Embedded System Design

TaeWook Kim & SeokHyun Hong

Hanyang University

Contents

1. CCS Tutorial
2. LED & Switch
3. IR Sensor

Original Lecture

Today’s Lecture is based on

- Running Code on the TI LaunchPad Board Using CCS

- GPIO

- Interfacing Input and Output

CCS TUTORIAL

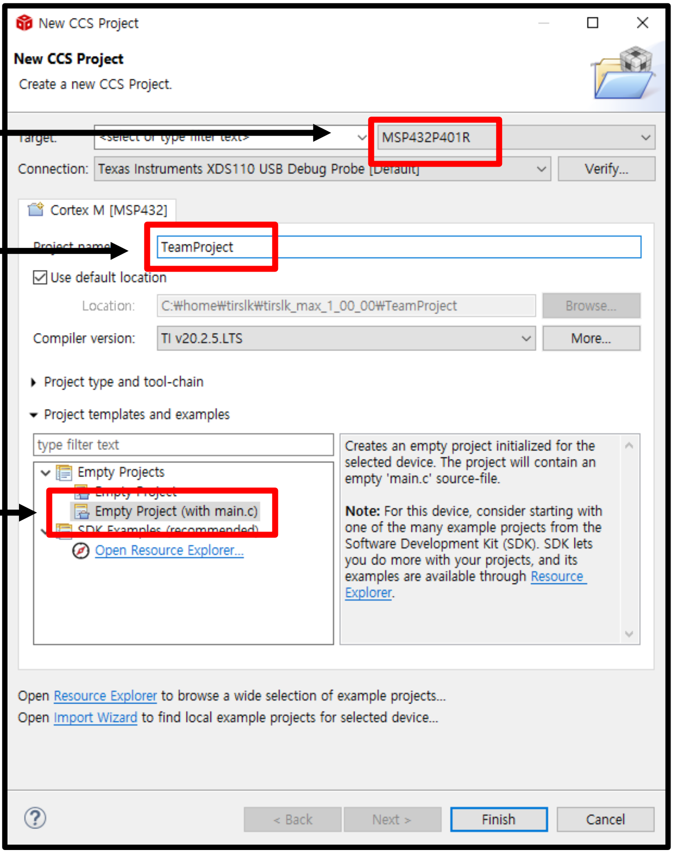
Create Our Project

- Click [File] -> [New] -> [CCS Project]

Select "MSP432P401R"

Project Name

Select "Empty Project (with main.c)"



New CCS Project

New CCS Project

Create a new CCS Project.

Target: <select or type filter text> MSP432P401R

Connection: Texas Instruments XDS110 USB Debug Probe [Default] Verify...

Cortex M [MSP432]

Project name: TeamProject

☑ Use default location

Location: C:₩home₩tirslk₩tirslk_max_1_00_00₩TeamProject Browse...

Compiler version: TI v20.2.5.LTS More...

▸ Project type and tool-chain

▾ Project templates and examples

type filter text

- Empty Projects
  - Empty Project
  - Empty Project (with main.c)
- SDK Examples (recommended)
  - Open Resource Explorer...

Creates an empty project initialized for the selected device. The project will contain an empty 'main.c' source-file.

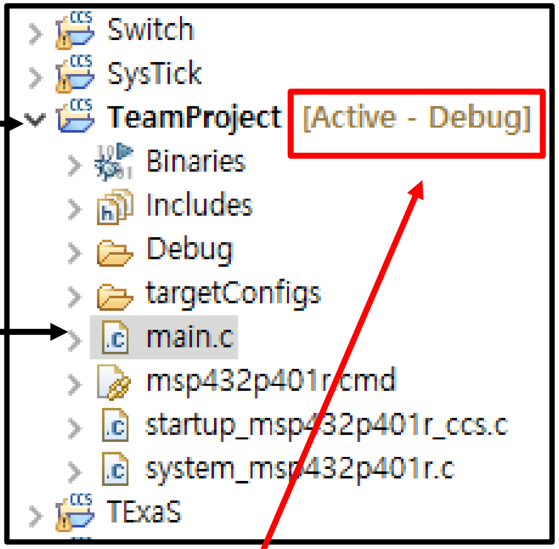
Note: For this device, consider starting with one of the many example projects from the Software Development Kit (SDK). SDK lets you do more with your projects, and its examples are available through Resource Explorer.

Open Resource Explorer to browse a wide selection of example projects...

Open Import Wizard to find local example projects for selected device...

Write a Simple Program

- Open Our Project



Open our project

Select main.c

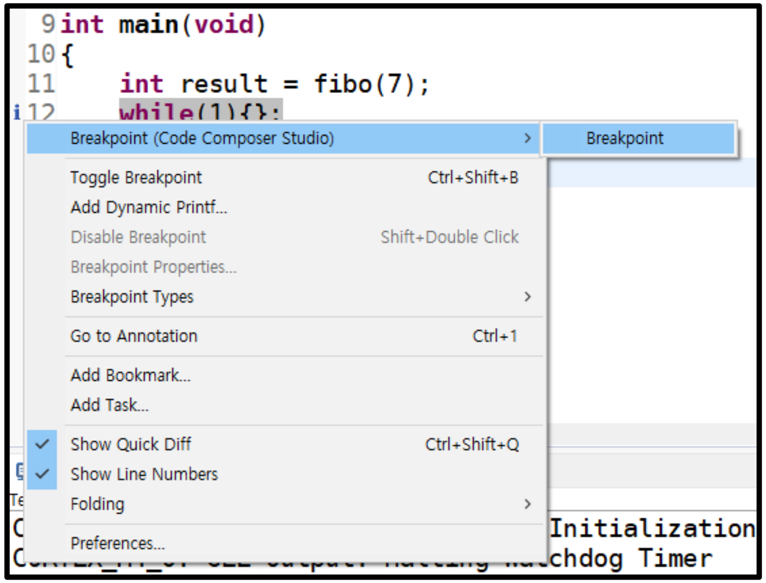
Important! Please check that “Active-Debug” is enabled

Write a Simple Program

- Write a Simple Fibonacci Program

```
[01] // main.c
[02] #include “msp.h”
[03]
[04] int fibo (int num) {
[05]    if (num <= 1) return 0;
[06]    else if (num == 2) return 1;
[07]    return fibo(num-1) + fibo(num-2);
[08] }
[09]
[10] int main(void) {
[11]   int result = fibo(7);
[12]   while(1) {};
[13] }
```

Write a program



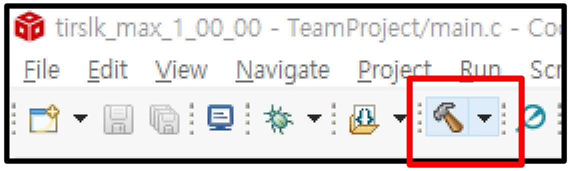
```
 9 int main(void)
10 {
11     int result = fibo(7);
12     while(1){};
```

Make a break point at line 12

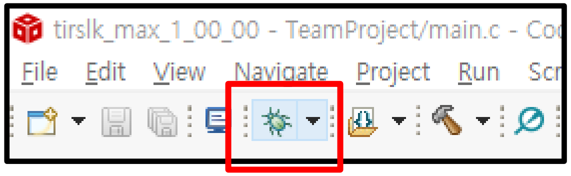
[Right Click at Line 12] -> [Breakpoint] ->[Breakpoint]

Debugging

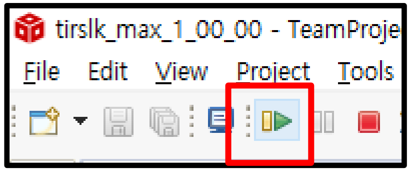
- Compile and Debug



Compile



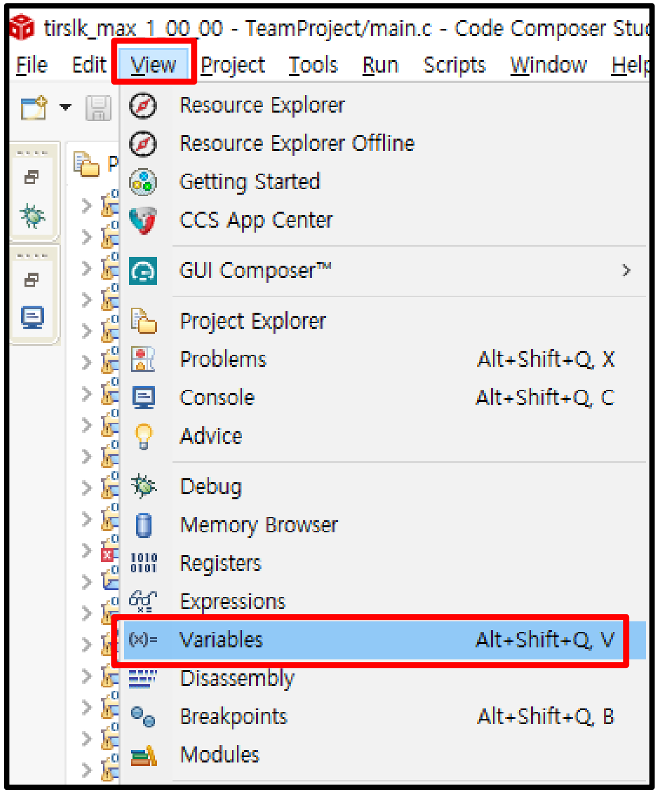
Debug

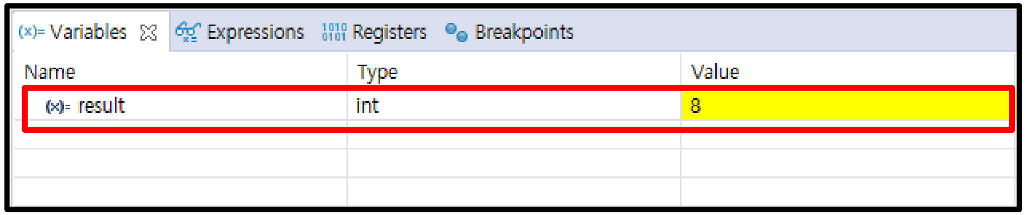


Run a Program

Debugging

- Check the “result” value by clicking [View] -> [Variables]



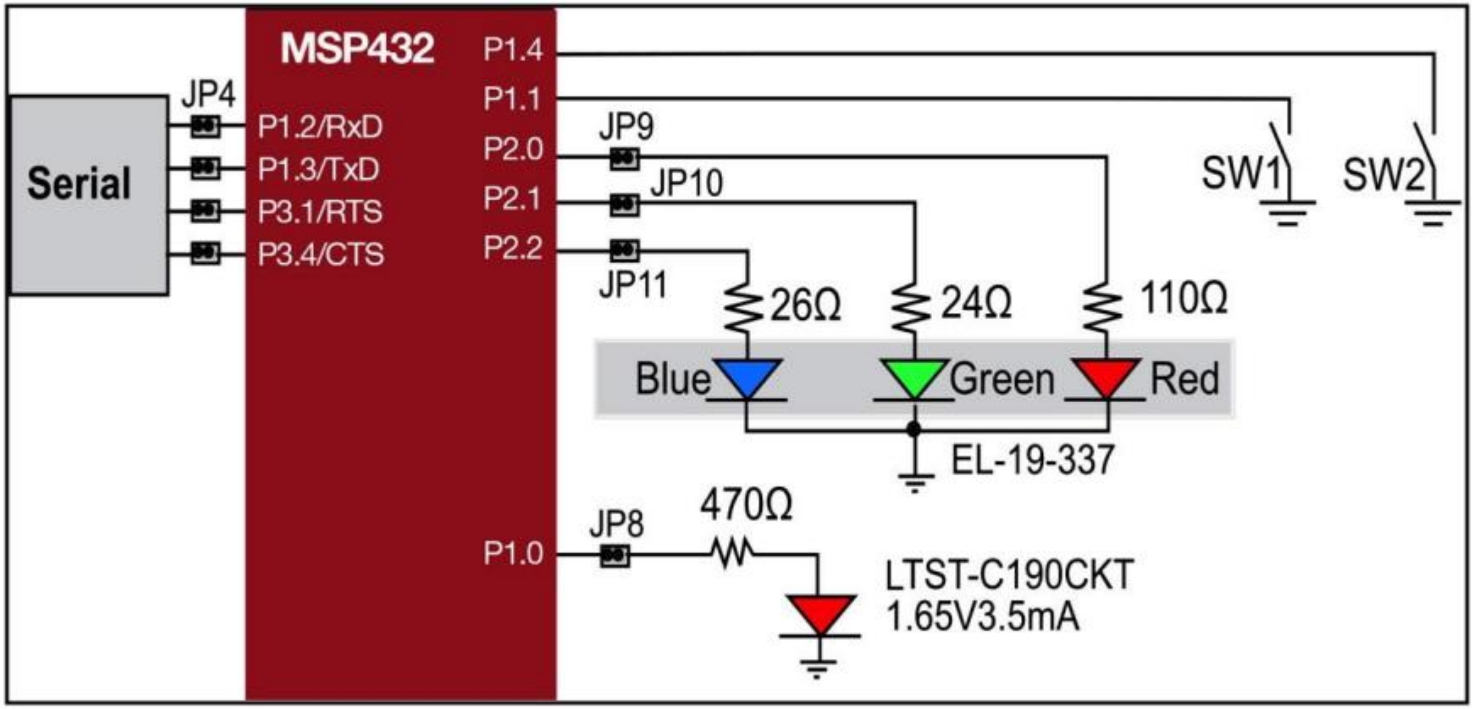


Name	Type	Value
(x)= result	int	8

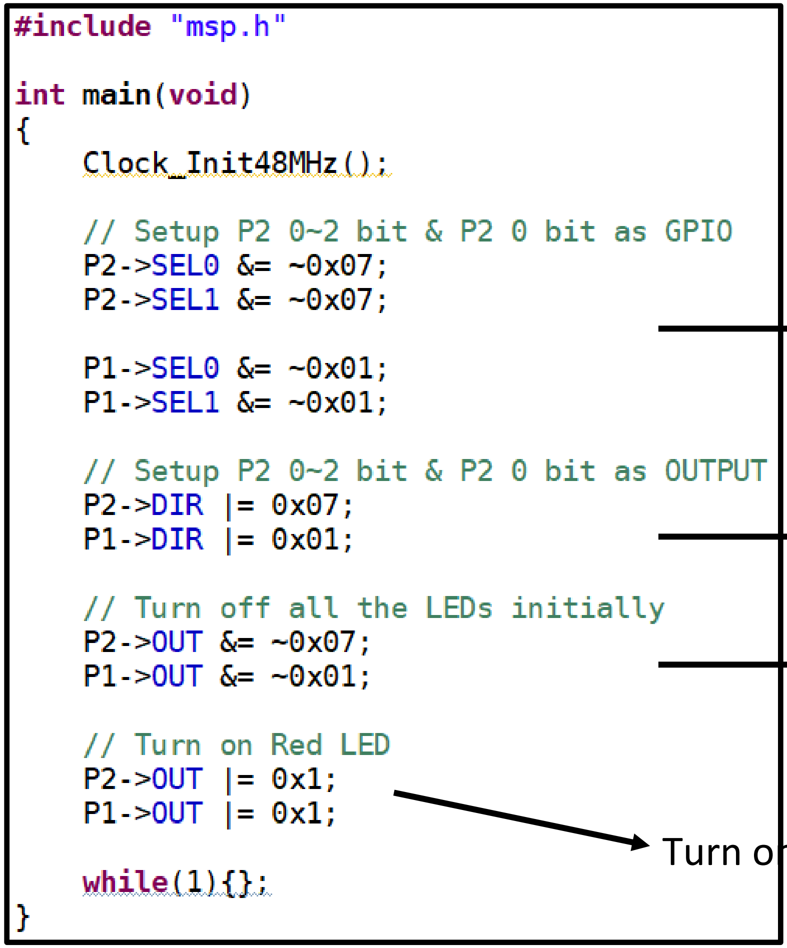
You can see that “result” has been changed to “8”

LED & SWITCH

MSP432 LaunchPad LED & Switch



Basic LED Control



```
#include "msp.h"

int main(void)
{
    Clock_Init48MHz();

    // Setup P2 0~2 bit & P2 0 bit as GPIO
    P2->SEL0 &= ~0x07;
    P2->SEL1 &= ~0x07;

    P1->SEL0 &= ~0x01;
    P1->SEL1 &= ~0x01;

    // Setup P2 0~2 bit & P2 0 bit as OUTPUT
    P2->DIR |= 0x07;
    P1->DIR |= 0x01;

    // Turn off all the LEDs initially
    P2->OUT &= ~0x07;
    P1->OUT &= ~0x01;

    // Turn on Red LED
    P2->OUT |= 0x1;
    P1->OUT |= 0x1;

    while(1){};
}
```

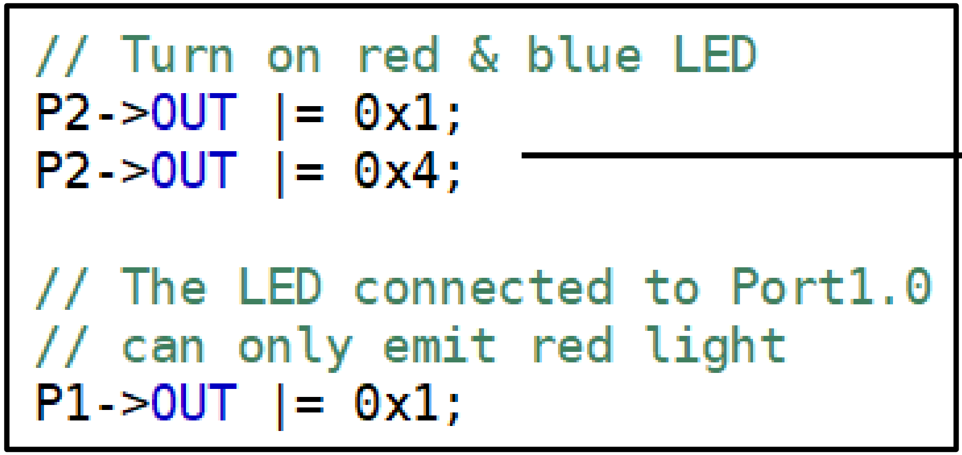
Setup Port1 0 bit, Port2 0~2bit as GPIO

Setup Port1 0 bit, Port2 0~2bit as Output

Turn off all the LEDs initially

Turn on RED LEDs

LED Control 1



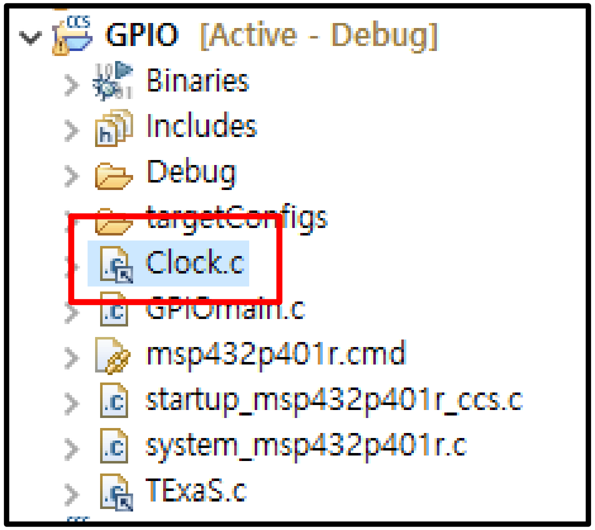
```
// Turn on red & blue LED
P2->OUT |= 0x1;
P2->OUT |= 0x4;

// The LED connected to Port1.0
// can only emit red light
P1->OUT |= 0x1;
```

See the light color of LED

The LED which is connected to Port1.0 can only emit red light

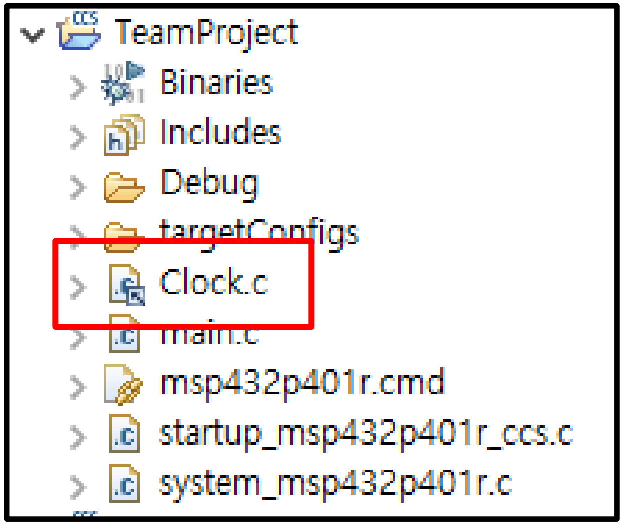
But the LEDs connected to Port2 0~2bit can emit RGB light

LED Control 2



GPIO [Active - Debug]
- Binaries
- Includes
- Debug
- targetConfigs
- Clock.c
- GPIOmain.c
- msp432p401r.cmd
- startup_msp432p401r_ccs.c
- system_msp432p401r.c
- TExaS.c

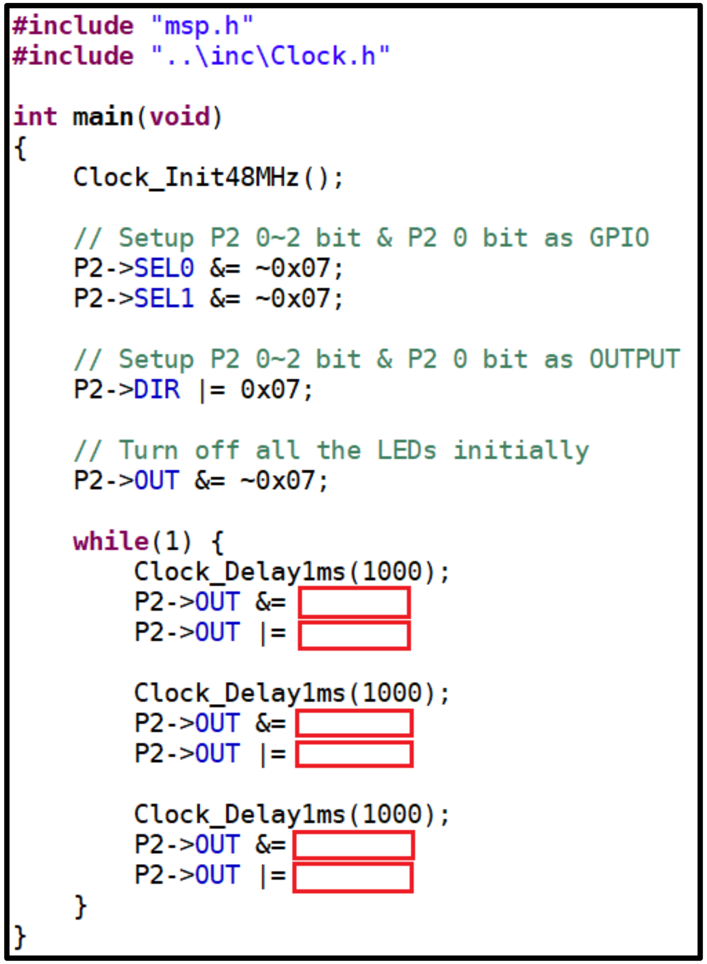
Copy “Clock.c” from GPIO project



TeamProject
- Binaries
- Includes
- Debug
- targetConfigs
- Clock.c
- main.c
- msp432p401r.cmd
- startup_msp432p401r_ccs.c
- system_msp432p401r.c

Paste “Clock.c” to our project

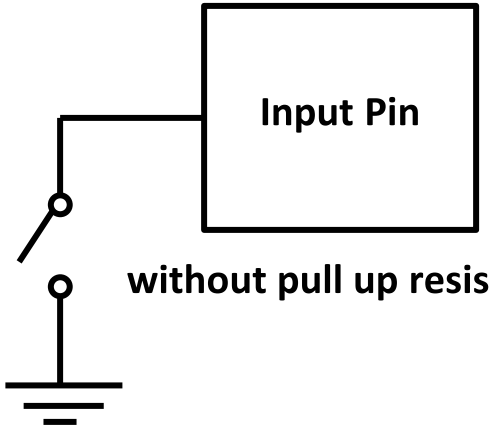
LED Control 2



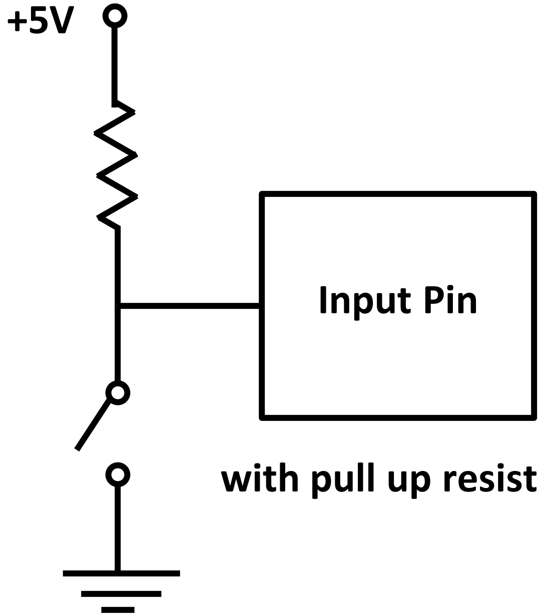
```
#include "msp.h"
#include "..\inc\Clock.h"

int main(void)
{
    Clock_Init48MHz();

    // Setup P2 0~2 bit & P2 0 bit as GPIO
    P2->SEL0 &= ~0x07;
    P2->SEL1 &= ~0x07;

    // Setup P2 0~2 bit & P2 0 bit as OUTPUT
    P2->DIR |= 0x07;

    // Turn off all the LEDs initially
    P2->OUT &= ~0x07;

    while(1) {
        Clock_Delay1ms(1000);
        P2->OUT &= □
        P2->OUT |= □

        Clock_Delay1ms(1000);
        P2->OUT &= □
        P2->OUT |= □

        Clock_Delay1ms(1000);
        P2->OUT &= □
        P2->OUT |= □
    }
}
```

Make the colors of the LEDs in the order of red, green and blue

Basic Switch Control



without pull up resistor

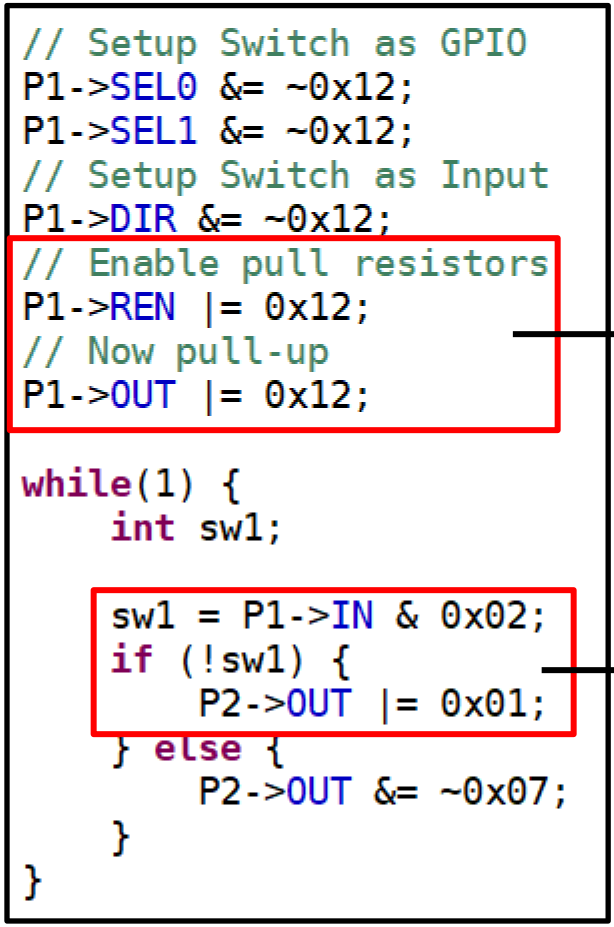


with pull up resistor

Floating!

+5V (High)

Basic Switch Control

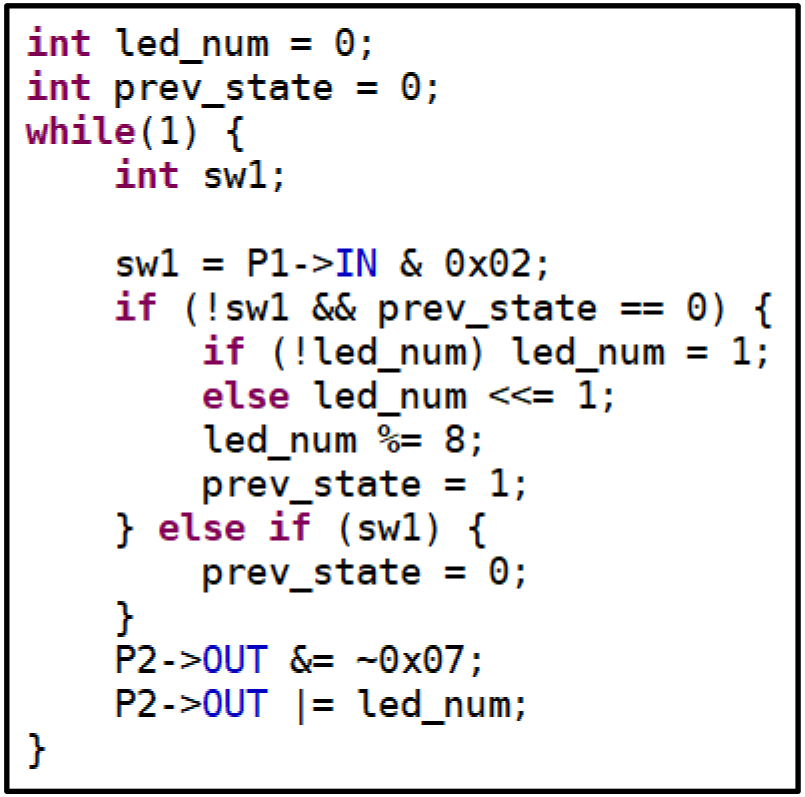


```
// Setup Switch as GPIO
P1->SEL0 &= ~0x12;
P1->SEL1 &= ~0x12;
// Setup Switch as Input
P1->DIR &= ~0x12;
// Enable pull resistors
P1->REN |= 0x12;
// Now pull-up
P1->OUT |= 0x12;

while(1) {
    int sw1;

    sw1 = P1->IN & 0x02;
    if (!sw1) {
        P2->OUT |= 0x01;
    } else {
        P2->OUT &= ~0x07;
    }
}
```

Setup pull-up resistors

Turn on LED when pressed the switch 1

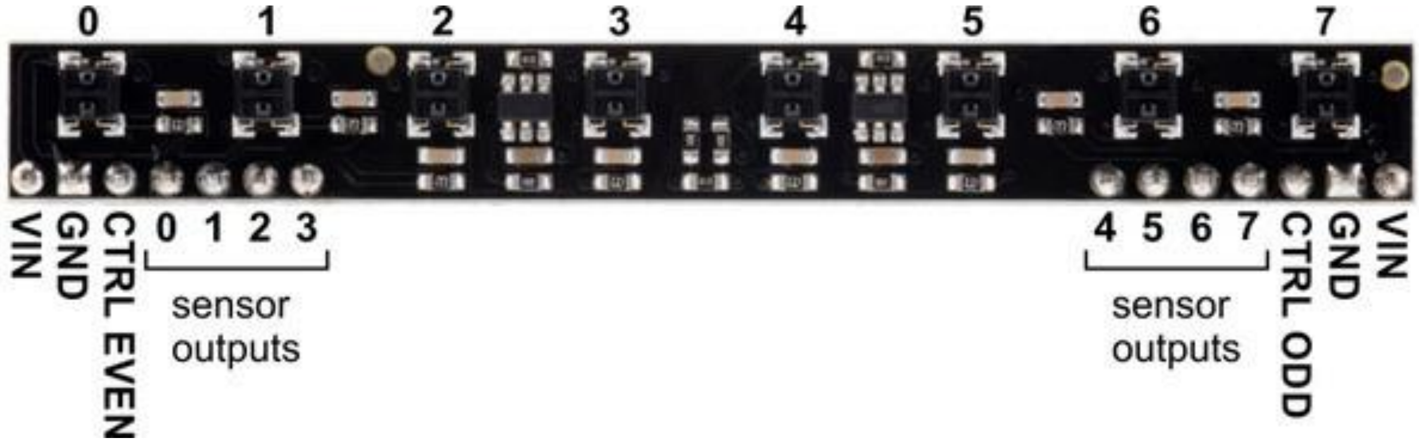
Control LED with Switch



```c
int led_num = 0;
int prev_state = 0;
while(1) {
    int sw1;

    sw1 = P1->IN & 0x02;
    if (!sw1 && prev_state == 0) {
        if (!led_num) led_num = 1;
        else led_num <<= 1;
        led_num %= 8;
        prev_state = 1;
    } else if (sw1) {
        prev_state = 0;
    }
    P2->OUT &= ~0x07;
    P2->OUT |= led_num;
}
```

IR SENSOR

QTRX Sensor

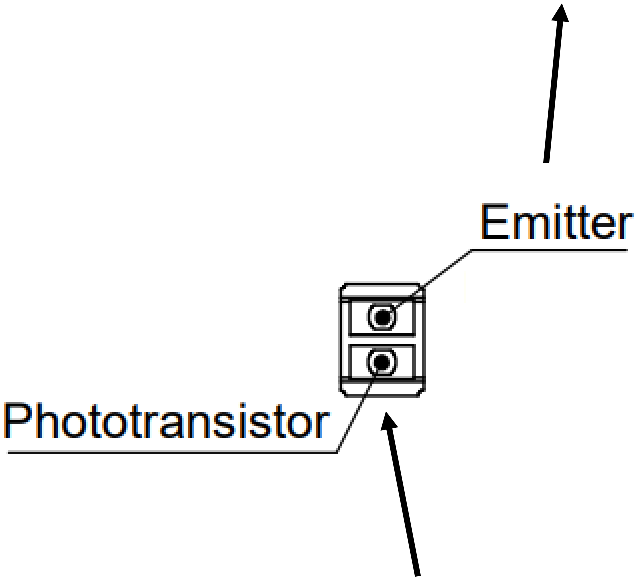


QTRX Sensor is a Infra Red Sensor

It cannot detect visible sensor!

QTRX Sensor

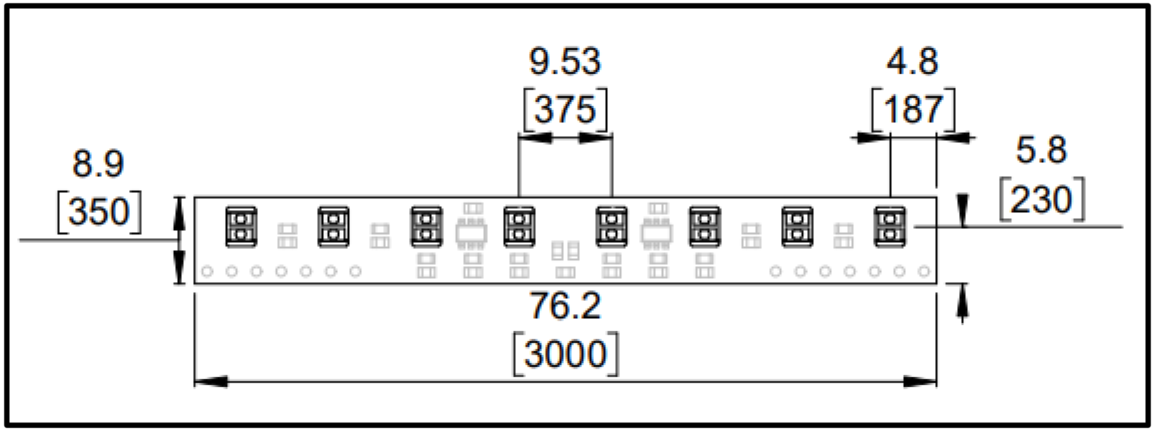
Infra Red light comes from here

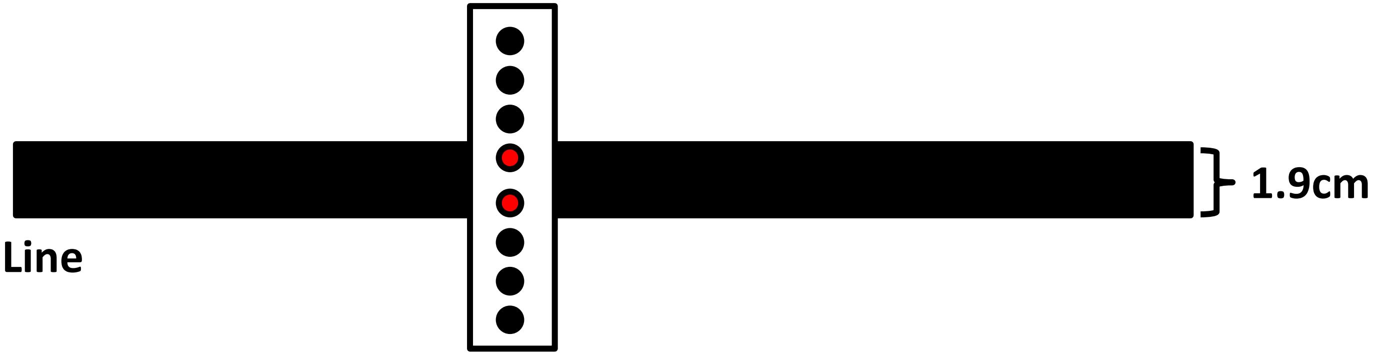


IR light should go to phototransistor

View QRTX Sensor with IR Camera

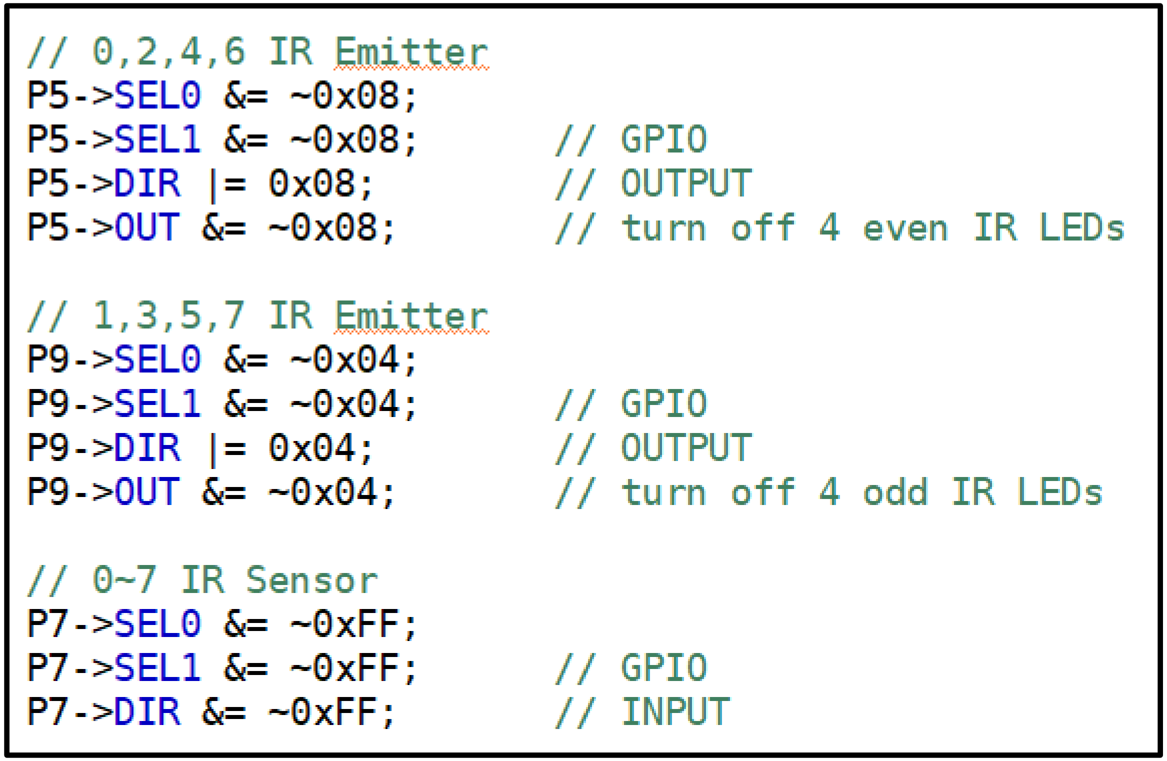
QTRX Sensor





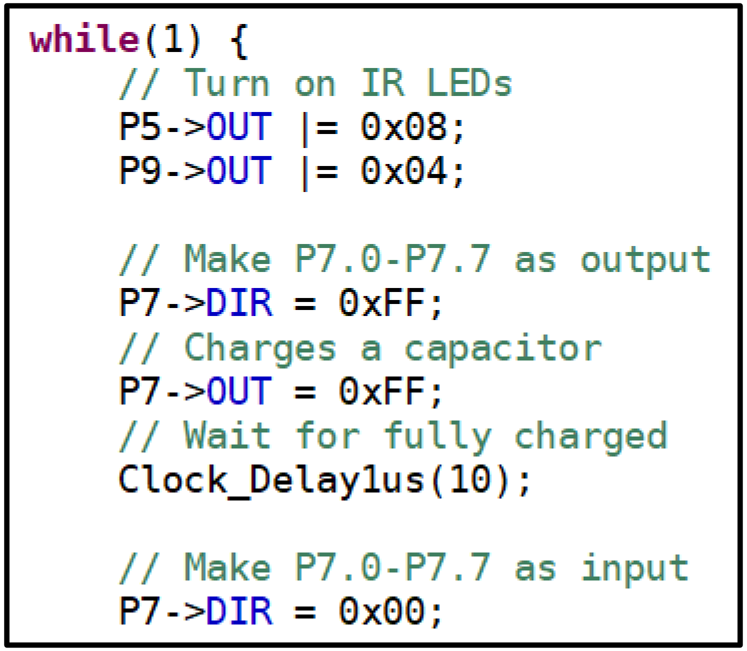
QTRX Sensor

Basic IR Sensor Control

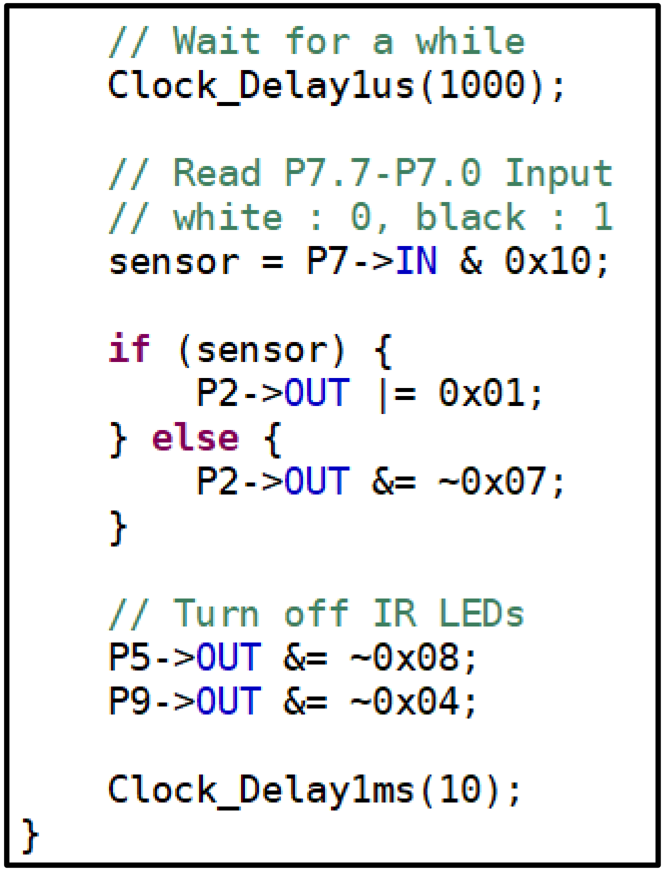


```
// 0,2,4,6 IR Emitter
P5->SEL0 &= ~0x08;
P5->SEL1 &= ~0x08;      // GPIO
P5->DIR |= 0x08;        // OUTPUT
P5->OUT &= ~0x08;       // turn off 4 even IR LEDs

// 1,3,5,7 IR Emitter
P9->SEL0 &= ~0x04;
P9->SEL1 &= ~0x04;      // GPIO
P9->DIR |= 0x04;        // OUTPUT
P9->OUT &= ~0x04;       // turn off 4 odd IR LEDs

// 0~7 IR Sensor
P7->SEL0 &= ~0xFF;
P7->SEL1 &= ~0xFF;      // GPIO
P7->DIR &= ~0xFF;       // INPUT
```

Basic IR Sensor Control



```
while(1) {
    // Turn on IR LEDs
    P5->OUT |= 0x08;
    P9->OUT |= 0x04;

    // Make P7.0-P7.7 as output
    P7->DIR = 0xFF;
    // Charges a capacitor
    P7->OUT = 0xFF;
    // Wait for fully charged
    Clock_Delay1us(10);

    // Make P7.0-P7.7 as input
    P7->DIR = 0x00;
```

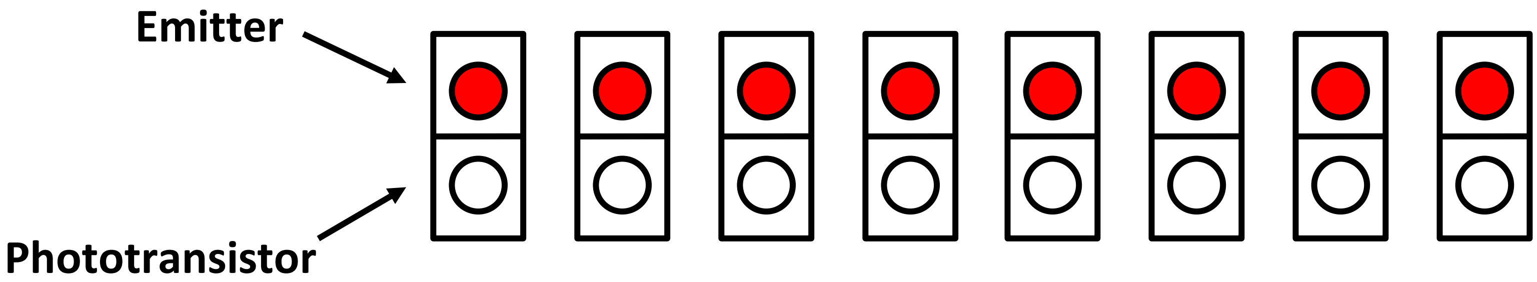


```
    // Wait for a while
    Clock_Delay1us(1000);

    // Read P7.7-P7.0 Input
    // white : 0, black : 1
    sensor = P7->IN & 0x10;

    if (sensor) {
        P2->OUT |= 0x01;
    } else {
        P2->OUT &= ~0x07;
    }

    // Turn off IR LEDs
    P5->OUT &= ~0x08;
    P9->OUT &= ~0x04;

    Clock_Delay1ms(10);
}
```

Basic IR Sensor Control

1) Turn on IR LEDs

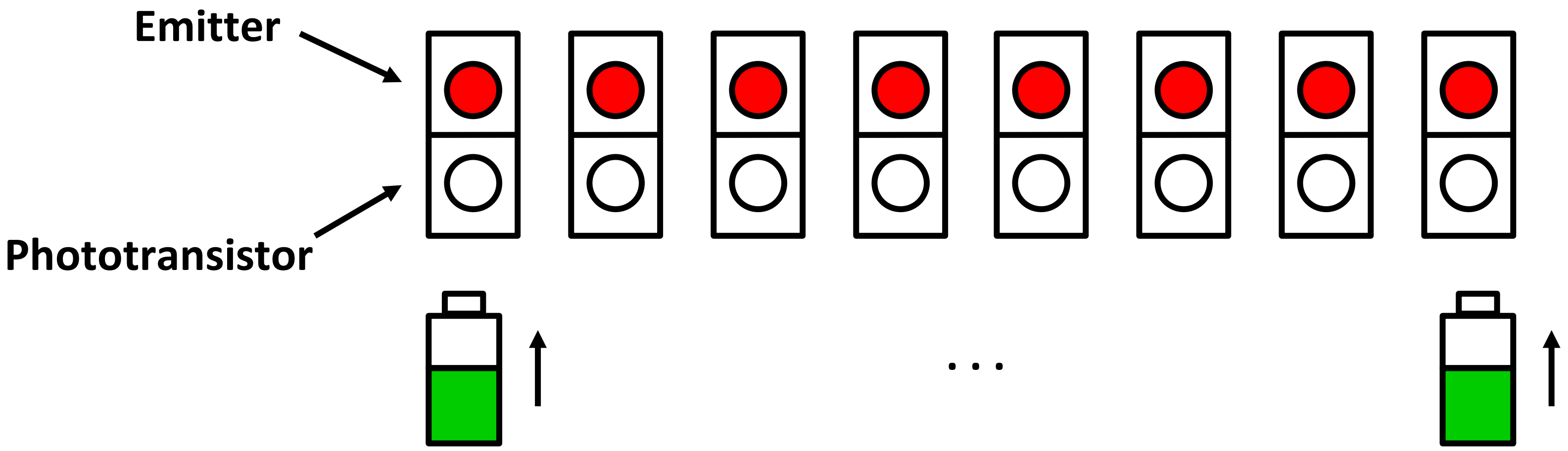
- Turn on both even and odd emitters



Basic IR Sensor Control

2) Charge Capacitors

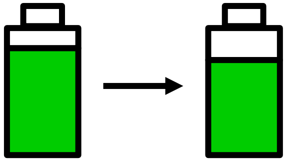
- To charge, we should change P7->DIR to output and charge a capacitor through P7->OUT = 0xFF
- We need to wait for fully charged



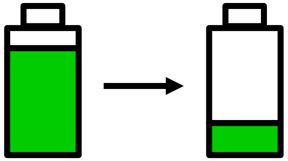
Basic IR Sensor Control

3) Wait for a while after fully charged

- Capacitor is discharged slowly in a natural situation. But it is very slow
- When IR Sensor gets a light, it discharges capacitor
- Use above property, we can distinguish between white and black surfaces



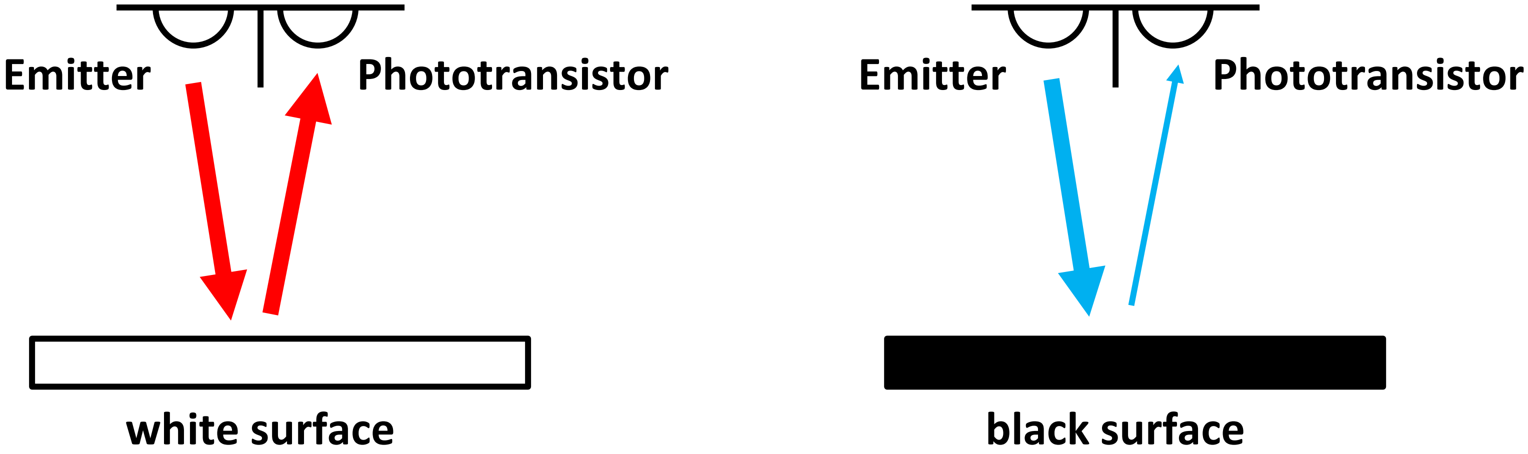
No IR Light



With IR Light

Basic IR Sensor Control

3) Wait for a while after fully charged


Emitter

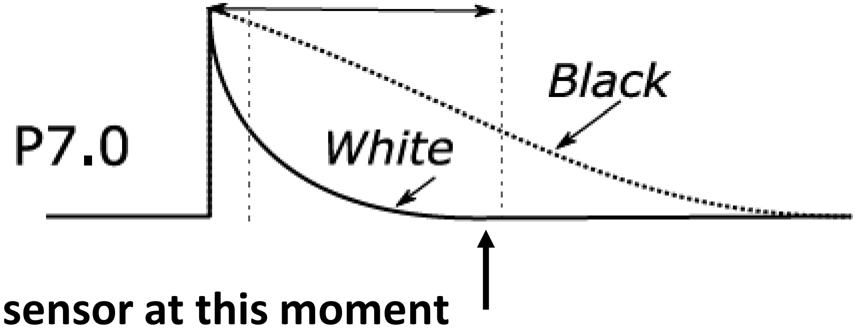
Phototransistor

white surface

Emitter

Phototransistor

black surface



We have to read a sensor at this moment

Basic IR Sensor Control

4) Read Sensor

- Make Port7 as input and read Port 7
- When we read 0, it means white
- When we read 1, it means black

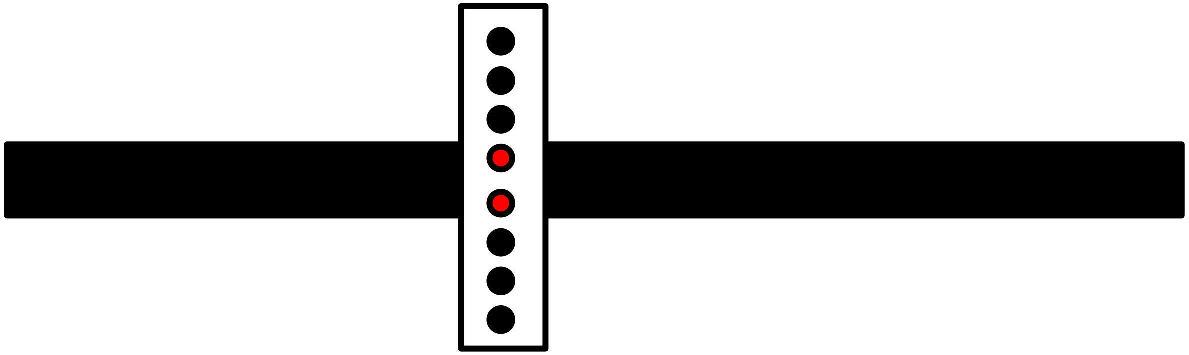
Basic IR Sensor Control

5) Turn off IR LEDs

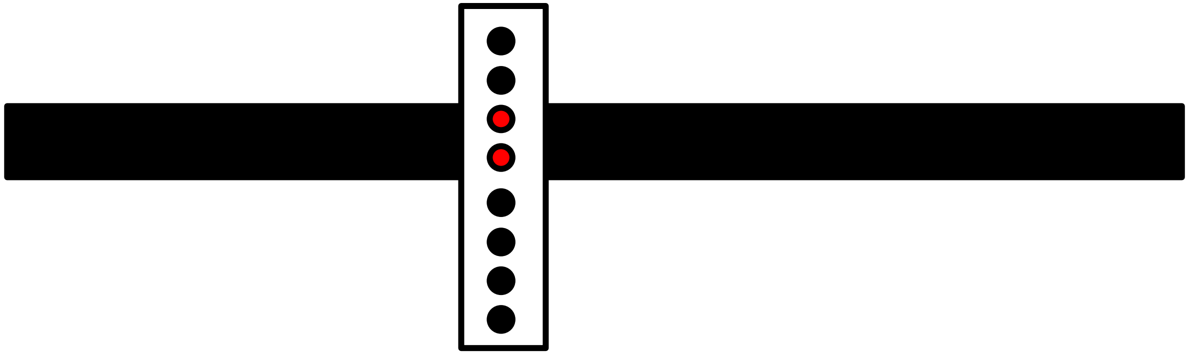
- To save energy, turn off IR LEDs and sleep for a while

IR Sensor Control Practice

Turn on LED when the line is located at the center of the robot



Turn on LED!



Turn off LED!

Thank You