



드론으로 배우는
프로그래밍 교실

Ch3-2. LED깜박이기



01 아두이노와 컴퓨터 연결	01
USB연결, 보드 및 포트 설정	02
아두이노 드라이버 업데이트	03
컴파일 및 업로드	04
02 LED 깜빡이기 프로그램	05
Setup()과 loop()	06
LED 깜빡이기 예제 작성하기	07
예제 가져오기	08
03 아두이노 하드웨어 구성	09
아두이노 핀 구성	10
메인 보드 핀 구성	11
브레드보드 및 LED	12
저항이란?	13
저항 값 읽기	14
LED 하드웨어 구성	15
LED 속도 변화 해보기	16
LED 속도 변화 해보기 정답	17



드론으로 배우는
프로그래밍 교실

초판발행 2016년 9월 23일
지은이 이상준 | 펴낸이 CodingBird
펴낸곳 WHIT | 주소 안산시 한양대학로55 창업보육센터 B01

Published by WHIT. Printed in Korea
Copyright © 2016 CodingBird & WHIT

이 책의 저작권은 CodingBird와 WHIT에 있습니다.
저작권법에 의해 보호를 받는 저작물이므로
무단 복제 및 무단 전제를 금합니다.

01 아두이노와 컴퓨터 연결



아두이노에 프로그램을 어떻게 전달할까요?

아두이노와 컴퓨터를 연결해주는 것은 하드웨어적으로는 USB케이블이 있고, 소프트웨어적으로는 드라이버가 있습니다.

아두이노라는 장치를 컴퓨터에서 인식하기 위해서는 드라이버가 깔려 있어야 합니다.

USB연결, 보드 및 포트 설정

USB 연결

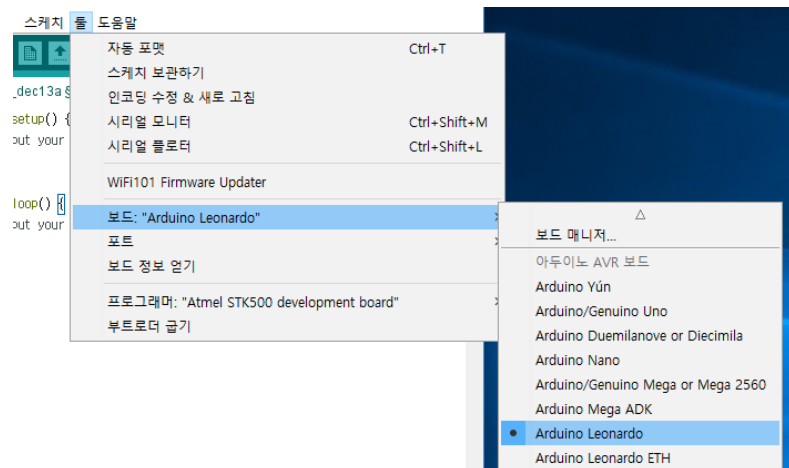
- 1 아두이노와 컴퓨터를 USB케이블을 통해 연결합니다.



<그림1-1> 아두이노 USB 연결

보드 설정

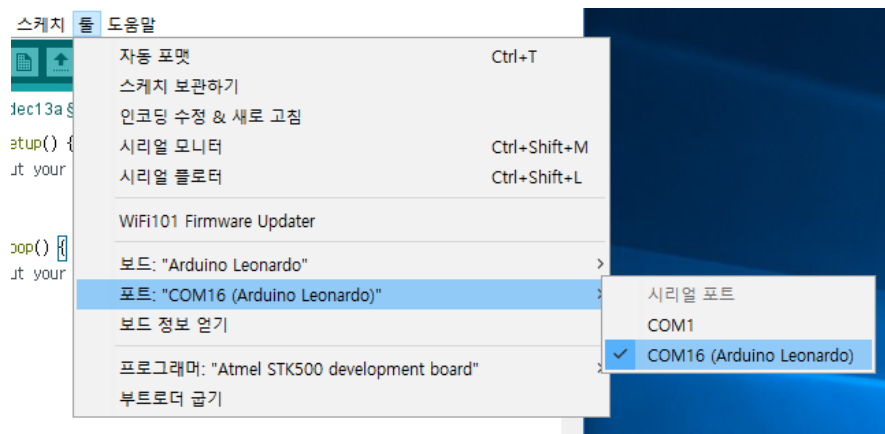
- 2 메뉴 바의 도구(툴) – 보드 – Arduino Leonardo를 선택합니다.



<그림1-2> 아두이노 보드 선택

포트 설정

- 3 메뉴 바의 도구 – 포트 – 아두이노의 COM을 선택합니다.



<그림1-3> 아두이노 포트 선택

- 4 업로드를 클릭하여 프로그램을 아두이노에 넣습니다.

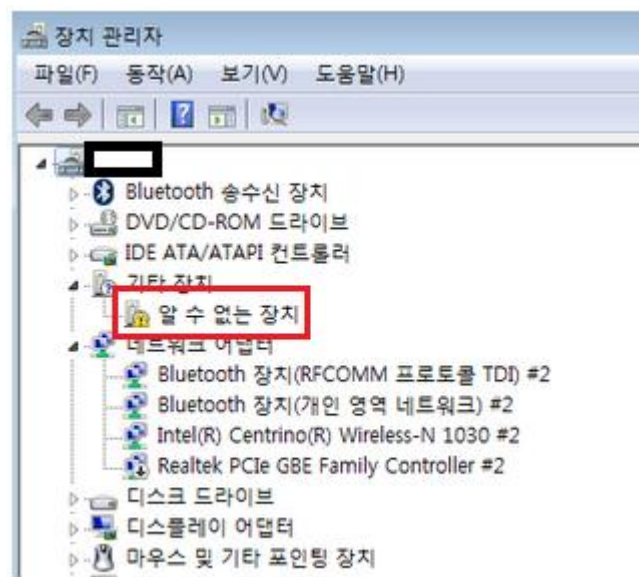


아두이노 드라이버 업데이트

드라이버 업데이트

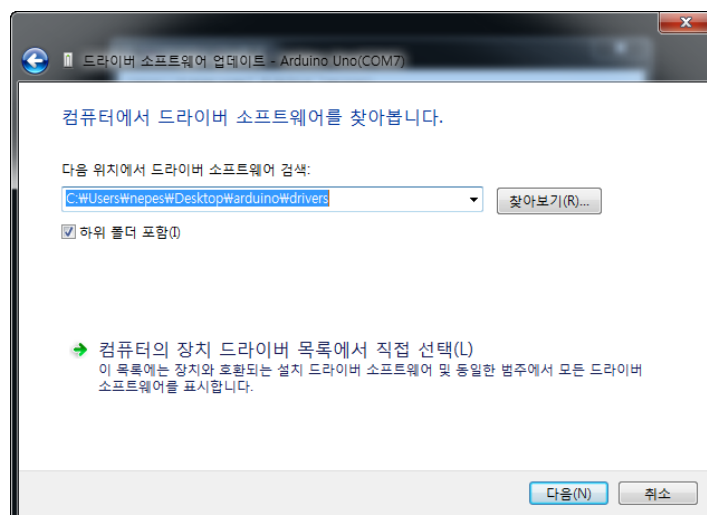
아두이노를 처음 연결할 경우 알 수 없는 장치가 연결되었다고 뜨는 경우가 종종 있습니다. 이럴 경우 장치 드라이버를 업데이트 시켜 줘야 합니다.

- 1 시작 – 장치관리자 를 통해 장치관리자를 엽니다.



<그림1-4> 장치 관리자

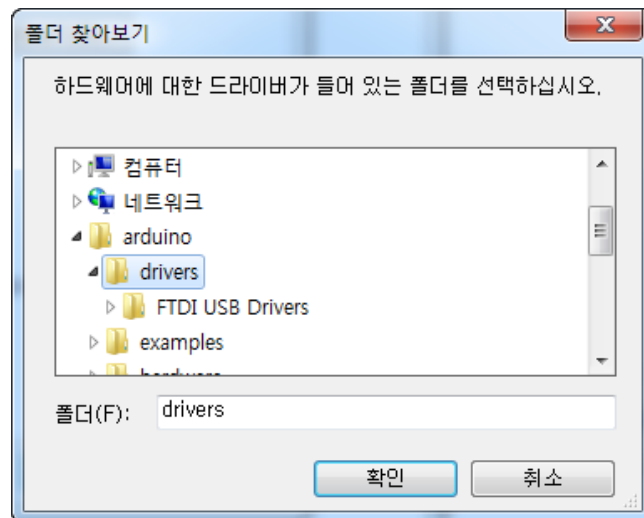
- 2 알 수 없는 장치를 우 클릭 합니다.
- 3 드라이버 소프트웨어 업데이트를 클릭합니다.



<그림1-5> 드라이버 소프트웨어 업데이트

컴파일 및 업로드

- 4 다운로드 받은 폴더 안의 drivers 까지의 경로를 입력합니다. Ex) C:\Users\m\desktop\arduino\drivers



<그림1-6> 드라이버 소프트웨어 업데이트

- 5 확인을 누르고 설치가 완료되길 기다립니다.

컴파일 업로드

컴파일은 작성된 코드가 아두이노로 들어갈 때 이상이 없는지 에러를 체크하는 과정입니다. 만약, 컴파일이 없다면 에러가 있는 프로그램이 들어가 아두이노를 망가뜨릴 수 있습니다.

업로드는 코드를 실제 아두이노에 넣는 것입니다. 아래 그림의 화살표 버튼을 누르면 컴파일과 업로드가 같이 진행됩니다.



<그림1-7> 컴파일 및 업로드 버튼

02 LED 깜빡이기 프로그램



아두이노를 처음 다뤄본다면 꼭 한번 접하게 되는 프로그램이 LED 깜빡이기입니다.

아두이노의 LED를 1초 간격으로 깜빡이는 예제를 통해서 아두이노라는 하드웨어를 어떻게 소프트웨어로 제어하는지에 대한 감을 잡을 수 있을 겁니다.

Setup()과 loop()

아두이노 기초 함수

아두이노 프로그램은 쉽게 이해할 수 있는 단순한 구조로 구성되어 있으며, 크게 2개의 파트로 나뉘어져 있습니다.

```
void setup()
{
    기본 세팅;
}
void loop()
{
    실행 함수;
}
```

setup()

setup()은 프로그램이 시작할 때, 사용됩니다.
pin mode를 초기화하거나 serial를 초기화합니다.
내용이 없어도 무조건 있어야 프로그램이 실행됩니다.

loop()

setup()실행 후, loop()함수는 반복적으로 정해진 명령어를 실행합니다.

<그림2-1> 아두이노 기초 함수

LED 깜빡이기 예제 작성하기

LED
깜빡이기

- ① 다음과 같이 코드를 작성하여 아두이노에 업로드합니다.

```
ch3_2_1_blink
1 void setup() {
2   pinMode(3, OUTPUT);
3 }
4
5 void loop() {
6   digitalWrite(3, HIGH);
7   delay(1000);
8   digitalWrite(3, LOW);
9   delay(1000);
10 }
```

<그림2-2> LED Blink 예제

```
void setup(){
  pinMode(3, OUTPUT); //3번 핀을 출력으로 설정
}

void loop()
{
  digitalWrite(3, HIGH); //3번 핀을 HIGH로 출력
  delay(1000);           // 1초간 멈춤
  digitalWrite(3, LOW);  //3번 핀을 LOW로 출력
  delay(1000);           // 1초간 멈춤
}
```

꿀TIP

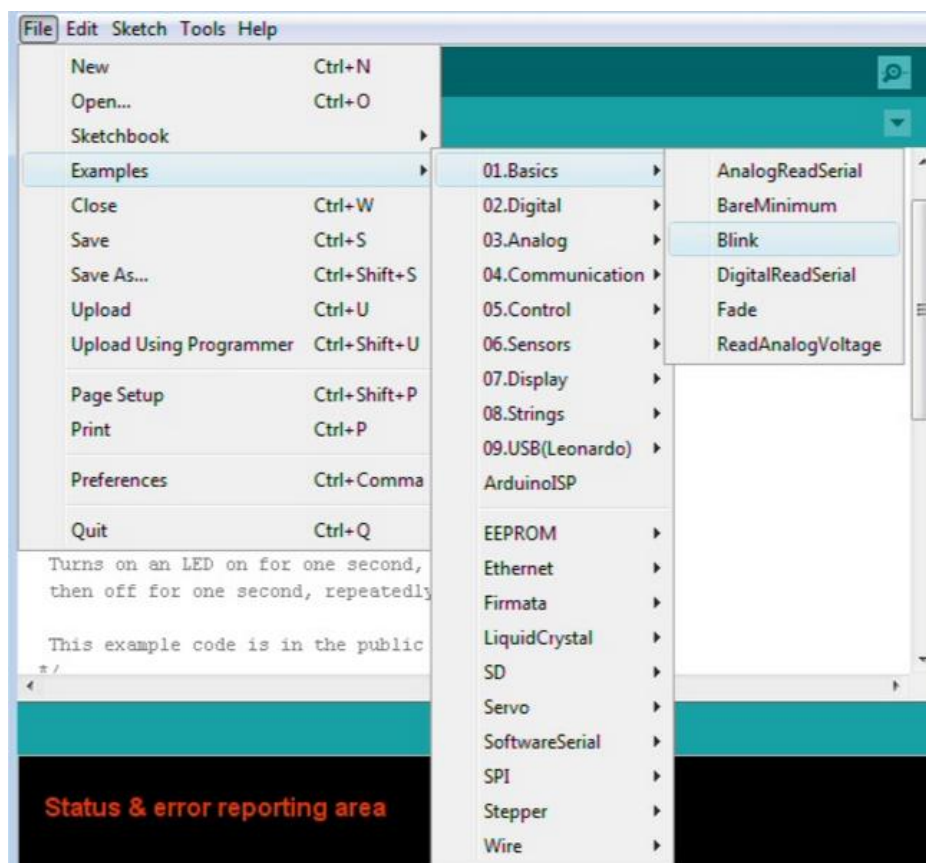
프로젝트 저장

업로드 시 프로젝트를
저장할 지 물어보는데,
저장을 해 두는게
좋습니다.

LED
깜빡이기
해석

예제
가져오기

메뉴 바 - 파일 - 예제 - 01.Basics - Blink 를 통해서 LED 깜빡이기 예제를 가져 옵니다.



<그림2-3> LED Blink 예제 가져오기

아두이노에는 미리 작성되어 있는 다양한 예제가 있습니다.

프로그래밍을 처음 배울 때에는 무엇부터 해야 할지 모르는 경우가 많고, 어떻게 프로그래밍을 시작해야 할지 막막한 경우가 많습니다. 이럴 때 예제를 참고하면 도움이 많이 됩니다.

아두이노에서 기본적으로 제공 해주는 예제에는 Serial통신, LED Fade, analog 등 많은 예제들이 있습니다.

03 아두이노 하드웨어 구성



아두이노의 하드웨어는 어떻게 구성될까요?

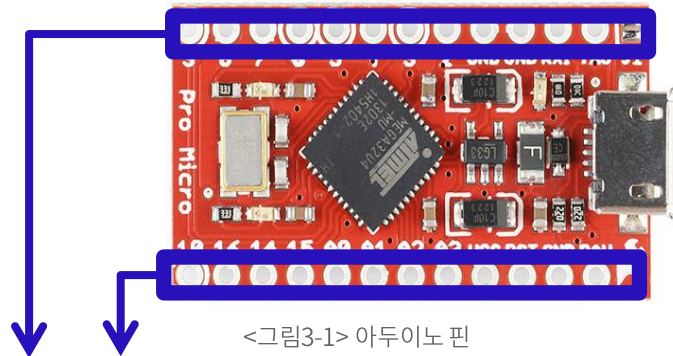
아두이노 스케치를 통해서 소프트웨어는 제작하였는데, 실제로 LED를 깜빡이기 위해서는 하드웨어적인 구성이 필요합니다.

아두이노가 LED를 제어할 때는 핀을 사용합니다.

핀을 사용하여 입력과 출력을 받아 외부 센서를 제어하게 됩니다.

아두이노
핀 구성

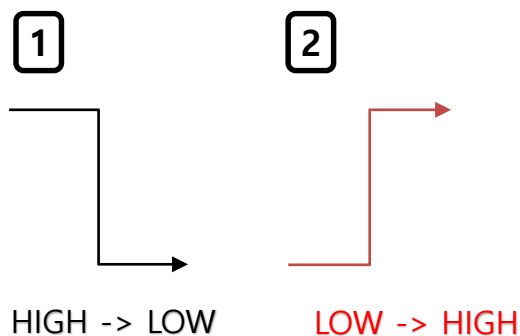
아두이노의 좌우로 매겨져 있는 번호를 아두이노의 핀이라고 합니다. 이 핀을 통해서 전류를 내보내 모터를 제어하거나, 전압을 읽어서 센서 값을 받아 들입니다.



<그림3-1> 아두이노 핀

아두이노의 핀을 통해서 원하는 명령을 내릴 수 있습니다. 크게 나누면 핀을 통해서 입력을 받거나 출력을 내보냅니다.

핀은 크게 digital pin과 analog pin으로 구분할 수 있는데, digital pin의 경우 LOW인 상태와 HIGH인 상태를 가집니다. LOW는 전압이 낮은 상태이고 HIGH는 전압이 높은 상태입니다.



<그림3-2> LOW와 HIGH인 상태

예를 들어

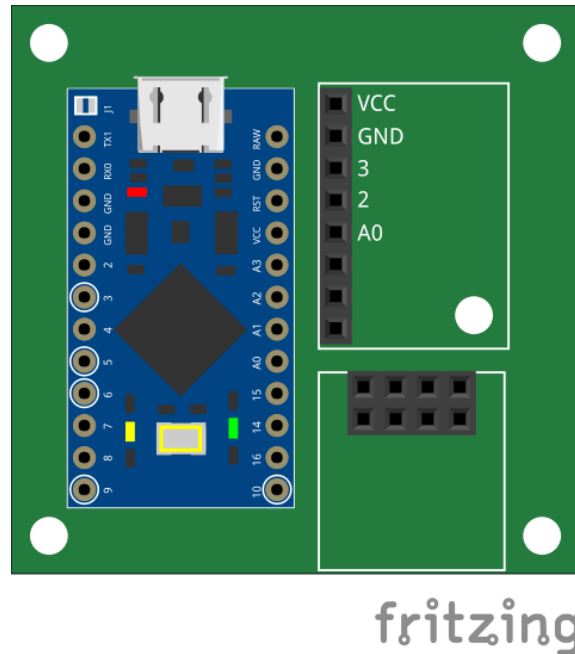
digitalWrite(3,HIGH);

와 같은 명령어의 경우 3번 핀을 HIGH로 만들어 전압을 높은 상태로 출력하라는 뜻입니다.

메인 보드 핀 구성

메인 보드

메인 보드는 내부적으로 아두이노의 핀들과 연결되어 있습니다. 그 중에서 우리는 VCC, GND, 3, 2, A0 핀을 사용할 것입니다.



<그림3-3> 아두이노와 메인 보드 연결

메인 보드 핀 특징

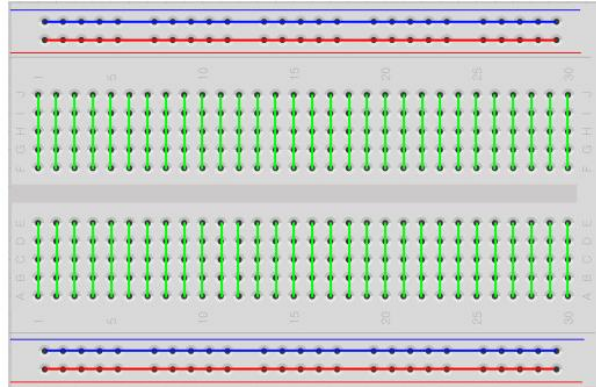
메인 보드의 핀들은 아두이노의 핀들과 연결되어 있고, 이들은 각각의 특징을 가지고 있습니다.

- VCC는 5V의 전압을 유지합니다.
- GND는 0V의 전압을 유지합니다.
- 3번핀은 디지털 입출력과 아날로그 출력으로 사용 가능합니다
- 2번핀은 디지털 입출력으로 사용 가능합니다.
- A0핀은 아날로그 값을 읽어 들일 수 있습니다.

핀과 관련된 내용은 뒷장에서 차근차근 배워볼 것입니다. 지금은 베이스보드에 이런 핀들이 있구나 정도만 알아두면 충분합니다.

브레드보드

브레드보드



<그림3-4> 브레드보드

브레드보드는 빵판이라고도 불리며, 시제품을 만드는 데 쓰는 재활용할 수 있는 무 땀납 장치입니다.

중간의 **초록 선들** 구역은 세로로 연결되어 있습니다. 세로로 5칸 씩 구멍이 있고 부품 영역이라고 합니다.

위의 5칸과 아래의 5칸은 서로 전기가 통하지 않습니다.

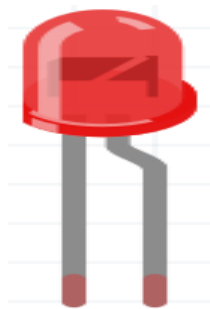
파란 선 : GND(접지선)입니다.

빨간 선 : VCC(전압)입니다.

초록 선 : 세로로 같은 핀을 나타냅니다.

LED

발광 다이오드, LED(Light Emitting Diode)



<그림3-5> LED

긴 쪽 : +

발광 다이오드는 순방향으로 전압을 가할 때 발광하는 반도체 소자입니다.

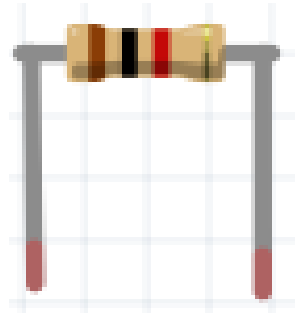
전류량에 비례해서 빛이 발생합니다.

긴 쪽을 +(전압이 높은 부분)과 연결해야 합니다.

저항이란?

저항

저항



<그림3-6> 탄소 피막 저항

저항은 전류의 흐름을 방해하는 정도를 나타내는 물리량입니다.

저항의 주된 용도는

- 1) 전류량 제어
- 2) 전압의 분배입니다.

옴의 법칙

옴의 법칙

$$I = \frac{V}{R}$$

<그림3-7> 옴의 법칙

옴의 법칙은 전압과 전류, 저항 사이의 관계를 나타낸 식입니다.

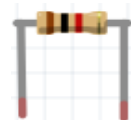
전압이 일정하다는 가정 하에 저항이 커지면 전류가 작아지게 됩니다.

저항의 종류

저항의 종류



<그림3-8> 권선 저항



<그림3-9> 탄소 피막 저항



<그림3-10> 가변 저항

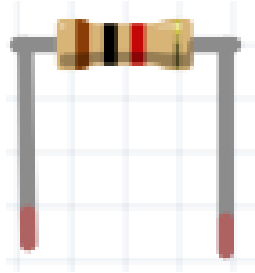
종류	설명&용도	장점&단점
권선 저항	도선을 길게 만든 저항	단순한 구조 고주파 회로에서 잡음 발생
탄소 피막 저항	DIP타입으로 제작 피막의 흠으로 저항크기 조절	저렴한 가격 온도에 따른 저항 값 변화가 크고 노이즈有
가변 저항	저항 값이 변하는 저항, 라디오 등 볼륨조절기에 사용됨	다양한 용도로 쓰일 수 있습니다.

<그림3-11> 저항 종류 비교

저항 값 읽기

저항값 읽기

Q 이 저항의 저항 값은?



_____ Ω

색 상 COLOR	저 항 환 산 표			
	첫째 수	둘째 수	셋째 수(곱하는 수)	오차표시
검 정(흑색)	0	0	1	
밤 색(갈색)	1	1	10	
빨 강(적색)	2	2	100	
주황색(동색)	3	3	1000	
노 랑(황색)	4	4	10000	
초록색(녹색)	5	5	100000	
파랑색(청색)	6	6	1000000	
보라색(자색)	7	7	10000000	
회 색(회색)	8	8	100000000	
흰 색(백색)	9	9	1000000000	
금 색			0.1	± 5%
은 색			0.01	± 10%
무 색				± 20%

<그림3-12> 저항 값 계산 표

꿀TIP

띠가 5개면

세 번째 띠까지 숫자를
세고, 네 번째 띠를
곱하는 수로 합니다.

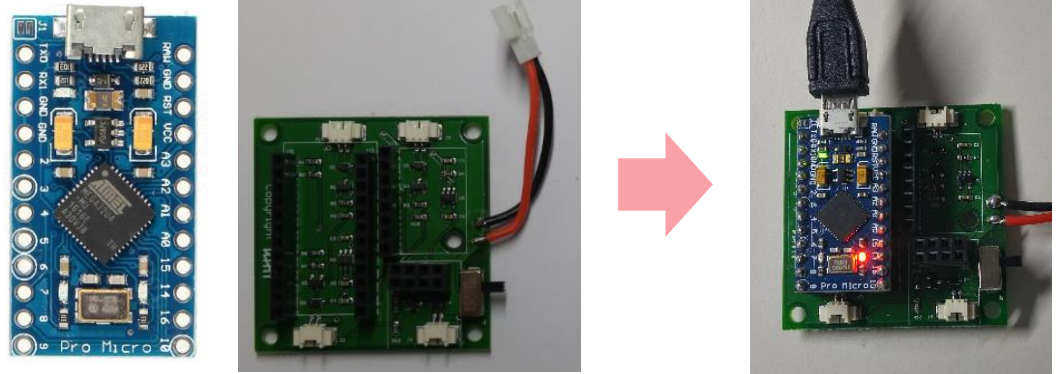
Ex) 주주검검금
 $330 \times 1 = 330\Omega$

A 첫 번째 띠가 밤색(갈색)이므로 첫 번째 수는 1
두 번째 띠가 검은색(흑색)이므로 두 번째 수는 0
세 번째 띠가 빨강색(적색)이므로 곱하는 수는 100
 $10 \times 100 = 1000\Omega = 1K\Omega$

LED 하드웨어 구성

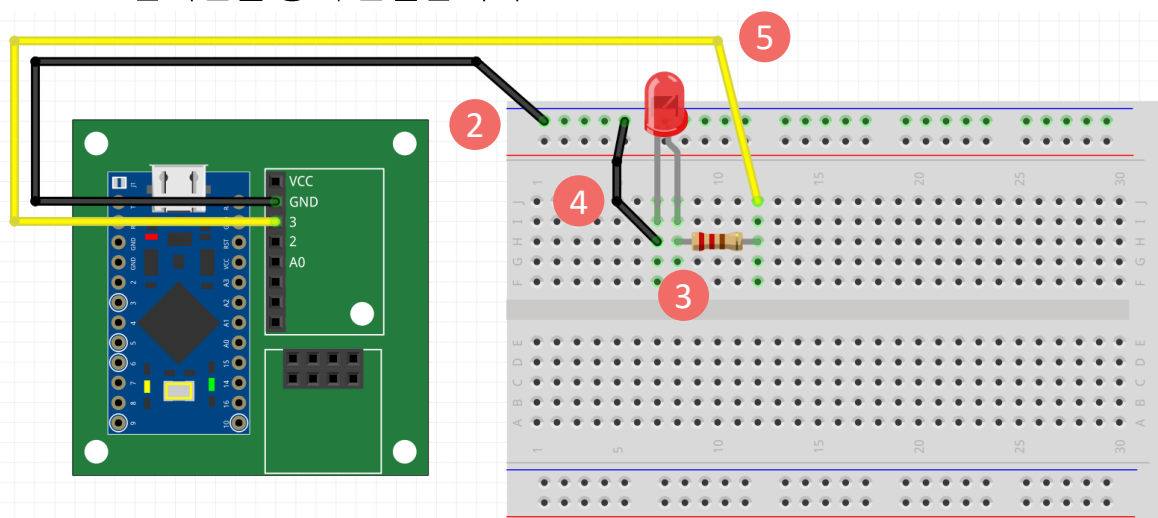
최종 구성

- 1 UBS가 연결되어 있는 아두이노를 메인 보드에 끼웁니다.(방향에 유의합니다)



<그림3-13> 아두이노와 메인 보드 연결

- 2 메인 보드의 GND핀(위에서2번째)을 빵판의 파란줄과 점퍼선을 통해 연결합니다.
- 3 LED와 저항을 빵판에 그림과 같이 꽂아 넣습니다.
(LED의 오른쪽과 저항의 왼쪽이 세로로 같은 라인에 위치)
- 4 LED의 -(짧은 쪽)와 빵판의 파란줄을 점퍼선을 통해 연결합니다.
- 5 메인 보드의 3번핀(위에서 3번째)을 저항의 우측과 점퍼선을 통해 연결합니다.



<그림3-14> 메인 보드와 LED 연결

꿀TIP LED 연결

LED는 다리가 긴 쪽이 +입니다.
전기는 +에서 -로 흐릅니다.

fritzing

- 6 만약, LED가 깜빡이지 않으면, 업로드가 되었는지, 선 연결이 잘 되었는지 확인합니다.

LED 속도 변화 해보기

LED 속도
변화

지금 아두이노는 1초간 켜지고 1초간 꺼지는 프로그램대로 동작하고 있습니다.

delay()의 괄호 안에 적절한 값을 적은 뒤 업로드하여 LED의 속도를 변화시켜 봅시다.

Q 0.5초마다 깜빡이게 만들어보세요.

```
ch3_2_3_blink
1 void setup() {
2   pinMode(3, OUTPUT);
3 }
4
5 void loop() {
6   digitalWrite(3, HIGH);
7   delay(    );
8   digitalWrite(3, LOW);
9   delay(    );
10 }
```

<그림3-15> LED속도 변화 해보기

LED 속도 변화 해보기 정답

A delay()의 괄호안에 1000 대신 500을 넣으면 0.5초마다 깜빡이게 됩니다.

delay()함수는 시간을 ms단위로 받기 때문에, 1000을 넣으면 1000ms = 1s로 1초가 되게 됩니다.

```
ch3_2_3_blink
1 void setup() {
2     pinMode(3, OUTPUT);
3 }
4
5 void loop() {
6     digitalWrite(3, HIGH);
7     delay(500);
8     digitalWrite(3, LOW);
9     delay(500);
10 }
```

<그림3-16> LED속도 변화 해보기 정답



WHIT