Line Tracer 05

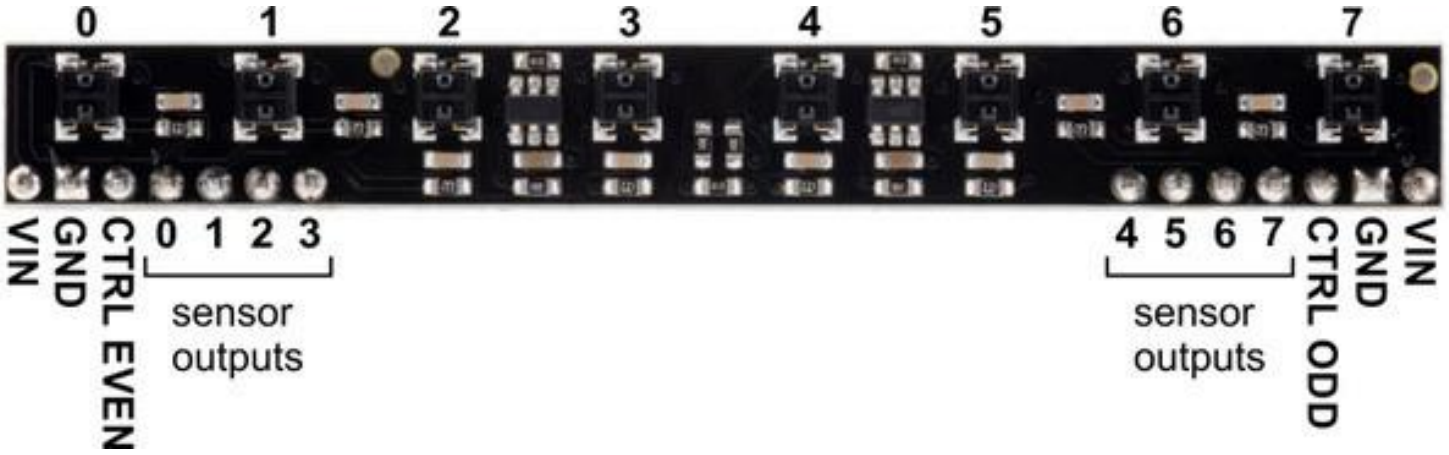
- IR Sensor -

This lecture is based on

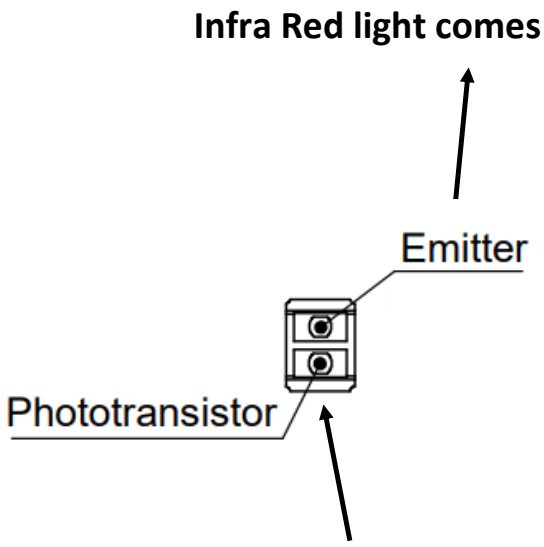
- GPIO

1. QTRX Sensor

About QTRX Sensor

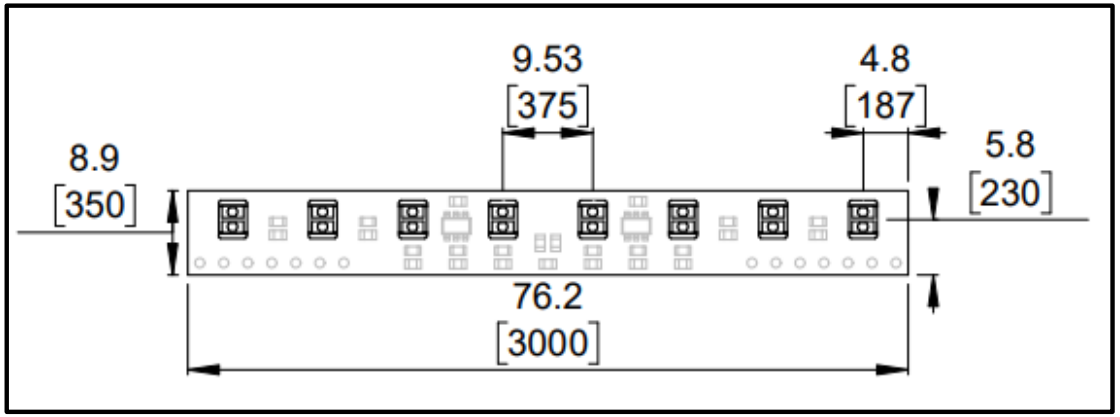


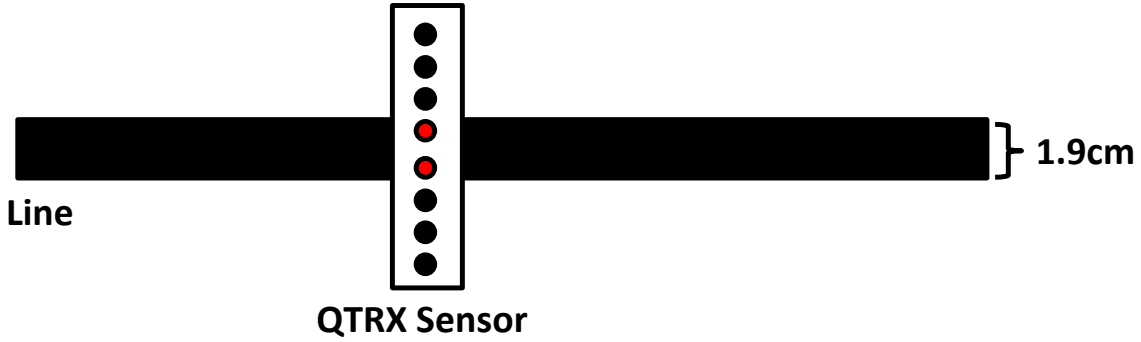
About QTRX Sensor



View QRTX Sensor with IR Camera

About QTRX Sensor





2. IR Sensor Implementation

IR Sensor Initialization

```
// 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
```

IR Sensor Basic Usage

```
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;
```

You should turn on the power!

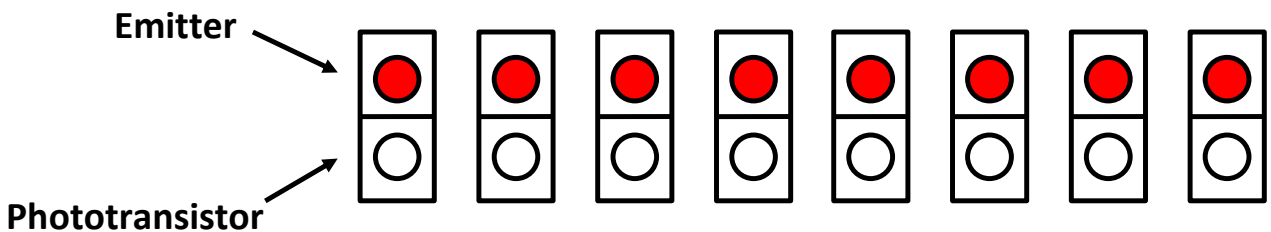
```
    // Wait for a while
    Clock_Delay1us(1000);

    // 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);
}
```

IR Sensor Basic Usage

1) Turn on IR LED

- Turn on both even and odd emitters

```
// Turn on IR LEDs
P5->OUT |= 0x08;
P9->OUT |= 0x04;
```

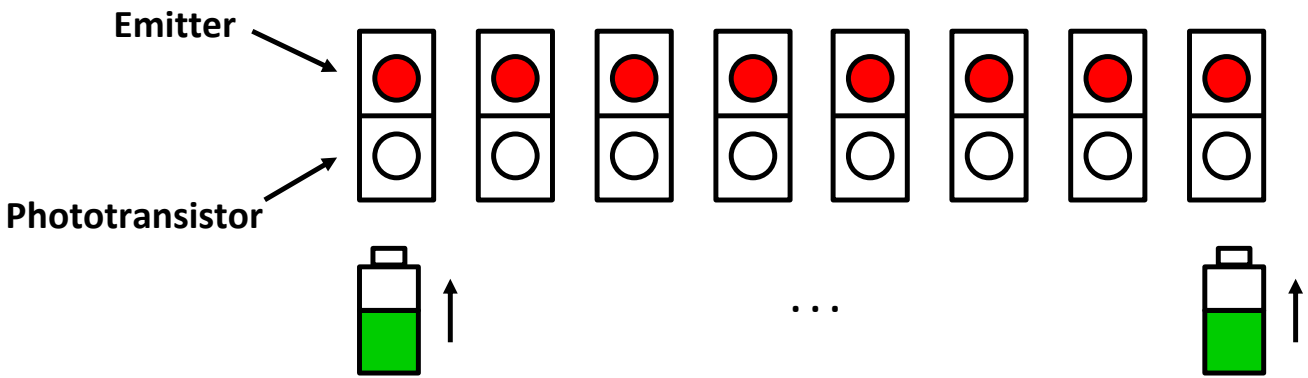


IR Sensor Basic Usage

2) Charge Capacitors

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

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

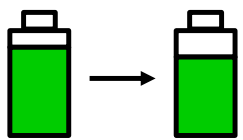


IR Sensor Basic Usage

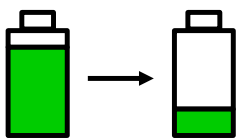
3) Wait for a while after fully charged

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

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

// Wait for a while
Clock_Delay1us(1000);

// Read 5th sensor, not entire
sensor = P7->IN & 0x10;
```



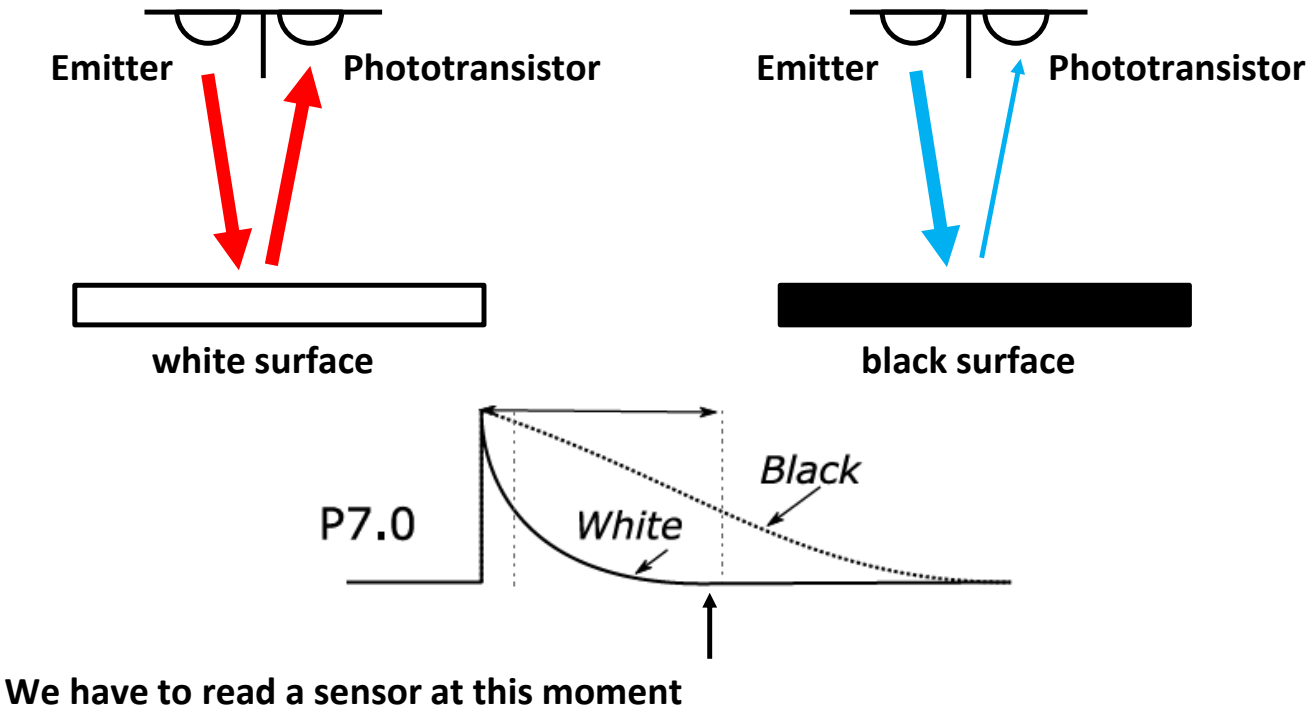
No IR Light



With IR Light

IR Sensor Basic Usage

3) Wait for a while after fully charged



IR Sensor Basic Usage

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

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

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

IR Sensor Basic Usage

5) Turn Off LEDs

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

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

Clock_Delay1ms(10);
```

Tip for Setting Waiting Constant

```
while (1) {
    P5->OUT |= 0x08;
    P9->OUT |= 0x04;

    P7->DIR = 0xFF;
    P7->OUT = 0xFF;

    Clock_Delay1us(10);

    P7->DIR = 0x00;

    int i;
    for (i = 0; i < 10000; i++) {
        sensor = P7->IN & 0x10;
        if (!sensor) {
            printf("Timing Constant : %d\n", i);
            break;
        }
        Clock_Delay1us(1);
    }

    P5->OUT &= ~0x08;
    P9->OUT &= ~0x04;

    Clock_Delay1ms(10);
}
```

Timing Constant : 1713
Timing Constant : 1671
Timing Constant : 1689
...

No Reflection

Timing Constant : 311
Timing Constant : 305
Timing Constant : 310
...

White Surface

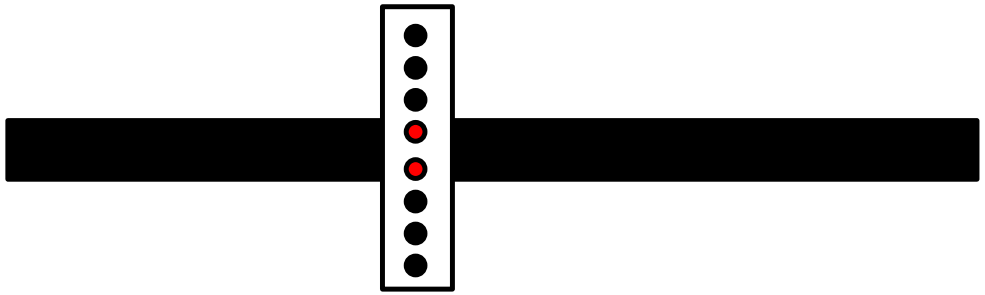
Timing Constant : 790
Timing Constant : 785
Timing Constant : 791

Black Surface

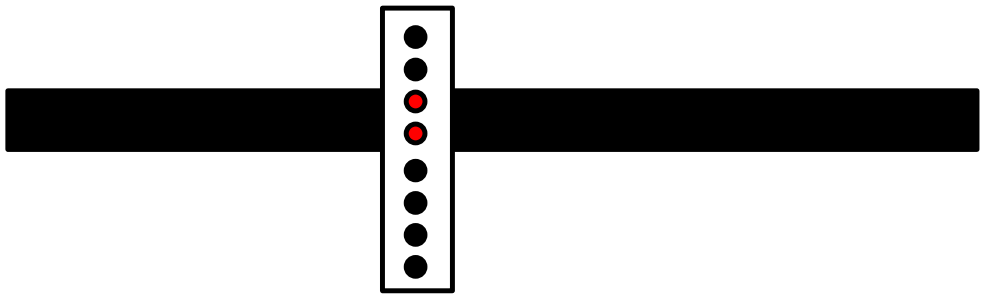
3. IR Sensor Activity

Line Follower – Sensor(not assignment)

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



Turn on LED!



Turn off LED!