

임베디드컴퓨팅

Embedded Computing
(0009488)

Tri-Color LED & Touch sensor

2022년 2학기

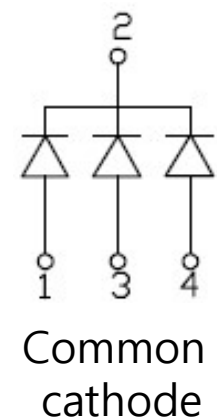
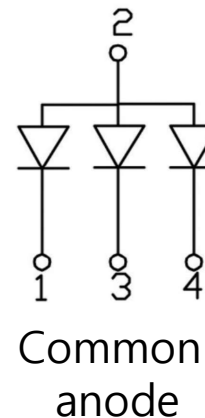
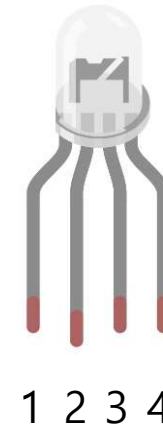
정보기술대학 정보통신공학과

김 영 필

ypkim@inu.ac.kr

Tri-color LED

- An electronic device that outputs various colors by mixing **RGB Values**.
- Various colors?
 - PWM signals are used
- Total of 4 pins, and two types depending on what is connected to **pin 2**.
 - **Common anode (+)** type
 - diode anode commonly connected to pin 2
 - **Common cathode (-)** type
 - diode cathode commonly connected to pin 2



Electrical characteristics of Tri-color LED

- The forward voltage of 2V is applied to the pin 1 (**Red**), and 3.2V to the pin 3 (**GREEN**), and the pin 4 (**BLUE**).

ITEMS	Color	Symbol	Condition	Min.	Typ.	Max.	Unit
Forward Voltage	Red	V_f	$I_f=20\text{mA}$	1.8	2.0	2.2	V
	Green			3.0	3.2	3.4	
	Blue			3.0	3.2	3.4	

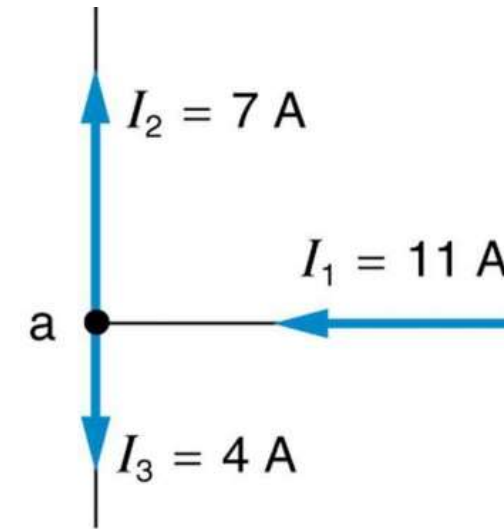
- Each pin is required to connect a resistor with appropriate resistance size, and Kirchhoff's Law and Ohm's Law are used to find the size of the resistance.
 - Ohm's Law
 - V (voltage) = I (current) x R (resistance)

Kirchhoff's Law

- Kirchhoff's 1st Law

- (Current Law)

- The sum of the currents flowing into the junction of the circuit is equals to the sum of the currents flowing out.

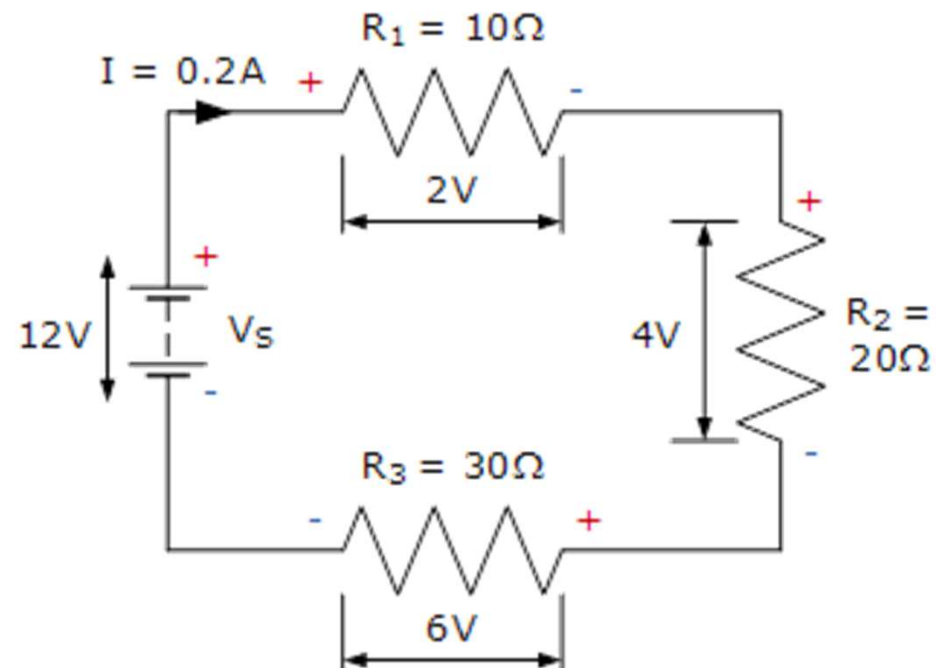


$$I_1 = I_2 + I_3$$

- Kirchhoff's 2nd Law

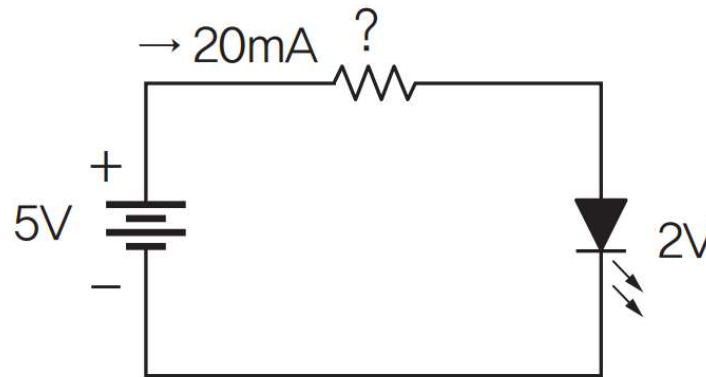
- (Voltage Law)

- The sum of the voltages applied to the circuit is equal to the sum of the voltage drops of each element.



Resistance of Red pin

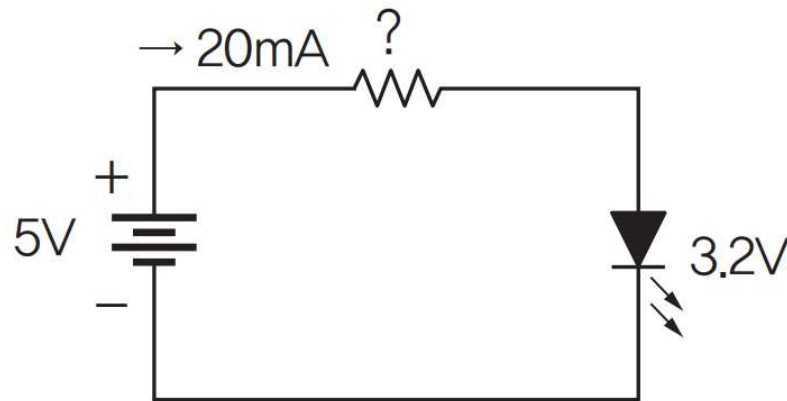
- When a voltage of 5V is applied, a current of about 20mA flows through the circuit.
- A voltage of **2V** is applied to the **pin 1 (Red)**.
- We can draw the circuit diagram to find the size of the resistor to be connected to the pin 1 is as follows.



- By voltage law, the voltage drop on the resistor is $5V - 2V = 3V$
- By ohm's law, the size of resistance $R = \frac{V}{I} =$
 $= 150\Omega$

Resistance of Green / Blue pin

- When a voltage of 5V is applied, a current of about 20mA flows through the circuit.
- A voltage of **3.2V** is applied to **the pin 3 (Green) or pin 4 (Blue)**.
- We can draw the circuit diagram to find the size of the resistor to be connected to the pin 3 or 4 is as follows.

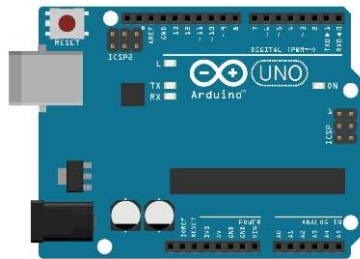


- By voltage law, the voltage drop on the resistor is $5V - 3.2V = 1.8V$
- By ohm's law, the size of resistance $R = \frac{V}{I} =$
 $= 90\Omega$

Control tri-color LED

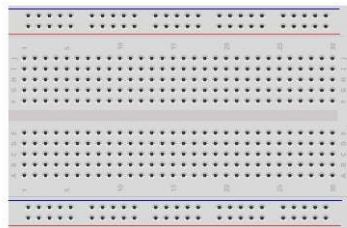
- Let's control tri-color LED to change color
 - Red and green lights sequentially at 1 second intervals.
- Required H/W components

Arduino Uno
board



x1

Breadboard



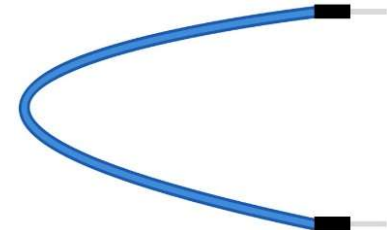
x1

Tri-color LED
(Common cathode)



x1

Jumper cable
(Male-Male)



x4

Resistor
(160Ω)



x1

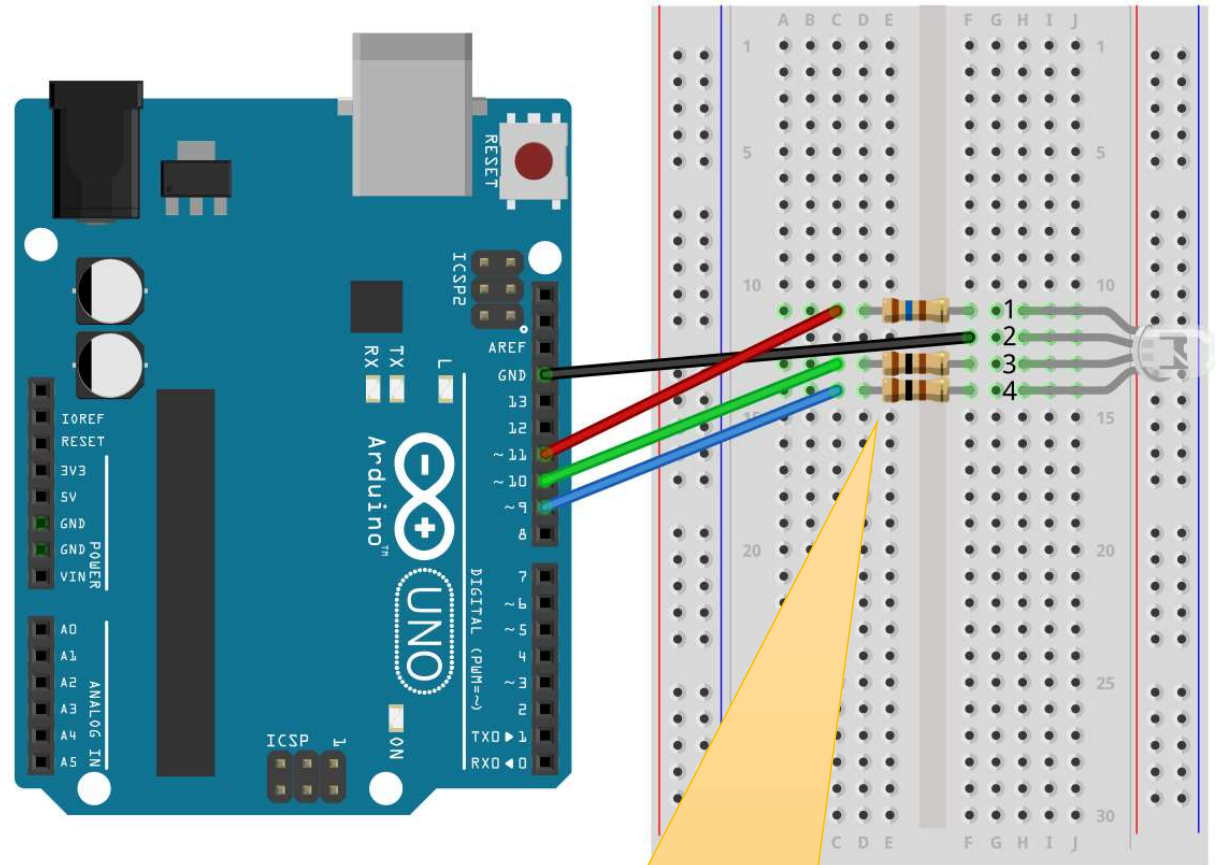
Resistor
(100Ω)



x2

Circuit wiring setup

Tri-color LED	Arduino pin
Red (Pin 1)	PWM 11
GND (Pin 2)	GND
Green (Pin 3)	PWM 10
Blue (Pin 4)	PWM 9



Resistors:
Red pin = 160 ohm
Green and Blue pin = 100 ohm

Basic setup for control tri-color LED

```
#define RED_PIN 11
#define GREEN_PIN 10
#define BLUE_PIN 9

void setup() {
  pinMode(RED_PIN, OUTPUT);
  pinMode(GREEN_PIN, OUTPUT);
  pinMode(BLUE_PIN, OUTPUT);
}
```

digital pin 9, 10, 11 are used for pulse width modulation (PWM)

We can use analogWrite()

set each pin mode to output

Loop code for control tri-color LED

```
void loop() {  
  analogWrite(RED_PIN, 255);  
  analogWrite(GREEN_PIN, 0);  
  analogWrite(BLUE_PIN, 0);  
  delay(1000);  
  analogWrite(RED_PIN, 0);  
  analogWrite(GREEN_PIN, 0);  
  analogWrite(BLUE_PIN, 0);  
  delay(1000);  
}
```

RGB for Red is



RGB for Green is



Let's run!

What happens?

Lab: Discomfort index indicator light

- Changing the color of the three-color LED according to the level of discomfort index (DI)
- Let's define the color as the level of discomfort index
 - extreme ($DI \geq 80$) → red
 - high ($\geq 75, < 80$) → orange
 - normal ($\geq 68, < 75$) → green
 - low (< 68) → blue
- Assume that DI increases from 60 to 90
 - increment it by 2 per second
- Display the current status via serial communication

Basic setup for DI indicator light

```
#define RED_PIN 11
```

```
#define GREEN_PIN 10
```

```
#define BLUE_PIN 9
```

```
void colorChange(int redValue, int  
greenValue, int blueValue);
```

```
void setup() {  
    pinMode(RED_PIN, OUTPUT);  
    pinMode(GREEN_PIN, OUTPUT);  
    pinMode(BLUE_PIN, OUTPUT);  
    Serial.begin(9600);  
}
```

declare and define a RGB color
control function

Function definition for DI indicator light

```
void colorChange(int redValue, int  
greenValue, int blueValue)
```

```
{
```

```
    analogWrite
```

```
    analogWrite
```

```
    analogWrite
```

```
}
```

three function parameters for
red, green, blue value

change a tri-color LED by
analogWrite()

Main loop for DI indicator light

```
void loop() {  
  for(int di=60; di<90; di+=2) {  
    if(di >= 80) {  
      colorChange(  
    } else if(di >= 75) {  
      colorChange(  
    } else if(di >= 68) {  
      colorChange(  
    } else {  
      colorChange(  
    }  
    Serial.print("DI = ");  
    Serial.println(di);  
    delay(1000);  
  }  
}
```

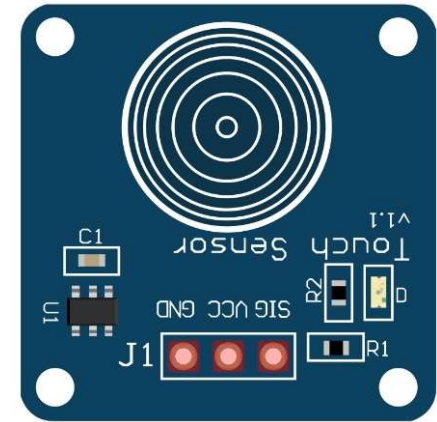
increase DI from 60 to 90
increment step is 2

Check the RGB values for Red,
Orange, Green and Blue

Display current status via serial
monitor

Touch sensor

- A sensor that detects a touch with an electrical signal that changes when a certain part is pressed.
- Consists of three pins:
 - **VCC:** a pin that applies + power (operating voltage = 2.0 ~ 5.5V, connect it to the 3.3V or 5V pin on the Arduino)
 - **GND:** a grounding pin (connected to the GND of the Arduino)
 - **SIG:** a pin that outputs a digital signal depending on the presence or absence of a touch.
 - Touched: HIGH (1) signal
 - Otherwise: LOW (0) signal
 - Connects to the digital pins of the Arduino.

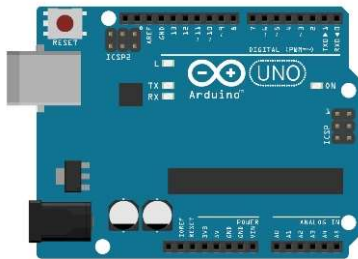


TTP223B

Checking touch event

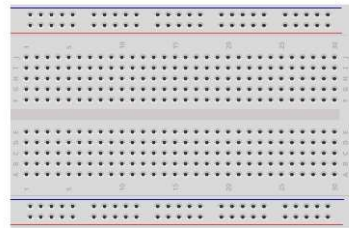
- Let's display the message 'Touched' to the serial monitor when a touch sensor is pressed

Arduino Uno Board



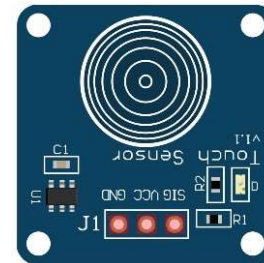
x1

Breadboard (400 pins)



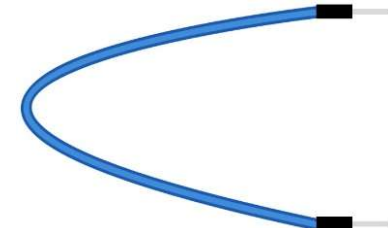
x1

Touch sensor (TTP223B)



x1

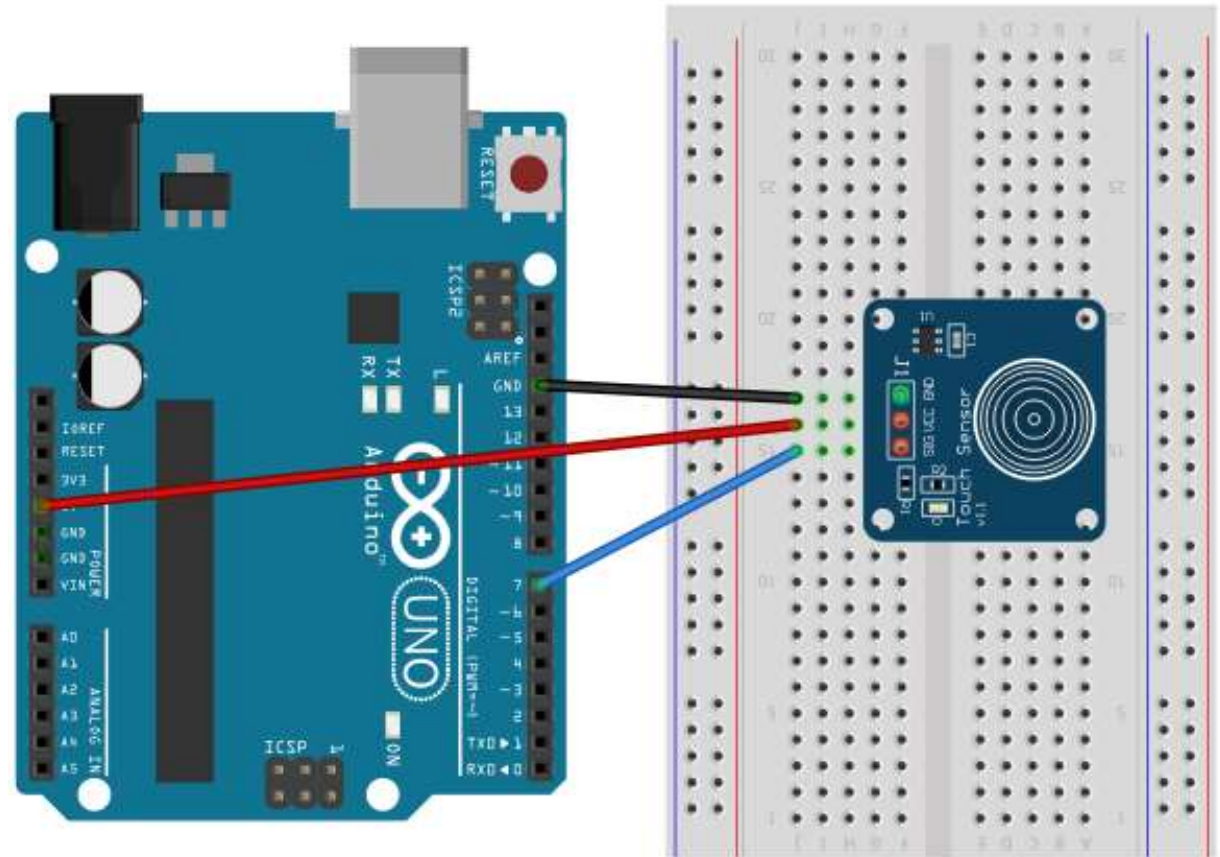
Jumper cable: Male-Male



x3

Circuit wiring setup

Touch sensor	Arduino pin
VCC	5V
GND	GND
SIG	digital 7



Basic setup for checking touch event

```
#define TOUCH_STATE 7
```

```
void setup() {  
  pinMode(TOUCH_STATE, INPUT);  
  Serial.begin(9600);  
}
```

We use digital pin 7 for touch sensor

We read data on digital pin

Main loop for checking touch event

```
void loop() {
```

```
  int touchValue =
```

store data value from the digital pin connected to touch sensor into a variable

```
  if(touchValue == HIGH) {
```

```
    Serial.println("Touched");
```

Display current status via serial monitor

```
    delay(200);
```

Wait for 200 ms to check next touch event

```
  }
```

```
}
```

Assignment: Touch Tri-LED

- Requirements

- Based on the today's examples, write a sketch program as follows.
 - Every second, check the touch event.
 - Every touch, change LED color sequentially: Off, Red, Orange, Green, Blue
 - Display counter and "touched" message via Serial communication
- Write block-type comments in the top of your source code, which includes "your student no., your name, writing date, what you feel about this assignment, etc."

- Results

- (a source code file) sketch source code (*"sketchfilename.ino"*)
- (a short video) a short demo video capture showing how to setup and what to run (max. 1GB file).