Line Tracer 03

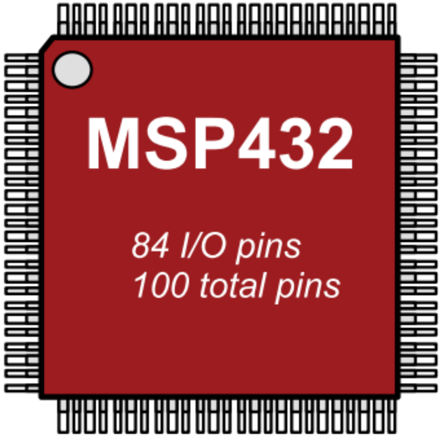
- LED & Switch -

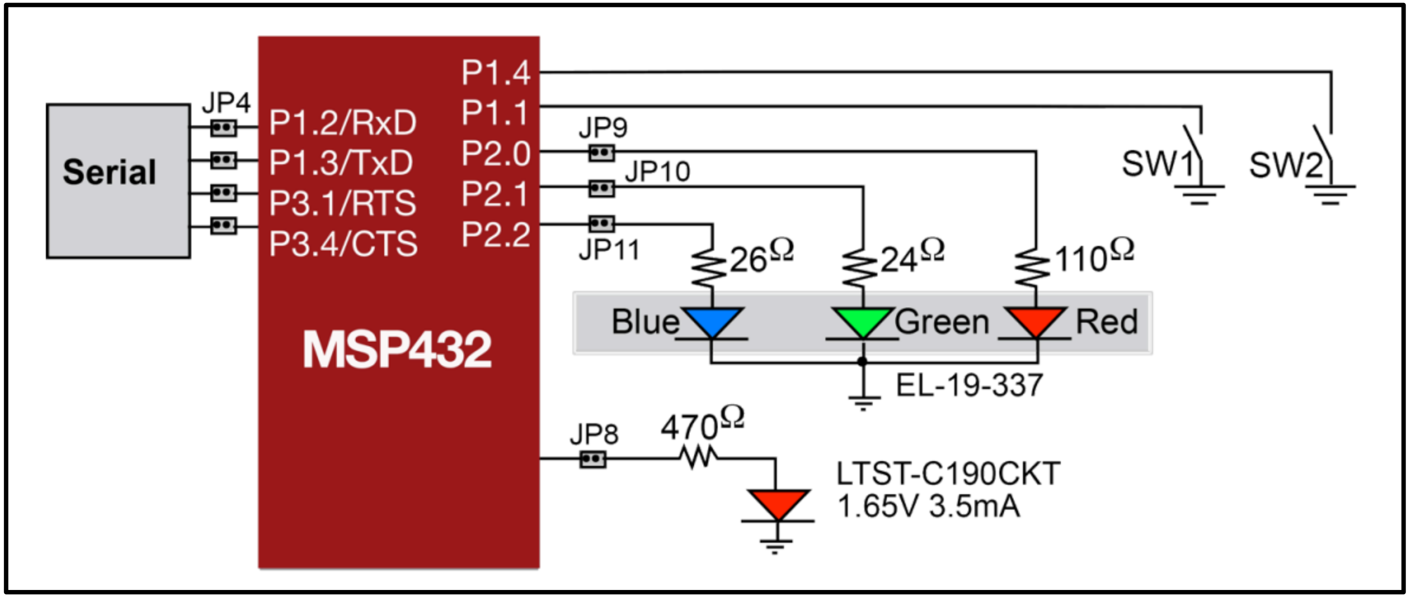
This lecture is based on

- GPIO
- Interfacing Input and Output

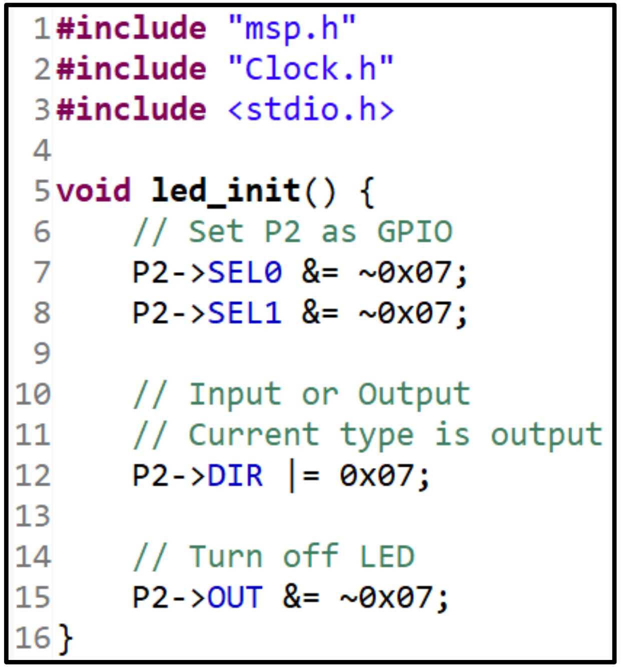
1. LED

LED Initialization

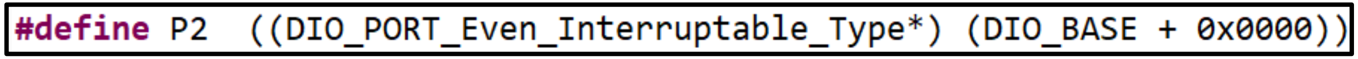




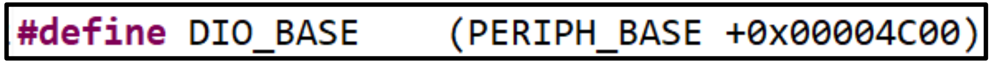
LED Initialization



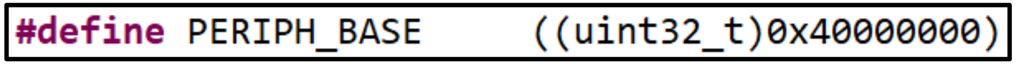
```c
#include "msp.h"
#include "Clock.h"
#include <stdio.h>

void led_init() {
    // Set P2 as GPIO
    P2->SEL0 &= ~0x07;
    P2->SEL1 &= ~0x07;

    // Input or Output
    // Current type is output
    P2->DIR |= 0x07;

    // Turn off LED
    P2->OUT &= ~0x07;
}
```



```c
#define P2  ((DIO_PORT_Even_Interruptable_Type*) (DIO_BASE + 0x0000))
```

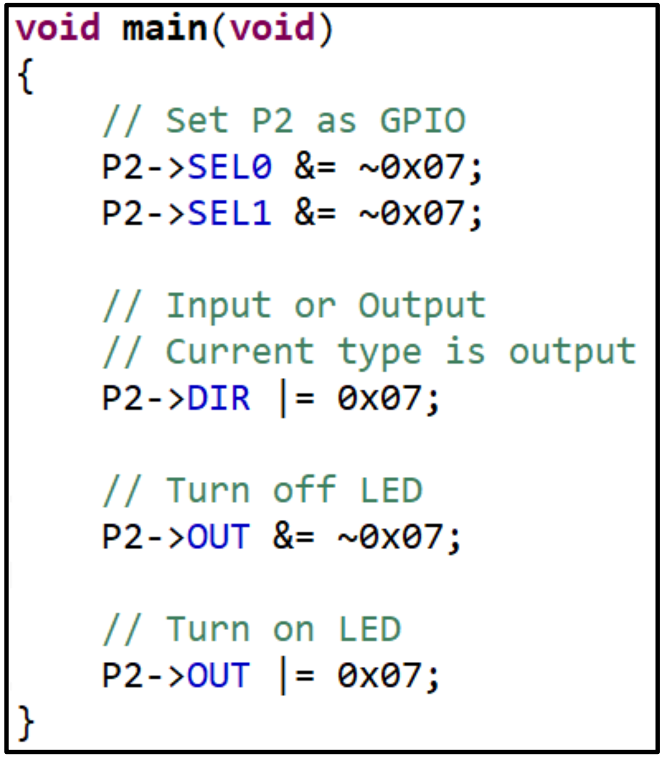


```c
#define DIO_BASE     (PERIPH_BASE +0x00004C00)
```



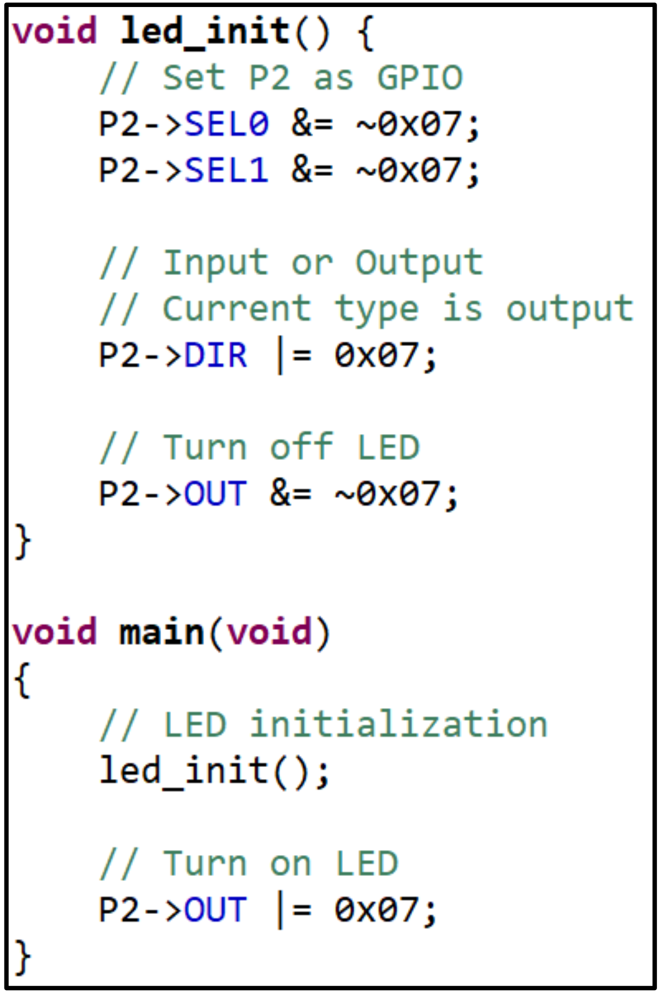
```c
#define PERIPH_BASE      ((uint32_t)0x40000000)
```

LED Initialization



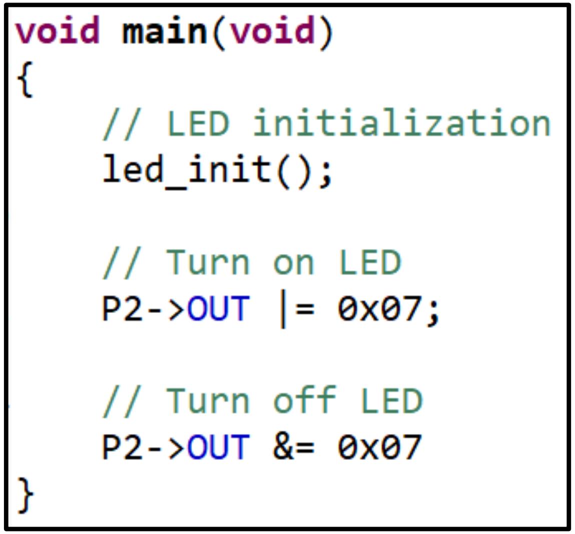
```c
void main(void)
{
    // Set P2 as GPIO
    P2->SEL0 &= ~0x07;
    P2->SEL1 &= ~0x07;

    // Input or Output
    // Current type is output
    P2->DIR |= 0x07;

    // Turn off LED
    P2->OUT &= ~0x07;

    // Turn on LED
    P2->OUT |= 0x07;
}
```

VS



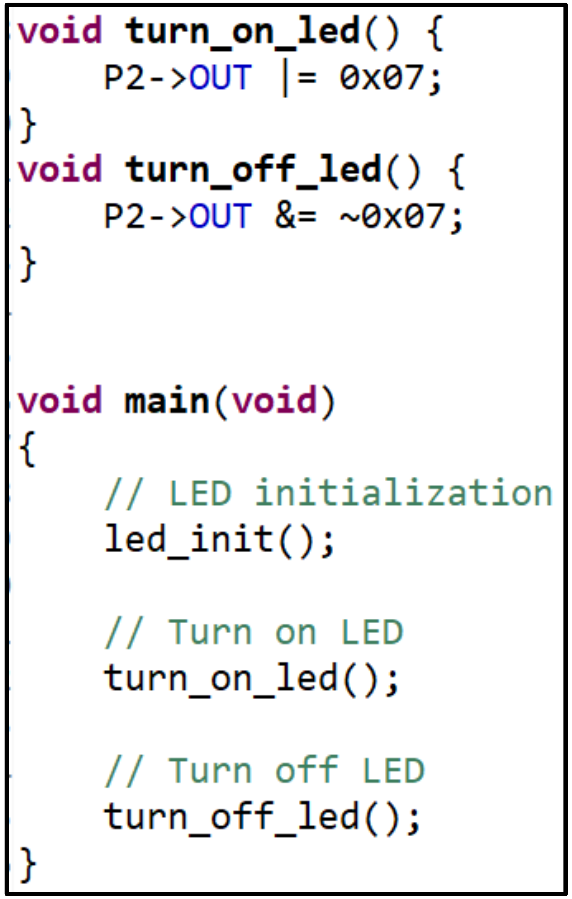
```c
void led_init() {
    // Set P2 as GPIO
    P2->SEL0 &= ~0x07;
    P2->SEL1 &= ~0x07;

    // Input or Output
    // Current type is output
    P2->DIR |= 0x07;

    // Turn off LED
    P2->OUT &= ~0x07;
}

void main(void)
{
    // LED initialization
    led_init();

    // Turn on LED
    P2->OUT |= 0x07;
}
```

Turn On & Off LED



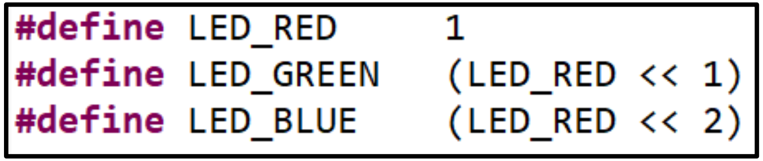
```c
void main(void)
{
    // LED initialization
    led_init();

    // Turn on LED
    P2->OUT |= 0x07;

    // Turn off LED
    P2->OUT &= 0x07
}
```

VS

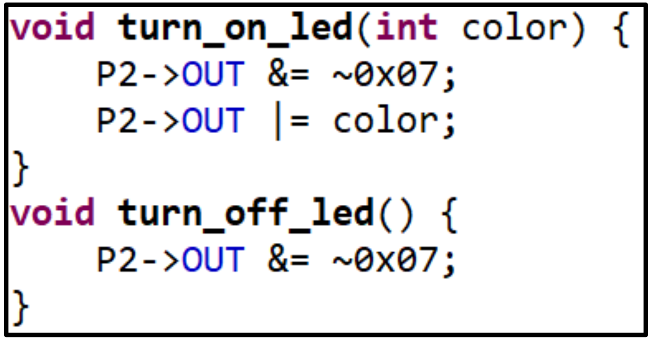


```c
void turn_on_led() {
    P2->OUT |= 0x07;
}
void turn_off_led() {
    P2->OUT &= ~0x07;
}


void main(void)
{
    // LED initialization
    led_init();

    // Turn on LED
    turn_on_led();

    // Turn off LED
    turn_off_led();
}
```

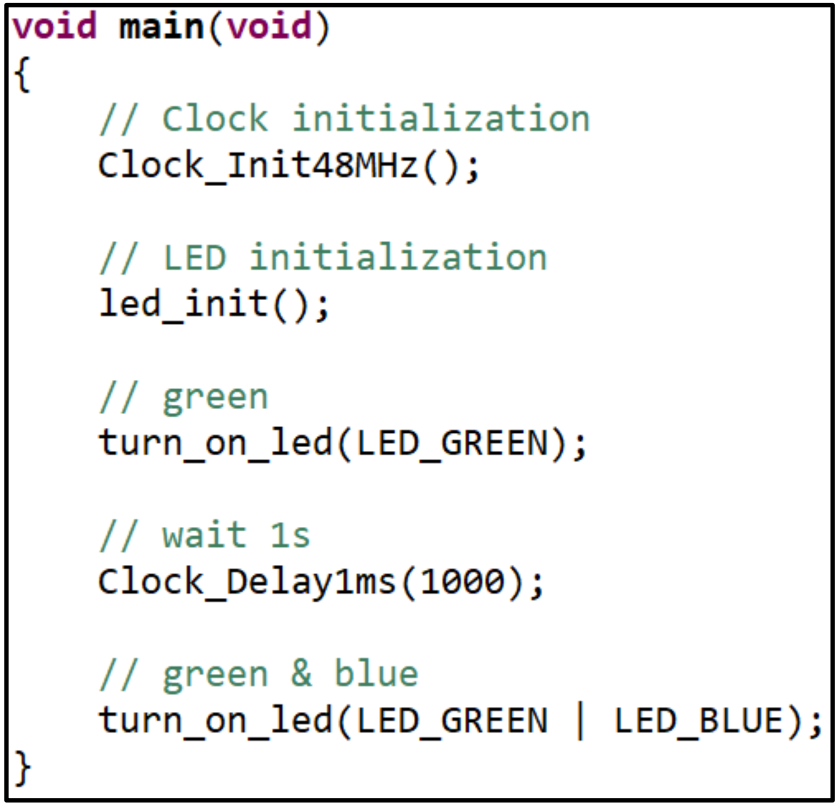
Turn On & Off LED



```c
#define LED_RED     1
#define LED_GREEN   (LED_RED << 1)
#define LED_BLUE    (LED_RED << 2)
```



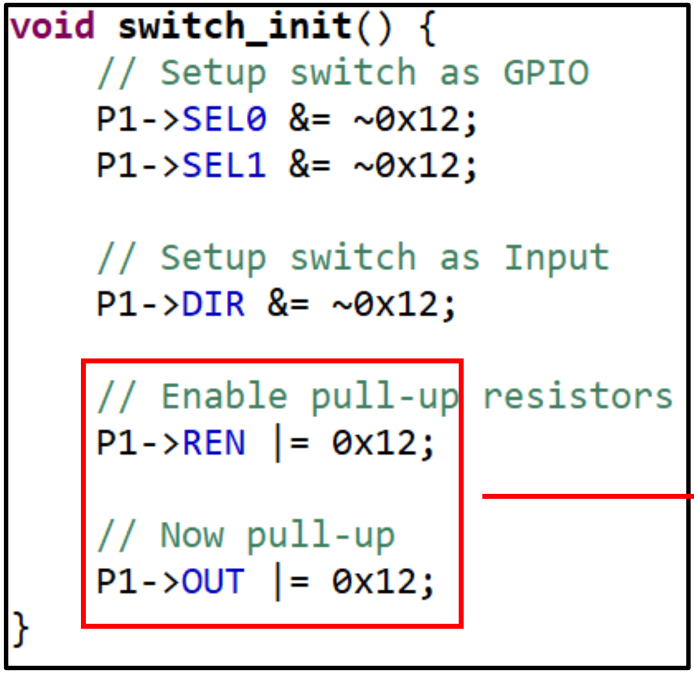
```c
void turn_on_led(int color) {
    P2->OUT &= ~0x07;
    P2->OUT |= color;
}
void turn_off_led() {
    P2->OUT &= ~0x07;
}
```



```c
void main(void)
{
    // Clock initialization
    Clock_Init48MHz();

    // LED initialization
    led_init();

    // green
    turn_on_led(LED_GREEN);

    // wait 1s
    Clock_Delay1ms(1000);

    // green & blue
    turn_on_led(LED_GREEN | LED_BLUE);
}
```

2. Switch

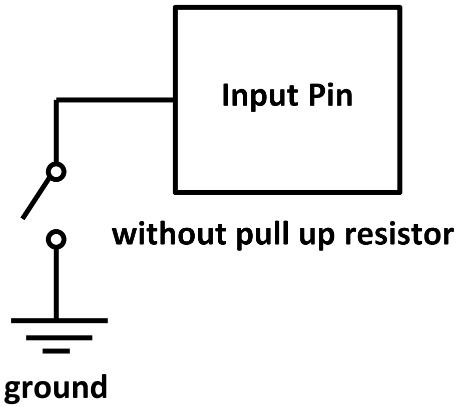
Switch Initialization



```
void switch_init() {
    // Setup switch as GPIO
    P1->SEL0 &= ~0x12;
    P1->SEL1 &= ~0x12;

    // Setup switch as Input
    P1->DIR &= ~0x12;

    // Enable pull-up resistors
    P1->REN |= 0x12;

    // Now pull-up
    P1->OUT |= 0x12;
}
```

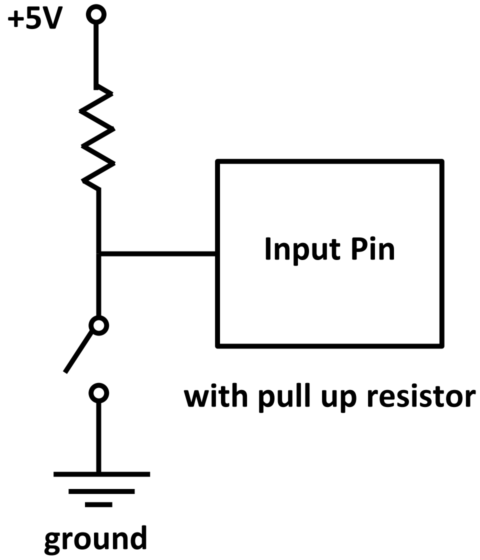
???

Switch Initialization



without pull up resistor

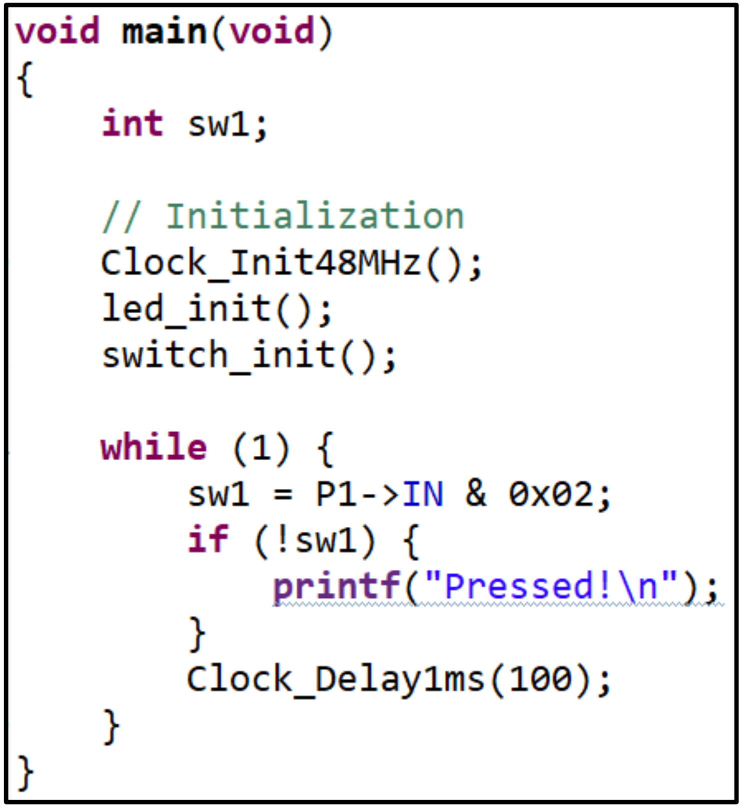
Floating!



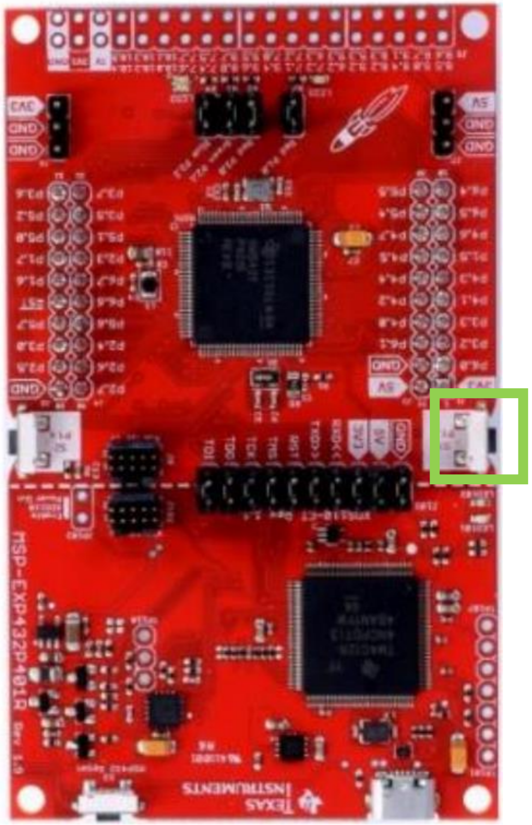
with pull up resistor

+5V (High)

Switch Example



```c
void main(void)
{
    int sw1;

    // Initialization
    Clock_Init48MHz();
    led_init();
    switch_init();

    while (1) {
        sw1 = P1->IN & 0x02;
        if (!sw1) {
            printf("Pressed!\n");
        }
        Clock_Delay1ms(100);
    }
}
```



Assignment

1. Turn on the LED when pressed -> Next week
2. Make the LED R->G->B each time pressed -> 11/30 X 가산점

The first assignment is due next week.(Deadline: Today ~ 10/4 23:59)

In the case of the second assignment, not submitting it has no significant effect,
but if it is submitted, additional points will be awarded.(Deadline: Today ~ 11/30 23:59)

Both assignments can be submitted after discussion with each other.
It doesn't matter if the members have the same code.