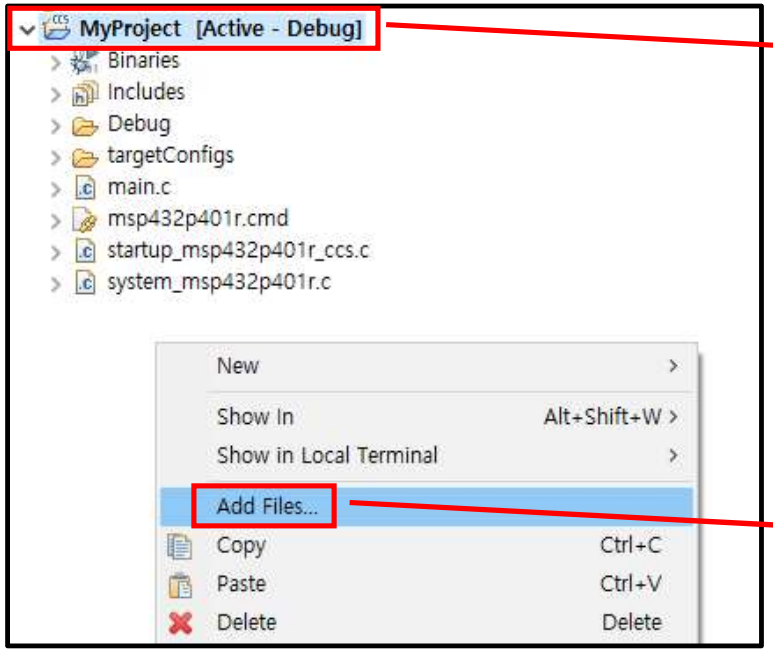
Line Tracer 02

- How to write Line Tracer in C -

Import Clock Library

-> Add Clock Library Files



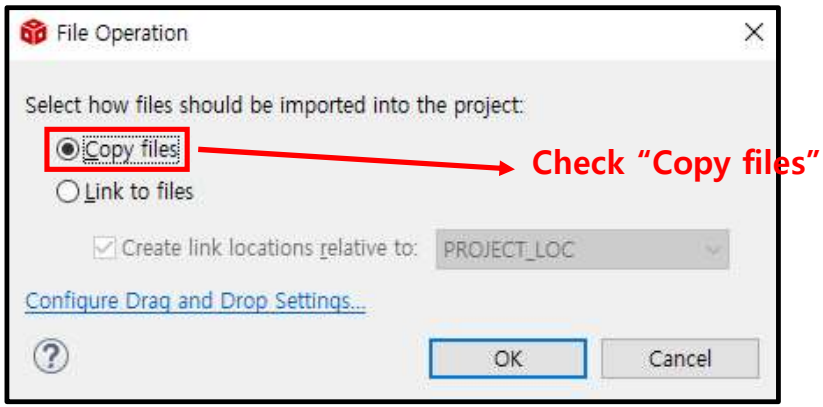
Click "MyProject" to activate

Select "Add Files..."

Import Clock Library

-> Select “Clock.c” and “Clock.h”

-> Copy them to the project folder



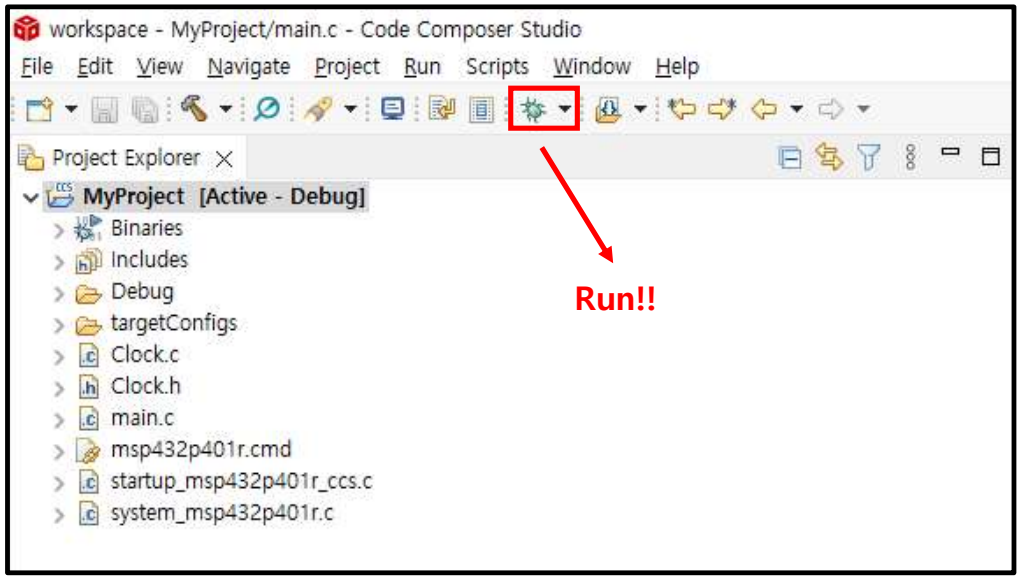
hello world

-> Write a simple program which is printing “hello world!”

-> Run the program clicking the debug button

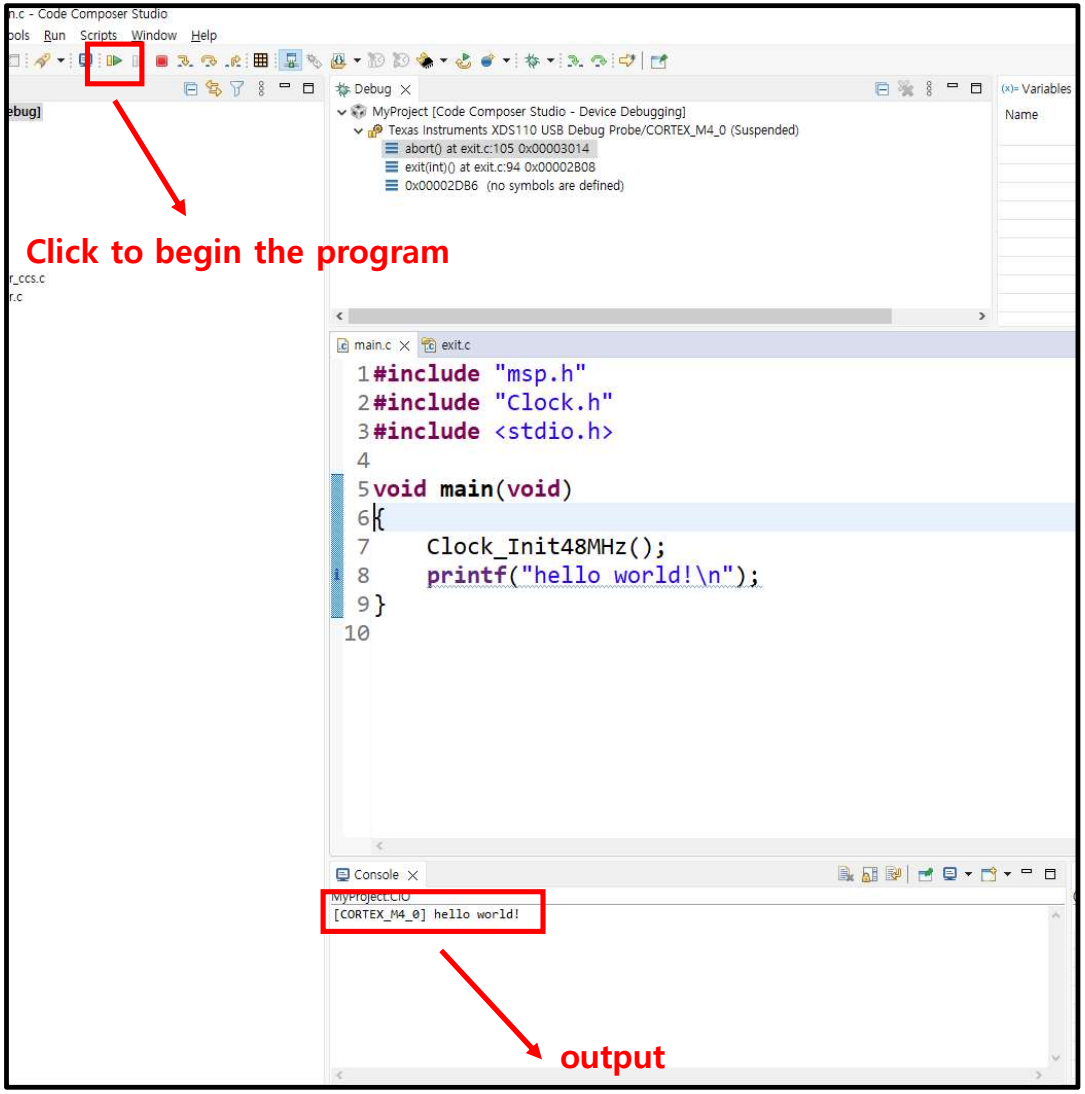
```
1#include "msp.h"
2#include "Clock.h"
3#include <stdio.h>
4
5void main(void)
6{
7    Clock_Init48MHz();
8    printf("hello world!\n");
9}
```

Enter the function name correctly.



hello world

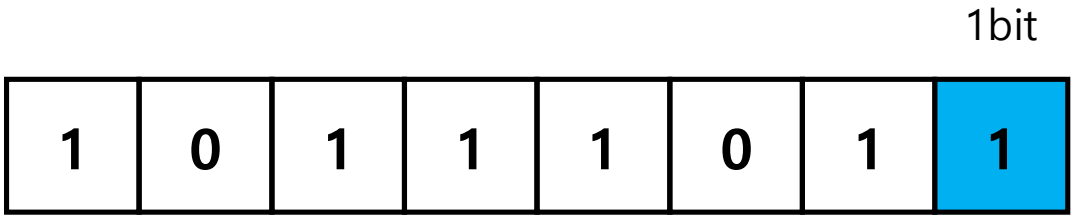
-> You can see the output



1. Bitwise Operation

Byte, Bit

1 Byte = 8 Bit



2^8 = 256

Bitwise Operation

AND

1 & 1 = 1

1 & 0 = 0

0 & 1 = 0

0 & 0 = 0

XOR

1 ^ 1 = 0

1 ^ 0 = 1

0 ^ 1 = 1

0 ^ 0 = 0

OR

1 | 1 = 1

1 | 0 = 1

0 | 1 = 1

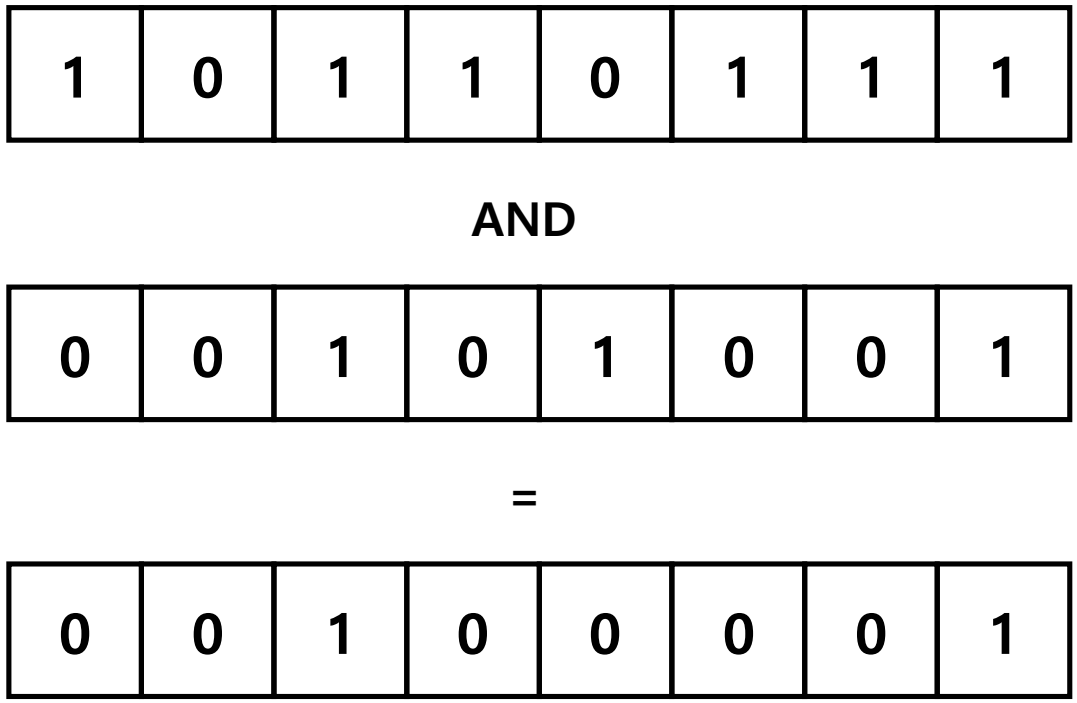
0 | 0 = 0

NOT

~0 = 1

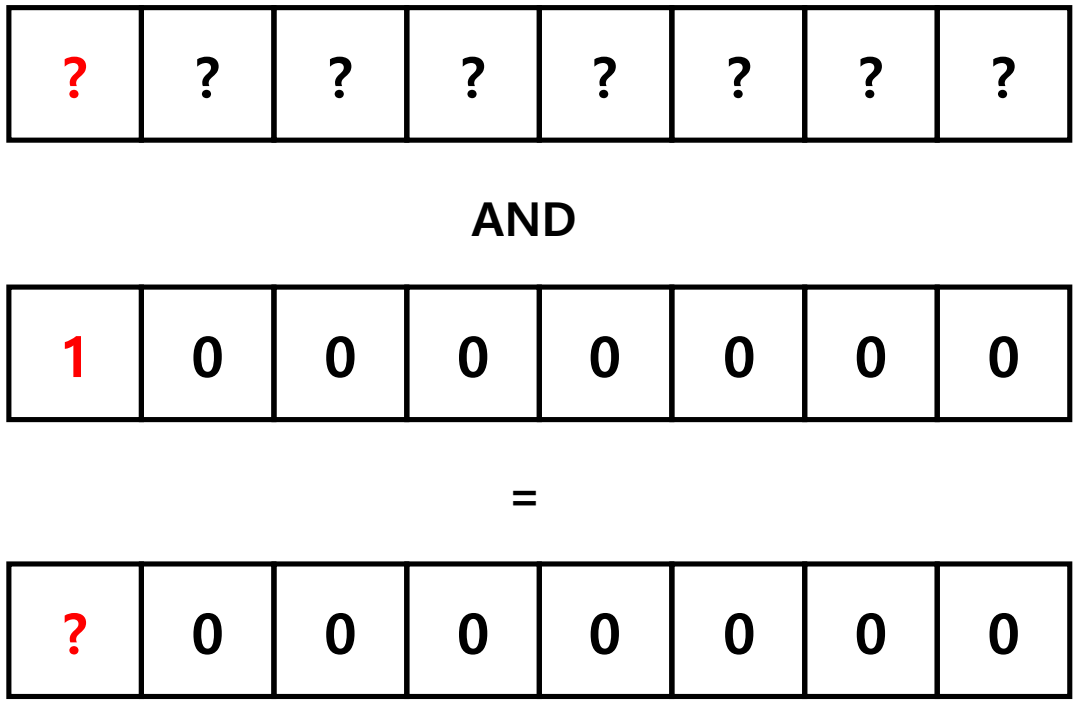
~1 = 0

Bitwise Operation



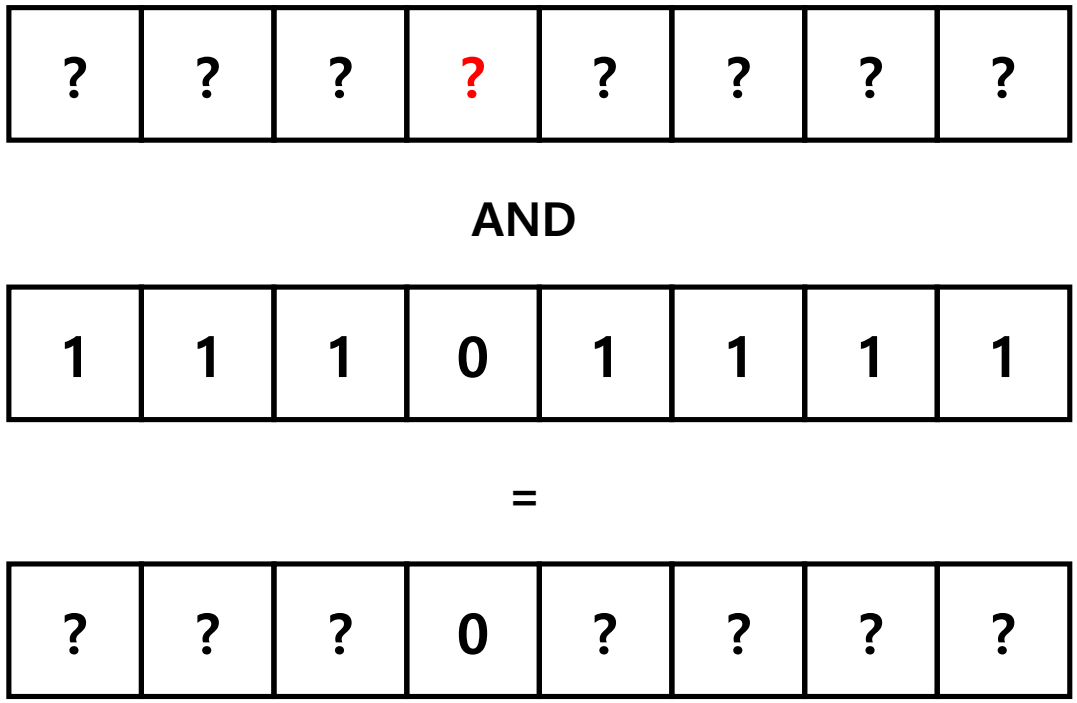
0b10110111
&
0b00101001
=
0b00100001

Bitwise Operation



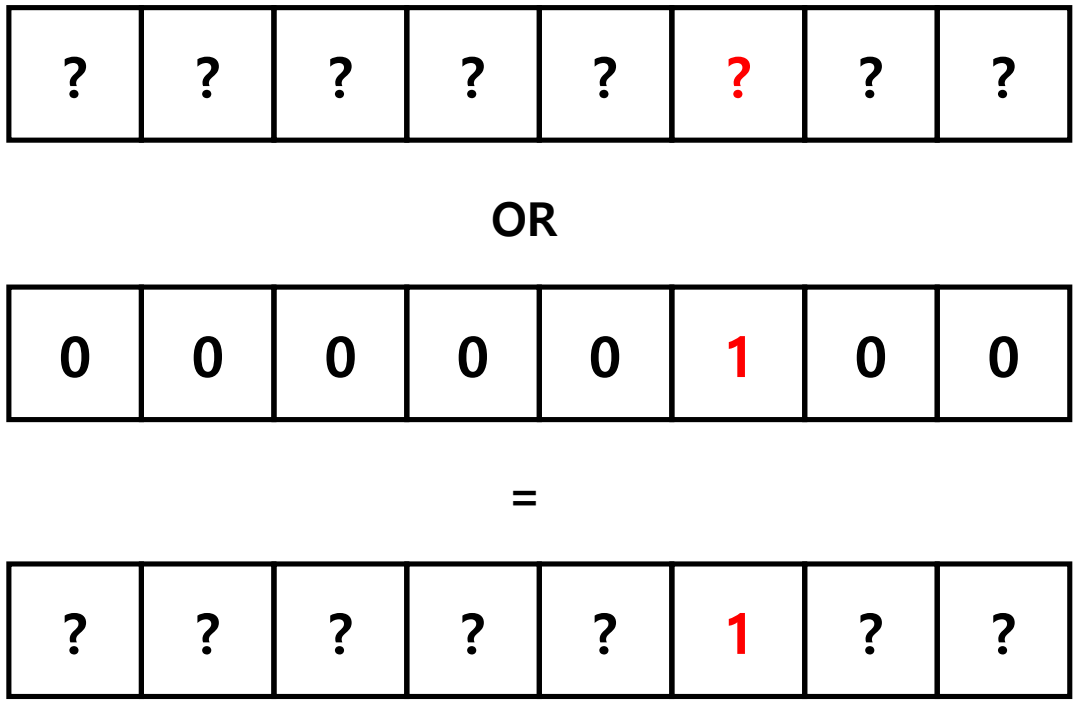
Extract specific bits

Bitwise Operation



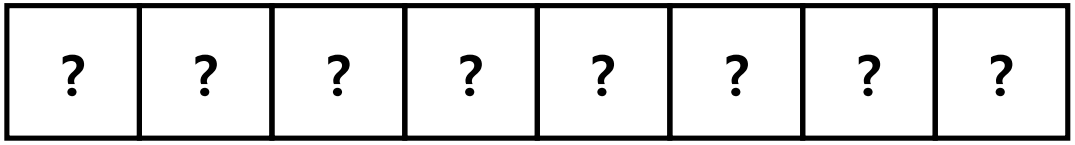
Setting a specific bit to 0

Bitwise Operation



Setting a specific bit to 1

Register setting

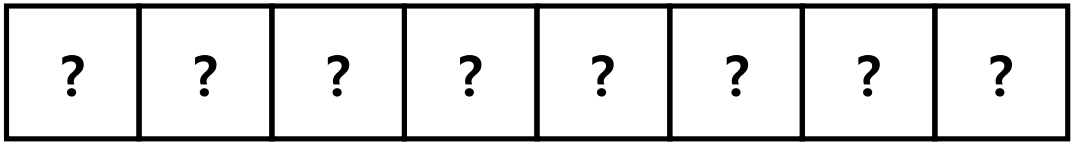


Bitwise Operation Example

```
1#include "msp.h"
2#include "Clock.h"
3#include <stdio.h>
4
5void main(void)
6{
7    // Clock Initialization
8    Clock_Init48MHz();
9
10   // LED Init
11   P2->SEL0 &= ~0x07;
12   P2->SEL1 &= ~0x07;
13   P2->DIR |= 0x07;
14   P2->OUT &= ~0x07;
15
16   // Turn On LED
17   P2->OUT |= 0b00000001;
18}
```

The LEDs on the board consist of P2.0, P2.1, and P2.2.

P2->OUT



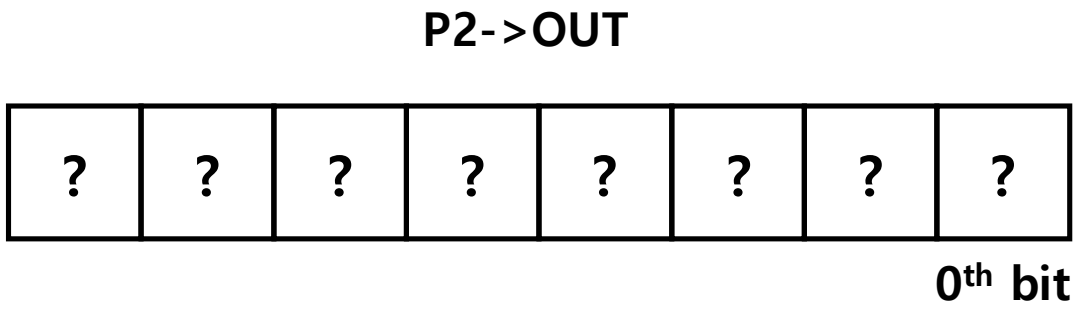
0^{th} bit

0^{th} bit : 1 -> Turn On LED

0^{th} bit : 0 -> Turn Off LED

Bitwise Operation Example

```
1#include "msp.h"
2#include "Clock.h"
3#include <stdio.h>
4
5void main(void)
6{
7    // Clock Initialization
8    Clock_Init48MHz();
9
10   // LED Init
11   P2->SEL0 &= ~0x07;
12   P2->SEL1 &= ~0x07;
13   P2->DIR |= 0x07;
14   P2->OUT &= ~0x07;
15
16   // Turn On LED
17   P2->OUT |= 0b00000001;
18}
```

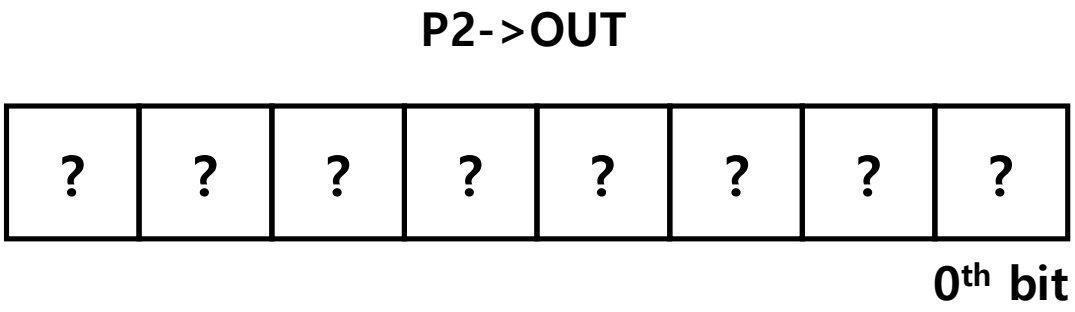


0^{th} bit : 1 -> Turn On LED

0^{th} bit : 0 -> Turn Off LED

Bitwise Operation Example

```
5 void main(void)
6 {
7     // Clock Initialization
8     Clock_Init48MHz();
9
10    // LED Init
11    P2->SEL0 &= ~0x07;
12    P2->SEL1 &= ~0x07;
13    P2->DIR |= 0x07;
14    P2->OUT &= ~0x07;
15
16    // Turn On LED
17    P2->OUT |= 0b00000001;
18
19    // Wait 1s
20    Clock_Delay1ms(1000);
21
22    // Turn Off LED
23    P2->OUT &= 0b11111110;
24 }
```



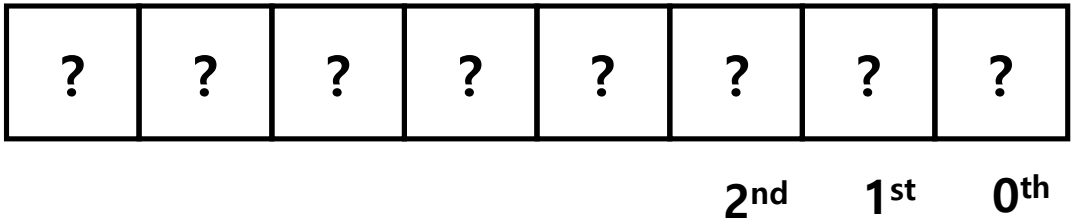
0th bit : 1 -> Turn On LED

0th bit : 0 -> Turn Off LED

Bitwise Operation Example

```
5  void main(void)
6  {
7      // Clock Initialization
8      Clock_Init48MHz();
9
10     // LED Init
11     P2->SEL0 &= ~0x07;
12     P2->SEL1 &= ~0x07;
13     P2->DIR |= 0x07;
14     P2->OUT &= ~0x07;
15
16     // LED BLUE
17     P2->OUT |= 0b00000100;
18 }
```

P2->OUT



0th bit : RED

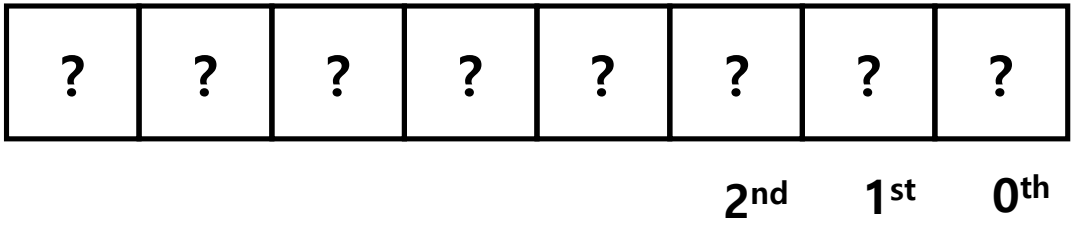
1st bit : GREEN

2nd bit : BLUE

Bitwise Operation Example

```
5  void main(void)
6  {
7      // Clock Initialization
8      Clock_Init48MHz();
9
10     // LED Init
11     P2->SEL0 &= ~0x07;
12     P2->SEL1 &= ~0x07;
13     P2->DIR |= 0x07;
14     P2->OUT &= ~0x07;
15
16     // LED BLUE
17     P2->OUT |= 0b00000100;
18     // LED GREEN
19     P2->OUT |= 0b00000010;
20
21     Clock_Delay1ms(1000);
22
23     // How about this
24     P2->OUT = 0b00000100;
25     P2->OUT = 0b00000010;
26 }
```

P2->OUT



0th bit : RED

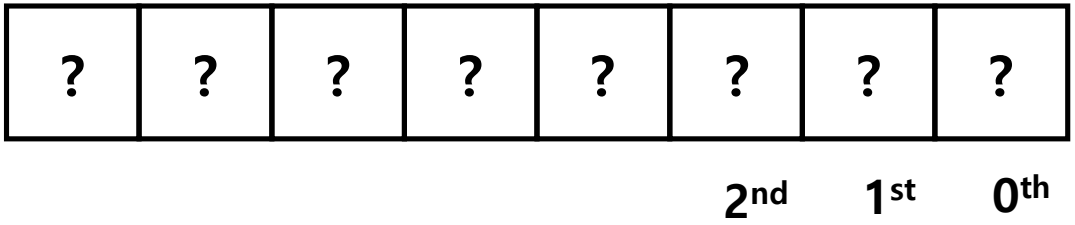
1st bit : GREEN

2nd bit : BLUE

Bitwise Operation Example

```
5  void main(void)
6  {
7      // Clock Initialization
8      Clock_Init48MHz();
9
10     // LED Init
11     P2->SEL0 &= ~0x07;
12     P2->SEL1 &= ~0x07;
13     P2->DIR |= 0x07;
14     P2->OUT &= ~0x07;
15
16     while (1) {
17         if (P2->OUT & 0b0010) {
18             P2->OUT &= ~0b10;
19         } else {
20             P2->OUT |= 0b10;
21         }
22         Clock_Delay1ms(1000);
23     }
24 }
```

P2->OUT



0th bit : RED

1st bit : GREEN

2nd bit : BLUE

2. Debugging

How to Debug

To make LED Flashing

```
5 void main(void)
6 {
7     int i;
8
9     Clock_Init48MHz();
10
11    P2->SEL0 &= ~0x07;
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 }
```

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

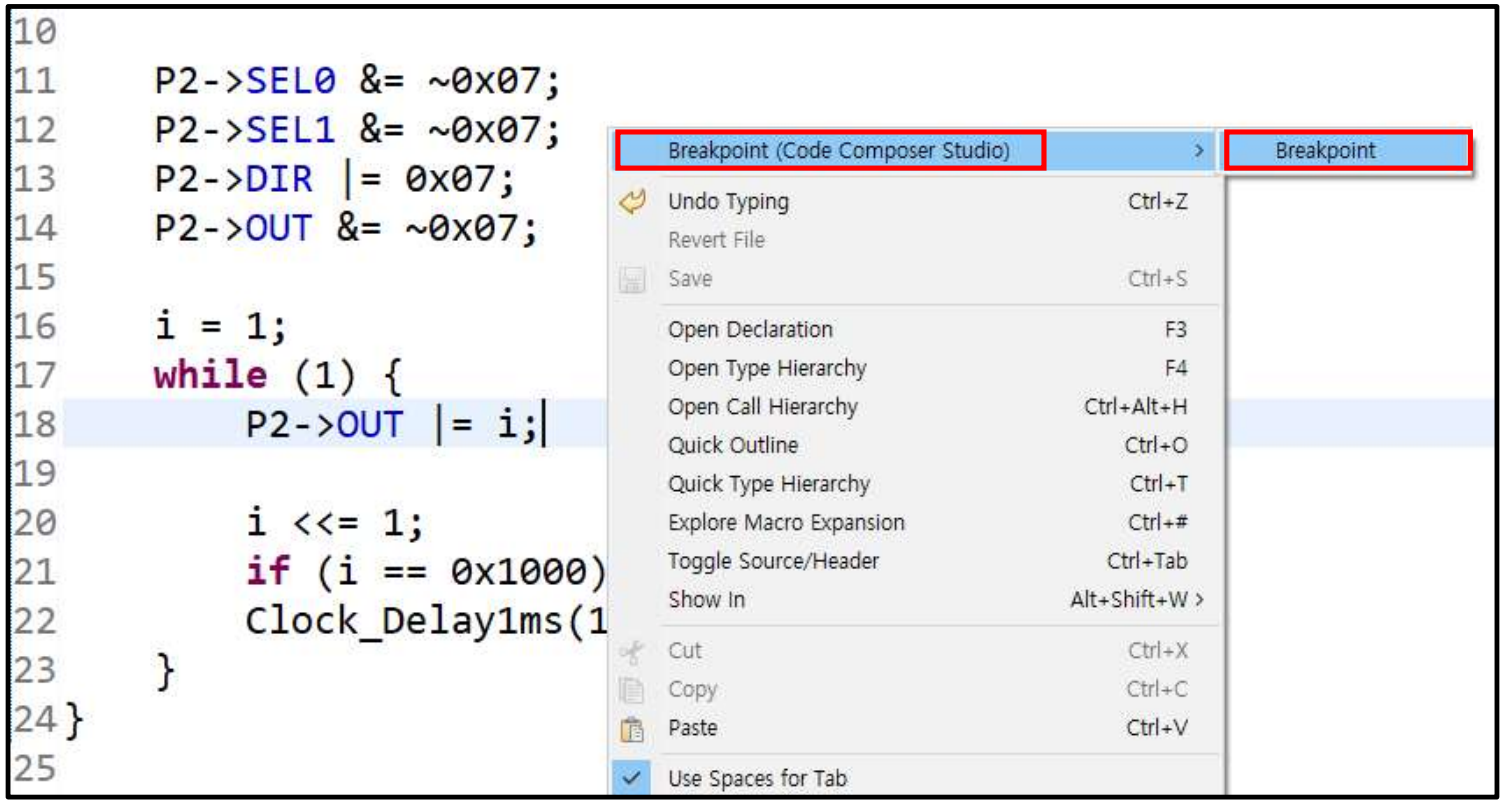
P2->OUT : ff

P2->OUT : ff

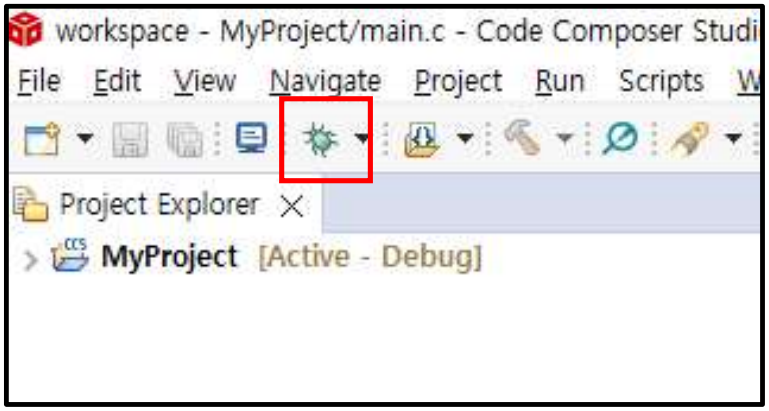
P2->OUT : ff

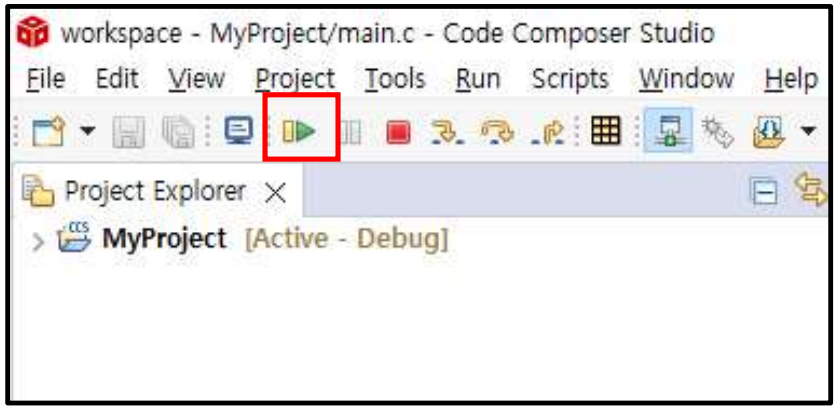
...

Debugger

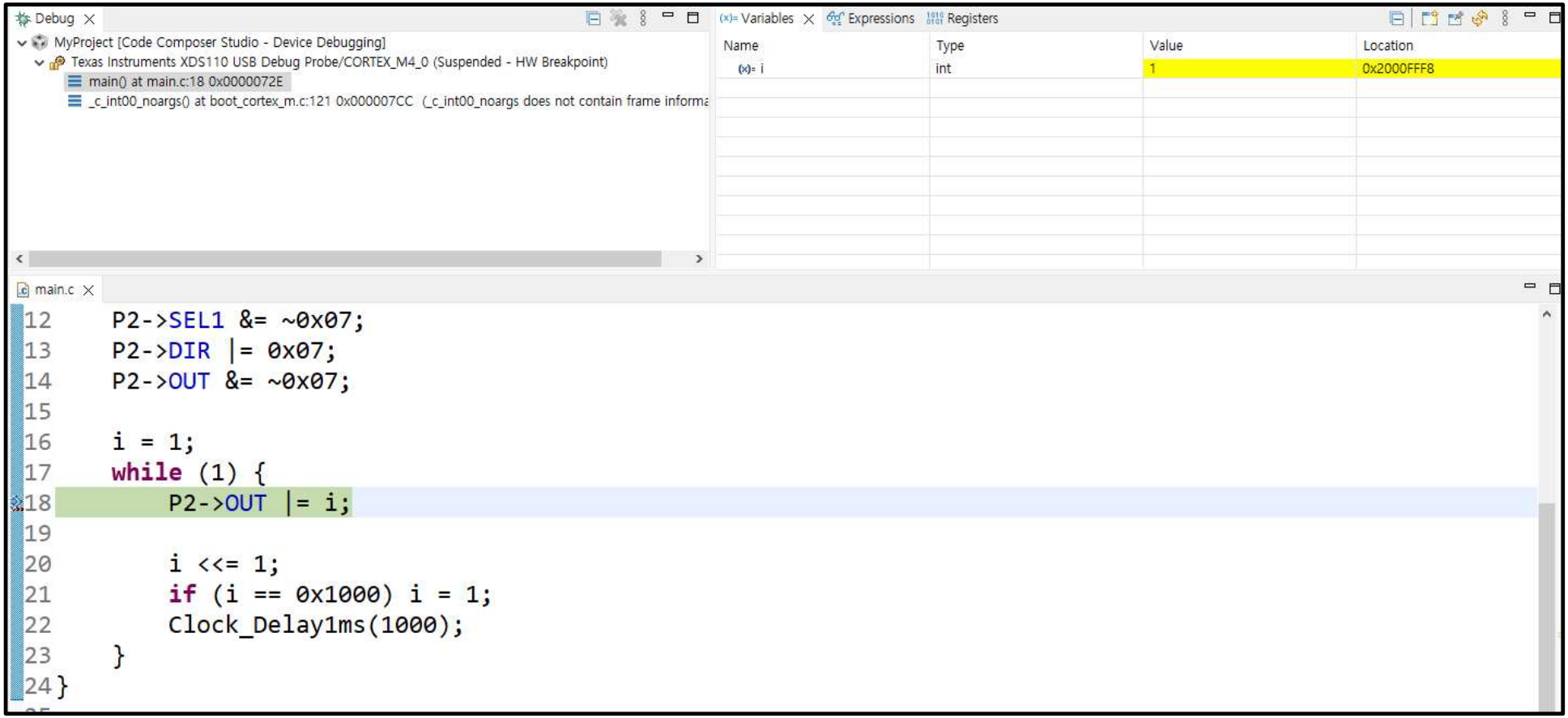

10
11 P2->SEL0 &= ~0x07;
12 P2->SEL1 &= ~0x07;
13 P2->DIR |= 0x07;
14 P2->OUT &= ~0x07;
15
16 i = 1;
17 while (1) {
18 P2->OUT |= i;
19
20 i <<= 1;
21 if (i == 0x1000)
22 Clock_Delay1ms(1
23 }
24}
25
Breakpoint (Code Composer Studio)
Breakpoint
Undo Typing Ctrl+Z
Revert File
Save Ctrl+S
Open Declaration F3
Open Type Hierarchy F4
Open Call Hierarchy Ctrl+Alt+H
Quick Outline Ctrl+O
Quick Type Hierarchy Ctrl+T
Explore Macro Expansion Ctrl+#
Toggle Source/Header Ctrl+Tab
Show In Alt+Shift+W
Cut Ctrl+X
Copy Ctrl+C
Paste Ctrl+V
Use Spaces for Tab

Debugger

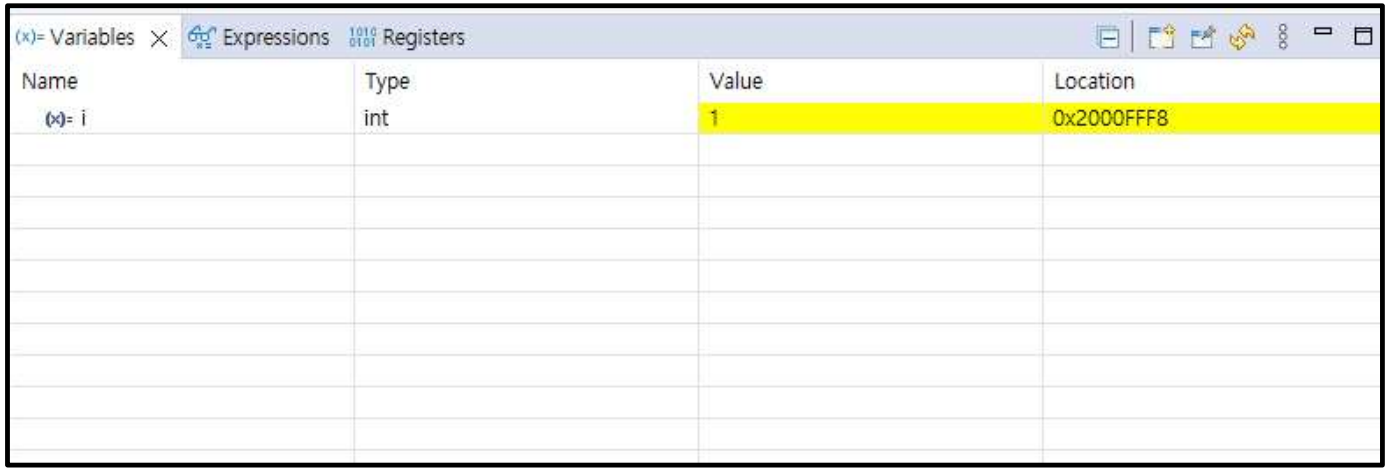

workspace - MyProject/main.c - Code Composer Studi
File Edit View Navigate Project Run Scripts W
Project Explorer
MyProject [Active - Debug]

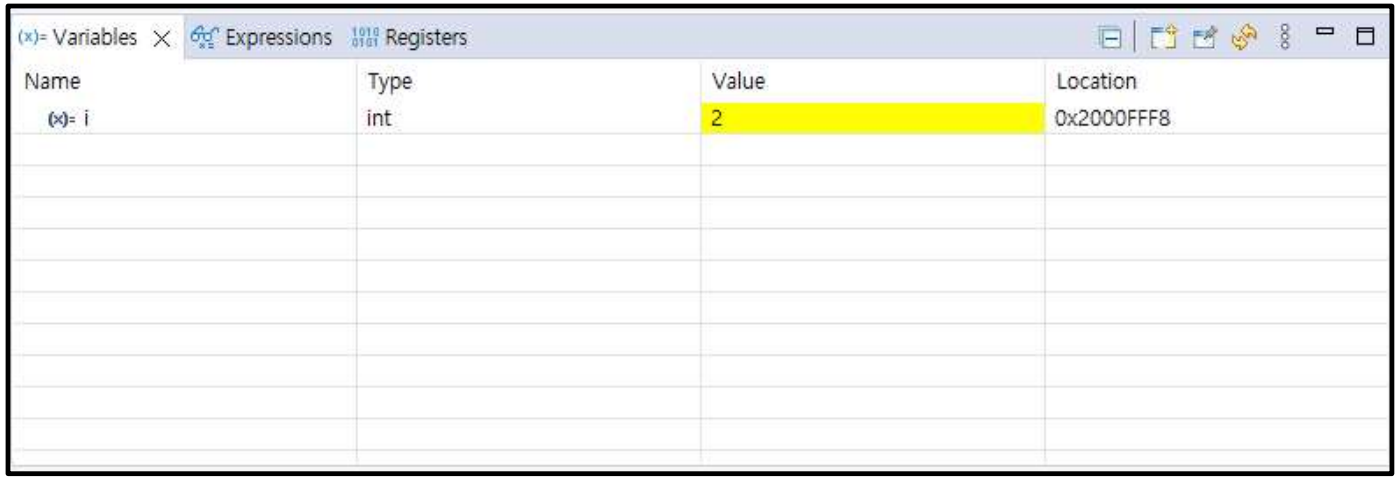

workspace - MyProject/main.c - Code Composer Studio
File Edit View Project Tools Run Scripts Window Help
Project Explorer
MyProject [Active - Debug]

Debugger


Debug
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
Variables
Expressions
Registers
Name
Type
Value
Location
i
int
1
0x2000FFF8
main.c
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 }

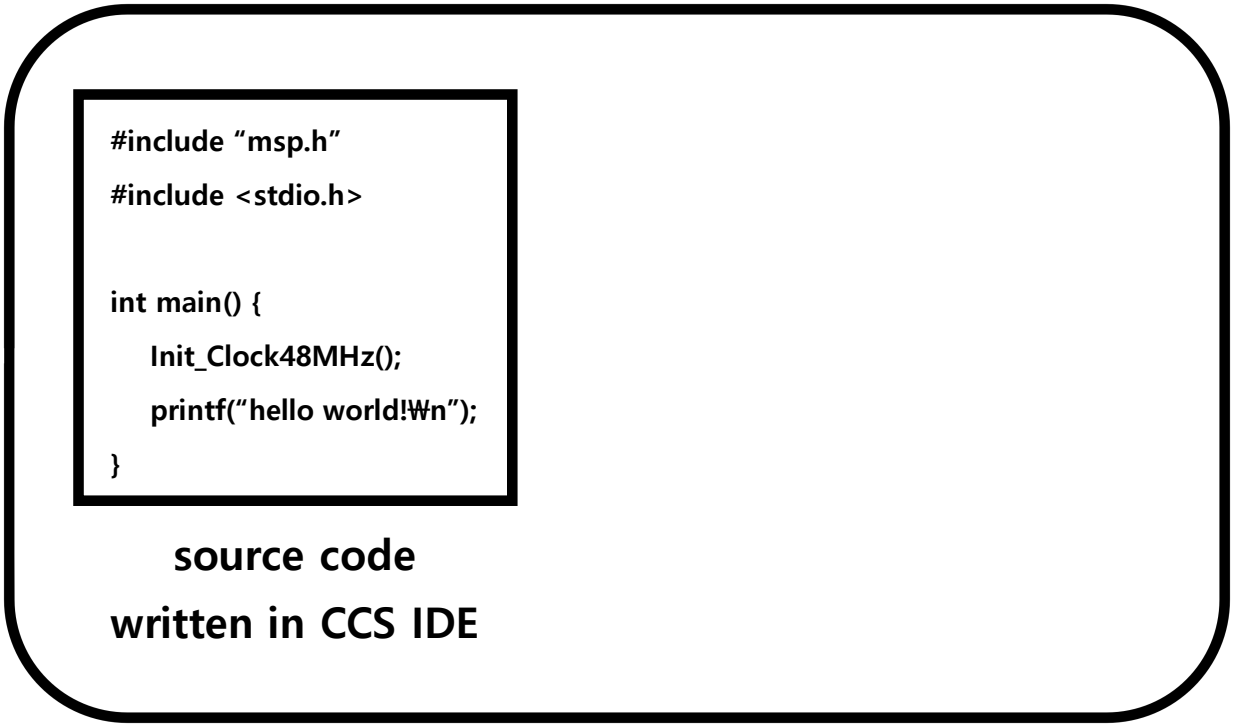
Debugger


Variables
Expressions
Registers
Name
Type
Value
Location
i
int
1
0x2000FFF8

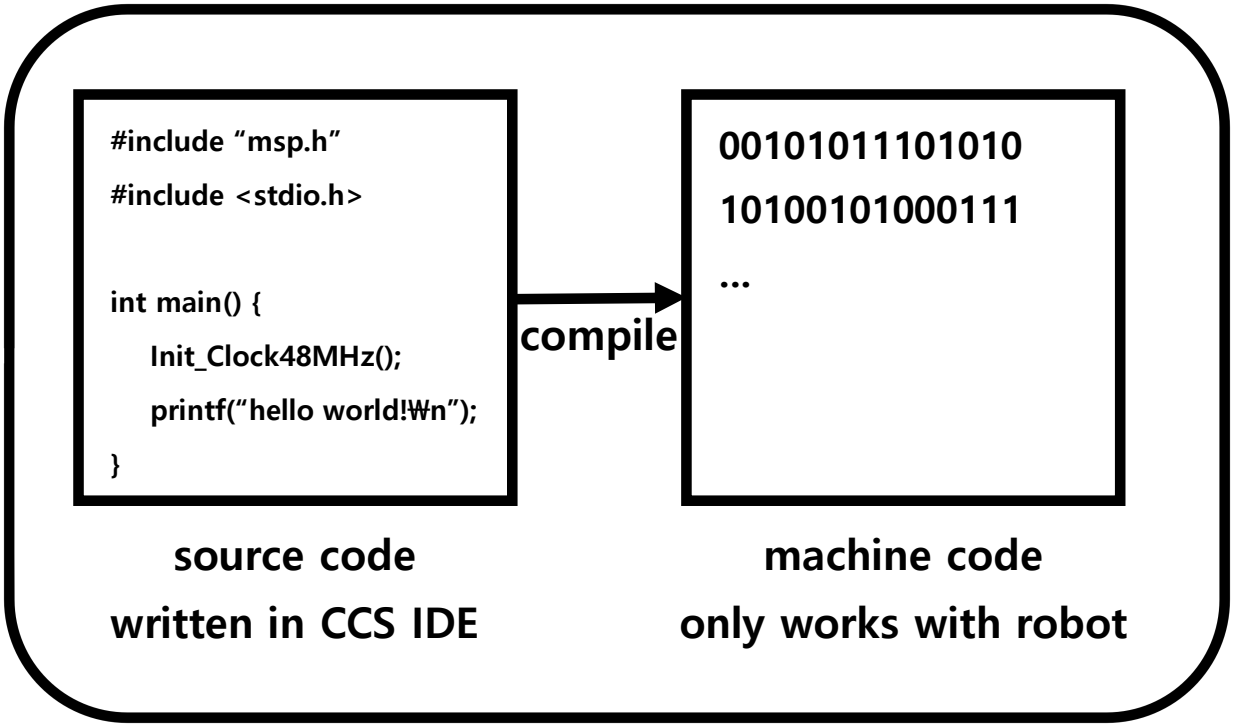

Variables
Expressions
Registers
Name
Type
Value
Location
i
int
2
0x2000FFF8

3. How Robot Works?

How to Upload Source Code



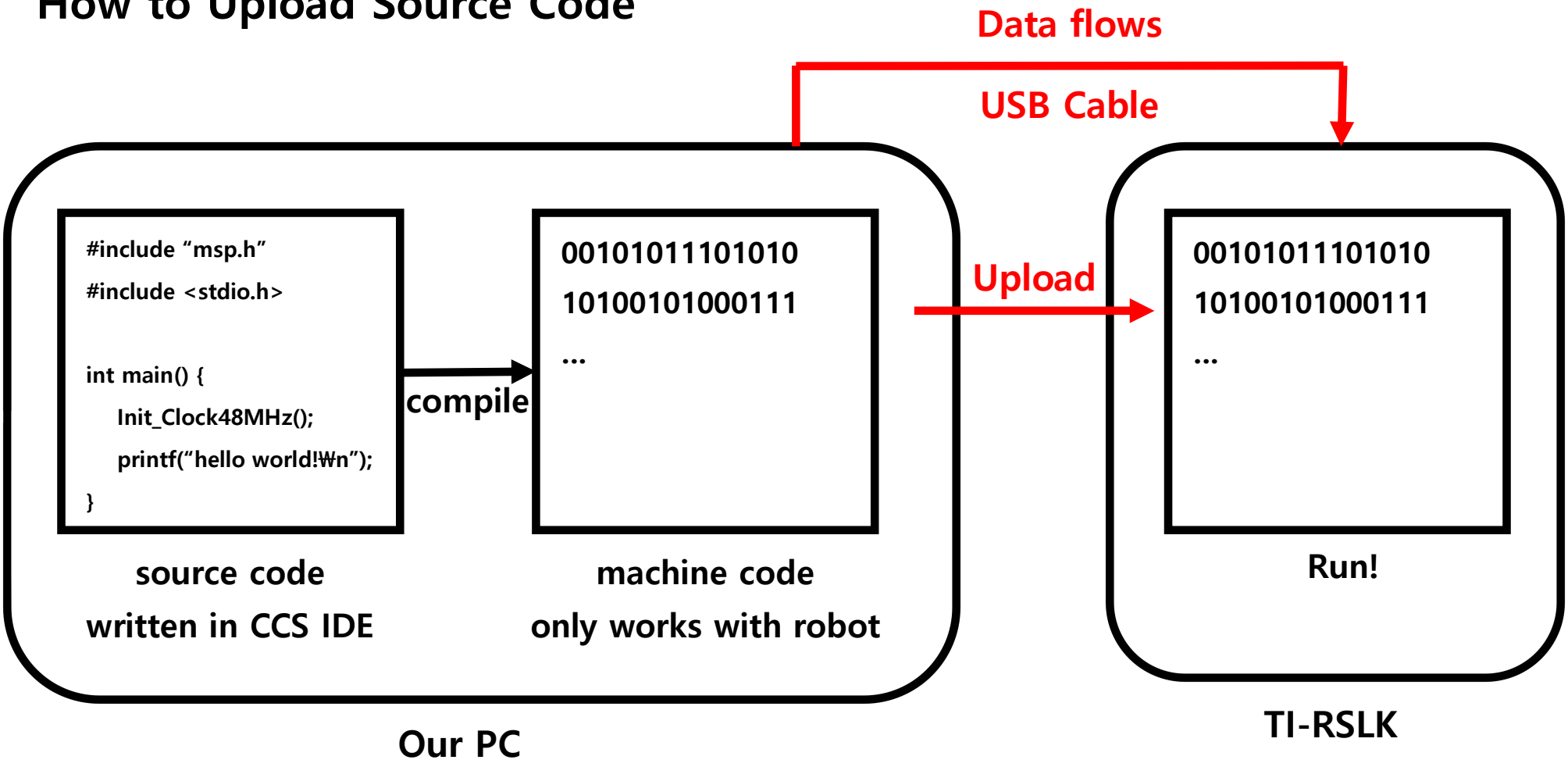
How to Upload Source Code



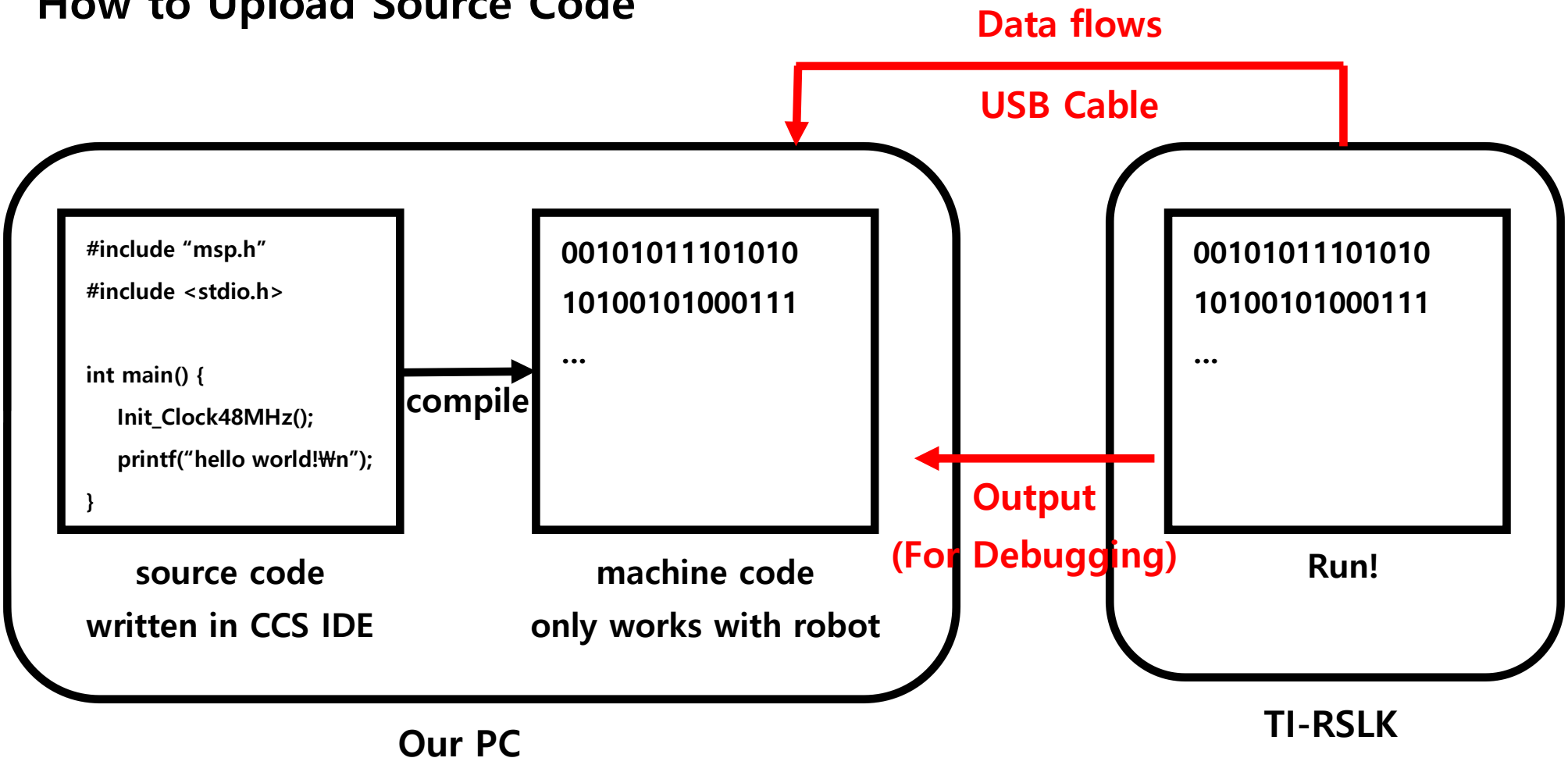
buid & upload

build

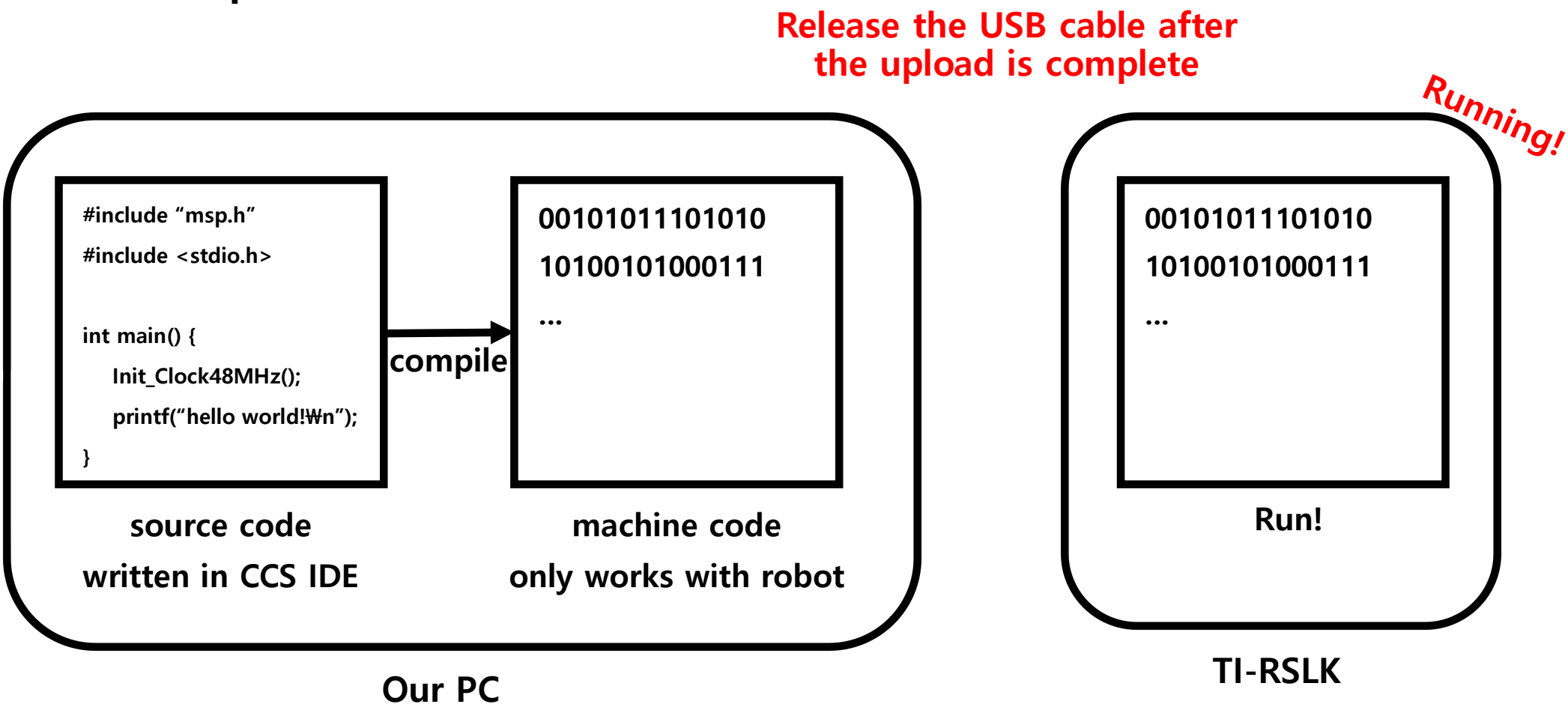
How to Upload Source Code



How to Upload Source Code



How to Upload Source Code



4. Assignment

Bitwise Operation Example

Make the LED

Red -> (3s) -> Blue -> (2s) -> Green -> (1s) -> Red -> ...

Try changing the LED color in the order of Colors should not overlap!

Deadline: ~ 9/27 11:59 PM

Submit : LMS

You can discuss or talk in groups

Only one member of the group needs to submit.

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

Getting Started with Practice

When you're done, be sure to return it.