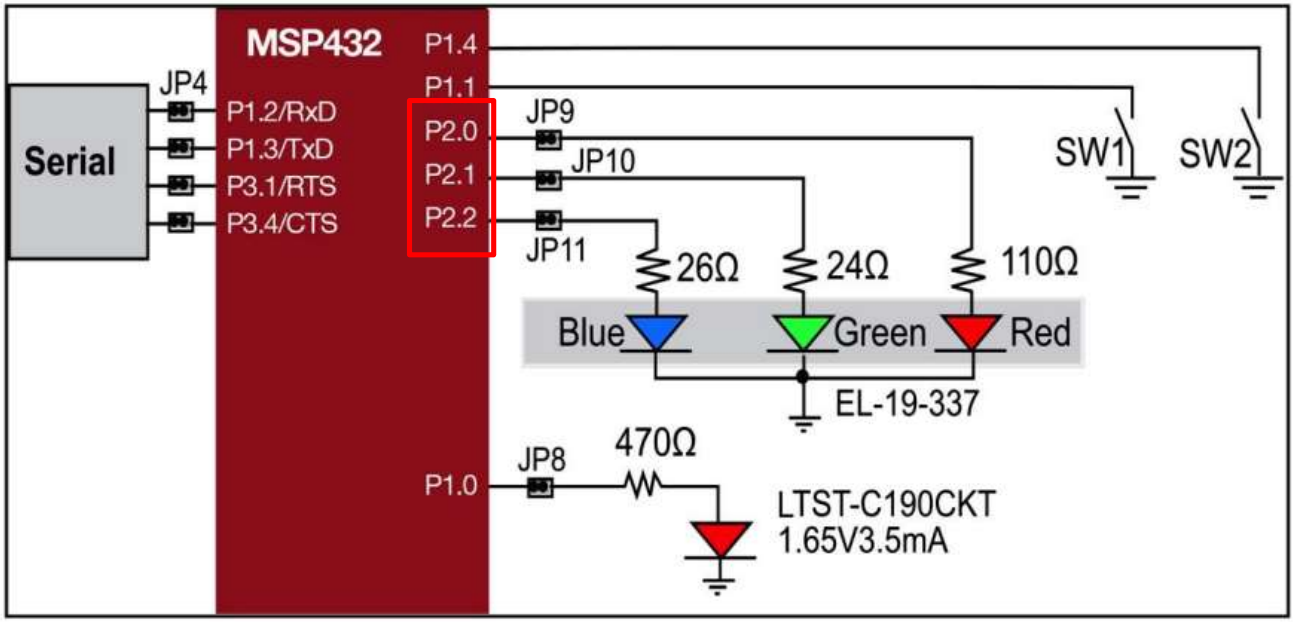
Line Tracer 03

- Systick Timer -

This lecture is based on

- Systick Timer

MSP432 LaunchPad LED & Switch



1. What is Systick Timer

About Systick Timer

What is Systick Timer?

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

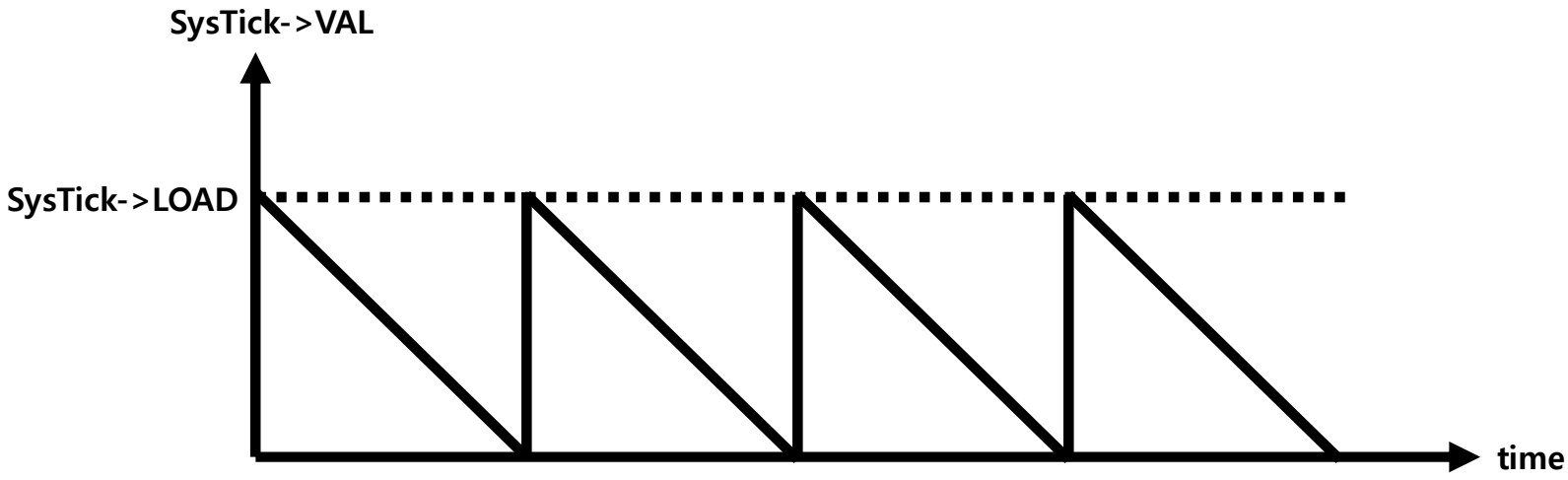
Where is Systick Timer used?

- 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



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

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 Initialization

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

void systick_wait1ms() {
    SysTick->LOAD = [redacted];
    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

```
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();
    }
}
```

Systick Timer Activity

Print how long time has passed when the button is pressed