Line Tracer 04

- Systick Timer -

Notice

- Midterm Exam
    - DATE : 10/17
    - TIME : 8:00 AM

This lecture is based on

- **Systick Timer**

1. What is Systick Timer

About Systick Timer

What is Systick Timer?

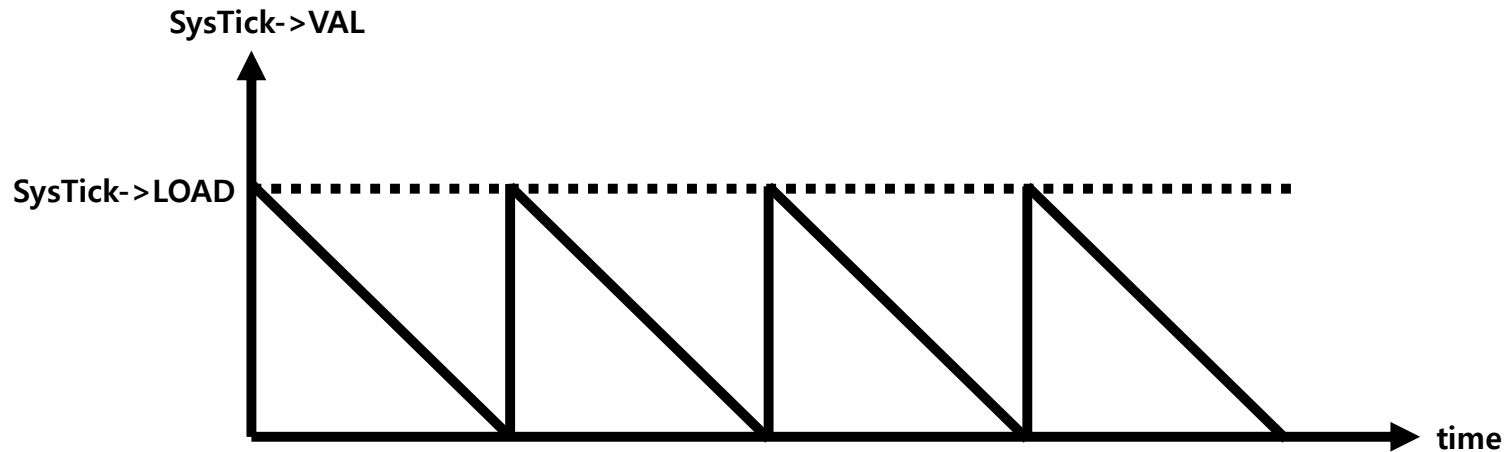- **A simple timer that performs timer/counter operation in all ARM**

Where is Systick Timer used?

- **Create time delays**
- **Generate periodic interrupts**

How It Works?

How Systick Timer Works?

- **24-bit** down counter decrements at **bus clock frequency**
- The counter, VAL, goes from **LOAD -> 0**

Background

What is Clock Cycle?

- The amount of time between two pulses of an oscillator

➡ **"The smallest unit of processor activity"**

What is Clock Frequency (Clock Speed)?

- The frequency at which the clock generator of a processor can generate pulses

➡ **"How many cycles done per second"**
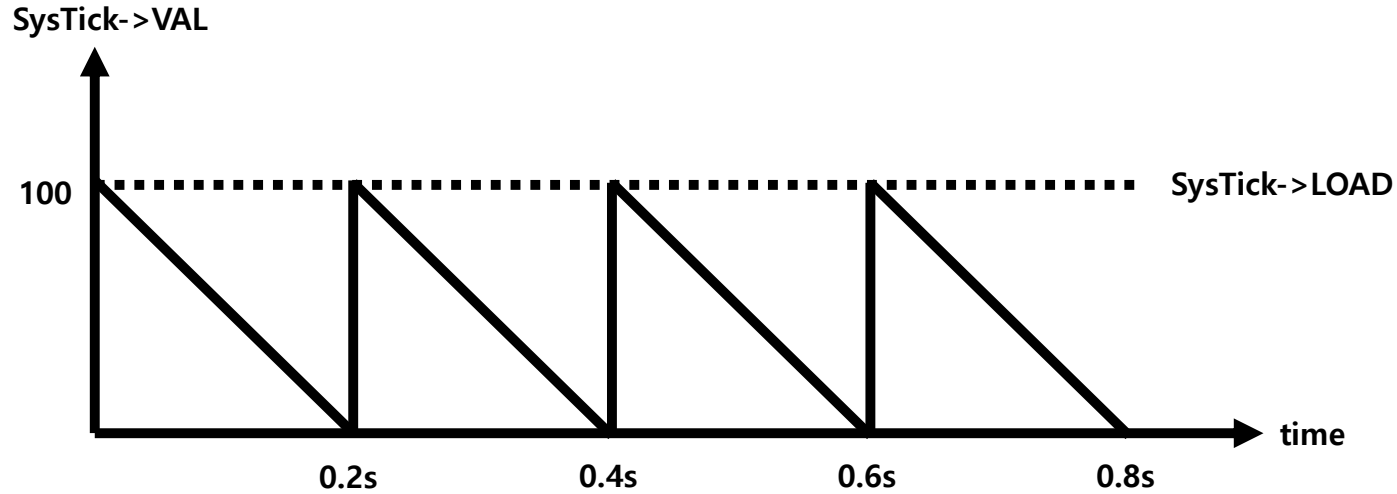
ex1) 100Hz   -> 1 cycle done in 0.01s

3.4GHz -> $3.4 * 10^9$ cycles done in 1s

ex2) MUL Rd, Rn, Rm   -> 2 cycle

3.4GHz System, -> $1.7 * 10^9$ MUL instructions / s

SysTick->LOAD = 100, Clock Speed = 500Hz

SysTick->VAL decreases by 1 per 0.002s

SysTick->VAL decreases by 100 (reset) per 0.2s

How Many Reset Occurs?

- SysTick->LOAD has maximum value (0xFFFFFF)

- Processor Clock is 48MHz

⟶   0xFFFFFF = 16,777,216

48MHz = 48,000,000

16,777,216 / 48,000,000 = 0.349525333...

Reset occurs approximately every 349ms!

2. SysTick Timer Implementation

Systick Timer Register

31-24	23-17	16	15-3	2	1	0	Name
0	0	COUNT	0	CLK_SRC	INTEN	ENABLE	SysTick->CTRL
0	24-bit RELOAD value						SysTick->LOAD
0	24-bit CURRENT value of SysTick Counter						SysTick->VAL

COUNT = Returns 1 if timer counted to 0 since last time this was read

CLK_SRC = Indicates the clock source, 0 = external clock, 1 = processor clock

INTEN = 0 : Disable Interrupt, 1 : Enable Interrupt

ENABLE = 0 : counter disabled, 1 : counter enabled

Systick Timer Initialization

```c
void systick_init(void) {
    SysTick->LOAD = 0x00FFFFFF;
    SysTick->CTRL = 0x00000005;
}

void systick_wait1ms() {
    SysTick->LOAD =          ;
    SysTick->VAL = 0;
    while((SysTick->CTRL & 0x00010000) == 0) {};
}

void systick_wait1s() {
    int i;
    int count = 1000;

    for (i = 0; i < count; i++) {
        systick_wait1ms();
    }
}
```

Systick Timer Example
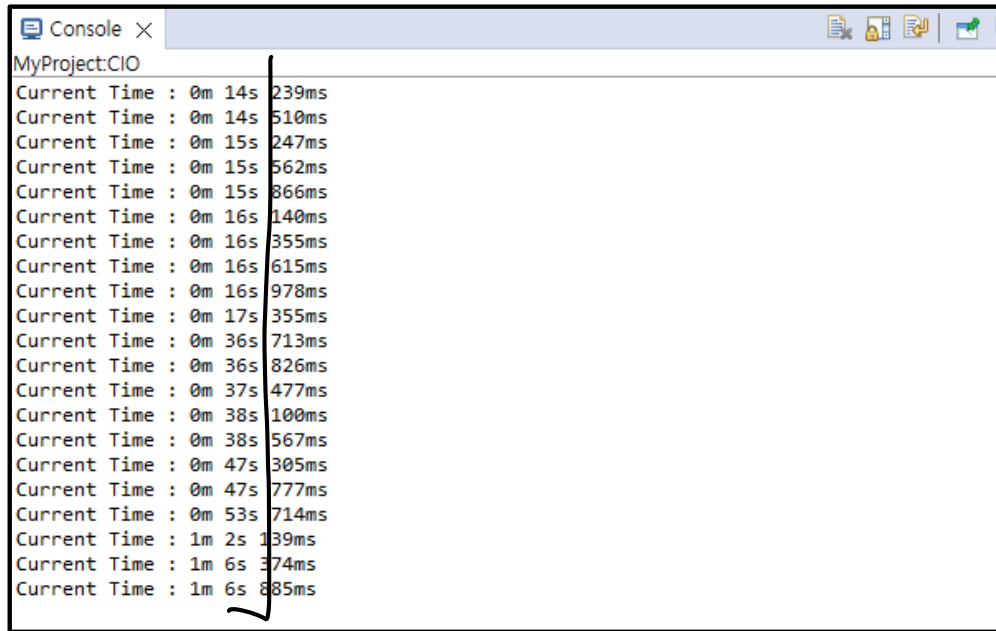
```c
void main(void)
{
    // Initialization
    Clock_Init48MHz();
    led_init();
    switch_init();
    systick_init();

    while (1) {
        turn_on_led(LED_RED);
        systick_wait1s();
        turn_on_led(LED_GREEN);
        systick_wait1s();
    }
}
```

3. SysTick Timer Activity

Systick Timer Activity(Not Assignment)

Print how long time has passed when the button is pressed

```
Console ×
MyProject:CIO
Current Time : 0m 14s 239ms
Current Time : 0m 14s 510ms
Current Time : 0m 15s 247ms
Current Time : 0m 15s 562ms
Current Time : 0m 15s 866ms
Current Time : 0m 16s 140ms
Current Time : 0m 16s 355ms
Current Time : 0m 16s 615ms
Current Time : 0m 16s 978ms
Current Time : 0m 17s 355ms
Current Time : 0m 36s 713ms
Current Time : 0m 36s 826ms
Current Time : 0m 37s 477ms
Current Time : 0m 38s 100ms
Current Time : 0m 38s 567ms
Current Time : 0m 47s 305ms
Current Time : 0m 47s 777ms
Current Time : 0m 53s 714ms
Current Time : 1m 2s 139ms
Current Time : 1m 6s 374ms
Current Time : 1m 6s 885ms
```