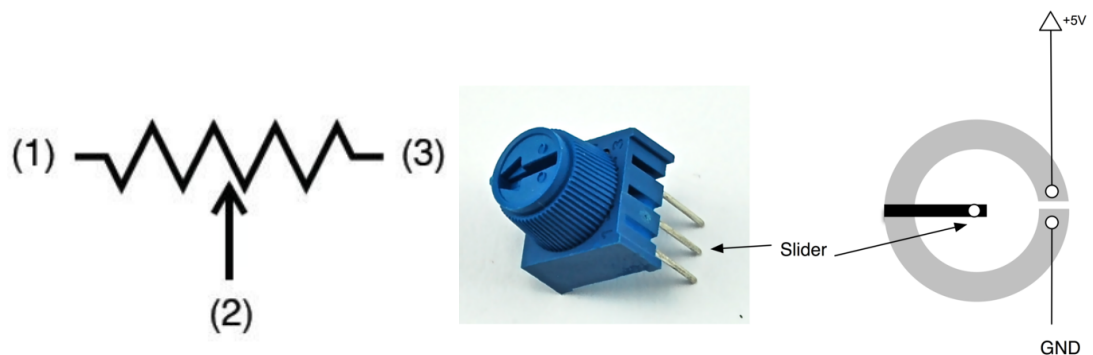


다. 가변저항

이번 작업에서는 가변저항을 사용하여 아날로그 입력에 따라 LED의 점멸 주기를 변경해 보자.

1) LED 점멸주기 변경하기

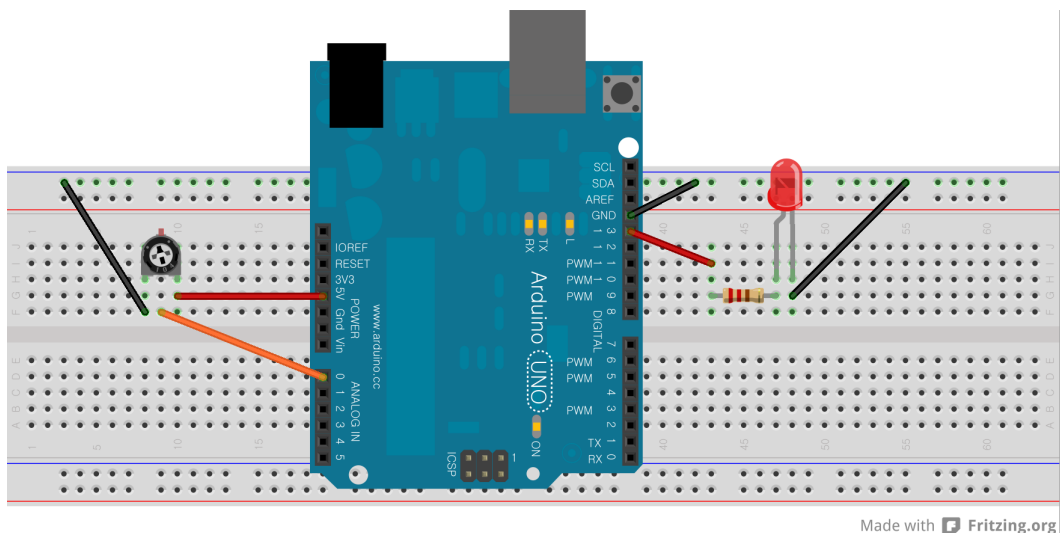
가변저항은 3개의 핀으로 구성되어 있으며, R1-R3의 저항은 동일하며 이는 가변 저항 전체의 저항 값이 된다. 보통 중앙에 위치한 2번은 저항을 회전시킴에 따라 일종의 와이퍼가 좌우로 움직여 저항의 변화를 발생시키게 된다. 즉, R1-R2와 R2-R3의 저항 값들이 변하며 전체 저항의 값은 일정하게 된다.



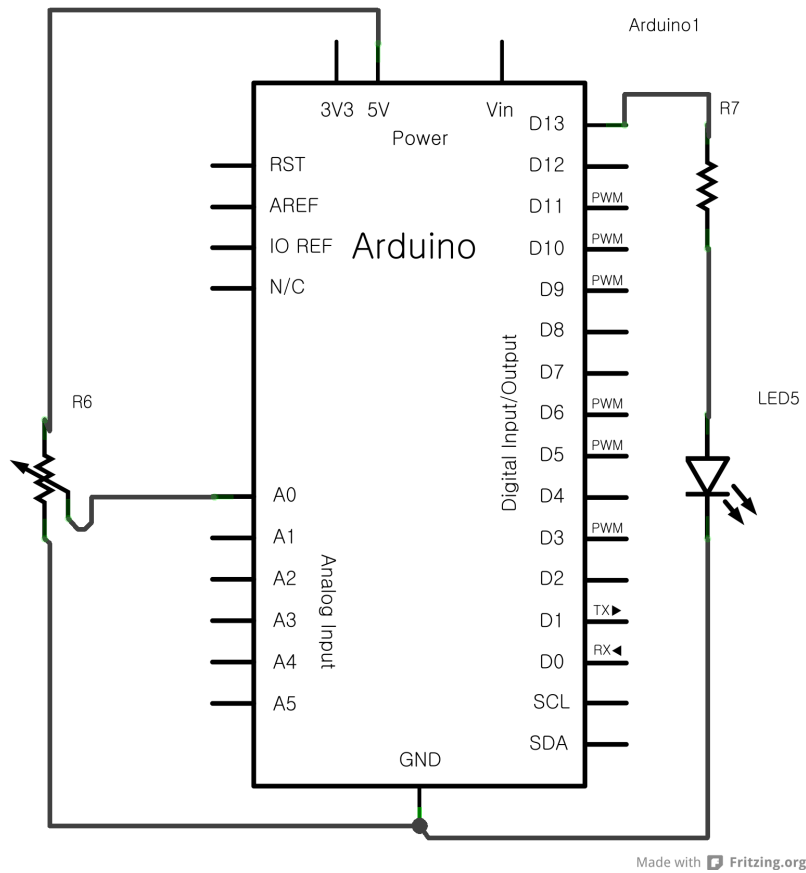
필요부품

아두이노, 가변저항(10Kohm), LED, 저항

브레드보드



스케메틱



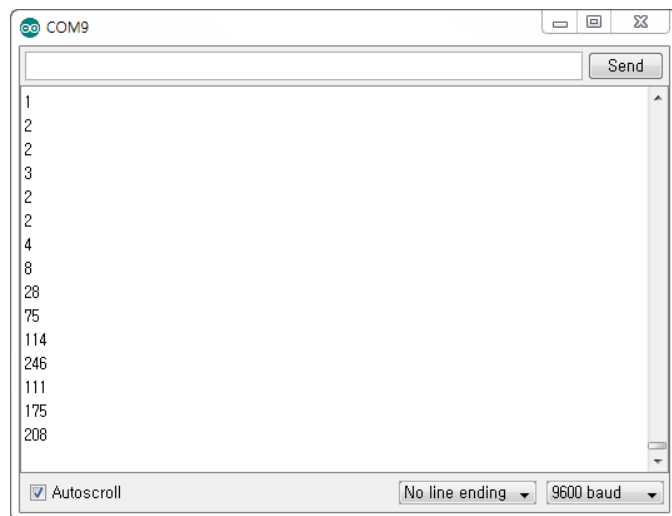
프로그램

- 가변저항에서 읽은 저항 값에 따라서 LED의 점멸 주기를 변경 시킨다.

```
int analogPin = 0;
int led = 13;
int val = 0;

void setup() {
  Serial.begin(9600);
  pinMode(led, OUTPUT);
}

void loop() {
  val = analogRead(analogPin);
  Serial.println(val);
  digitalWrite(led, HIGH);
  delay(val);
  digitalWrite(led, LOW);
  delay(val);
}
```



2) 프로세싱으로 시각화

프로세싱은 시리얼 포트를 통해 아두이노가 보내온 가변저항의 값을 이용하여 원을 그리고 해당 원의 크기를 가변 저항의 값으로 그린다. 따라서, 큰 저항 값이 전송되었을 때는 원의 지름이 커지고 반대인 경우에는 작아지게 된다.

가) 프로그램

- 가변저항에서 읽은 저항 값에 따라서 LED의 점멸 주기를 변경 시킨다.

```
import processing.serial.*;

int lf = 10; // Linefeed in ASCII
String myString = null;
Serial myPort; // Serial port you are using
float num;

void setup() {
  background(255,255,255);
  size(600,600);
  println(Serial.list());
  myPort = new Serial(this, Serial.list()[4], 9600); //자신의 포트에 맞는 번호 선택
}

void draw() {
  while (myPort.available() > 0) {
    myString = myPort.readStringUntil(lf);
    if (myString != null) {
      num=float(myString); // Converts and prints float
      background(255,255,255);
      fill(#ffcb01);
      ellipse(width/2,height/2,num/2,num/2);
    }
  }
  myPort.clear();
}
```

나) 실행결과

- 아두이노의 시리얼 모니터 창을 닫아야 한다. 한 번에 하나의 프로그램만이 시리얼 모니터를 이용할 수 있다. 가변저항을 움직임에 따라 프로세싱은 서로 다른 크기의 원을 표시해주게 된다.

