

Open Source

무료로 사용하고 수정하고 재배포 할 수 있는 소프트웨어 및 하드웨어 자원, 주로 소프트웨어 및 하드웨어,

Electronics

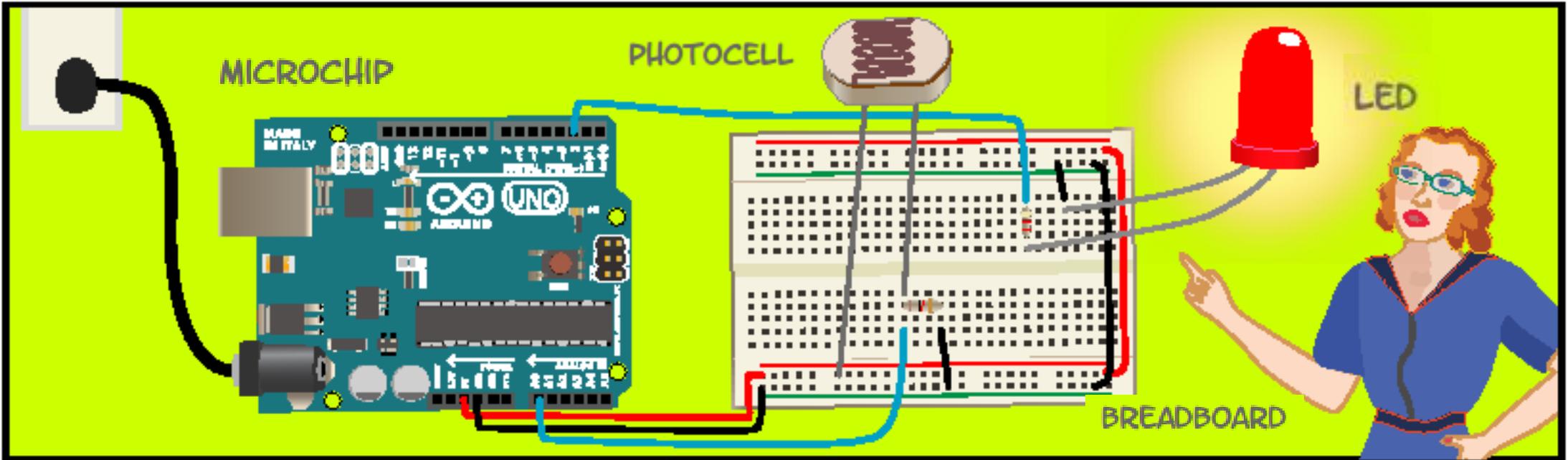
전자의 흐름으로 만들어진 전기 에너지를 이용하는 기술

Prototype

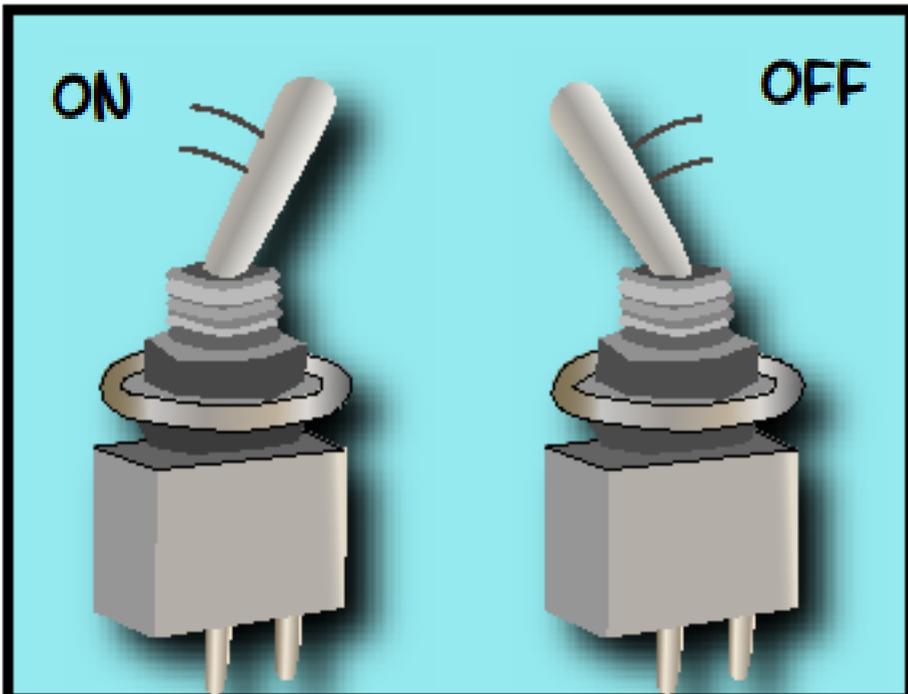
다른 것들에 대한 기준이나 표준이 될 수 있는 원래의 형태

Platform

다른 소프트웨어가 실행될 수 있는 소프트웨어 프레임 워크와 하드웨어 아키텍처



아두이노에는 여러분의 프로그램을 돌릴 수 있는 매우 작은 컴퓨터인 마이크로칩(MICROCHIP)이 들어 있습니다. 여러분은 아두이노에 센서를 연결할 수 있으며 그것을 이용해 상태를 측정할 수 있습니다.(예를 들어 방이 얼마나 밝은지를 측정하는 것처럼요) 또한 이러한 상황에 따라 다른 물체를 어떻게 반응하게 할 것인가도 제어할 수 있습니다.(예를 들어 방이 어두워지면 LED가 켜지게 하는 것이죠.)

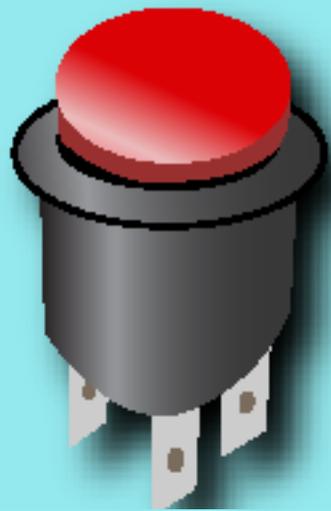


또는 스위치를 누르는 것과 같이 간단한 동작에 따라 반응하게 할 수도 있습니다.

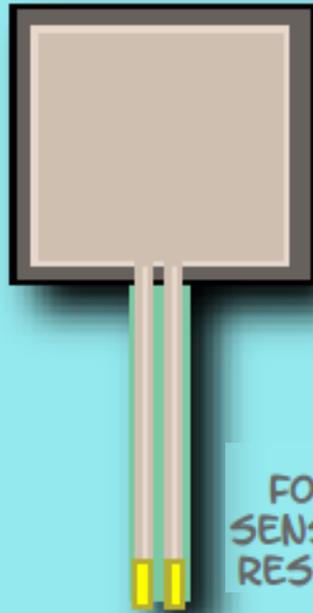


마우스는 데스크탑 컴퓨터를 위한 공통 입력 장치이며 모니터는 일반적인 출력 장치예요.

마이크로컨트롤러는 컴퓨터처럼 입력과 출력을 사용합니다. 입력이 사용자와 환경으로부터 정보를 수집하는 동안 출력은 수집된 정보로 어떤 일을 수행합니다.



MOMENTARY SWITCH



FORCE SENSITIVE RESISTOR

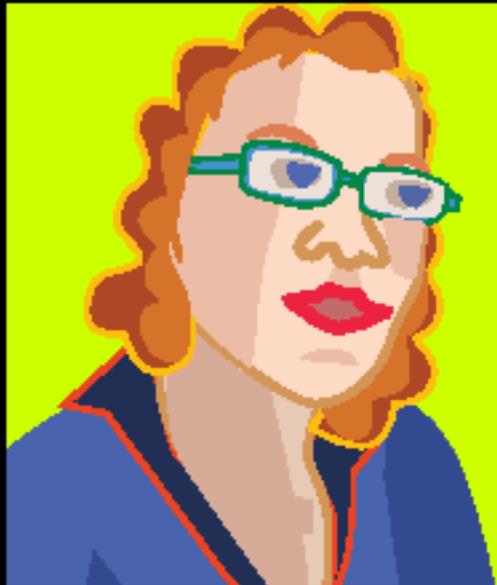
스위치 또는 센서는 아두이노의 입력장치가 될 수 있습니다.



DC MOTOR



우리가 키고 끄고 조절하고자 하는 모든 물체는 출력장치가 될 수 있으며, 그것은 모터 또는 심지어 컴퓨터가 될 수도 있습니다.



디지털과 아날로그의 입출력 차이는 무엇일까요?

입력과 출력은 디지털이나 아날로그가 될 수 있습니다. 디지털 정보는 참과 거짓으로 이루어진 이진법이며, 아날로그 정보는 연속적이며 범위 값을 가질 수 있습니다.

디지털 정보는 유한하고 별개의 값이에요. 모든 정보는 1 또는 0, on 또는 off 와 같은 두 가지 상태로 묘사되죠.

아날로그 정보는 지속적인 성격을 특징으로 가집니다. 가능한 값의 무한한 수를 가질 수 있어요.



스위치는 디지털 입력이고 센서는 아날로그 입력입니다. 아날로그 센서의 범위는 디지털 데이터로 변환에 의해 제한될 수 있습니다.

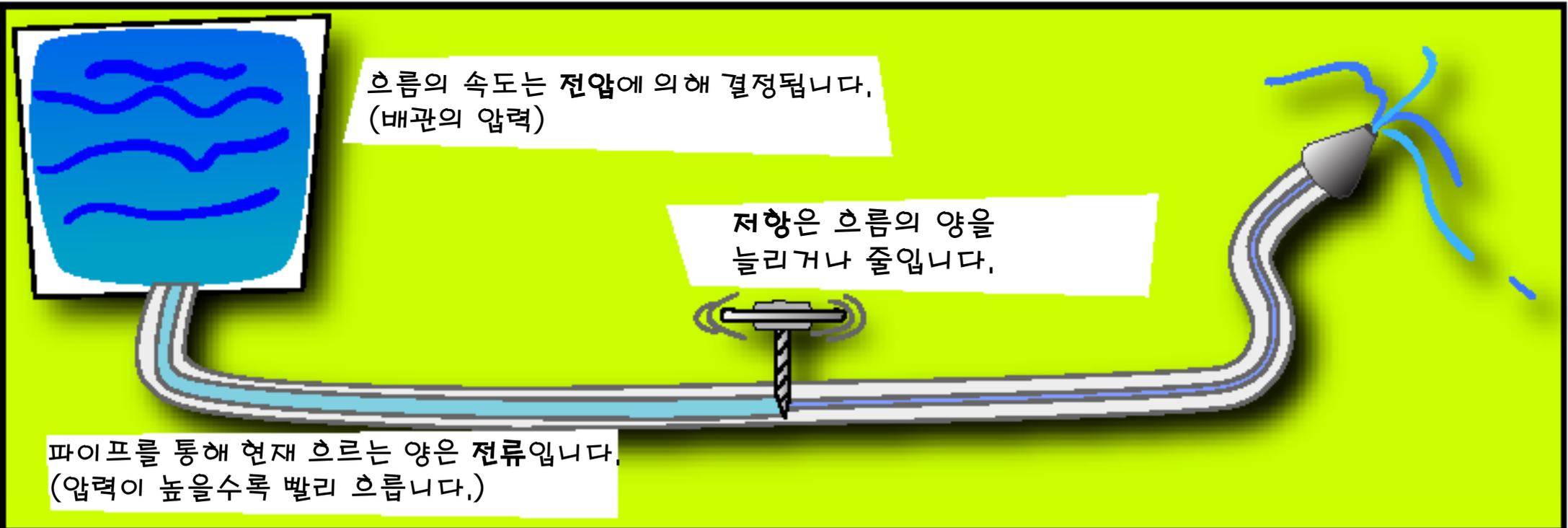


전압?
전류?
저항?
옴의 법칙?

아두이노를 연결하기 전에 우리는 전기 (그리고 전자에 대한)가 작동하는 몇 가지 조건과 원칙을 살펴봐야 합니다.

<p>Voltage (V) (전압은) 전류가 흐를 때 발생하는 전위의 차를 전압이라고 합니다. 단위는 볼트 (volts)입니다.</p>	<p>Current (I) (전류는) 전하가 이동하는 양을 말합니다. 단위는 암페어 (amperes) 또는 amps입니다.</p>	<p>Resistance (R) (저항은) 전류가 흐를 때 이를 방해하거나 반대로 유도하는 힘을 말합니다. 단위는 옴 (ohms)입니다.</p>
---	---	--

전기는 전도성 물질을 통해 흐르는 에너지의 흐름입니다.



흐름의 속도는 전압에 의해 결정됩니다.
(배관의 압력)

저항은 흐름의 양을 늘리거나 줄입니다.

파이프를 통해 현재 흐르는 양은 전류입니다.
(압력이 높을수록 빨리 흐릅니다.)

물 비유는 일반적으로 이런 용어를 설명하는데 사용됩니다. 여기 한가지 모델이 있습니다.

OHM'S LAW

$$\text{CURRENT} = \text{VOLTAGE} / \text{RESISTANCE} \\ (I = V/R)$$

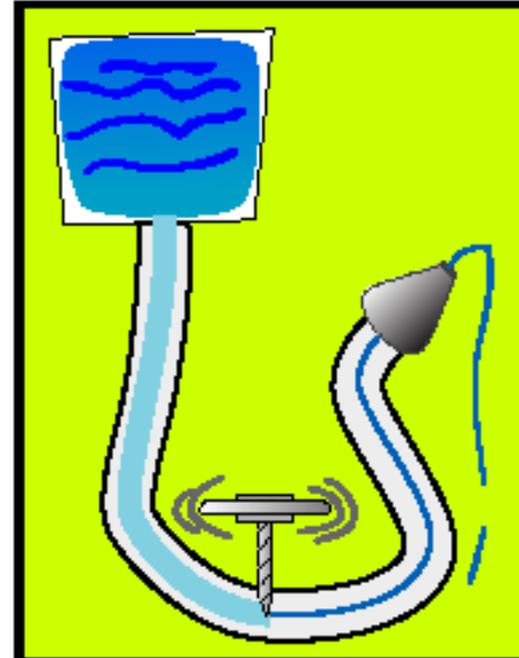
OR

$$\text{RESISTANCE} = \text{VOLTAGE} / \text{CURRENT} \\ (R = V/I)$$

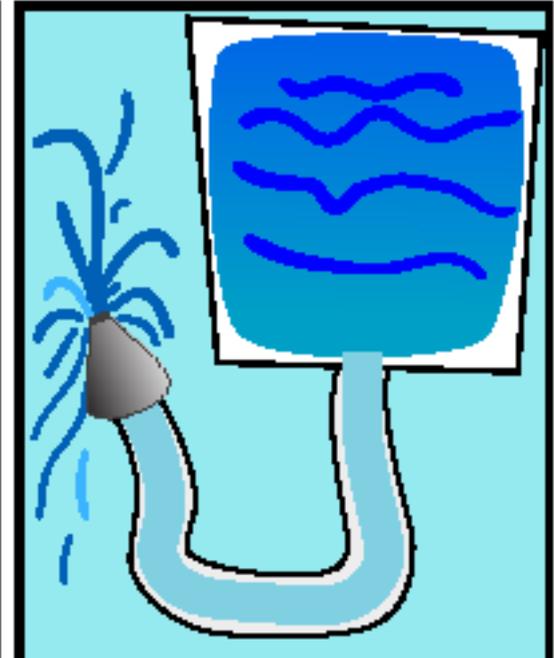
OR

$$\text{VOLTAGE} = \text{RESISTANCE} * \text{CURRENT} \\ (V = R*I)$$

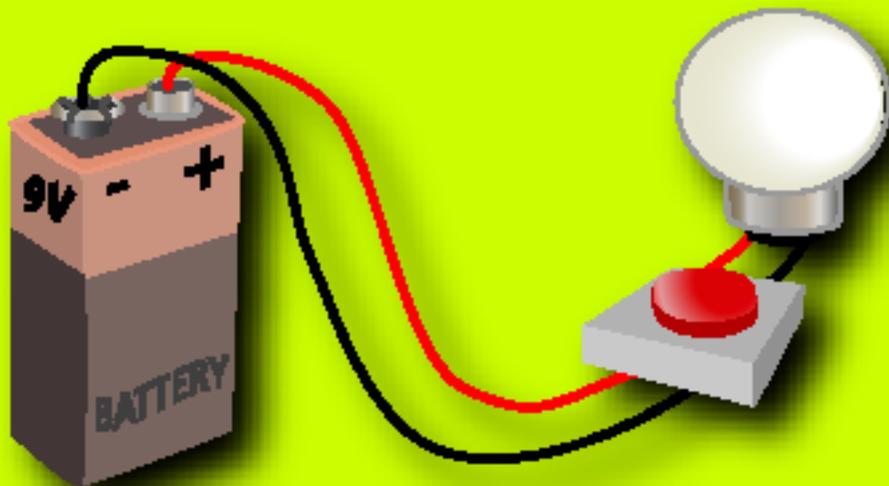
위는 독일 물리학자, Georg Ohm이 발견한 전압, 전류, 저항의 관계입니다.



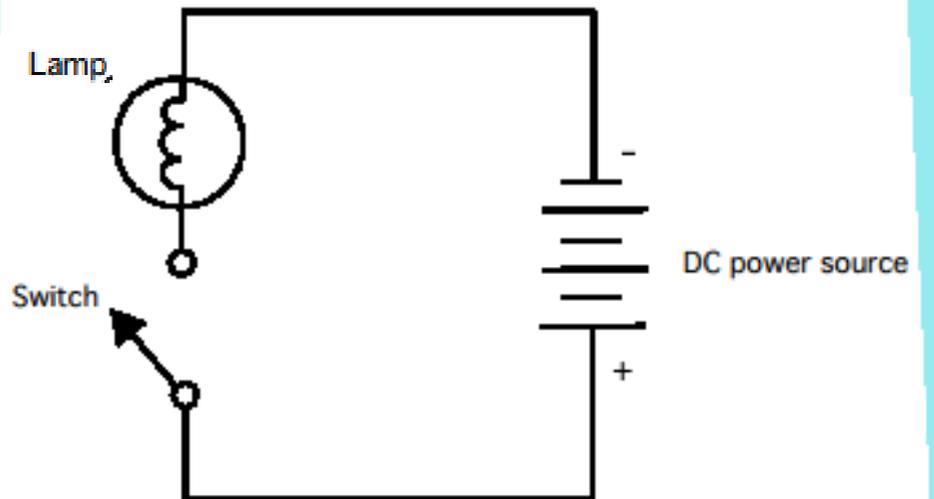
예를 들어, 저항을 늘리면 적게 흐르고



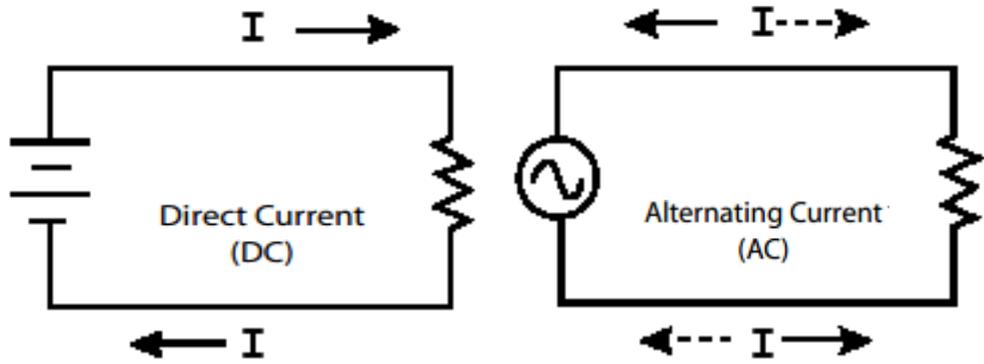
흐를 수 있는 양(전압)이 많으면 더 많이 흐르게 됩니다.



이제 간단한 회로를 살펴보겠습니다. 모든 회로는 에너지 공급(배터리)과 전기 부하(램프)로 달린 고리 형태여야 합니다. 전기 부하는 배터리의 전기 에너지를 전환하여 사용합니다. 이 회로에는 스위치도 있습니다.

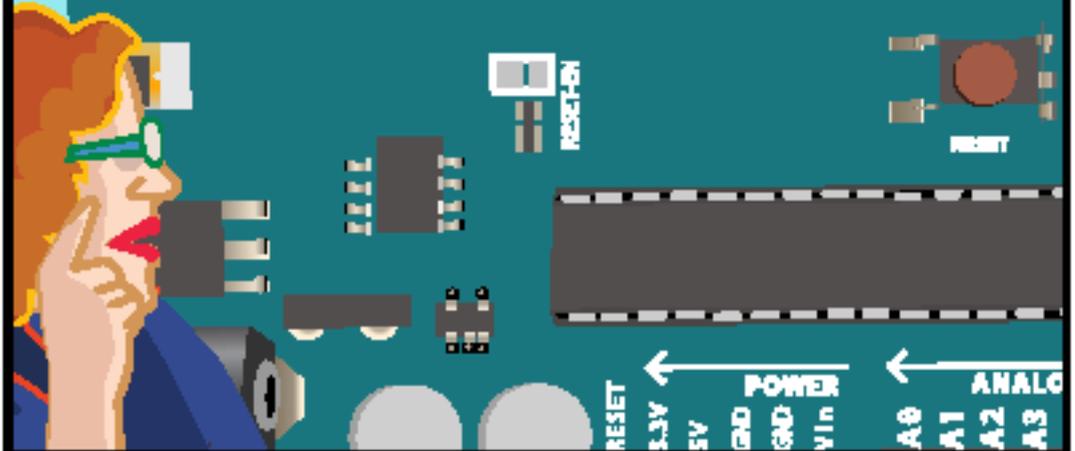


같은 회로의 회로도입니다. (전기 부품에 대한 간단한 심볼을 사용하여 회로를 나타냅니다.) 스위치가 닫히면, 전원 공급처로부터 전류가 흐르고 램프가 켜집니다.

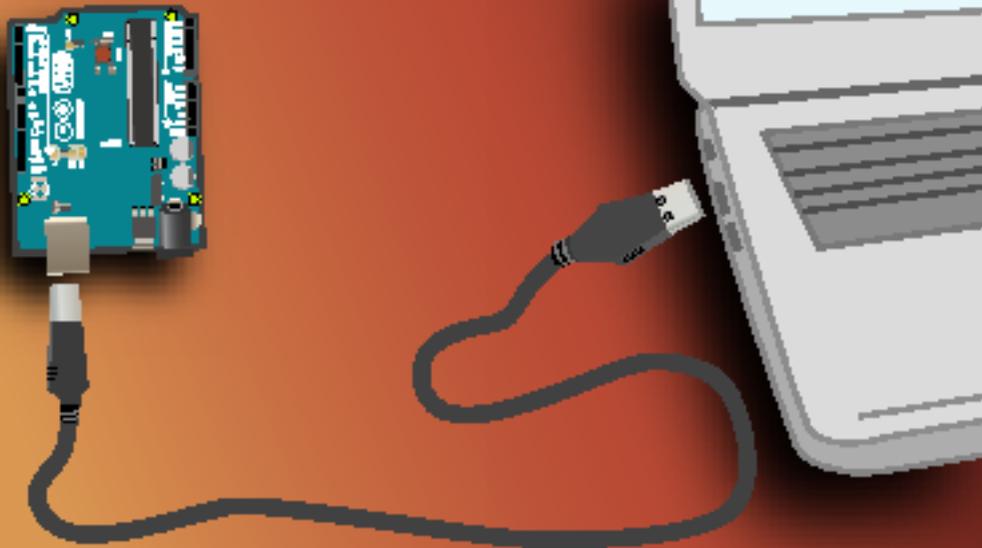


회로에는 흔히 직류(DC), 교류(AC), 두 가지 타입이 있습니다. DC(직류) 회로는 항상 한 방향으로 흐르며, AC(교류) 회로는 주기적으로 반대 방향으로 흐릅니다. 여기서 우리는 DC회로만을 이야기 할 것입니다.

이제 전기가 작동하는 원리에 대해 살펴보았으니 다시 아두이노로 돌아가 보시죠.



아두이노를 작동시키기 위해서는 전원이 꼭 필요합니다. 또한 프로그램을 하기 위해서는 컴퓨터에 아두이노를 연결해야 합니다.



USB케이블로 컴퓨터와 아두이노를 연결하는 것은 우리가 필요한 5볼트 전원을 공급해주고 프로그래밍을 시작할 수 있게 해줍니다.



DOWNLOAD HERE:

[HTTP://ARDUINO.CC/EN/MAIN/SOFTWARE](http://arduino.cc/en/main/software)

여러분은 아두이노를 프로그래밍 하기 위해 소프트웨어를 다운로드하고 설치해야 합니다. 위의 URL로 접속하면 무료로 다운로드 할 수 있으며, 이 프로그램은 맥OS X, 윈도우, 리눅스 운영체제에서 작동됩니다.

FOR INSTRUCTIONS ON HOW TO INSTALL
ARDUINO SOFTWARE ON A MAC:

[HTTP://WWW.ARDUINO.CC/EN/GUIDE/MACOSX](http://www.arduino.cc/en/guide/macOSX)

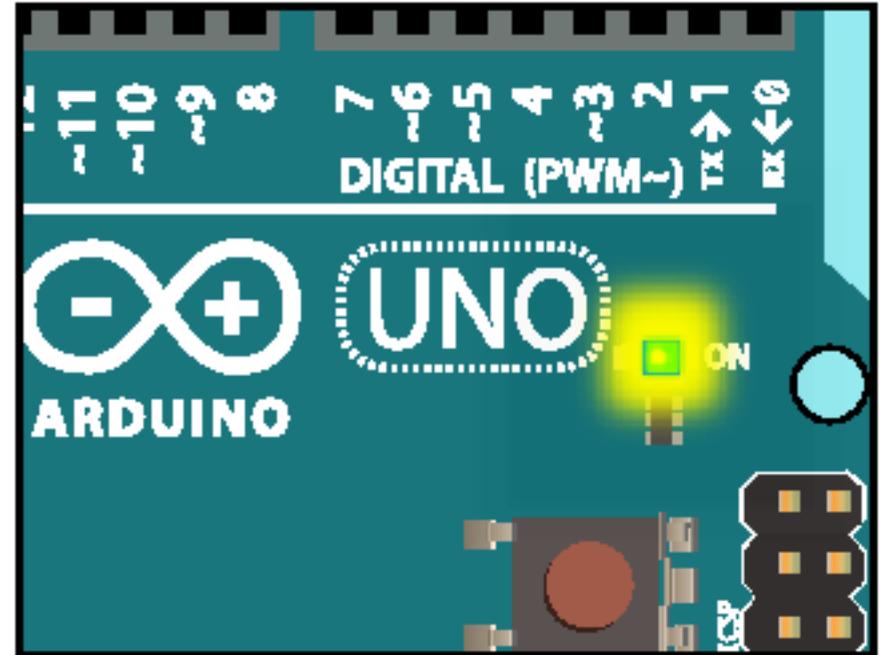
FOR INSTRUCTIONS ON HOW TO INSTALL
ON WINDOWS:

[HTTP://WWW.ARDUINO.CC/EN/GUIDE/WINDOWS](http://www.arduino.cc/en/guide/windows)

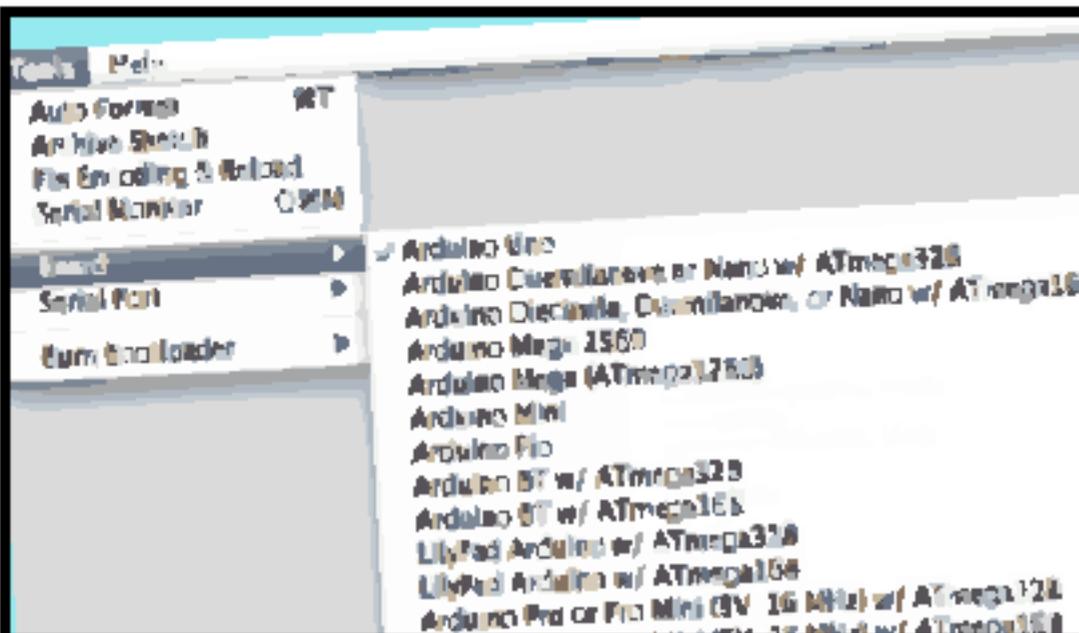
FOR INSTRUCTIONS ON HOW TO INSTALL
ON LINUX:

[HTTP://WWW.ARDUINO.CC/PLAYGROUND/LEARNING/LINUX](http://www.arduino.cc/playground/learning/linux)

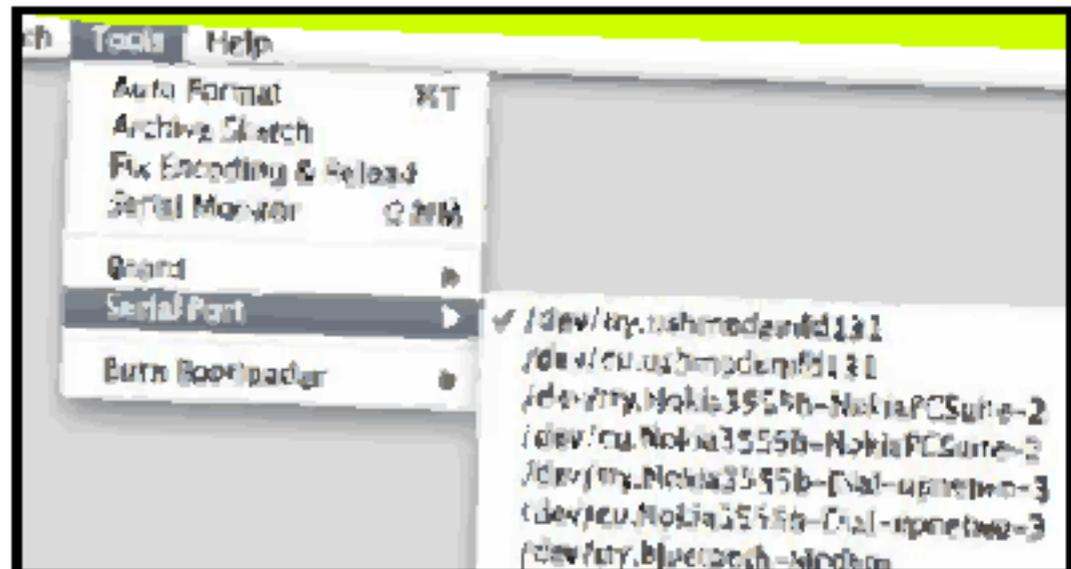
각 운영체제에 맞는 자세한 설치방법은 위 URL들을 참고하시기 바랍니다..



소프트웨어를 설치한 이후, 아두이노를 연결합니다. 보드의 ON이라고 표시된 LED에 불이 켜질 것입니다.



아두이노 소프트웨어를 작동시킨 후 Tools 메뉴에서 사용할 보드를 선택합니다. (Tools > Board). 예를 들어, Arduino Uno.

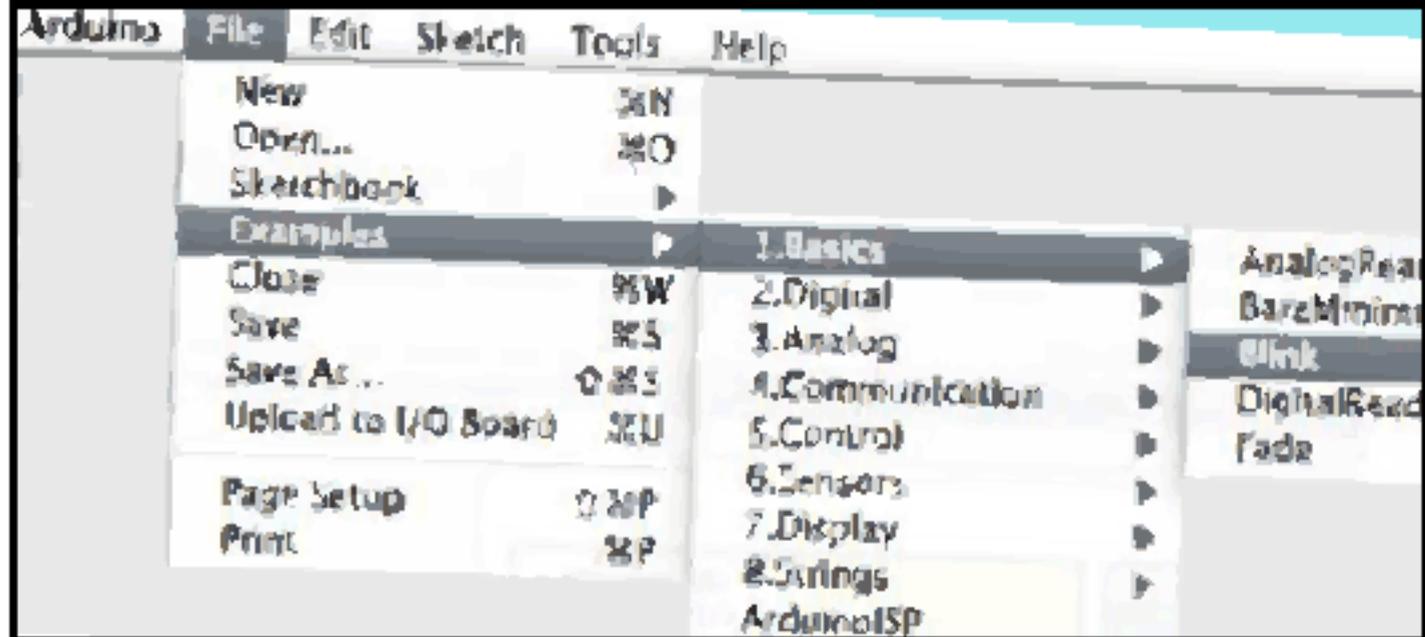


다음 시리얼 포트를 선택합니다. (Tools > Serial Port)
맥에서는 다음과 같이 나타납니다 /dev/tty.usbmodem.
윈도우에서는 COM3 또는 비슷한 형태로 나타납니다.

통합 개발
환경(IDE)이란
무엇인가요?



아두이노 소프트웨어를 다운로드 하면
IDE를 다운로드 한 것입니다. IDE는
컴파일러를 가진 텍스트 편집기와
몇가지 프로그래머를 도와주는 개발
소프트웨어로 구성되어 있습니다.



아두이노 IDE는 여러분이 스케치 또는 프로그램을 작성하고 그것을 보드에
업로드 할 수 있도록 해줍니다. 파일 메뉴에 있는 Blink 예제를 열어보시기
바랍니다. (File > Examples > 1. Basics > Blink.)



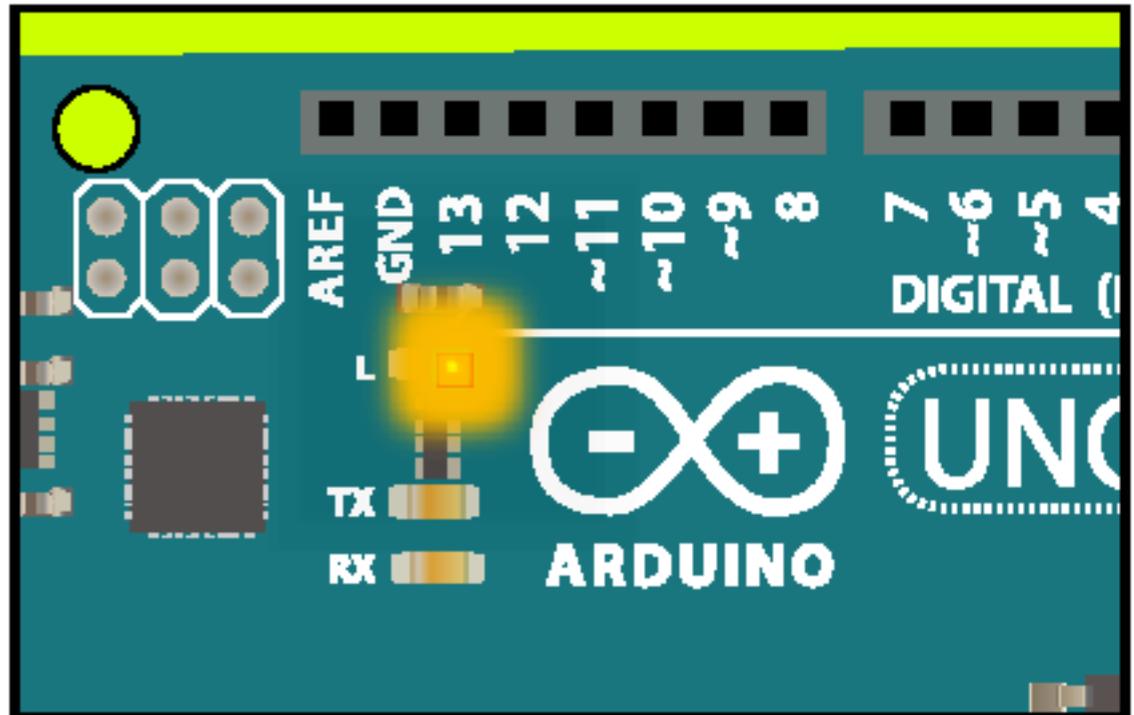
업로드 버튼

```
int ledPin = 13;

void setup() {
  pinMode(ledPin, OUTPUT);
}

void loop() {
```

스케치 코드를 아두이노 보드에 업로드 하기 위해
서는 윈도우 위에 있는 버튼 중에 업로드 버튼을
클릭하시면 됩니다. 업로드 완료 후 창 하단에
Done Uploading 라는 메시지가 뜰 것입니다.



아두이노에 있는 13번 핀 근처의 LED가 깜박이기 시작할 것입니다.

```

void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has LED connected on most Arduino boards
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH); // set the LED on
  delay(1000);           // wait for a second
  digitalWrite(13, LOW); // set the LED off
  delay(1000);           // wait for a second
}

```

스케치는 다른 어떤 프로그래밍 언어로 작성된 소프트웨어와 마찬가지로 컴퓨터를 위한 명령어의 집합입니다. LED Blink 예제를 자세히 보면 크게 `setup` 과 `loop`라는 두 가지 파트로 구성되어 있는 것을 알 수 있습니다.



Setup: 프로그램 시작 시 한번만 동작함



Loop: 반복해서 동작함

이 파트들은 모든 스케치에 나타나는 기본 함수 (Function)로 코드를 묶은 블록입니다. 또한 함수는 {}기호로 묶여집니다.

[HTTP://ARDUINO.CC/EN/REFERENCE/HOMEPAGE](http://arduino.cc/en/reference/homepage)



프로그래밍 언어를 배울 수 있는 아두이노 참조 가이드와 많은 다른 리소스를 아두이노 웹사이트에서 확인하시기 바랍니다.

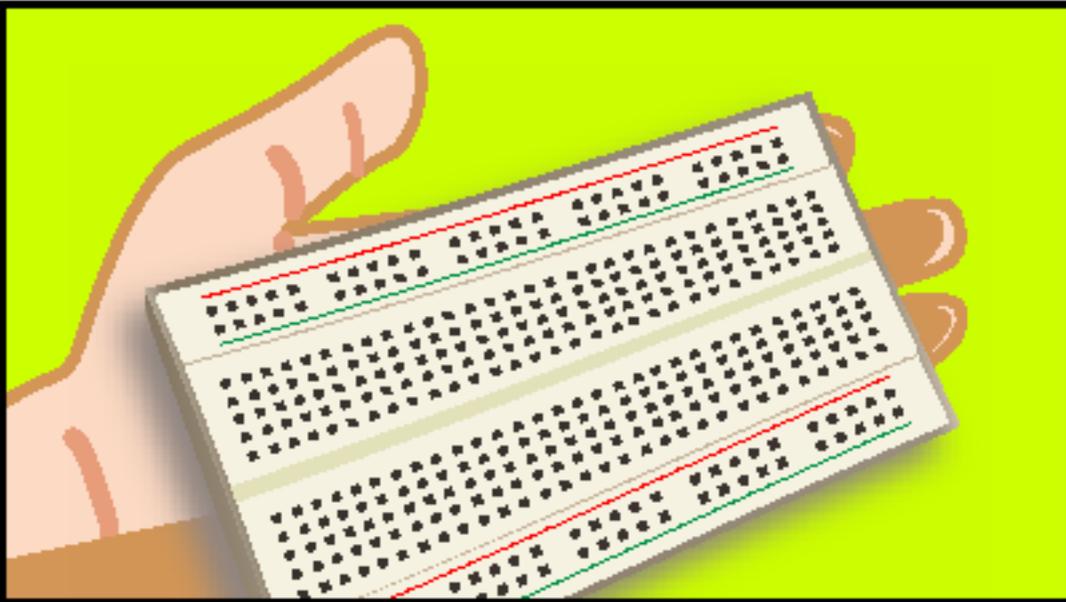
```

void setup() { //DECLARES BLOCK OF CODE
  pinMode(13, OUTPUT); //SETS PIN 13 TO OUTPUT
} //END BLOCK OF CODE

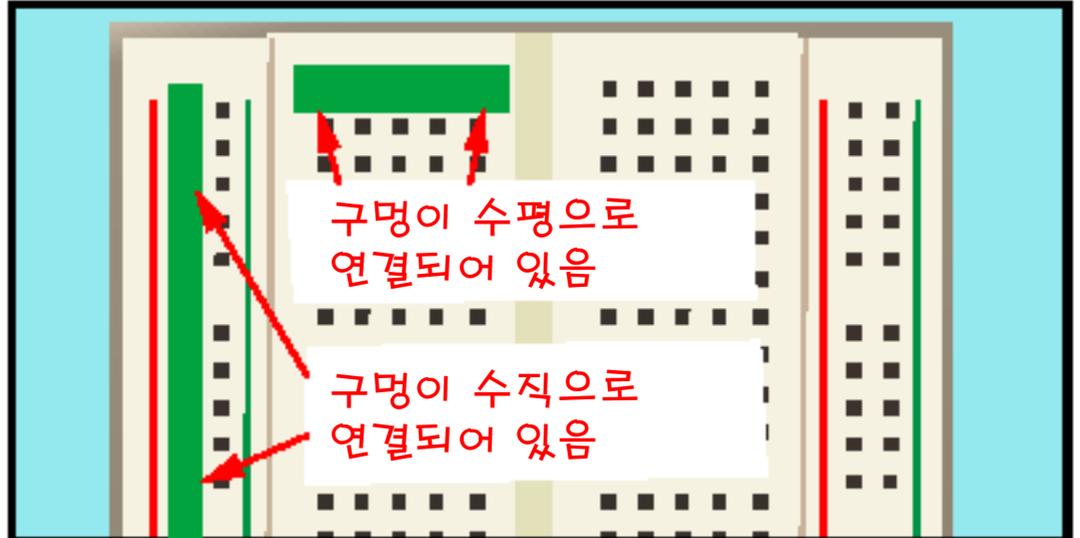
void loop() { //DECLARES BLOCK OF CODE
  digitalWrite(13, HIGH); //SETS PIN 13 HIGH
  delay(1000); //PAUSE 1 SECOND
  digitalWrite(13, LOW); //SETS PIN 13 LOW
  delay(1000); //PAUSE 1 SECOND
} //END BLOCK OF CODE

```

이제 이 간단한 스크립트의 각 라인에 표시된 주석을 통하여 각각의 의미를 파악해 보시기 바랍니다.



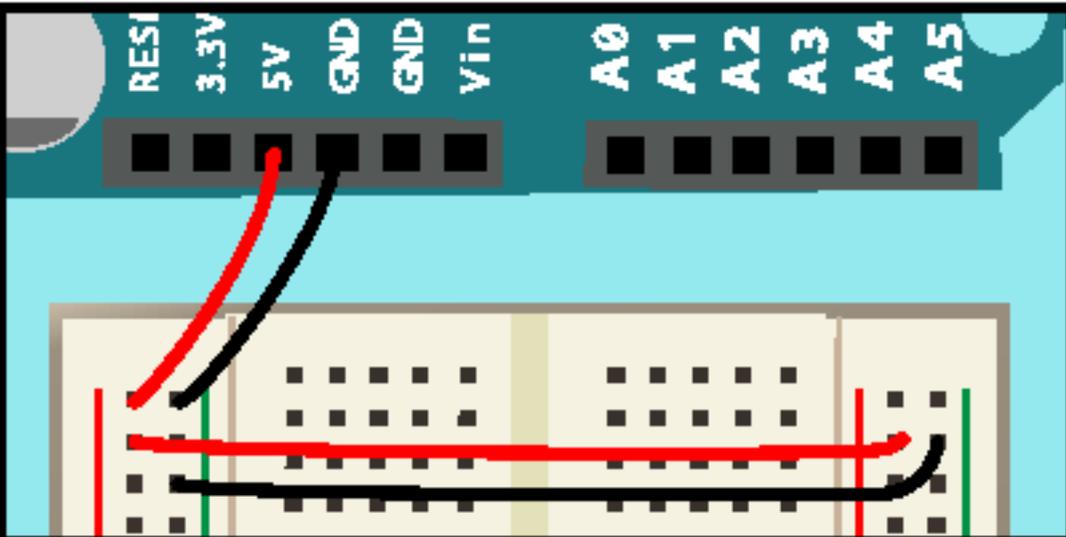
우리는 어떻게 아두이노 보드에 없는 객체를 제어할 수 있을까요? 우리는 무뎀납 브레드보드에 아두이노를 연결합니다. 브레드보드는 회로를 신속하게 설정하고 테스트 할 수 있게 해줍니다.



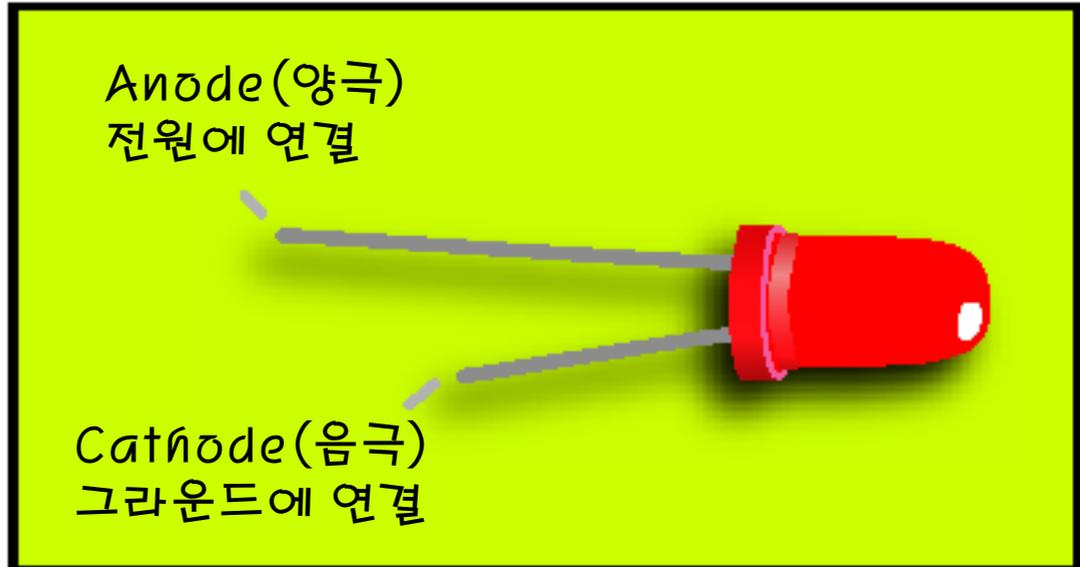
구멍이 수평으로 연결되어 있음

구멍이 수직으로 연결되어 있음

이 브레드보드는 왼쪽과 오른쪽 가장자리에 2개의 구멍이 있는 행을 가지고 있으며, 정 중앙을 기준으로 양 옆으로는 나란히 5줄의 구멍이 있는 행을 가지고 있습니다. 양 옆에 있는 2줄의 구멍은 수직으로 연결되어 있고, 정 중앙 양 옆의 나란한 5줄의 구멍은 수평으로 연결되어 있습니다.



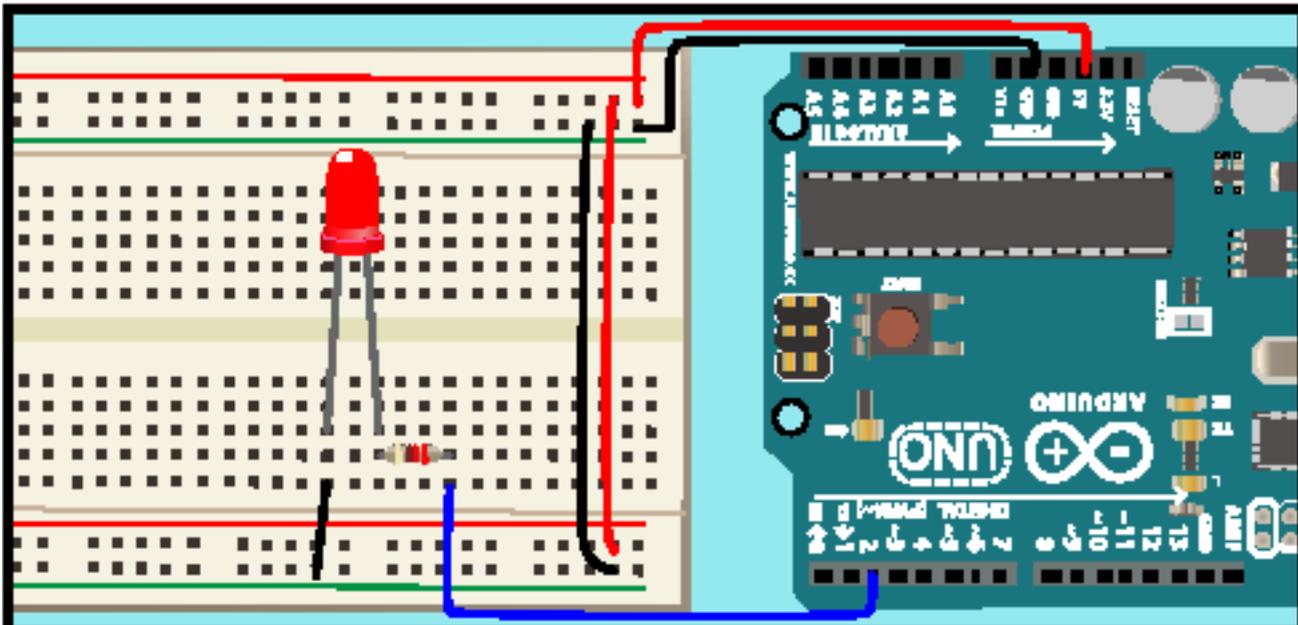
우리는 22게이지 와이어를 가지고 왼쪽, 오른쪽에 있는 수직적으로 연결된 줄들을 아두이노 보드에 있는 전원과 그라운드에 연결할 것입니다. 다른 구성품들은 필요할 때 중간에 있는 구멍들과 전원, 그라운드와 연결될 수 있습니다.



Anode(양극)
전원에 연결

Cathode(음극)
그라운드에 연결

LED(Light Emitting Diode)에 정방향으로 전류가 흐를 때, 불이 켜집니다. 우리는 브레드보드에 LED를 연결할 것입니다, 그러면 코드를 가지고 아두이노에서 이 LED를 제어 할 수 있습니다.



LED의 양극은 220 ohm 저항을 통해 아두이노의 2번 핀과 연결됩니다, 음극은 그라운드와 연결됩니다, 2번부터 13번 핀은 디지털 입력과 출력을 설정 할 수 있습니다, 스케치를 시작하기 위해 NEW 버튼을 눌러줍니다,

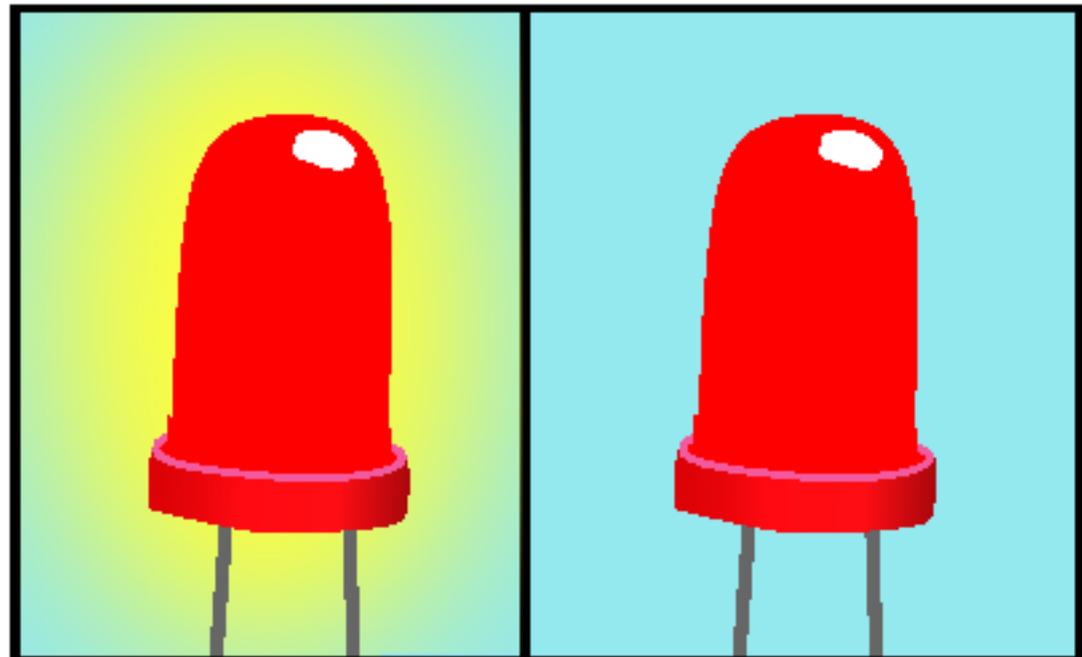
```
void setup() {
  pinMode(2, OUTPUT);
}

void loop() {
  digitalWrite(2, HIGH);
  delay(500);
  digitalWrite(2, LOW);
  delay(500);
}
```

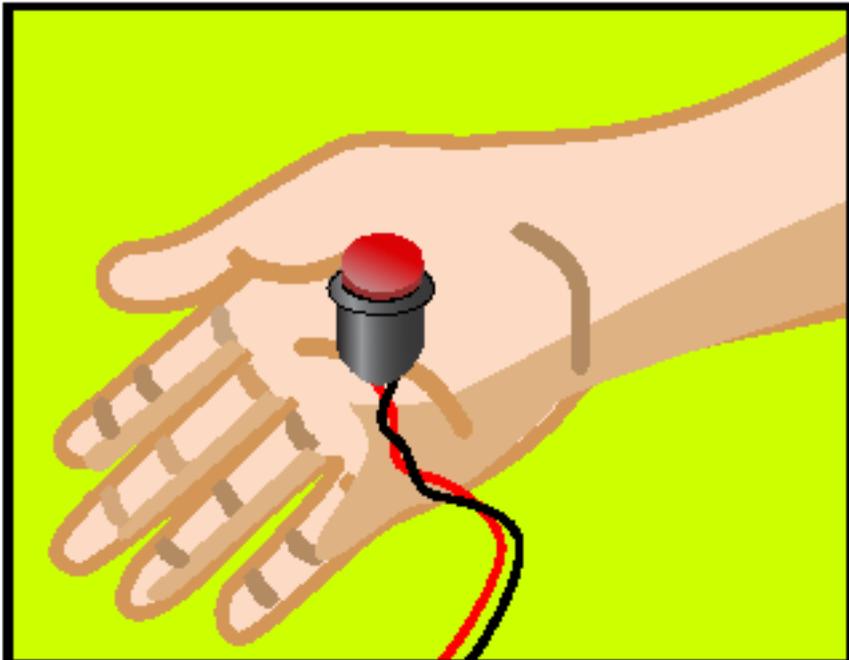
setup에서 2번 핀을 출력으로 설정했습니다, loop에서는 먼저 2번 핀을 led가 켜지도록 HIGH로 설정합니다, 다음 500 밀리세컨즈 (0.5초) 동안 일시 정지합니다, 2번 핀이 LOW로 설정되면 led는 꺼집니다, 그리고 다시 0.5초 동안 일시 정지합니다,



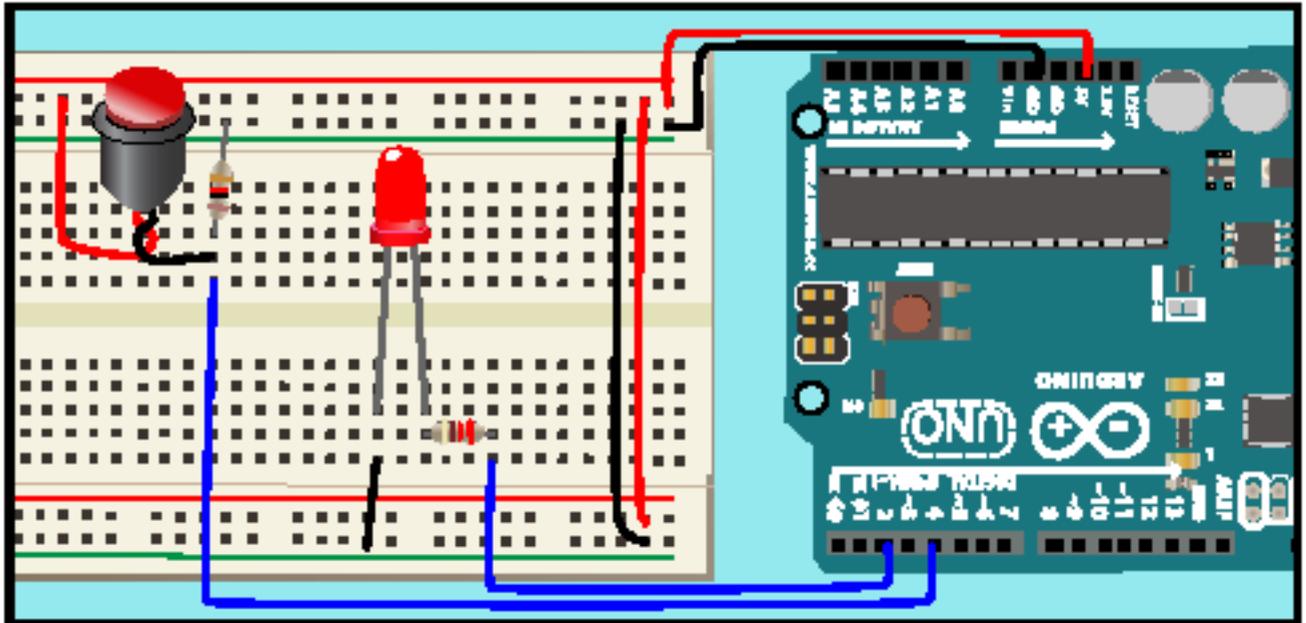
코드에 오류가 없는 지 확인 하기 위해 메뉴에 있는 확인 버튼을 누릅니다, 에러가 없다면 아두이노에 여러분의 프로그램을 넣기 위해 업로드 버튼을 누릅니다,



LED는 0.5초마다 꺼지고 켜짐을 반복함으로써 반짝이게 됩니다,



다음, 우리는 LED를 키고 끌 수 있도록 스위치를 디지털 입력으로 추가 연결할 것입니다.

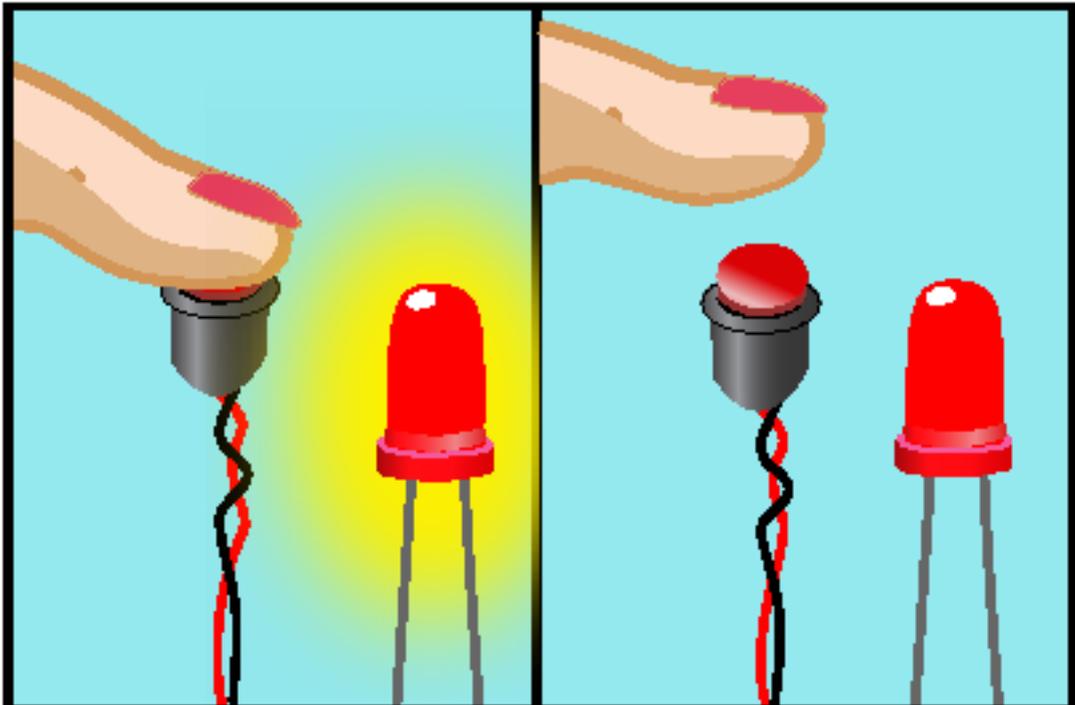


순간형 스위치의 한 쪽 끝을 아두이노의 4번 핀과 연결하고 같은 끝을 그라운드와 연결된 10k 저항과 연결합니다. 다른 쪽 끝은 전원과 연결합니다. LED는 앞에서 연결한 같은 핀에 연결된 상태로 유지합니다.

```
void setup() {
  pinMode(2, OUTPUT);
  pinMode(4, INPUT);
}

void loop() {
  if(digitalRead(4)){
    digitalWrite(2, HIGH);
  }else{
    digitalWrite(2, LOW);
  }
}
```

다음으로, 이제 코드를 작성합니다. setup에서 2번 핀은 출력, 4번 핀은 입력으로 설정합니다. loop에서는 if 문장을 사용합니다. 만약 4번 핀이 high이면 LED 핀이 high가 되도록 하고 반대이면 LED 핀이 low가 되도록 설정해서 LED가 꺼지도록 합니다.

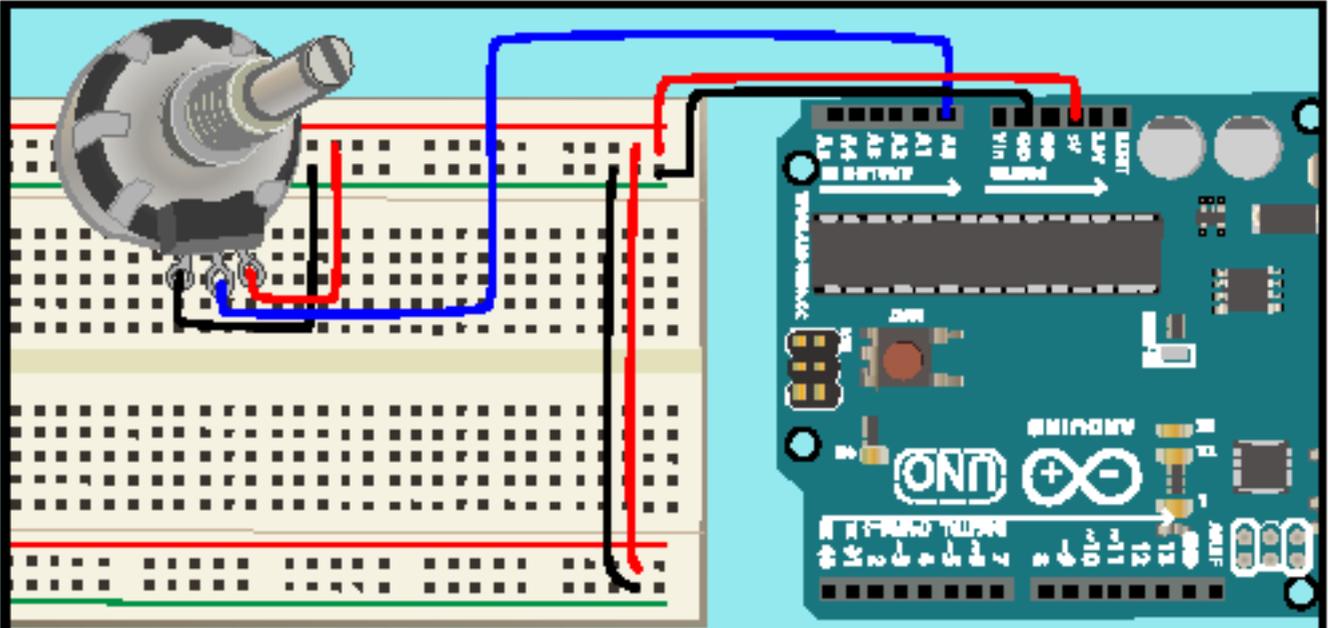


스위치가 눌리면 LED가 켜집니다.

potentiometer 또는 pot은 가변저항입니다, 저항의 양은 레버를 돌리는 방향이 바뀔 때 따라 증가하거나 감소합니다,



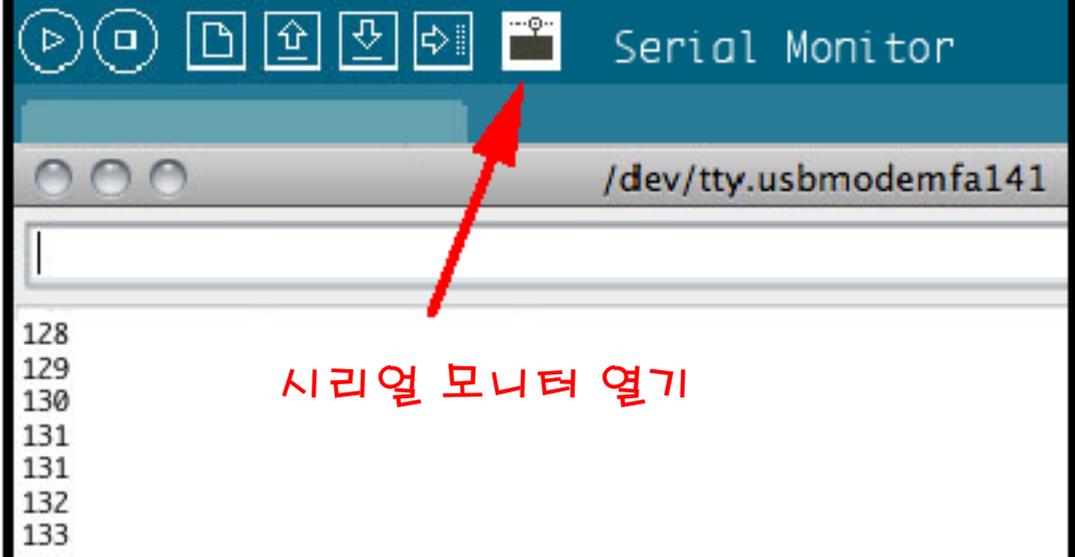
이제, 우리는 아날로그 입력을 설정할 것입니다, 우리는 potentiometer를 사용할 것입니다,



potentiometer의 가운데 핀을 아날로그 A0핀과 연결합니다, pot의 다른 한 끝은 전원과 연결하고 다른 쪽은 그라운드와 연결합니다,

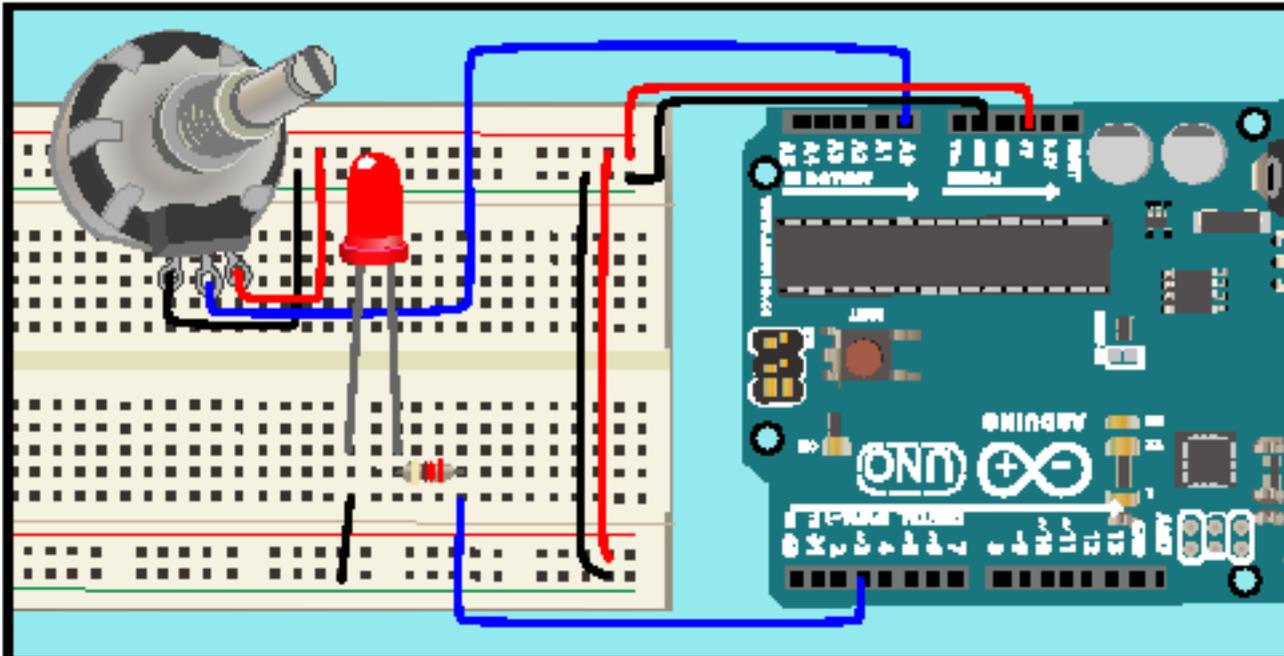
```
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  Serial.println(analogRead(A0));  
}
```

먼저, 우리는 시리얼 모니터를 사용해서 pot(가변저항)을 변화시킴으로써 얻을 수 있는 값의 범위를 확인해 볼 것입니다. 위의 코드에서, setup에는 전송 속도를 9600으로 설정하고 시리얼 객체를 초기화 하는 코드를 넣습니다. Loop에서는 아날로그 핀 A0에서 값을 읽습니다, 그리고 이 값을 println함수를 사용해서 시리얼 모니터에 출력할 것입니다,

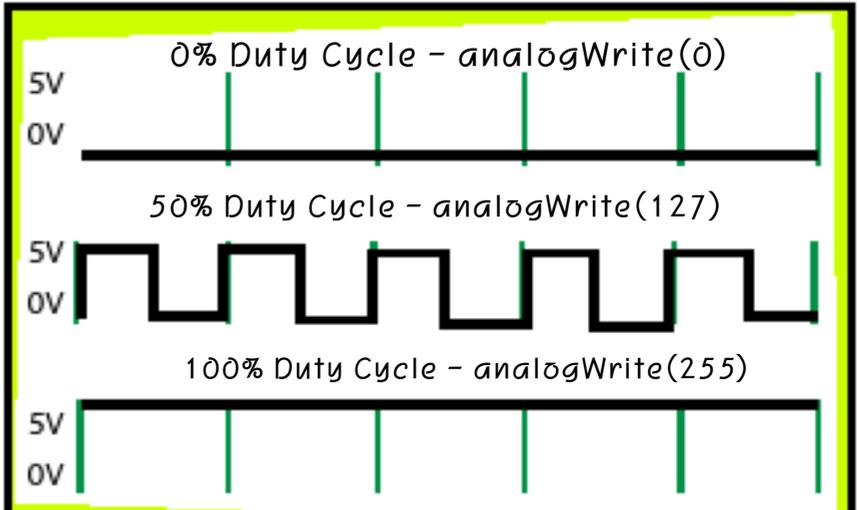


시리얼 모니터 열기

아두이노에 스크립트를 업로드 한 후, pot(가변저항)을 돌림에 따라 변하는 값을 보기 위해 시리얼 모니터 버튼을 클릭합니다, 창이 열리고, pot(가변저항)을 돌림에 따라 0에서 1024사이로 변하는 값들을 볼 수 있을 것입니다,



LED의 밝기를 조절하는 조광기처럼 가변저항으로부터 우리가 받게 되는 변화되는 값을 LED 밝기를 조절하는데 사용합니다. 양극을 저항을 통하여 3번핀과 연결하고 음극은 그라운드와 연결합니다.



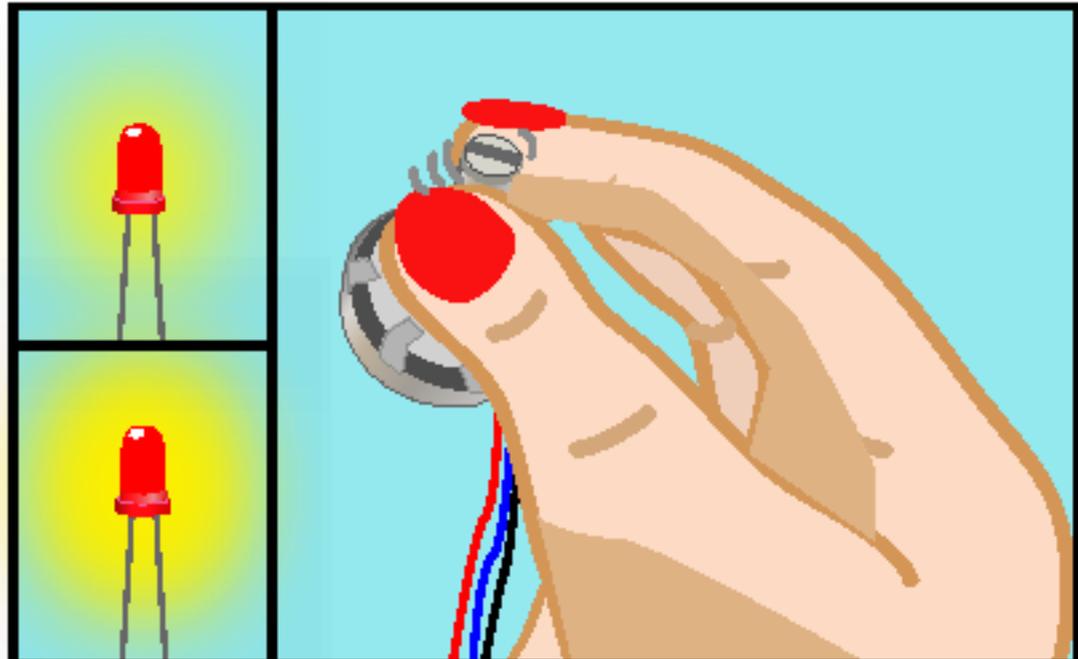
우리는 펄스폭 변조(PWM)를 사용할 것입니다. 이것은 on / off 하는 비율이나 duty cycles 를 다르게 바꾸거나, 전압을 조절함으로써 아날로그 값을 시뮬레이션 하는 방법입니다. 3,5,6,9,10,11번 핀을 pwm핀으로 사용할 수 있습니다.

```
int sensorValue = 0;

void setup() {
  pinMode(3,OUTPUT);
}

void loop() {
  sensorValue = analogRead(A0);
  analogWrite(3, sensorValue/4);
}
```

먼저, pot 값을 저장할 변수를 만듭니다. setup에서는 3번 핀을 출력으로 설정하고 loop에서는 a0핀의 값을 읽어서 변수에 저장합니다. 그리고 우리는 이 값을 3번 핀(LED 핀)에서 사용합니다. 우리는 값을 4로 나눠야만 합니다, 그렇게 함으로써 0에서 255까지의 값이나 바이트의 범위를 가질 수 있습니다.



POT(가변저항)을 돌림으로써 LED밝기를 완전히 꺼지게 하거나 최대한 밝게 조절 할 수 있습니다.



이게 끝입니다!
지금까지 보신 내용은
매우 간단한 소개이며
다음페이지의 링크로
가시면 더 많은 자료를
보실 수 있습니다.

LINKS

SOFTWARE

SOFTWARE DOWNLOAD

[HTTP://WWW.ARDUINO.CC/EN/MAIN/SOFTWARE](http://www.arduino.cc/en/main/software)

LANGUAGE REFERENCE

[HTTP://ARDUINO.CC/EN/REFERENCE/HOMEPAGE](http://arduino.cc/en/reference/homepage)

SUPPLIES

SPARKFUN ELECTRONICS

[HTTP://WWW.SPARKFUN.COM/](http://www.sparkfun.com/)

ADAFRUIT INDUSTRIES

[HTTP://ADAFRUIT.COM/](http://adafruit.com/)

MAKER SHED

[HTTP://WWW.MAKERSHED.COM/](http://www.makershed.com/)

JAMECO ELECTRONICS

[HTTP://WWW.JAMECO.COM/](http://www.jameco.com/)

TUTORIALS

ARDUINO SITE TUTORIALS

[HTTP://WWW.ARDUINO.CC/EN/TUTORIAL/HOMEPAGE](http://www.arduino.cc/en/tutorial/homepage)

LADY ADA

[HTTP://WWW.LADYADA.NET/LEARN/ARDUINO/](http://www.ladyada.net/learn/arduino/)

INSTRUCTABLES

[HTTP://WWW.INSTRUCTABLES.COM/TAG/TYPE-ID/
CATEGORY-TECHNOLOGY/CHANNEL-ARDUINO/](http://www.instructables.com/tag/type-id/category-technology/channel-arduino/)

BOOKS

GETTING STARTED WITH ARDUINO BY MASSIMO BANZI

MAKING THINGS TALK: USING SENSORS, NETWORKS, AND

ARDUINO TO SEE, HEAR, AND FEEL YOUR WORLD BY
TOM IGOE

PHYSICAL COMPUTING: SENSING AND CONTROLLING

THE PHYSICAL WORLD WITH COMPUTERS BY DAN
O'SULLIVAN & TOM IGOE

ARDUINO COOKBOOK BY MICHAEL MARGOLIS

All text and drawings by Jody Culkin
For more, check out jodyculkin.com

Special Thanks to Tom Igoe, Marianne petit, Calvin Reid, The faculty and staff of the interactive telecommunications program at nyu, particularly Dan o'sullivan, Danny rozin and Red burns. Thanks to Cindy karasek, chris Stein, sarah teitler, Kathy goncharov & zannah marsh. many, many thanks to the Arduino team for bringing us this robust and flexible open source platform. and thanks to the lively, active and ever growing arduino community.

Introduction to Arduino by Jody Culkin is licensed
under a Creative Commons

Attribution-NonCommercial-ShareAlike 3.0

Unported License.



한글화 by Kocoafab

<http://kocoafab.cc>

