한양대학교
HANYANG UNIVERSITY

피지컬 컴퓨팅

3. 피지컬 컴퓨팅 기초(2)

In Kyung, Choi

Ph.D | Assistant Professor

Division of Media, Culture and Design at College of Computing

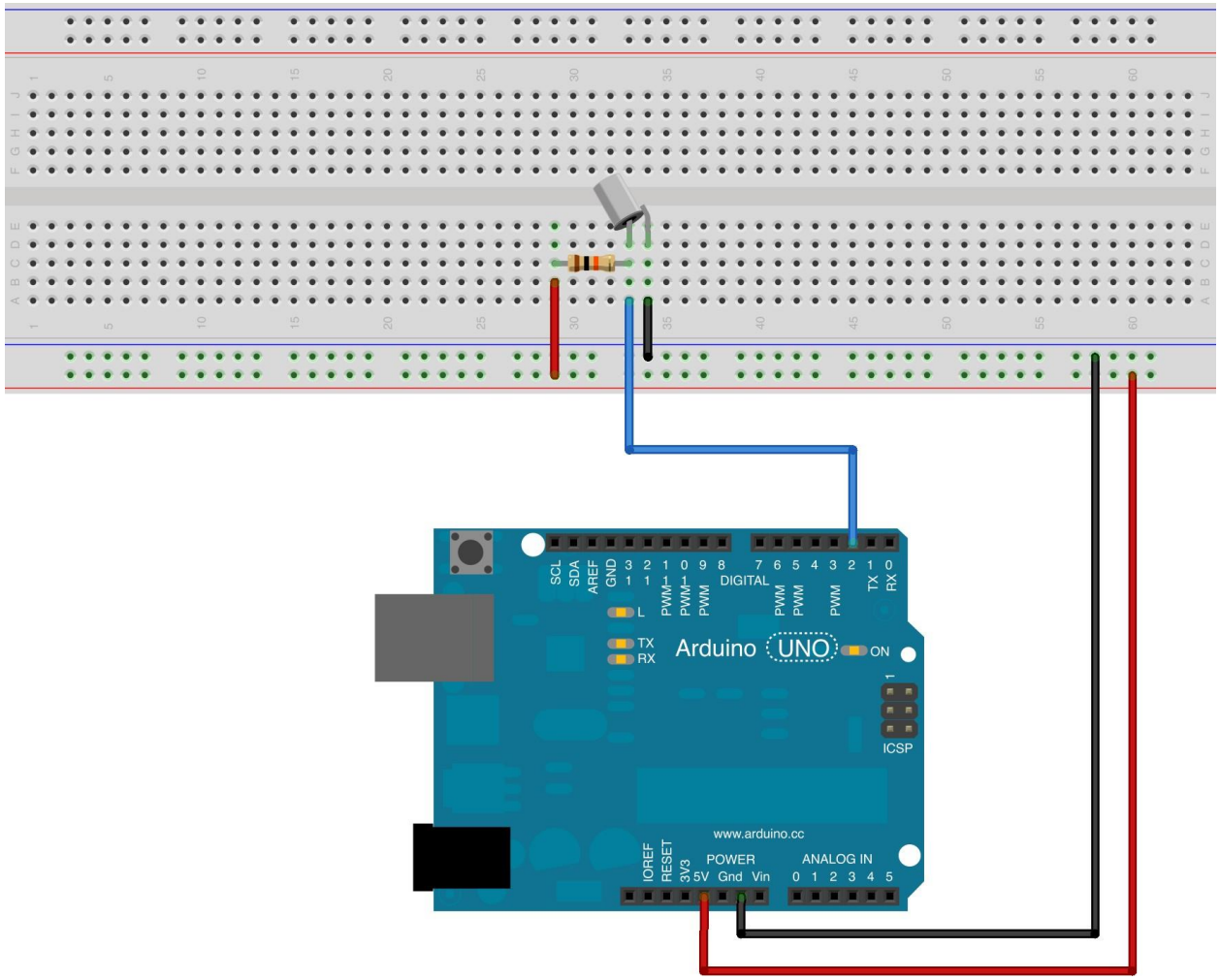
Hanyang University (ERICA)

경기도 안산시 상록구 한양대학로 55 한양대학교 에리카캠퍼스 학연산클러스터 618호

Tel: 031-400-1072| E-mail: ikchoi@hanyang.ac.kr

Digital Input-Output

2) Tilt Switch



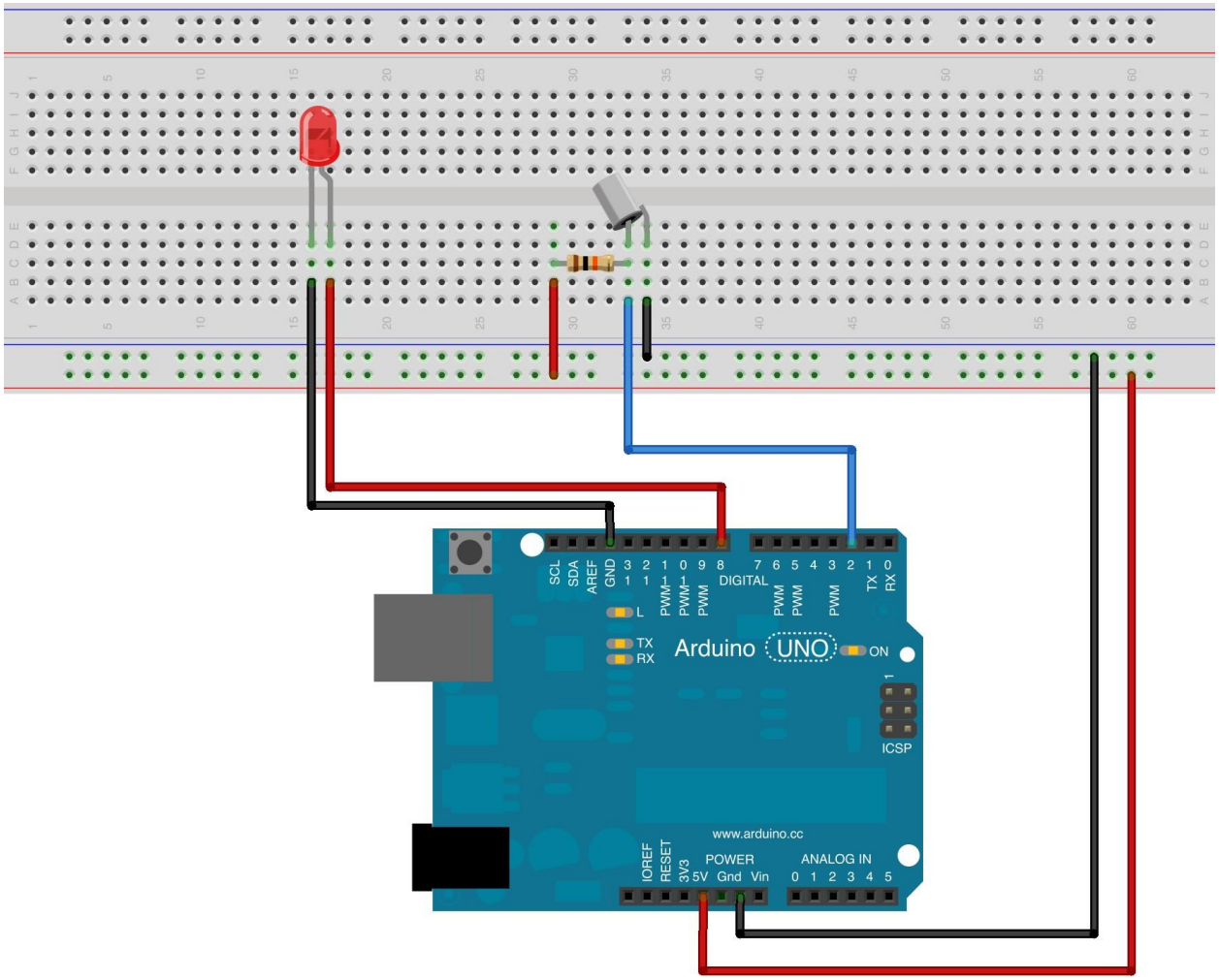
Digital Input-Output

2) Tilt Switch

```
int tilt = 2;
int sensorValue = 0;

void setup() {
  pinMode(tilt, INPUT);

  Serial.begin(9600);
}


void loop() {

  sensorValue = digitalRead(tilt);

  Serial.println(sensorValue);
  delay(1);
}
```

Digital Input-Output

2) Tilt Switch



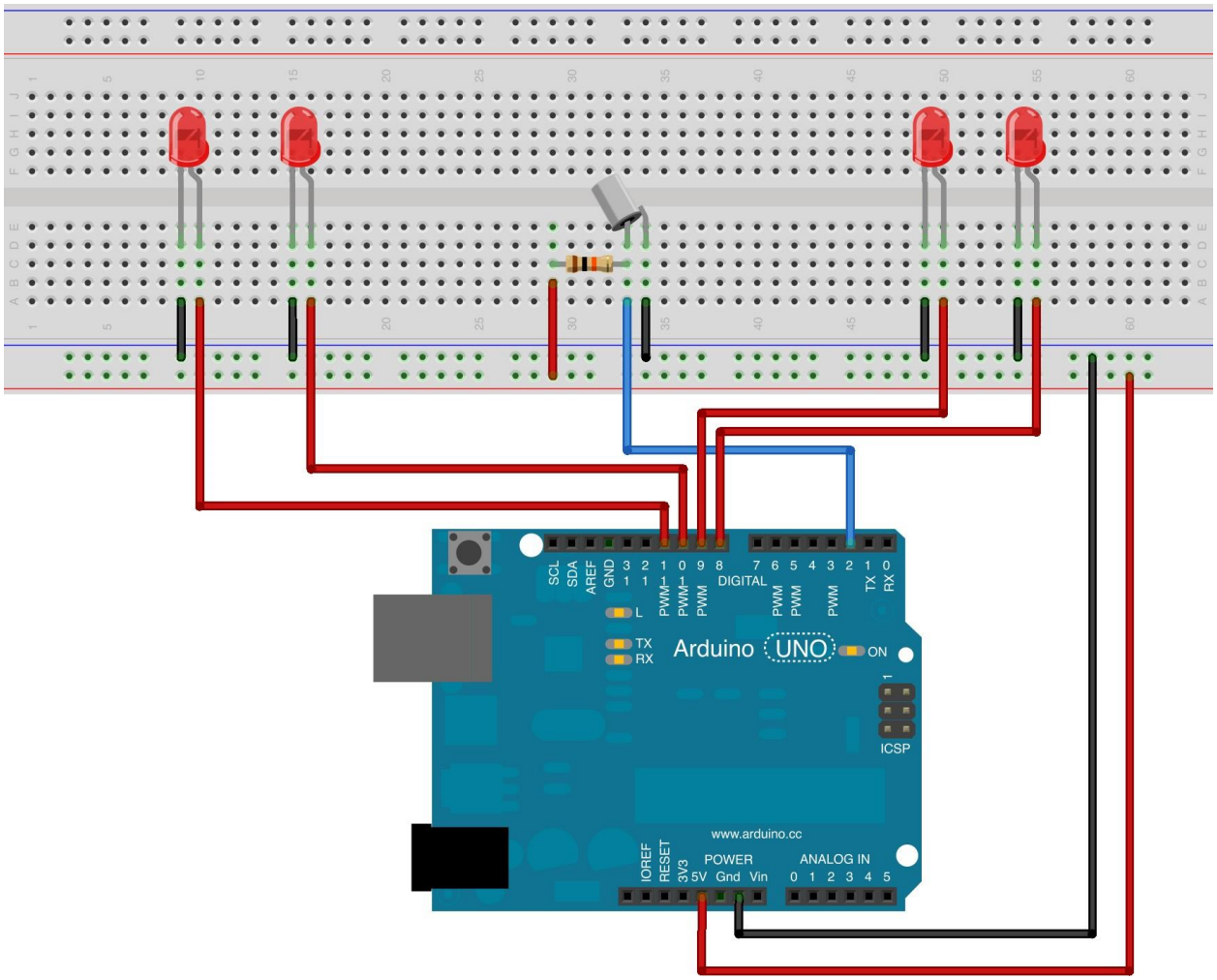
Digital Input-Output

2) Tilt Switch

```
const int Tilt = 2;
const int led1 =  8;

int buttonState = 0;

void setup() {

  pinMode(led1, OUTPUT);
  pinMode(Tilt, INPUT);
}

void loop() {
  buttonState = digitalRead(Tilt);

  if (buttonState == HIGH) {
    digitalWrite(ledPin1, HIGH);
  }
  else {
    digitalWrite(ledPin1, LOW);
  }
}
```

Digital Input-Output

2) Tilt Switch



Digital Input-Output

2) Tilt Switch

```
const int Tilt = 2;
const int led1 =  8;
const int led2 =  9;
const int led3 =  10;
const int led4 =  11;

int buttonState = 0;

void setup() {

  pinMode(led1, OUTPUT);
  pinMode(led2, OUTPUT);
  pinMode(led3, OUTPUT);
  pinMode(led4, OUTPUT);
  pinMode(Tilt, INPUT);
}

void loop() {

  buttonState = digitalRead(Tilt);

  if (buttonState == HIGH) {
    digitalWrite(ledPin1, HIGH);
    digitalWrite(ledPin2, HIGH);
    digitalWrite(ledPin3, LOW);
    digitalWrite(ledPin4, LOW);
  } else {
    digitalWrite(ledPin1, LOW);
    digitalWrite(ledPin2, LOW);
    digitalWrite(ledPin3, HIGH);
    digitalWrite(ledPin4, HIGH);
  }
}
```

< Review +@>

Operators

Arithmetic Operators

- = (assignment operator)
- + (addition)
- - (subtraction)
- * (multiplication)
- / (division)
- % (modulo)

Comparison Operators

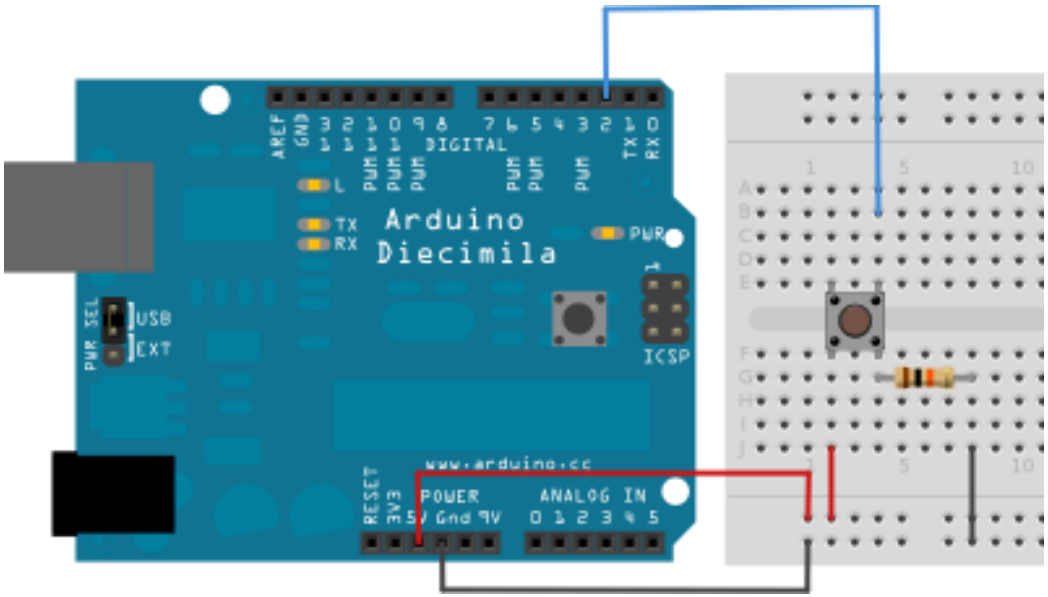
- == (equal to)
- != (not equal to)
- < (less than)
- > (greater than)
- <= (less than or equal to)
- >= (greater than or equal to)

Boolean Operators

- && (and)
- || (or)

< Review +@>

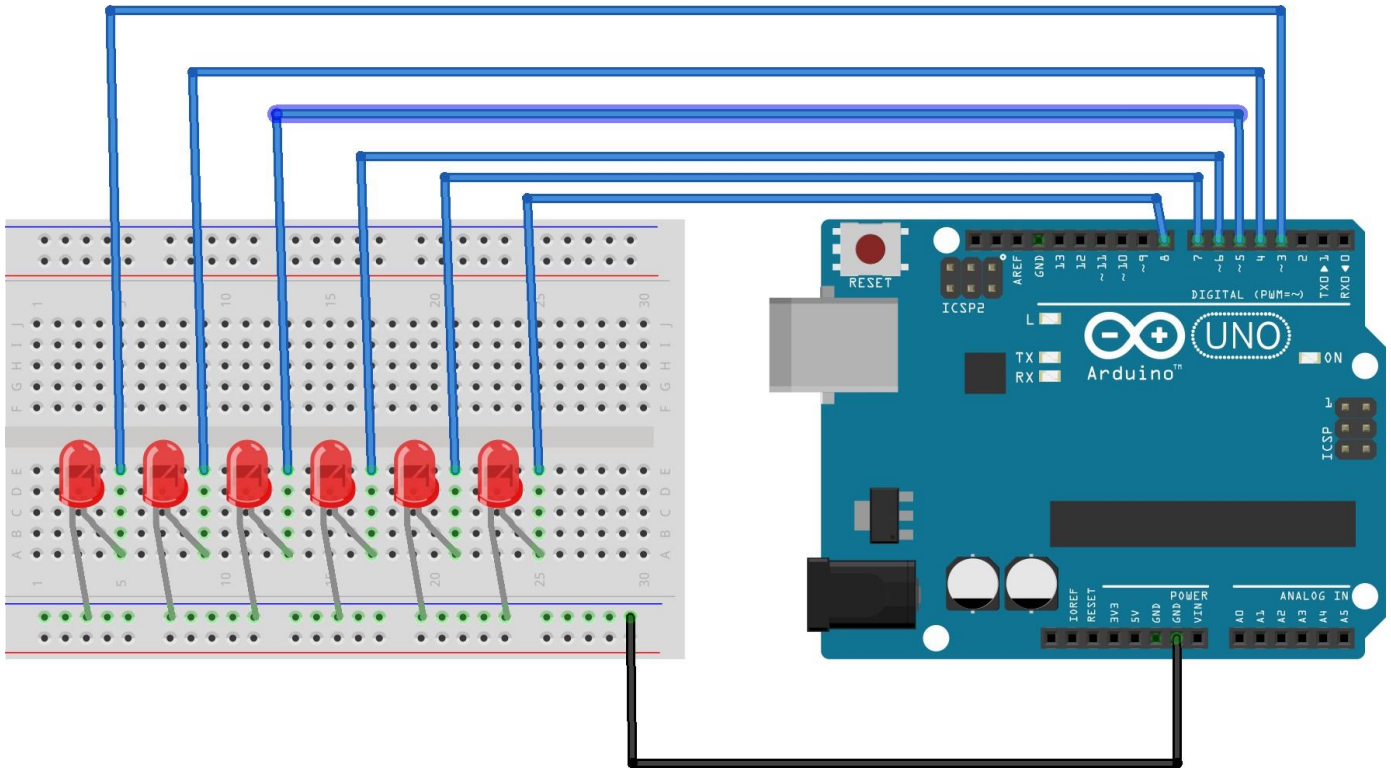
3) Button State 저장, 활용하기




```
int  buttonPin = 2;
int ledPin = 13;

int buttonPushCounter = 0;
int buttonState = 0;
int lastButtonState = 0;

void setup() {

  pinMode(buttonPin, INPUT);
  pinMode(ledPin, OUTPUT);

}


void loop() {

  buttonState = digitalRead(buttonPin);

  if (                                  ) {

    if (                          ) {
      buttonPushCounter++;
    }
    delay(50);

  }

  if                                    ) {
    digitalWrite(ledPin, HIGH);
  } else {
    digitalWrite(ledPin, LOW);
  }

 }


}
```

1. 다수의 조명 제어 with arduino

1–1.다수 led on/off



1. 다수의 조명 제어 with arduino

* 배열(Array)

Creating (Declaring) an Array

All of the methods below are valid ways to create (declare) an array.

```
int myInts[6];   배열을 선언하고 전부 0으로 초기화
int myPins[] = {2, 4, 8, 3, 6};     배열의 크기를 명시하지 않으면 초기화 하는 수만큼 크기를 설정
int mySensVals[6] = {2, 4, -8, 3, 2};  배열을 선언하고 바로 각각 초기화
char message[6] = "hello";
```

It also means that in an array with ten elements, index nine is the last element. Hence:

```
int myArray[10]={9,3,2,4,3,2,7,8,9,11};
     // myArray[9]    contains 11
     // myArray[10]   is invalid and contains random information (other memory address)
```

1. 다수의 조명 제어 with arduino

1–1.다수 led on/off

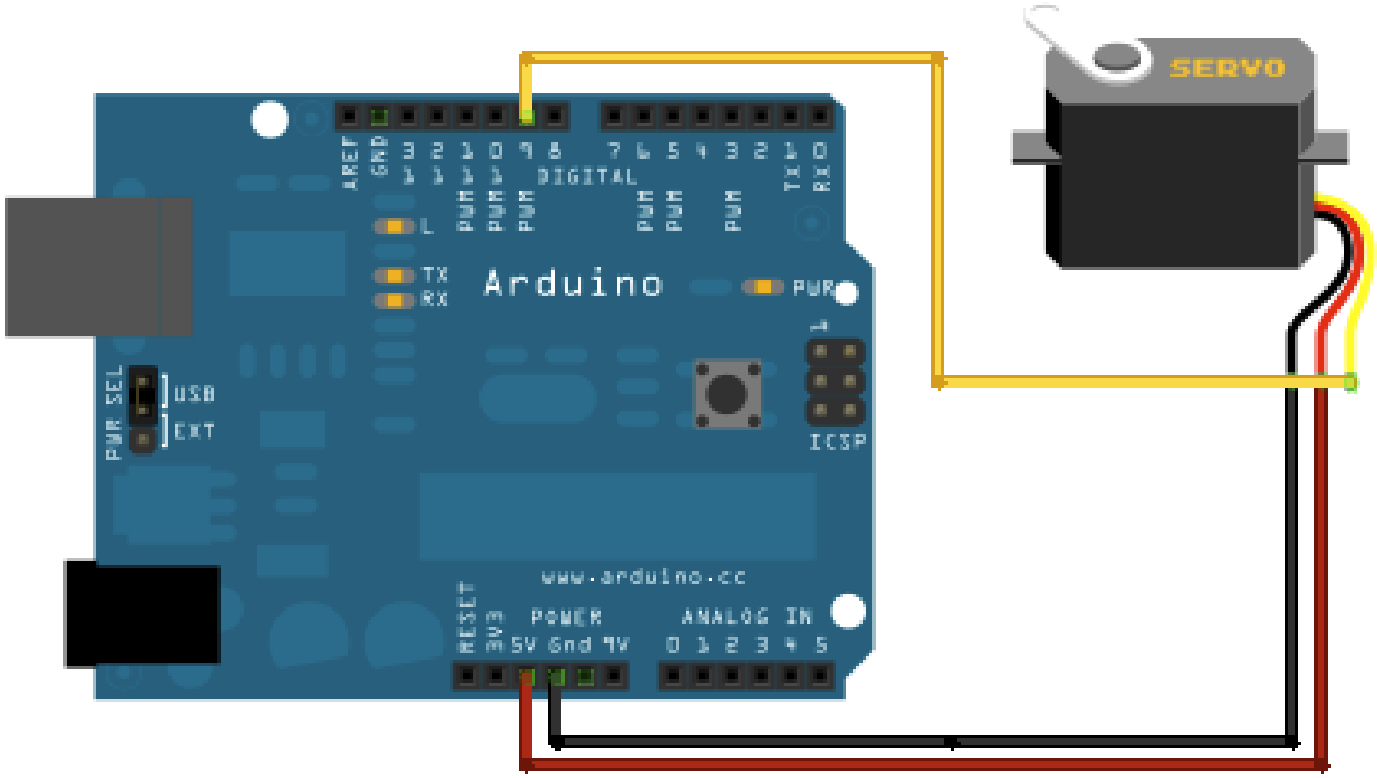
```
int ledPins[] = { 3, 4, 5, 6, 7, 8 };
int pinCount = 6;
void setup() {

  for (int thisPin = 0; thisPin < pinCount; thisPin++) {
    pinMode(ledPins[thisPin], OUTPUT);
  }
}

void loop() {

  for (int thisPin = 0; thisPin < pinCount; thisPin++) {

    digitalWrite(ledPins[thisPin], HIGH);
    delay(100);

    digitalWrite(ledPins[thisPin], LOW);

  }

  for (int thisPin = pinCount - 1; thisPin >= 0; thisPin--) {

    digitalWrite(ledPins[thisPin], HIGH);
    delay(100);

    digitalWrite(ledPins[thisPin], LOW);
  }
}
```

문제1.

- 연결된 LED 모두 아날로그 출력
- For문 활용하여 각 LED가 서서히 dimming되도록 수정

Servo motor

1) 서보모터 움직여보기



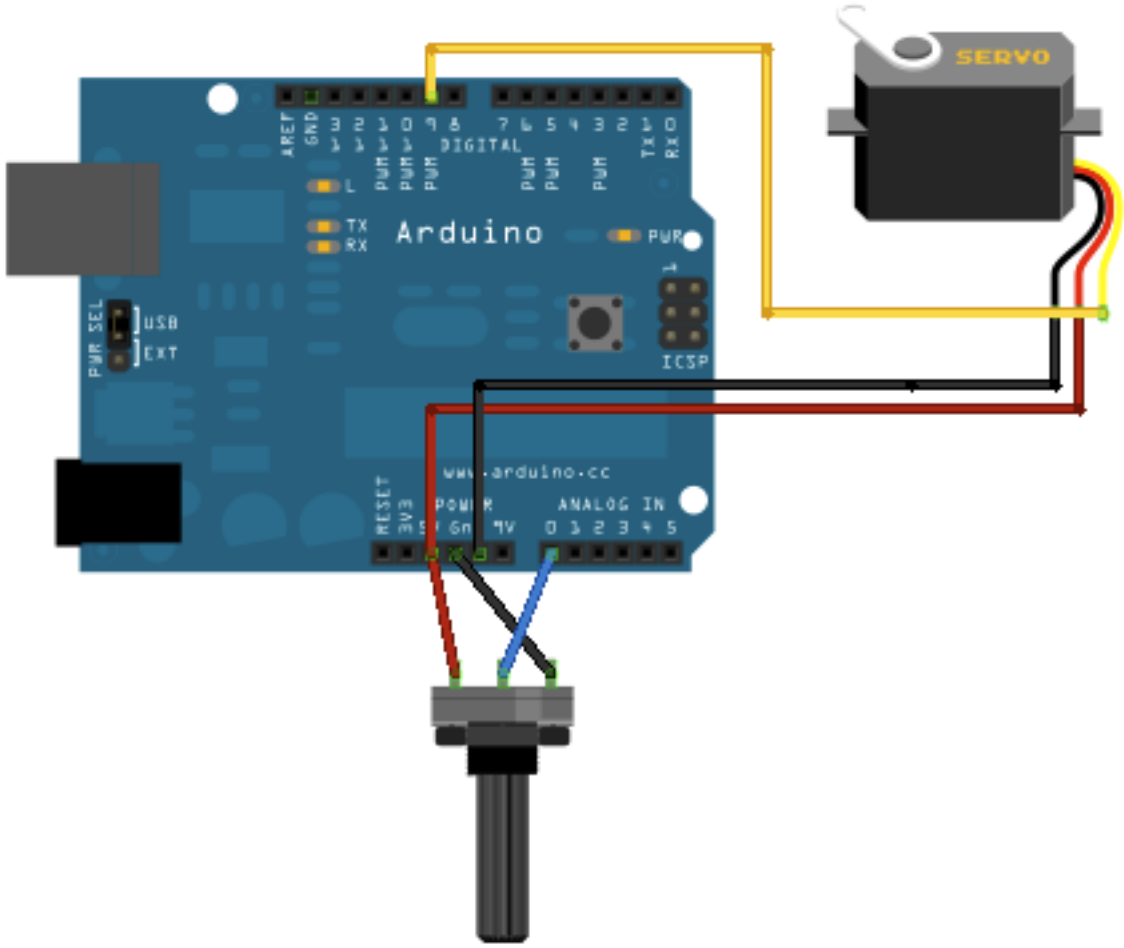
Servo motor

1) 서보모터 움직여보기

```
#include <Servo.h>

Servo myservo;  // create servo object to control a servo
                // a maximum of eight servo objects can be created

int pos = 0;    // variable to store the servo position

void setup()
{
  myservo.attach(9);  // attaches the servo on pin 9 to the servo object
}


void loop()
{
  for(pos = 0; pos < 180; pos += 1)  // goes from 0 degrees to 180 degrees
  {                                  // in steps of 1 degree
    myservo.write(pos);              // tell servo to go to position in variable 'pos'
    delay(15);                       // waits 15ms for the servo to reach the position
  }
  for(pos = 180; pos>=1; pos-=1)     // goes from 180 degrees to 0 degrees
  {
    myservo.write(pos);              // tell servo to go to position in variable 'pos'
    delay(15);                       // waits 15ms for the servo to reach the position
  }
}
```

Servo motor

2) 볼륨센서로 서보모터 움직여보기



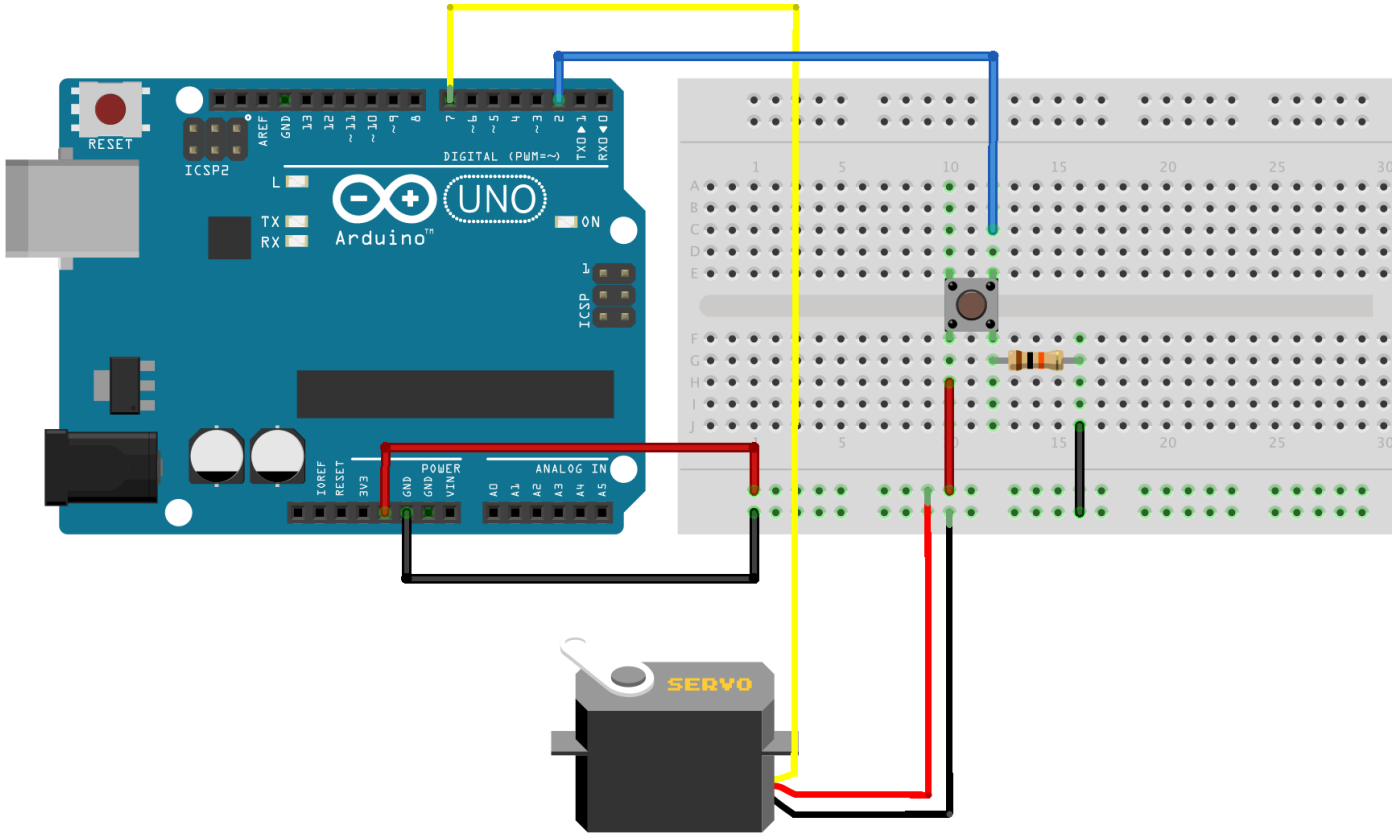
Servo motor

2) 볼륨센서로 서보모터 움직여보기

```
#include <Servo.h>

Servo myservo;  // create servo object to control a servo

int potpin = 0;  // analog pin used to connect the potentiometer
int val;    // variable to read the value from the analog pin

void setup()
{
  myservo.attach(9);  // attaches the servo on pin 9 to the servo object
}

void loop()
{
  val = analogRead(potpin);
  val = map(val, 0, 1023, 0, 179);
  myservo.write(val);
  delay(15);
}
```

1. Servo motor

3) 버튼으로 특정각도 지정하기



1. Servo motor

3) 버튼으로 특정각도 지정하기

```
#include <Servo.h>

Servo myservo;

int potpin = A0;
int val;
int btn = 7;

void setup() {
  myservo.attach(9);
  pinMode(btn, INPUT);
}

void loop() {
  val = digitalRead(btn);

  if (val == 1) {
    myservo.write(10);
  }
  else {
    myservo.write(100);
  }

  delay(15);
}
```

문제2.

- 빛센서 연결
- 핸드폰의 라이트가 들어오면 열리고,
- 라이트가 꺼지면 다시 닫히는 렌즈조리개를 제어하는 모터 구현