

* Configure SysTick with NVIC
* Create your own functions for configuration of interrupts

### 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 SysTick

## A. Register List

List of SysTick registers for this tutorial. [**Programming Manual** ch4.3, ch10.2]

|  |  |  |
| --- | --- | --- |
| Type | Register Name | Description |
| SYSCFG\_ | SysTick\_CTRL  SysTick\_LOAD  SysTick\_VAL | Clock Control and Status  Reload Value  Current Value |

Schematic

**텍스트, 도표, 평면도, 스크린샷이(가) 표시된 사진

자동 생성된 설명**

## B. Register Setting

**(RCC system clock)**

* + - * 1. PLL, HCLK= 84MHz

**(System Tick Configuration)**

* + 1. Disable SysTick Timer
       - SysTick->CTRL ENABLE=0

2. Choose clock signal: System clock or ref. clock(STCLK)

* + - * SysTick->CTRL CLKSOURCE = 0 or 1

3. Choose to use Tick Interrupt (timer goes 1->0)

* + - * SysTick->CTRL TICKINT = 0 or 1

4. Write reload Counting value (24-bit)

* + - * SysTick->LOAD RELOAD = (value-1)

5. Start SysTick Timer

* + - * SysTick->CTRL ENABLE=1

6. (option) Read or Clear current counting value

* + - * Read from SysTick->VAL
      * Write clears value

**(NVIC Configuration)**

* NVIC SysTick Interrupt priority
* NVIC SysTick Enable

# III. Tutorial

## A. Register Configuration

Fill in the table

|  |  |  |
| --- | --- | --- |
| **Port/Pin** | **Description** | **Register setting** |
| SYSTICK | Disable SysTick Timer | SysTick->CTRL |= 0<<0 |
| Choose clock signal: System clock | SysTick->CTRL |= 1<<2 |
| Write reload Counting value  Make it 1ms / 1s | SysTick-> LOAD = 16000000 / 1000 – 1 |
| Start SysTick Timer | SysTick-> CTRL |= 1<<0 |
| Read from SysTick value | val = SysTick -> VAL |
| Clear from SysTick value | SysTick ->VAL = 0 |
| NVIC | Set SysTick Interrupt priority =15 | NVIC\_SetPriority(SysTick\_IRQn, 16) |
| NVIC SysTick Enable | NVIC\_EnableIRQ(SysTick\_IRQn) |

## B. Programming

**Procedure**

* Create a new folder ‘**EC/Tutorial/TU\_ SysTick /**’
* Open the program ‘Keil uVision5’ and create a new project.
* Name the project as ‘**TU\_ SysTick**’.
* Create a new item called ‘**TU\_SysTick.c**’
* Use the given source code [Click here to download](https://github.com/ykkimhgu/EC-student/tree/main/tutorial/tutorial-student)
* This is an example code for turning LED on/off with the button input trigger.
* Fill in the empty spaces in the code.
* Run the program and check your result.
* Your tutorial report must be submitted to LMS

**Exercise**

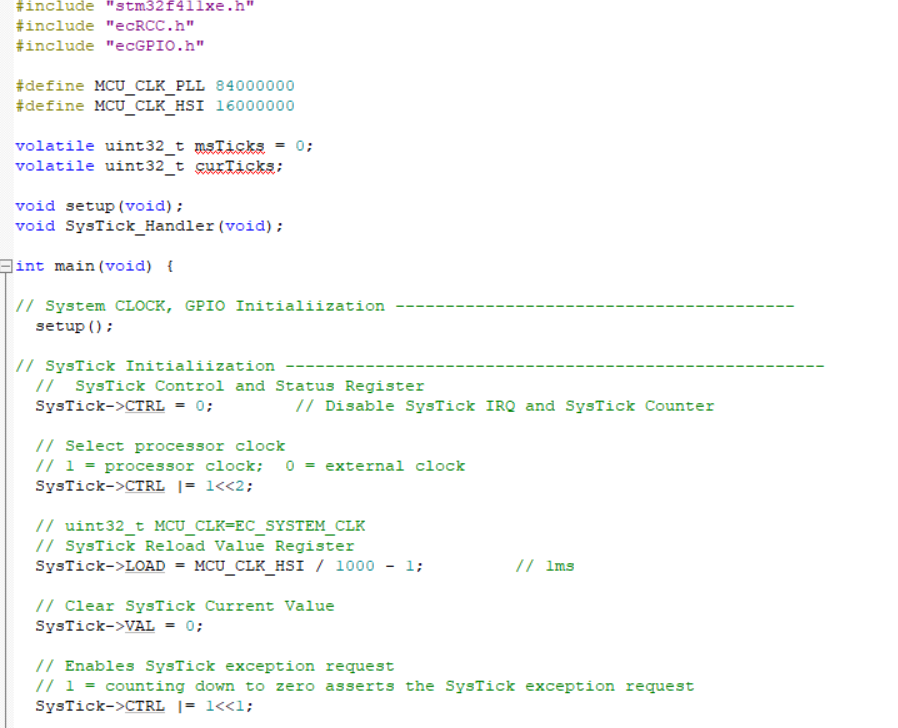
* Create a simple program that turns LED on/off at 1 second period.
* Set the SysTick to be 1msec.
* You can define the necessary timer or time wait function in the handler function of

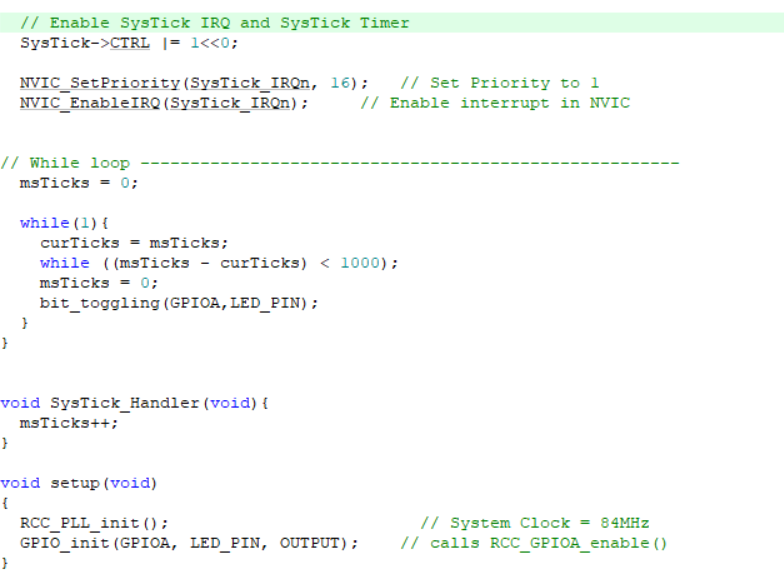
void Systick\_Handler(void)

Example:

void SysTick\_Handler(void){ msTicks++;}

* There are other methods for making time delay functions. [Check here for examples](https://ykkim.gitbook.io/ec/firmware-programming/peripheral-api/exti_systick#tutorial-systick-initiation)
* You can check some [sample codes here](https://github.com/ykkimhgu/EC-student/blob/main/include/lib-student/ecSysTick_student.c)





## Appendix

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

1. Pin Configuration of NUCLE-F401RE

**텍스트, 스크린샷, 폰트, 디자인이(가) 표시된 사진

자동 생성된 설명텍스트, 번호, 평행, 폰트이(가) 표시된 사진

자동 생성된 설명**

1. LED/Button Circuit Diagram

텍스트, 도표, 스크린샷, 라인이(가) 표시된 사진

자동 생성된 설명

텍스트, 도표, 지도, 평면도이(가) 표시된 사진

자동 생성된 설명