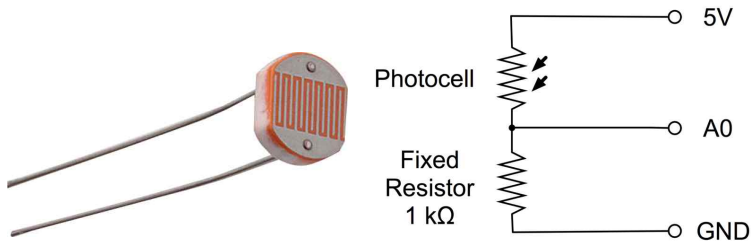


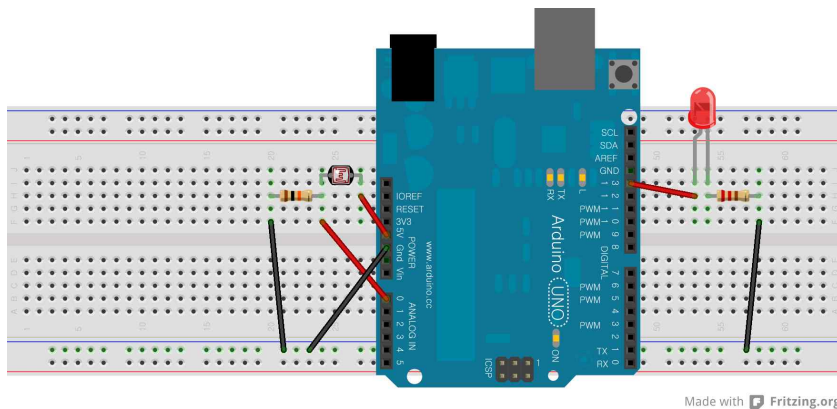
라. 조도 센서 입력

주위의 빛의 밝기를 재는 기본적인 방법은 포토레지스터를 이용한다. 빛의 세기가 증가하면 전기 저항이 낮아지는 특성으로 빛 의존성 저항(Light-Dependent Resistor)이라고 불린다. 조도 센서와 저항을 직렬로 연결하고 저항에 걸리는 전압의 변화량을 읽어 들여 빛의 밝기를 판단한다. 빛이 밝으면 LDR의 저항 값이 낮아지므로 A0의 입력은 높아지며, 어두워지면 저항 값이 증가하므로 A0의 입력은 낮아진다. 아래 그림에서 10킬로옴의 저항을 사용한다.

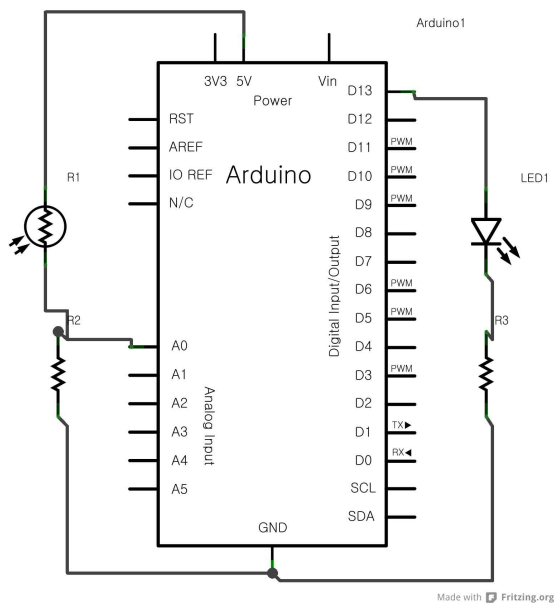


브레드보드

- LDR 센서 값을 읽어 들여 빛이 밝으면 LED를 어둡게, 어두우면 밝게 켜는 작업을 해보자.
또한 LDR 센서 값을 프로세싱에서 표현해보자.



스케메틱



프로그램

```
int sensorPin = 0;
int ledPin = 13;
int sensorValue = 0;

void setup() {

  pinMode(ledPin, OUTPUT);
  Serial.begin(9600);
}

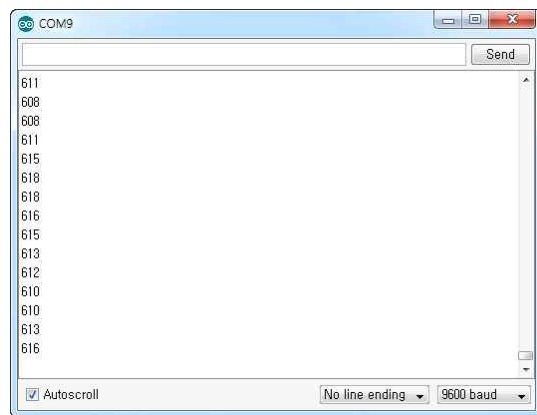
void loop() {

  sensorValue = analogRead(sensorPin);

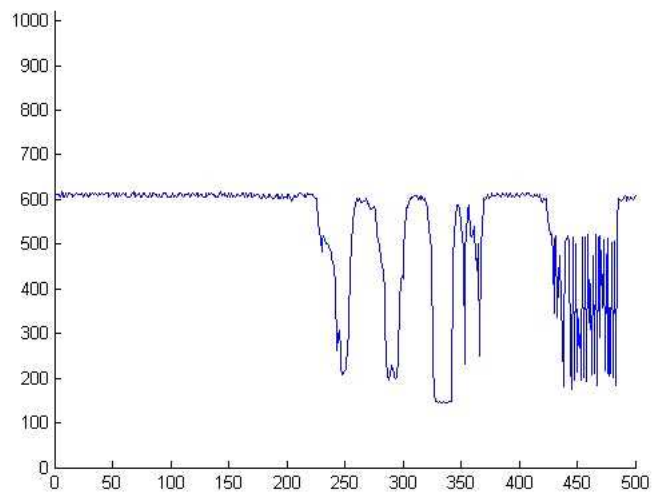
  Serial.println(sensorValue);
  int mapValue = map(sensorValue, 0,
1023, 255, 0);

  analogWrite(ledPin, mapValue);
  delay(500);

}
```



Graph using Matlab



1) 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.*;

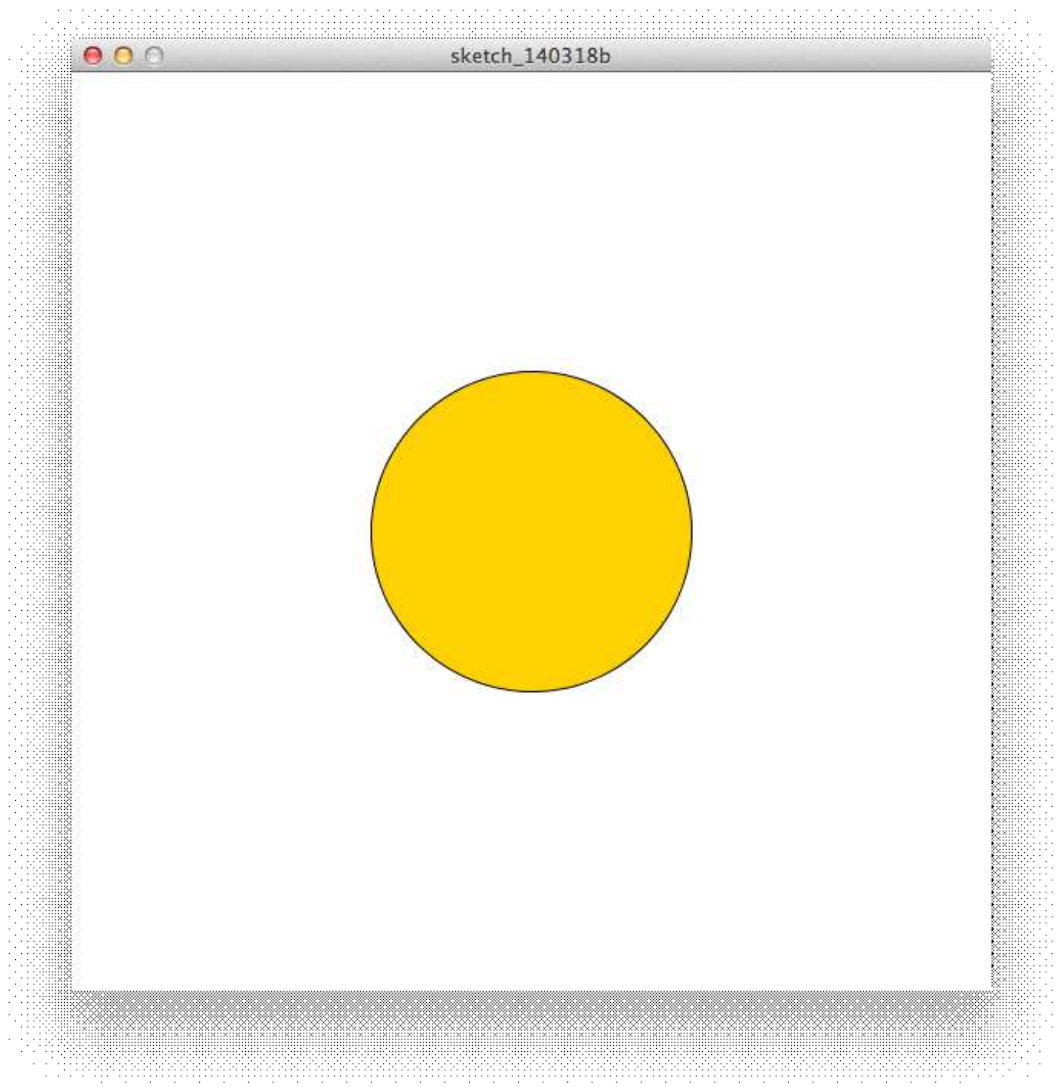
int lf = 10; // Linefeed in ASCII
String myString = null;
Serial myPort; // Serial port you are using
float num;

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. 빛 센서의 밝기 변화에 따라 프로세싱은 서로 다른 크기의 원을 표시해주게 된다.



2) 엑셀을 이용한 데이터의 표현

아두이노가 보내오는 데이터를 처리하기 위해서는 컴퓨터 상에서 다양한 프로그램을 이용할 수 있다. 엑셀은 숫자 데이터의 처리에 적합하며 각종 함수를 이용하여 데이터 처리 및 표 작업등에 유용하게 사용할 수 있다. 이번 작업에서는 엑셀을 이용하여 아두이노가 보내온 조도 센서 데이터의 처리를 한다. 이 작업은 앞으로의 다양한 추가 작업에 공통적으로 이용될 수 있다.

가) 프로그램

parallax daq는 아두이노와 엑셀을 연결할 수 있는 프로그램으로 아두이노 코드상에서 엑셀에 어떠한 데이터를 출력할지 정하며 컴퓨터와 아두이노의 시리얼통신을 이용한다. 실행 파일을 클릭하여 parallax를 설치하며, 매크로가 포함된 엑셀파일을 이용하여 아두이노의 데이터를 처리한다.



```

int x = 0;
int row = 0;
void setup() {
  Serial.begin(128000); // opens serial port, sets data rate to 9600 bps
  Serial.println("CLEARDATA");
  Serial.println("LABEL,Time,x,sin(x)");
}
void loop() {
  Serial.print("DATA,TIME,"); Serial.print(x); Serial.print(","); Serial.println(sin(x*PI/180));
  row++;
  x++;
  if (row > 360)
  {
    row=0;
    Serial.println("ROW,SET,2");
  }
  delay(100);
}

```

Serial.println("CLEARDATA");는 데이터를 보내기 전 엑셀에 있는 모든 데이터를 지우겠다는 것이며 Serial.println("LABEL,Time,x, sin(x));는 엑셀의 맨 윗줄에 라벨을 다는 것입니다. Loop 함수에서는 DATA, TIME 로 시작하는 다음 줄에 실제 데이터가 들어오게 된다. 데이터의 행의 수가 360이 넘으면 행의 값을 2로 하여 다시금 데이터가 입력되게 해주는 코드이다.

