

# PreLAB: SysTick

Name: Dongjun, Lee

ID: 22201042

## 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 the configuration of interrupts

## Hardware

- NUCLEO -F411RE

## Software

- VS code, CMSIS, EC\_HAL

## Documentation

- [STM32 Reference Manual](#)

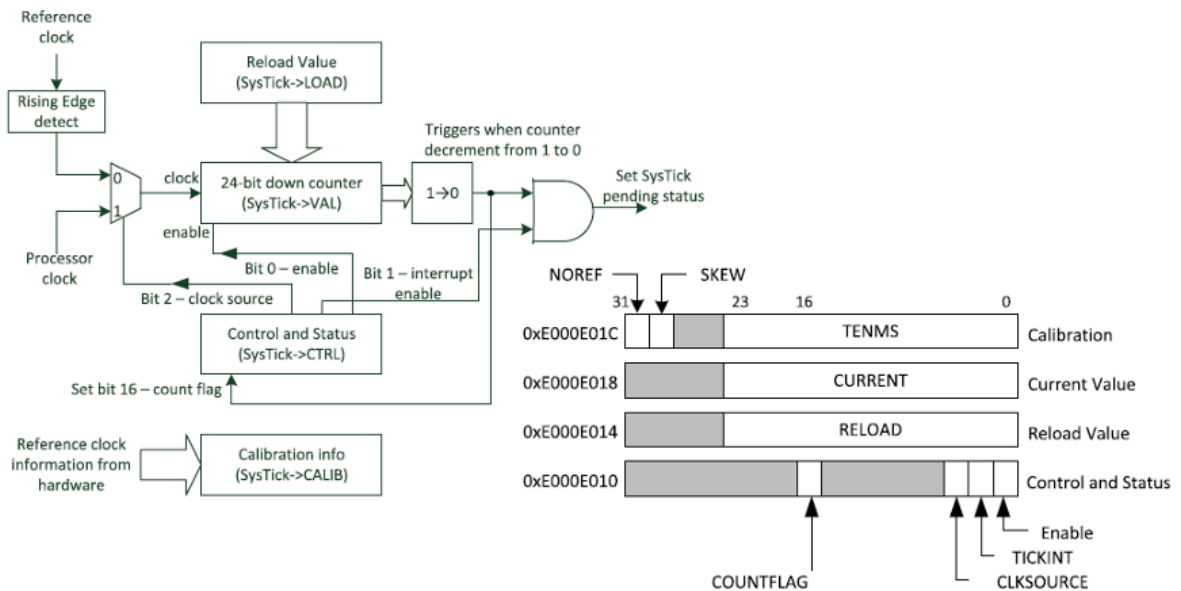
## II. Basics of SysTick

### A. Register List

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

Type	Register Name	Description
SYSCFG_	SysTick_CTRL	Clock Control and Status
	SysTick_LOAD	Reload Value
	SysTick_VAL	Current Value

## Schematic



**FIGURE 9.15**

A simplified block diagram of SysTick timer

## 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)

1. NVIC SysTick Interrupt priority
  2. NVIC SysTick Enable
- 

## III. Tutorial

### A. Programming

This is an example code for turning the LED on/off with the button input trigger with a wait function.

### Procedure

- Name the project as '**TU\_SysTick**' by creating a new folder as '**tutorial/TU\_SysTick**'
- Download the header library files and save under `include\`.
  - `ecSysTick2_student. ecSysTick2_student.c` : [Click here to download](#)
  - Rename the files as `ecSysTick2. ecSysTick2.c`
- Download the template code
  - `TU_SysTick_student.c` : [Click here to download](#)
- This is an example code for turning LED on/off with the button input trigger with a wait function.
- Fill in the empty spaces in the code.
- Run the program and check your result.
- Your tutorial report must be submitted to the LMS
- This is a sample program that turns LED on/off at 1 second period using SysTick

### Example Code

- Understand the code definition for void SysTick\_init() : in ecSysTick2.h
- Read the code definition for void delay\_ms( ) in ecSysTick2.h
- You can modify previous LAB code to include delay\_ms()

```

/**
*****
* @author  SSSLAB
* @Mod      2025-9-25 by YKKIM
* @brief    Embedded Controller: Tutorial ___
*          -  -----
*
*****
*/

#include "stm32f411xe.h"
#include "ecRCC2.h"
#include "ecGPIO2.h"
#include "ecSysTick2.h" // added

volatile uint32_t msTicks = 0;

void setup(void);

void main(void) {
    // System CLOCK, GPIO Initialiization -----
    -----
    setup();

    // While loop -----

    while(1){
        GPIO_write(PB_12, HIGH);
        delay_ms (1000);
        GPIO_write(PB_12, LOW);
        delay_ms (1000);
    }
}

void setup(void)
{
    RCC_PLL_init();           // System Clock = 84MHz
    //GPIO_init(PA_5, OUTPUT); // LED for Nucleo

```

```
    GPIO_init(PB_12, OUTPUT); // LED for Eval Board    JKIT
    SysTick_init();
}
```

```
tutorial > PreLAB_SysTick > C TU_SysTick_student.c > ...
1  /**
2  ****
3  * @author IILAB
4  * @Mod 2025-9-25 by YKKIM
5  * @brief Embedded Controller: Tutorial TU_SysTick_student
6  *
7  ****
8  */
9  #include "stm32f411xe.h"
10 #include "ecRCC2.h"
11 #include "ecGPIO2.h"
12 #include "ecSysTick2.h"
13 volatile uint32_t msTicks = 0;
14 void setup(void);
15 int main(void) {
16 // System CLOCK, GPIO Initialization -----
17     setup();
18 // While loop -----
19     while(1){
20         GPIO_write(PB_12, HIGH);
21         delay_ms (1000);
22         GPIO_write(PB_12, LOW);
23         delay_ms (1000);
24     }
25 }
26 void setup(void)
27 {
28     RCC_PLL_init(); // System Clock = 84MHz
29     //GPIO_init(PA_5, OUTPUT); // LED for Nucleo
30     GPIO_init(PB_12, OUTPUT); // LED for Eval Board    JKIT
31     SysTick_init();
32 }
33
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Environment	Status	Duration
PreLAB_SysTick	SUCCESS	00:00:03.900

===== 1 succeeded in 00:00:03.900 =====

Terminal will be reused by tasks, press any key to close it.

env:PreLAB\_SysTick (EC-Dongjun-042) Auto Ln 12, Col 24 Tab Size: 4 UTF-8 LF () C