Line Tracer 02

- How to write Line Tracer in C -

This lecture is based on

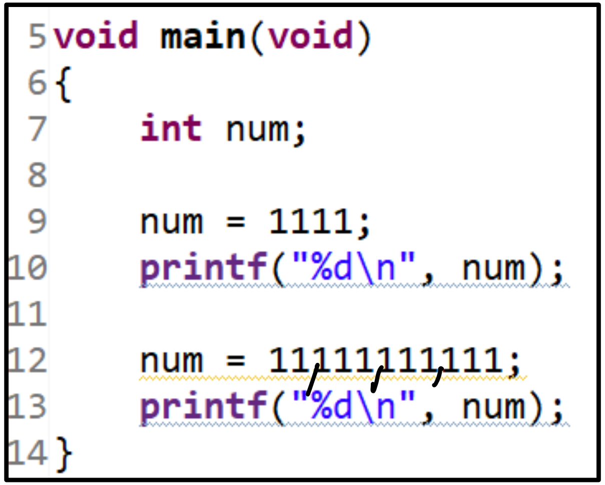
- [Running Code on the TI LaunchPad Board Using CCS]

1. Bitwise Operation

Byte, Bit

int

- integer type
- 4byte, 32bit
- ex) 123, 4833, 2147483647



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

    num = 1111;
    printf("%d\n", num);

    num = 11111111111;
    printf("%d\n", num);
}
```

```
1111

-1773790777
```

Byte, Bit

1 Byte = 8 Bit

1	0	1	1	1	0	1	1

2^8 = 256

Bitwise Operation

00000001 | 00000010

→ 00000011

AND

1 & 1 = 1

1 & 0 = 0

0 & 1 = 0

0 & 0 = 0

OR

1 | 1 = 1

1 | 0 = 1

0 | 1 = 1

0 | 0 = 0

XOR

1 ^ 1 = 0

1 ^ 0 = 1

0 ^ 1 = 1

0 ^ 0 = 0

NOT

~0 = 1

~1 = 0

Bitwise Operation

1	0	1	1	0	1	1	1

AND

0	0	1	0	1	0	0	1

=

0	0	1	0	0	0	0	1

이진수 표현

0b10110111

&

0b00101001

=

0b00100001

Bitwise Operation

?	?	?	?	?	?	?	?

AND

1	0	0	0	0	0	0	0

=

?	0	0	0	0	0	0	0

Bitwise Operation

특정비트를 0으로 세팅하고 싶을때

?	?	?	?	?	?	?	?

AND

1	1	1	0	1	1	1	1

=

?	?	?	0	?	?	?	?

Bitwise Operation

특정 비트를 1로 만들고 싶을때

?	?	?	?	?	?	?	?

OR

0	0	0	0	0	1	0	0

=

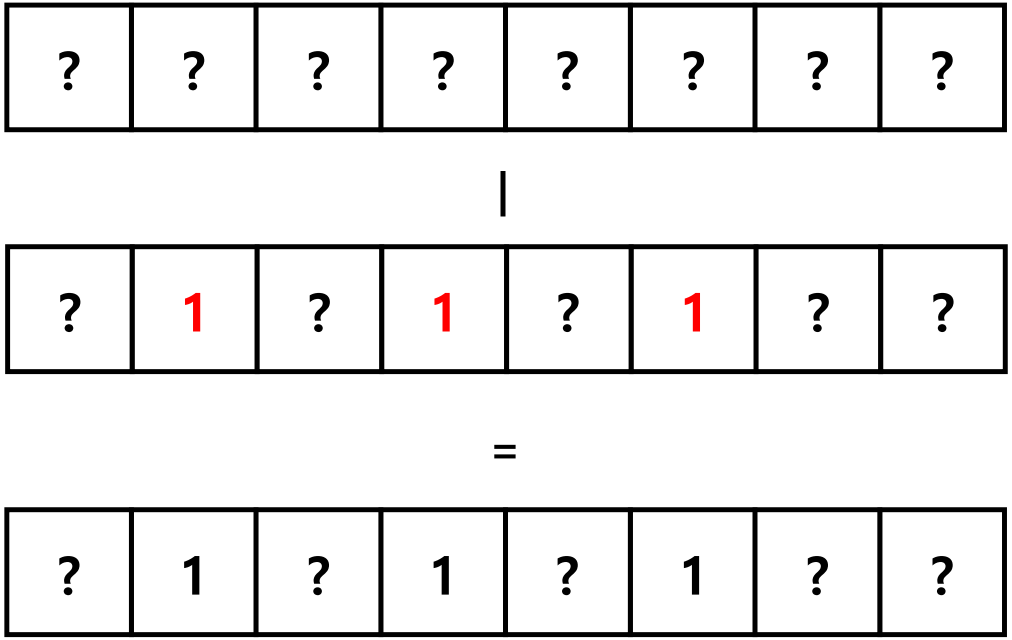
?	?	?	?	?	1	?	?

Register setting

레지스터 크기 : 8bit

?	?	?	?	?	?	?	?

Register setting



?	?	?	?	?	?	?	?

Before Register A(REA)

|

?	1	?	1	?	1	?	?

b01010100
or
0x54

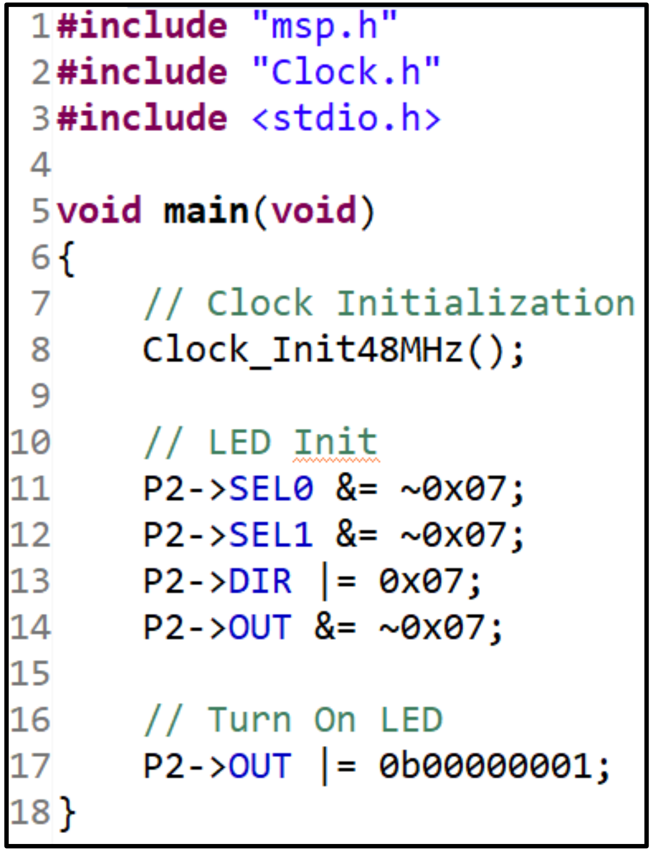
=

?	1	?	1	?	1	?	?

After Register A(REA)

Bitwise Operation Example

레지스터로 LED 조작



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

void main(void)
{
    // Clock Initialization
    Clock_Init48MHz();

    // LED Init
    P2->SEL0 &= ~0x07;
    P2->SEL1 &= ~0x07;
    P2->DIR |= 0x07;
    P2->OUT &= ~0x07;

    // Turn On LED
    P2->OUT |= 0b00000001;
}
```

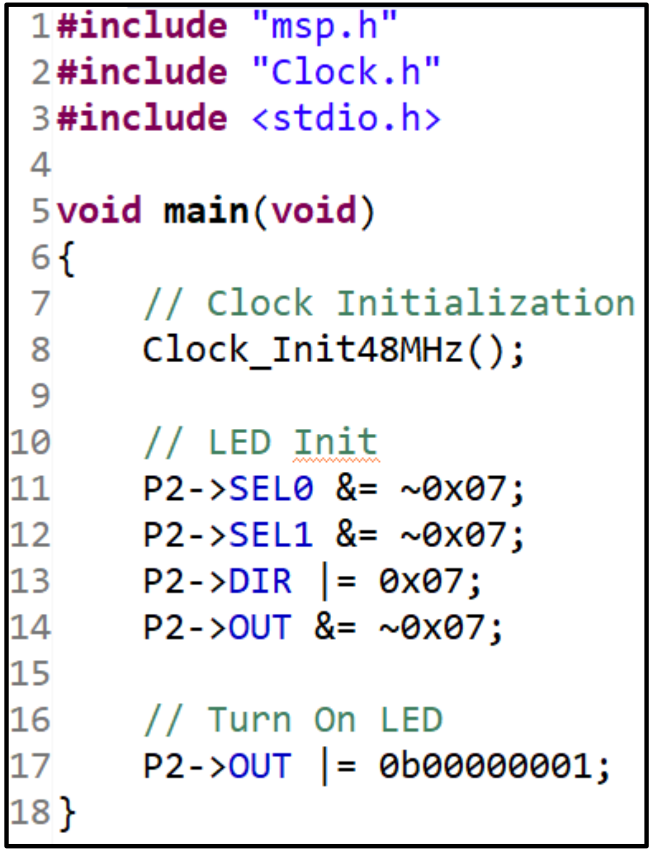
P2->OUT

?	?	?	?	?	?	?	?
							0th bit

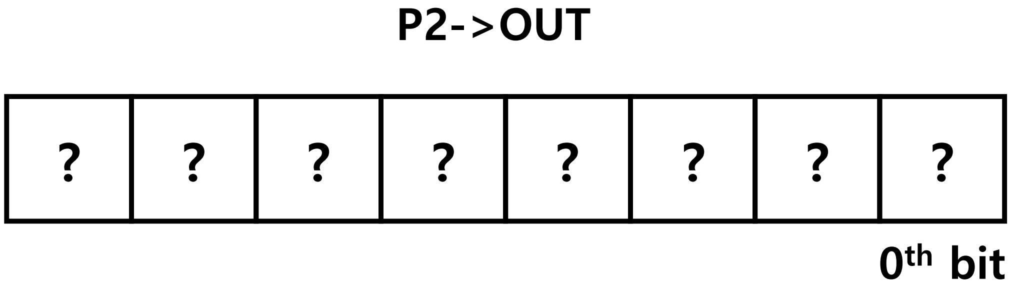
0th bit : 1 -> Turn On LED

0th bit : 0 -> Turn Off LED

Bitwise Operation Example



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

void main(void)
{
    // Clock Initialization
    Clock_Init48MHz();

    // LED Init
    P2->SEL0 &= ~0x07;
    P2->SEL1 &= ~0x07;
    P2->DIR |= 0x07;
    P2->OUT &= ~0x07;

    // Turn On LED
    P2->OUT |= 0b00000001;
}
```



P2->OUT

?	?	?	?	?	?	?	?

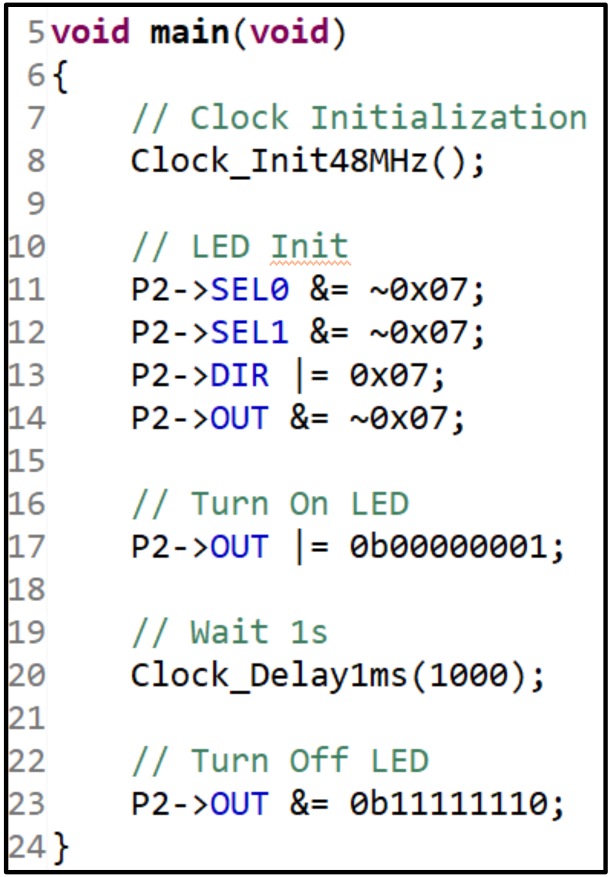
0th bit

0th bit : 1 -> Turn On LED

0th bit : 0 -> Turn Off LED

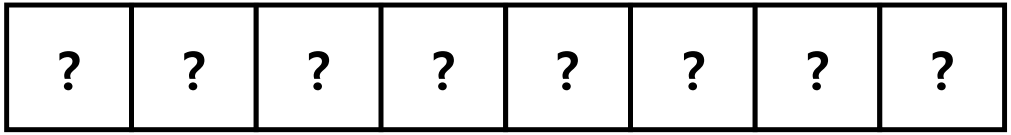
Bitwise Operation Example

1초뒤 LED 끄기



```
void main(void)
{
    // Clock Initialization
    Clock_Init48MHz();

    // LED Init
    P2->SEL0 &= ~0x07;
    P2->SEL1 &= ~0x07;
    P2->DIR |= 0x07;
    P2->OUT &= ~0x07;

    // Turn On LED
    P2->OUT |= 0b00000001;

    // Wait 1s
    Clock_Delay1ms(1000);

    // Turn Off LED
    P2->OUT &= 0b11111110;
}
```

P2->OUT



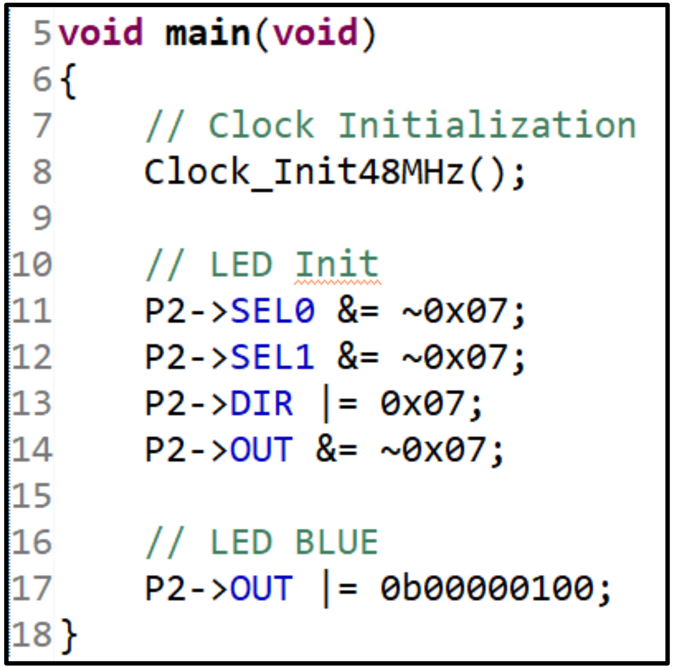
0th bit

0th bit : 1 -> Turn On LED

0th bit : 0 -> Turn Off LED

Bitwise Operation Example

RGB LED



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

    // LED Init
    P2->SEL0 &= ~0x07;
    P2->SEL1 &= ~0x07;
    P2->DIR |= 0x07;
    P2->OUT &= ~0x07;

    // LED BLUE
    P2->OUT |= 0b00000100;
}
```

P2->OUT

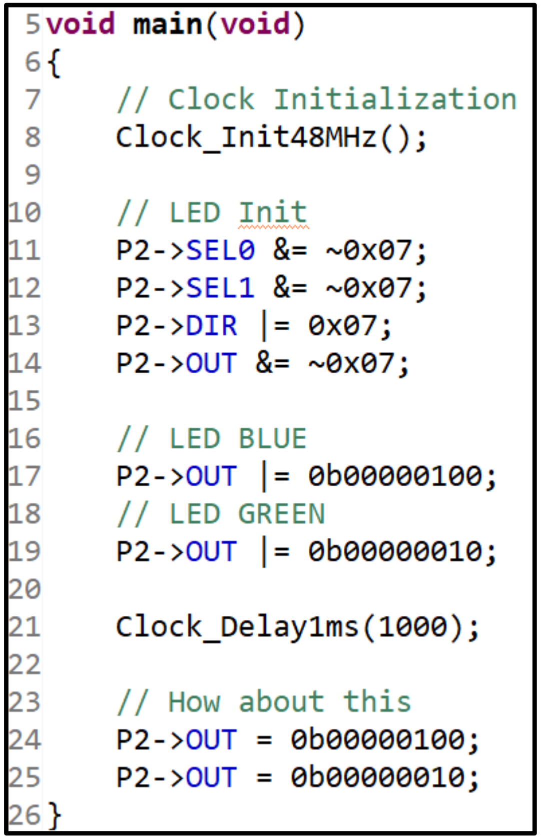
?	?	?	?	?	?	?	?
					2nd	1st	0th

0th bit : RED

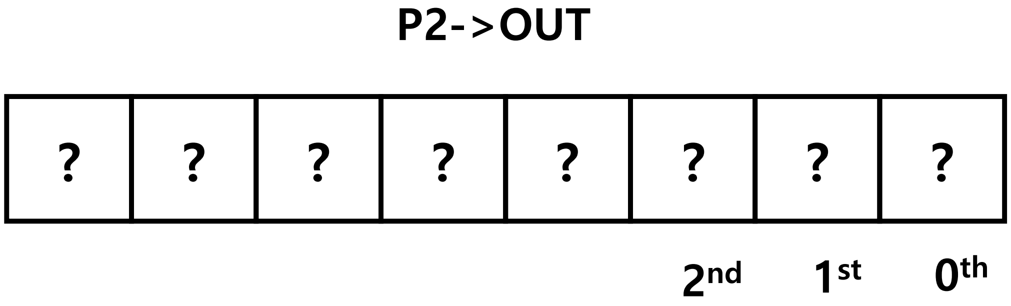
1st bit : GREEN

2nd bit : BLUE

Bitwise Operation Example



```
void main(void)
{
    // Clock Initialization
    Clock_Init48MHz();

    // LED Init
    P2->SEL0 &= ~0x07;
    P2->SEL1 &= ~0x07;
    P2->DIR |= 0x07;
    P2->OUT &= ~0x07;

    // LED BLUE
    P2->OUT |= 0b00000100;
    // LED GREEN
    P2->OUT |= 0b00000010;

    Clock_Delay1ms(1000);

    // How about this
    P2->OUT = 0b00000100;
    P2->OUT = 0b00000010;
}
```

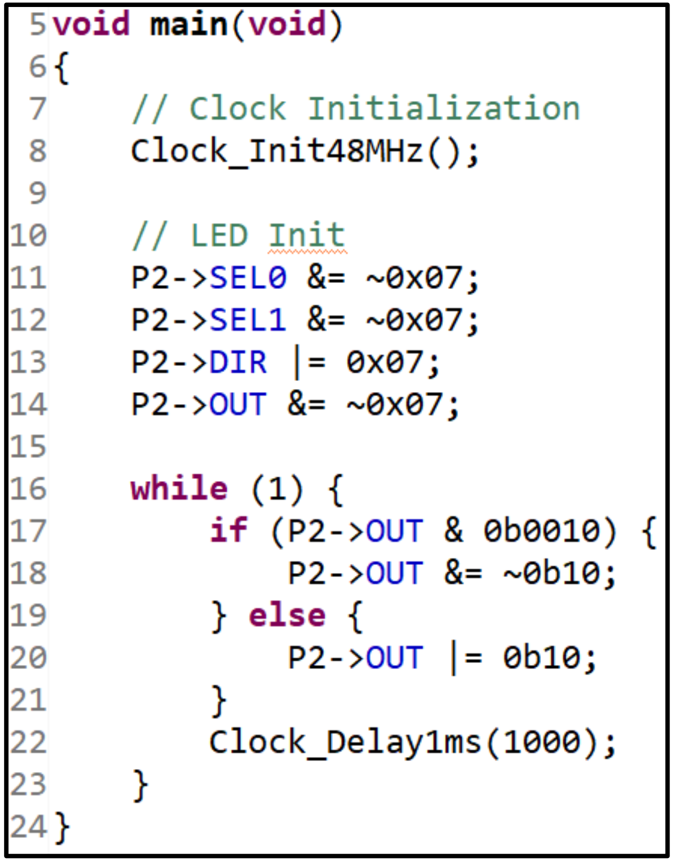


0th bit : RED

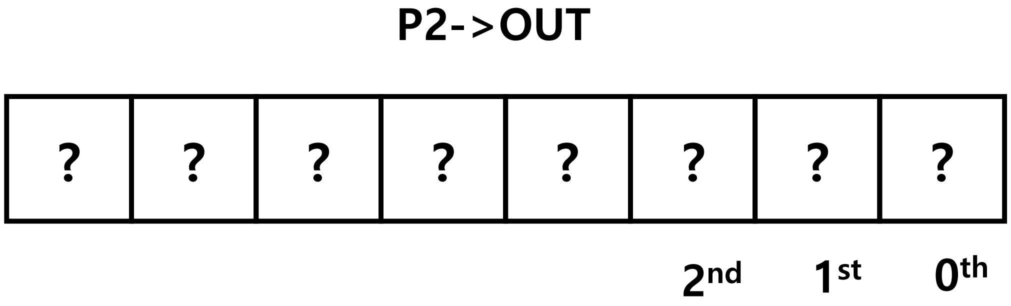
1st bit : GREEN

2nd bit : BLUE

Bitwise Operation Example



```
void main(void)
{
    // Clock Initialization
    Clock_Init48MHz();

    // LED Init
    P2->SEL0 &= ~0x07;
    P2->SEL1 &= ~0x07;
    P2->DIR |= 0x07;
    P2->OUT &= ~0x07;

    while (1) {
        if (P2->OUT & 0b0010) {
            P2->OUT &= ~0b10;
        } else {
            P2->OUT |= 0b10;
        }
        Clock_Delay1ms(1000);
    }
}
```



0th bit : RED

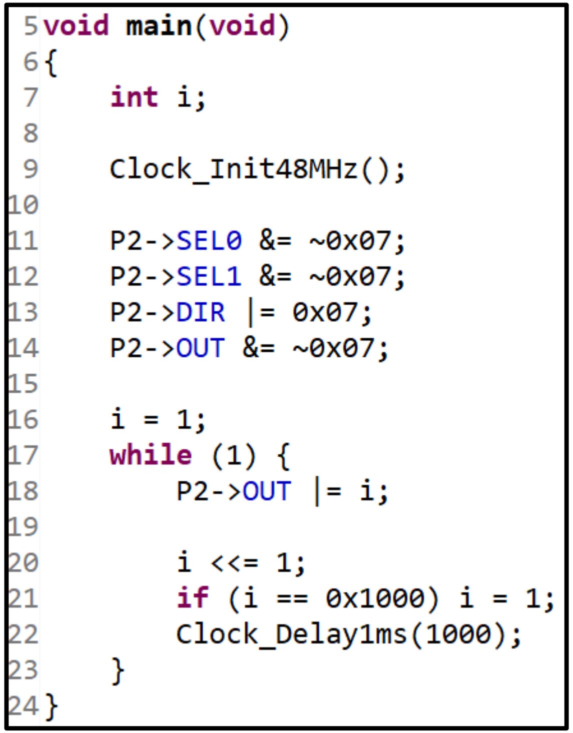
1st bit : GREEN

2nd bit : BLUE

2. Debugging

How to Debug

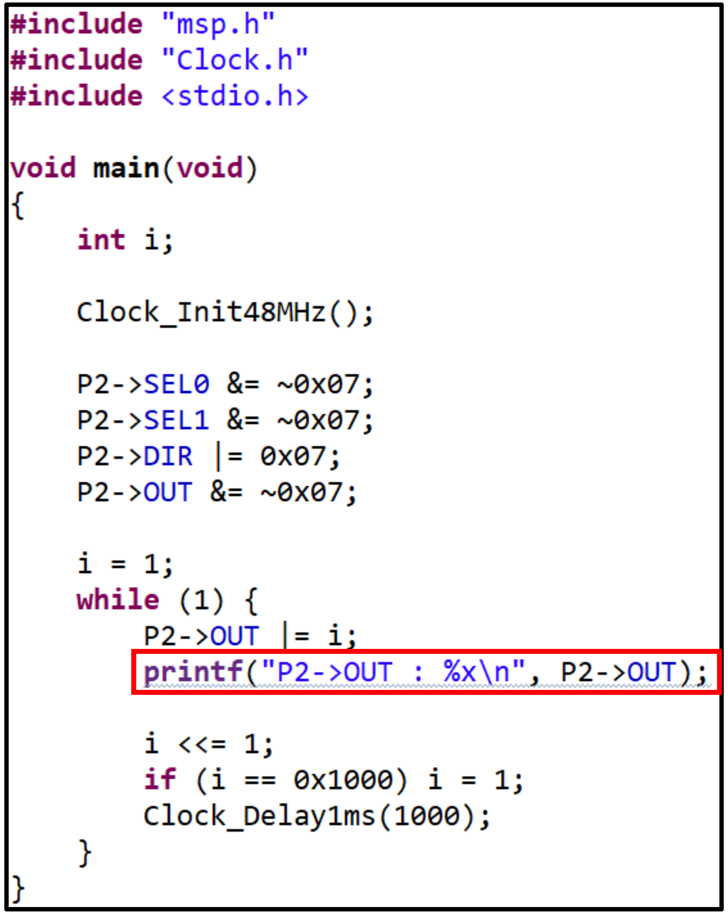
To make LED Flashing



```
void main(void)
{
    int i;

    Clock_Init48MHz();

    P2->SEL0 &= ~0x07;
    P2->SEL1 &= ~0x07;
    P2->DIR |= 0x07;
    P2->OUT &= ~0x07;

    i = 1;
    while (1) {
        P2->OUT |= i;

        i <<= 1;
        if (i == 0x1000) i = 1;
        Clock_Delay1ms(1000);
    }
}
```

Red -> (1s) -> Green -> (1s) -> Blue -> (1s) -> Red -> ...

Logging



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

void main(void)
{
    int i;

    Clock_Init48MHz();

    P2->SEL0 &= ~0x07;
    P2->SEL1 &= ~0x07;
    P2->DIR |= 0x07;
    P2->OUT &= ~0x07;

    i = 1;
    while (1) {
        P2->OUT |= i;
        printf("P2->OUT : %x\n", P2->OUT);

        i <<= 1;
        if (i == 0x1000) i = 1;
        Clock_Delay1ms(1000);
    }
}
```

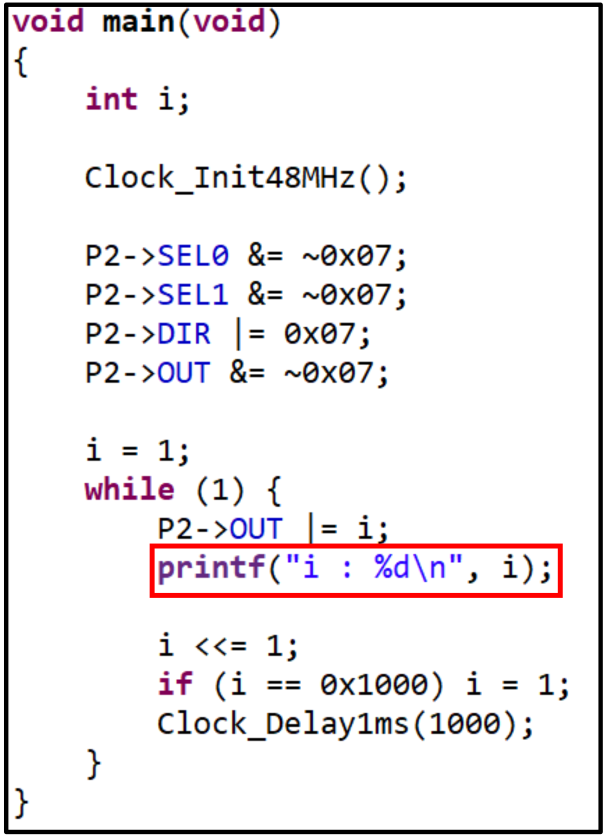
Expected Result

```
P2->OUT : f9
P2->OUT : fa
P2->OUT : fc
P2->OUT : f9
P2->OUT : fa
…
```

Actual Result

```
P2->OUT : f9
P2->OUT : fb
P2->OUT : ff
P2->OUT : ff
P2->OUT : ff
…
```

Logging



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

    Clock_Init48MHz();

    P2->SEL0 &= ~0x07;
    P2->SEL1 &= ~0x07;
    P2->DIR |= 0x07;
    P2->OUT &= ~0x07;

    i = 1;
    while (1) {
        P2->OUT |= i;
        printf("i : %d\n", i);

        i <<= 1;
        if (i == 0x1000) i = 1;
        Clock_Delay1ms(1000);
    }
}
```

Expected Result

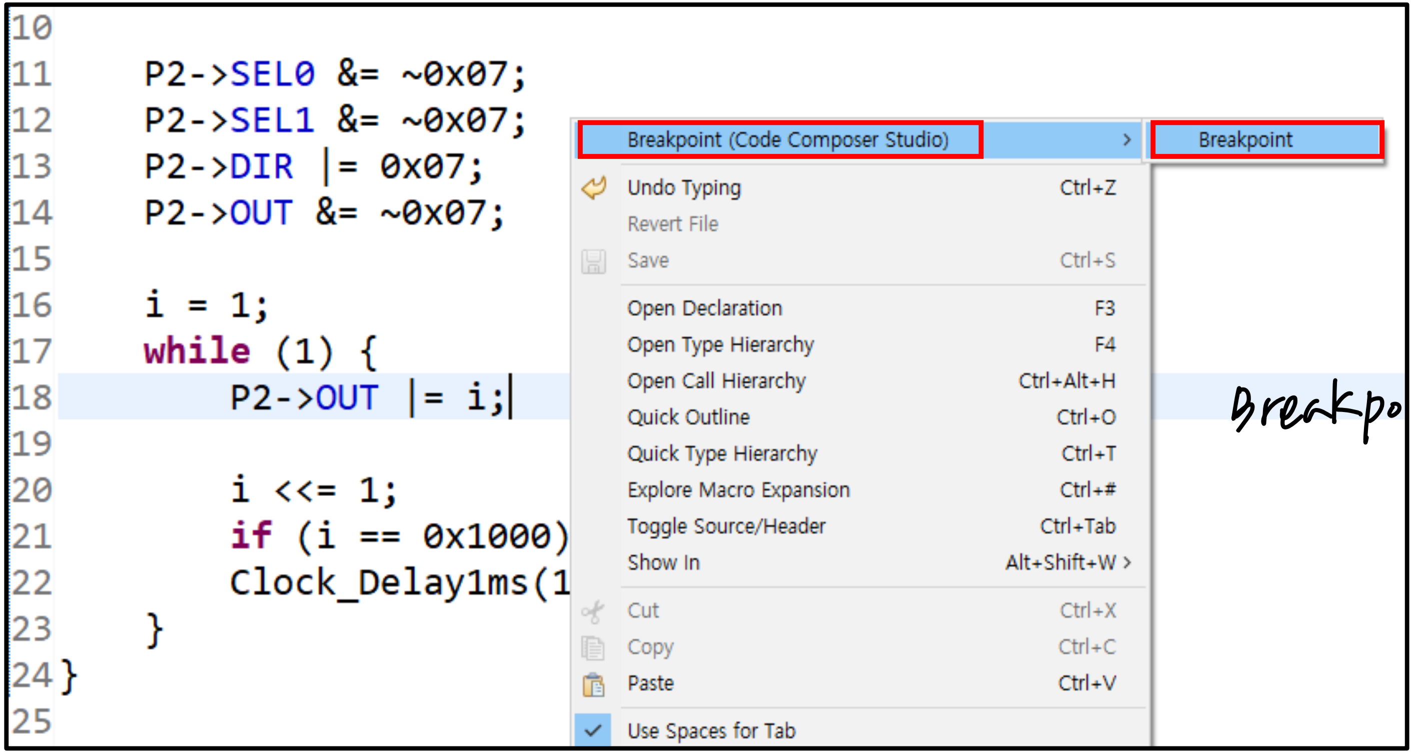
```
i : 1
i : 2
i : 4
i : 1
i : 2
...
```

Actual Result

```
i : 1
i : 2
i : 4
i : 8
i : 16
...
```

i 값 이상

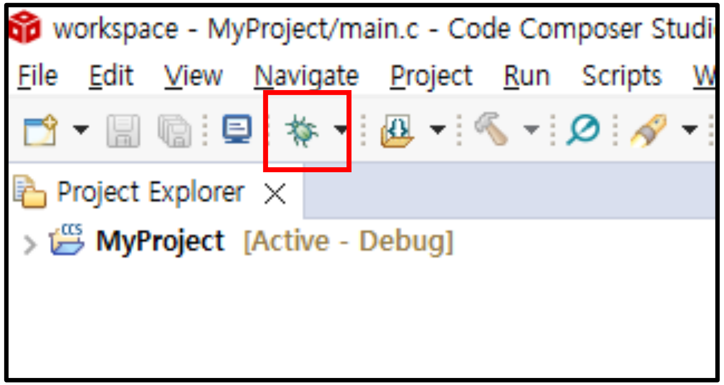
Debugger

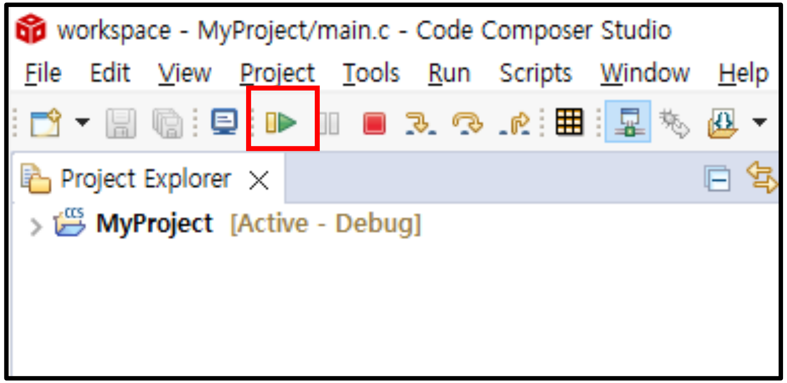


```
    P2->SEL0 &= ~0x07;
    P2->SEL1 &= ~0x07;
    P2->DIR |= 0x07;
    P2->OUT &= ~0x07;

    i = 1;
    while (1) {
        P2->OUT |= i;

        i <<= 1;
        if (i == 0x1000)
        Clock_Delay1ms(1
    }
}
```

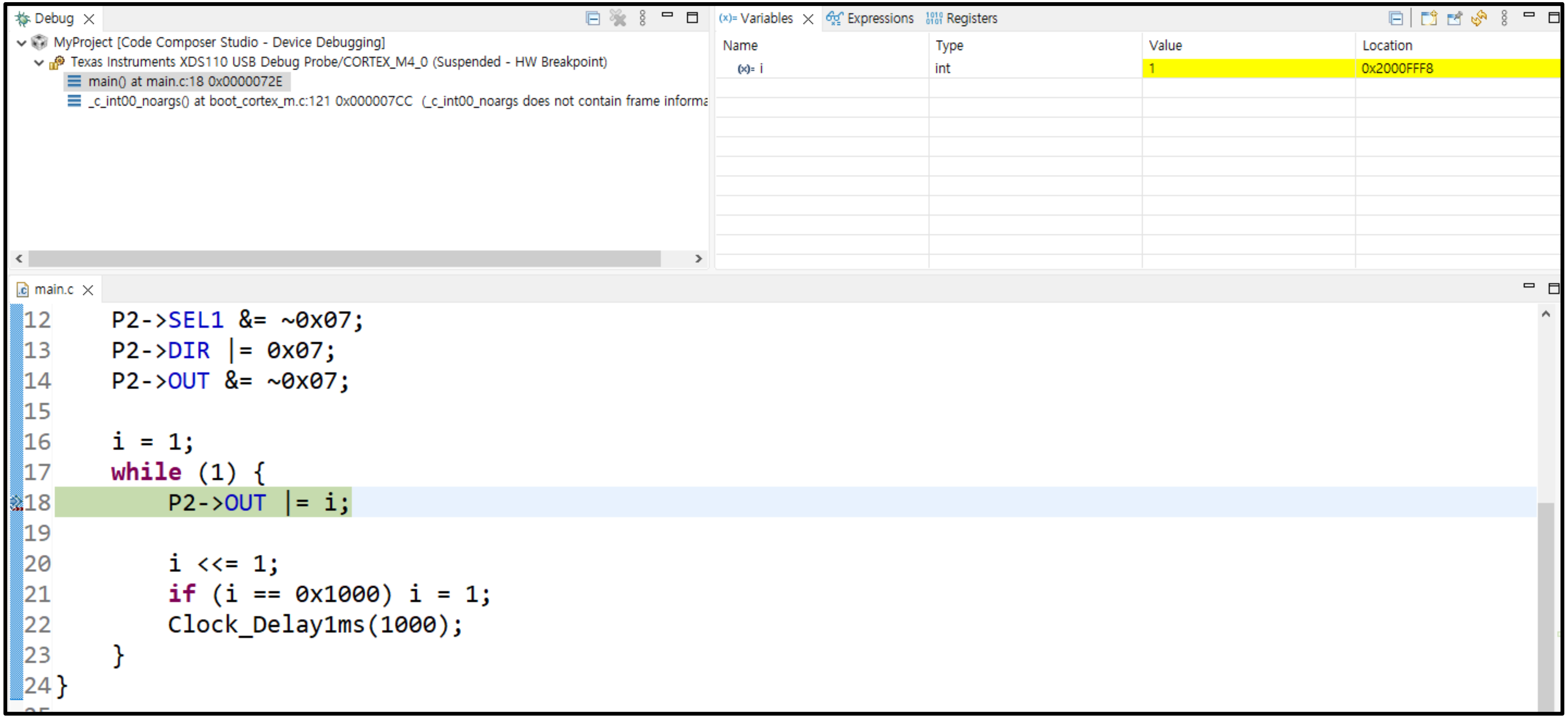
Breakpoint 설정

Debugger


workspace - MyProject/main.c - Code Composer Studi
File Edit View Navigate Project Run Scripts W
Project Explorer
MyProject [Active - Debug]


workspace - MyProject/main.c - Code Composer Studio
File Edit View Project Tools Run Scripts Window Help
Project Explorer
MyProject [Active - Debug]

Debugger

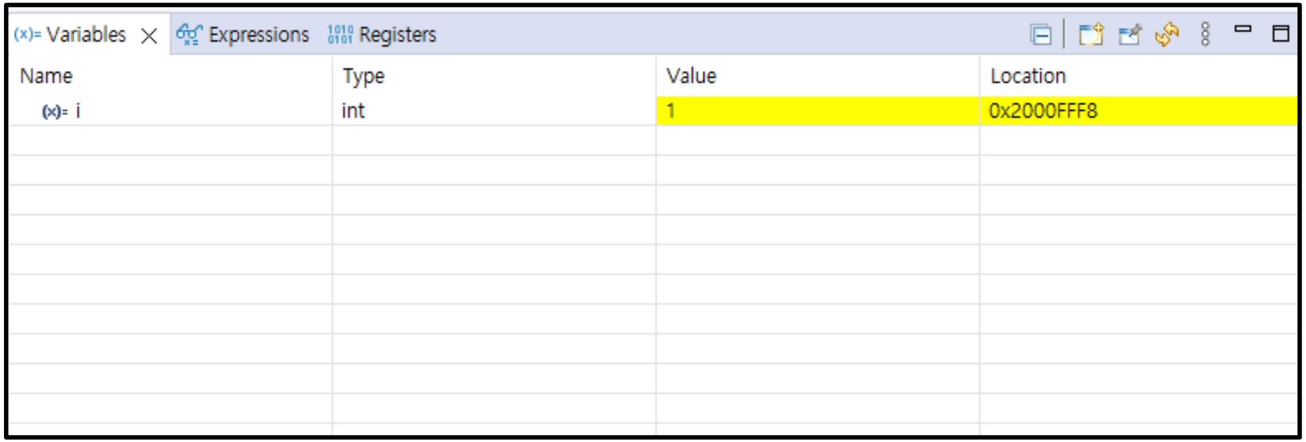


```
MyProject [Code Composer Studio - Device Debugging]
  Texas Instruments XDS110 USB Debug Probe/CORTEX_M4_0 (Suspended - HW Breakpoint)
    main() at main.c:18 0x0000072E
    _c_int00_noargs() at boot_cortex_m.c:121 0x000007CC  (_c_int00_noargs does not contain frame informa
```

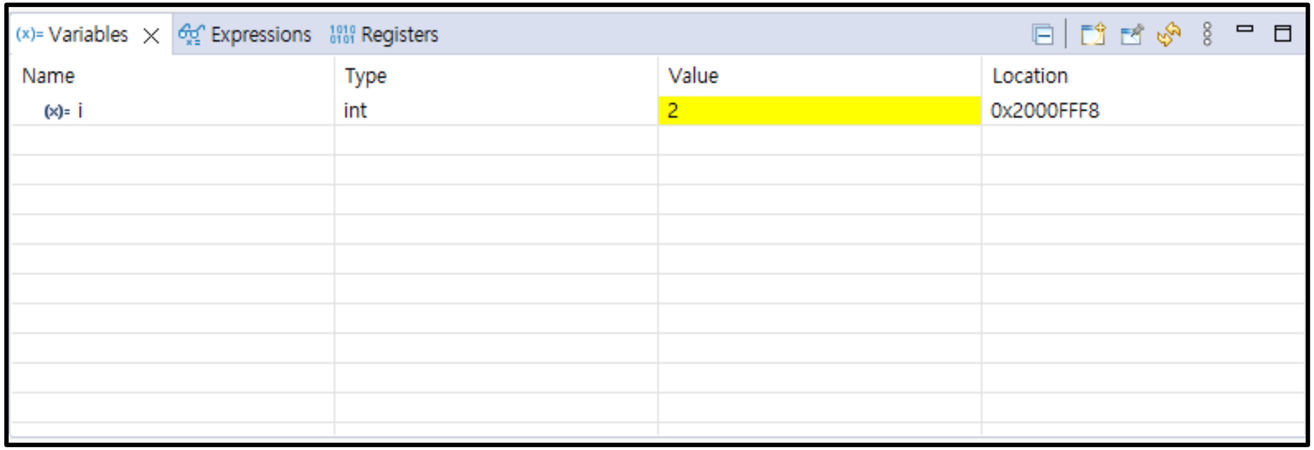
Name	Type	Value	Location
(x)= i	int	1	0x2000FFF8

```
12     P2->SEL1 &= ~0x07;
13     P2->DIR |= 0x07;
14     P2->OUT &= ~0x07;
15
16     i = 1;
17     while (1) {
18         P2->OUT |= i;
19
20         i <<= 1;
21         if (i == 0x1000) i = 1;
22         Clock_Delay1ms(1000);
23     }
24 }
```

Debugger



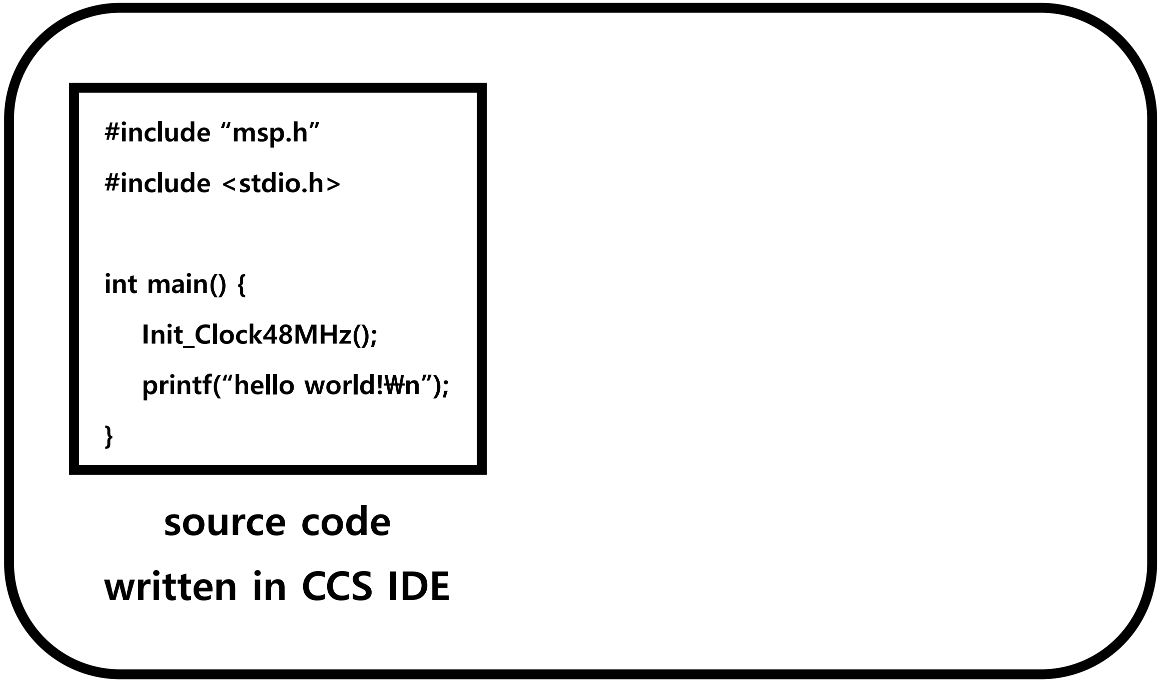
Name	Type	Value	Location
(x)= i	int	1	0x2000FFF8



Name	Type	Value	Location
(x)= i	int	2	0x2000FFF8

3. How Robot Works?

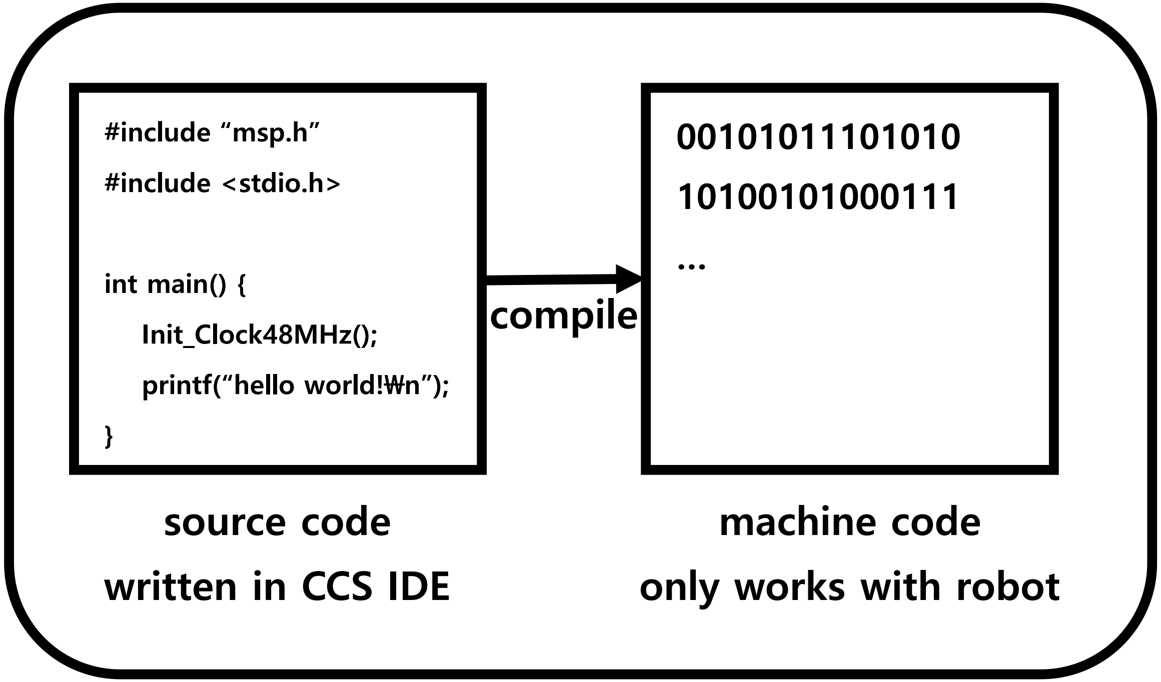
How to Upload Source Code



```
#include “msp.h”
#include <stdio.h>

int main() {
    Init_Clock48MHz();
    printf(“hello world!₩n”);
}
```

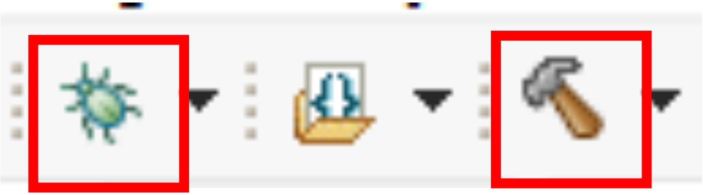
Our PC

How to Upload Source Code



```
#include “msp.h”
#include <stdio.h>

int main() {
    Init_Clock48MHz();
    printf(“hello world!₩n”);
}
```

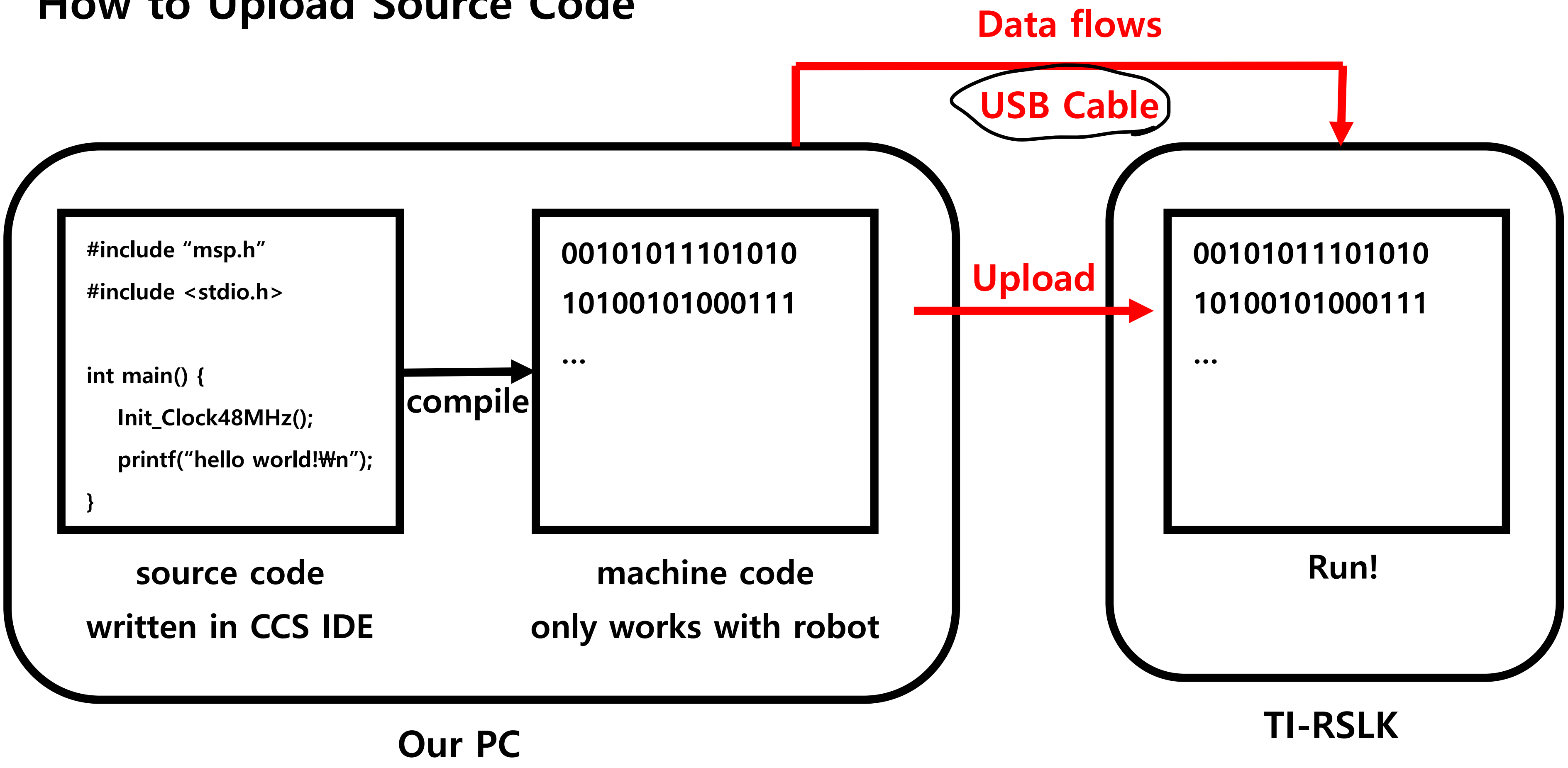
Our PC



build & upload

build

How to Upload Source Code



Data flows

USB Cable

```
#include “msp.h”
#include <stdio.h>

int main() {
    Init_Clock48MHz();
    printf(“hello world!₩n”);
}
```

compile

```
00101011101010
10100101000111
...
```

source code

written in CCS IDE

machine code

only works with robot

Upload

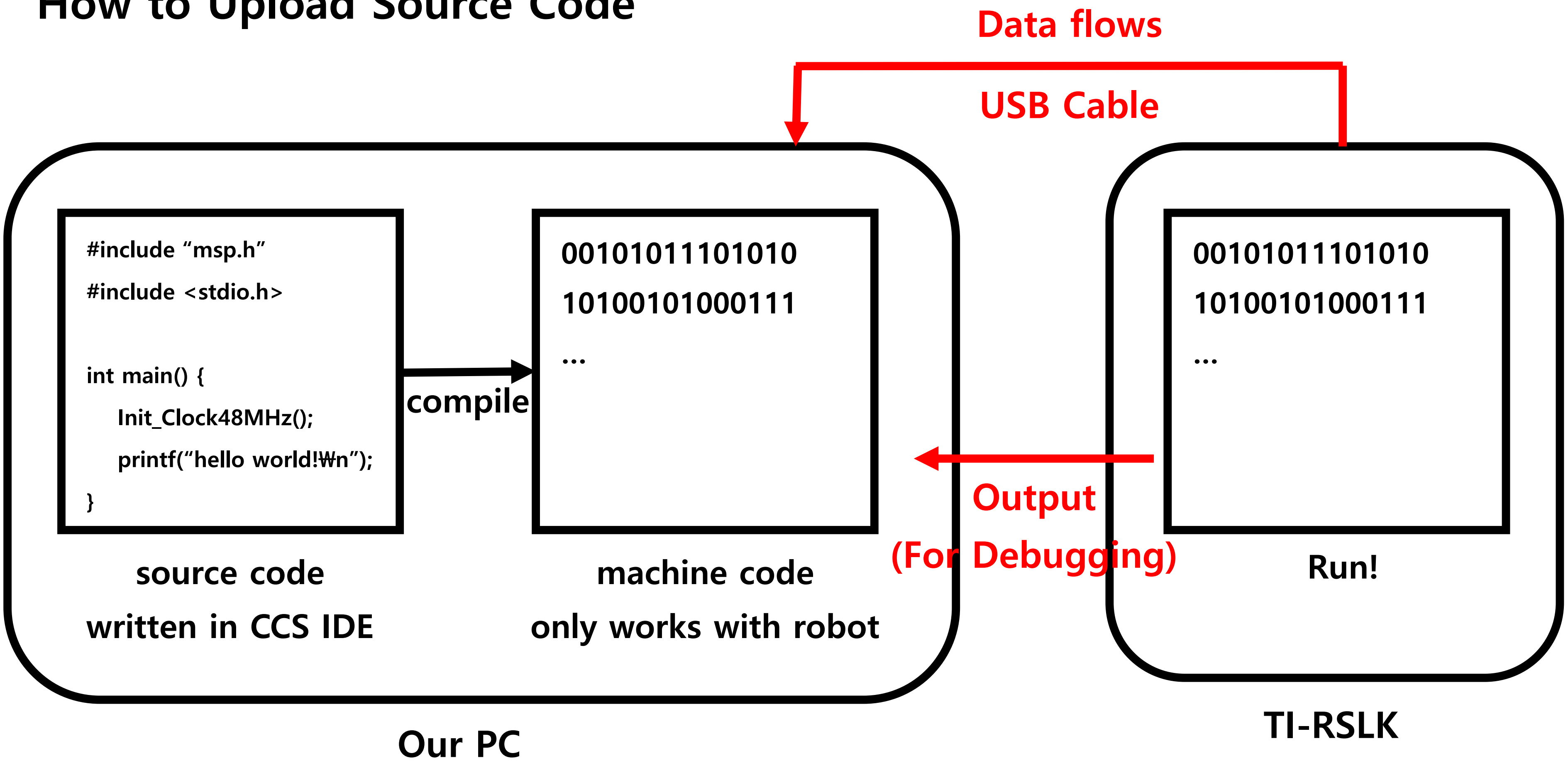
```
00101011101010
10100101000111
...
```

Run!

Our PC

TI-RSLK

How to Upload Source Code



```
#include “msp.h”
#include <stdio.h>

int main() {
    Init_Clock48MHz();
    printf(“hello world!₩n”);
}
```

```
00101011101010
10100101000111
…
```

```
00101011101010
10100101000111
…
```

4. Assignment

Bitwise Operation Example

Make the LED

Red -> (1s) -> Green -> (1s) -> Blue -> (1s) -> Red -> …

Try changing the LED color every 1 second in the order of

Deadline: ~ 9/26 11:59 PM

Submit : LMS

The assignment will be scored individually. (But You can discuss or talk in groups)

You can submit it in file *.C when you submit it.