

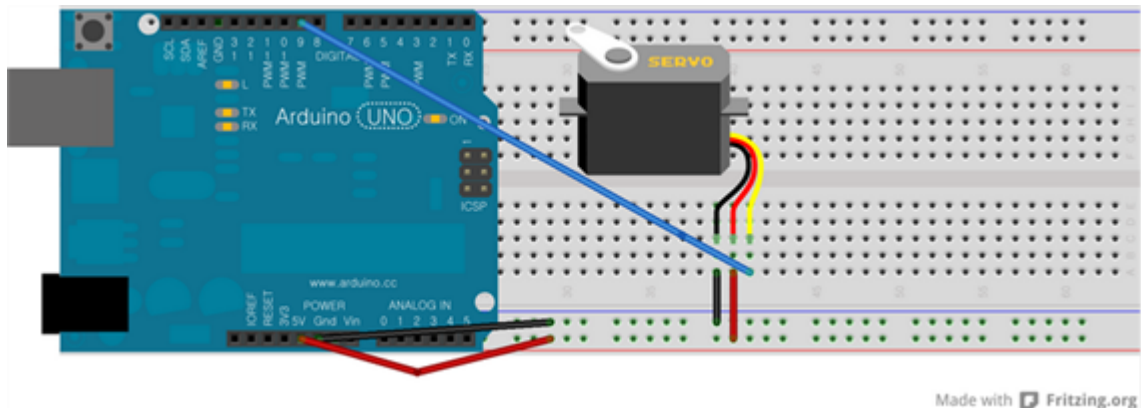
## 6. 아날로그 출력

### 가. 서보모터 제어

DC 모터는 단순히 전류를 흘려주면 전류가 공급되는 동안 모터는 회전을 하게 된다. 이에 반해 서보모터는 지속적으로 회전하는 것이 아니라, 자신이 원하는 위치로 모터를 이동시킬 수 있어 보다 정밀한 제어가 가능하게 된다. 서보 모터를 한쪽 끝에서 나머지 끝까지 이동시키고 다시 그 반대 동작을 지속적으로 하게 하는 작업이다.

#### 1) 하드웨어 연결하기

서보모터의 전원(빨간색), GND(검은색)을 아두이노의 전원과 GND에 연결하고, 나머지 한 선을 아두이노의 9번 포트에 연결한다. 대부분의 서보모터는 일정한 각도 이상으로 회전을 할 수는 없다. 본 프로젝트에서 사용하는 모터는 180도까지만 회전을 하며, 아두이노의 PWM을 통해 각도를 조정할 수 있다. delay()함수를 같이 사용하여 회전 속도를 조절하는 것도 가능하다.



## 2) 프로그램 작성

```
#include <Servo.h>

int servoPin = 9;
Servo servo;//모터 객체 선언
int angle = 0;

void setup() {
    servo.attach(servoPin);
}

void loop() {
    for (angle = 0; angle < 180; angle++) {
        servo.write(angle);
        delay(15);
    }
    for (angle = 180; angle > 0; angle--) {
        servo.write(angle);
        delay(15);
    }
}
```

모터를 컨트롤 하기 위해 복잡한 코드를 작성하는 대신 Servo.h 라이브러리를 이용하여 보다 편리하게 모터를 제어할 수 있다.

`servo.attach(servoPin);`

모터가 연결된 핀을 설정하여 준다. 서보 모터는 PWM 제어를 사용하기에 보드에서 PWM을 지원하는 핀에 연결하여 준다.

`servo.write(angle);`

모터가 회전할 각을 입력해준다.

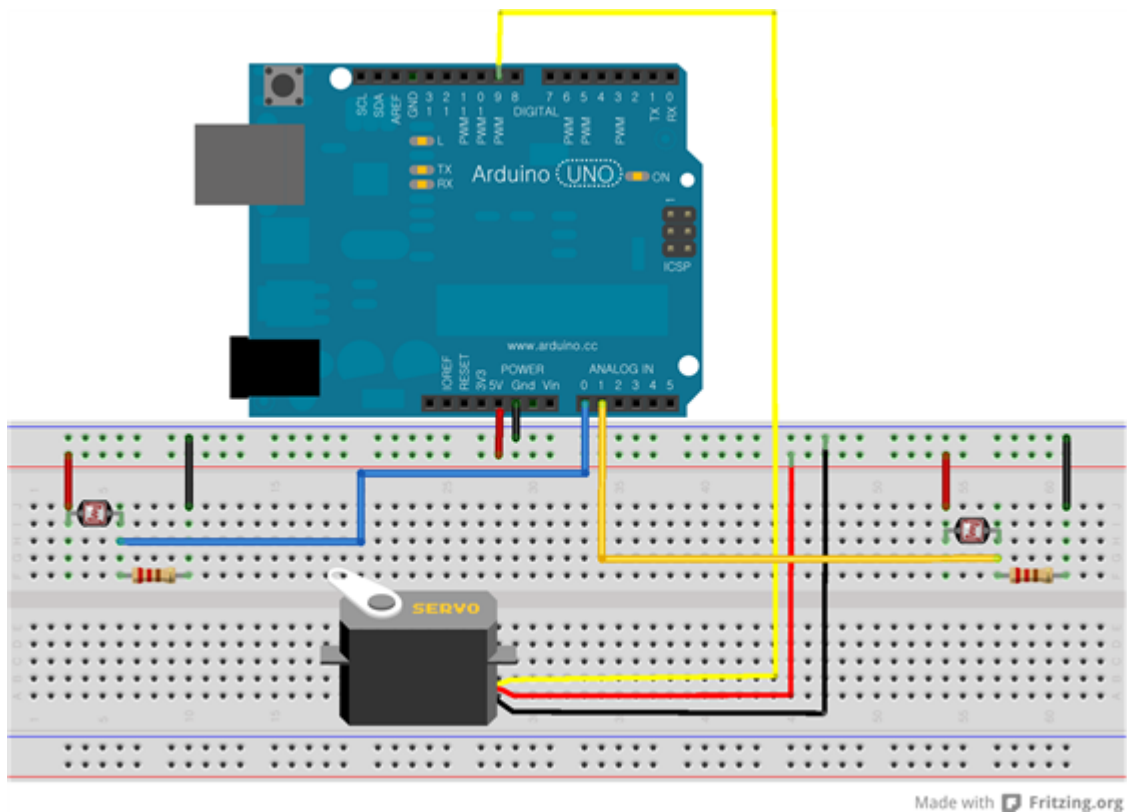
위 코드는 2개의 반복문을 통해 한쪽 0도에서 179도까지 그리고 180도에서 1도 까지 회전하는 코드이다.

### 3) 빛을 추적해보기

서보 모터의 기본적인 사용 방법을 이전 수업에서 학습하였다. 빛 센서 2개를 브레드 보드의 양단에 연결하고 서보모터를 이용하여 양쪽의 빛의 차이를 계산하여 해당 방향으로 모터를 회전 시키는 프로젝트이다.

#### 가) 하드웨어 연결하기

서보 모터의 전원(빨간색), GND(검은색)을 아두이노의 전원과 GND에 연결하고, 빛 센서 2개를 브레드 보드의 양단에 그림과 같이 설치하며, 10K옴 저항 2개를 직렬로 연결 시켜 준다. 양단의 저항에 걸리는 전압을 A0, A1에 연결하여 analogRead 함수를 통하여 그 값을 읽어준다.



## 나) 프로그램

```
#include <Servo.h>

int left = A0;
int right = A1;
int sensorValue = 0;
Servo myservo;
int value;
int center = 100;

void setup() {
    myservo.attach(9);
}

void loop() {
    int left = analogRead(A0); //왼쪽 빛 센서
    int right = analogRead(A1); //오른쪽 빛 센서
    value = (left - right) / 10;
    if (value == 0)
        myservo.detach();
    else
        myservo.attach(9);

    if (value > 10)
        value = -70;
    else
        value = 70;
    myservo.write(center+value);
    delay(10);
}
```