

2022 메이커 스페이스

크리스마스 무드등을 위한 아두이노 기초 지식

패스파인더 노 승 희

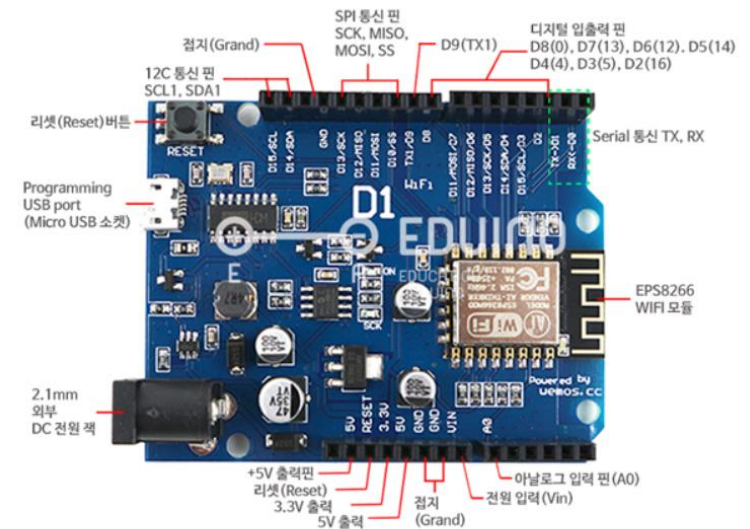


아두이노란

- 오픈소스를 지향하는 마이크로 컨트롤러를 내장한 기기 제어용 기판
- 컴퓨터 메인보드의 단순 버전으로 이 기판에 다양한 센서나 부품 등의 장치를 연결할 수 있음
- 이탈리아 디자인 전문학교에서 예술과 IT를 융합하여 교육하던 마시모 반지 교수가 예술학도라도 접근하기 쉽고 저렴한 전자 교육용 제품을 고민하며 탄생
- 전기/전자/컴퓨터를 모르는 일반인 누구나 쉽게 접근 가능

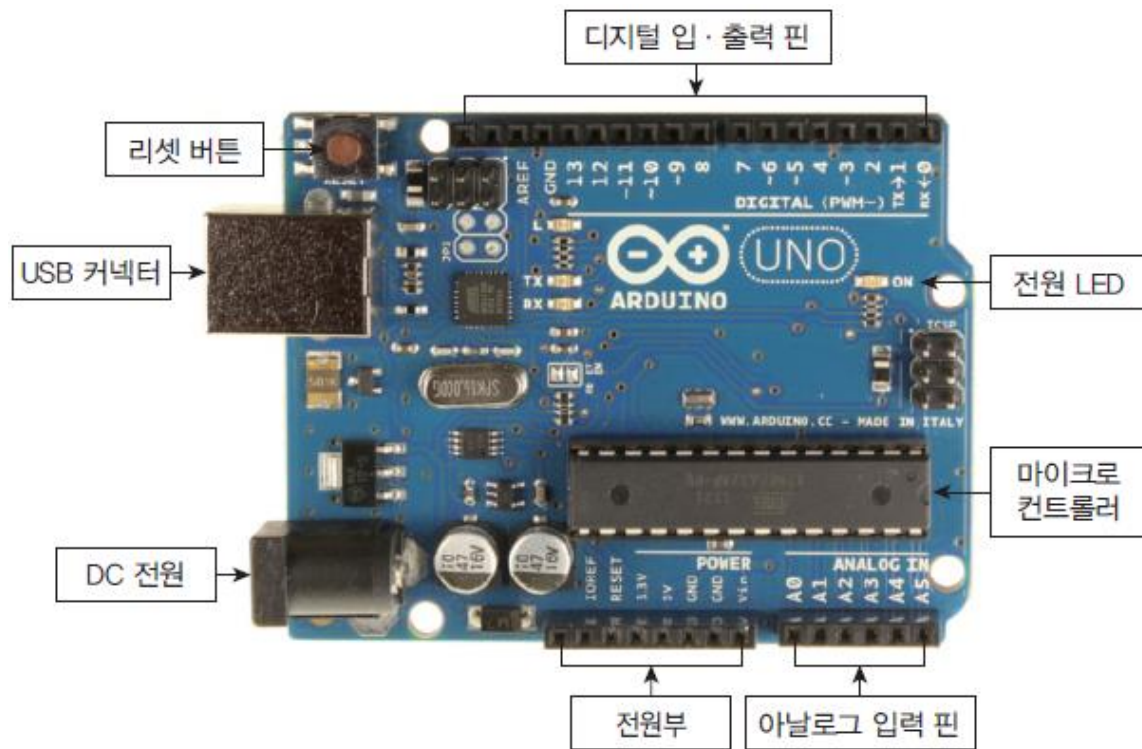


아두이노 종류



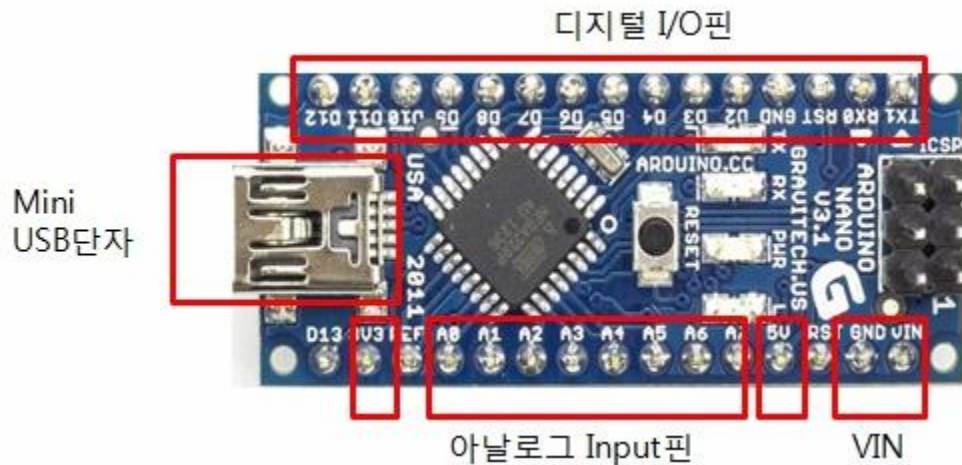
<WIFI ESP8266 D1 R1 보드 핀맵>

아두이노 우노(UNO)



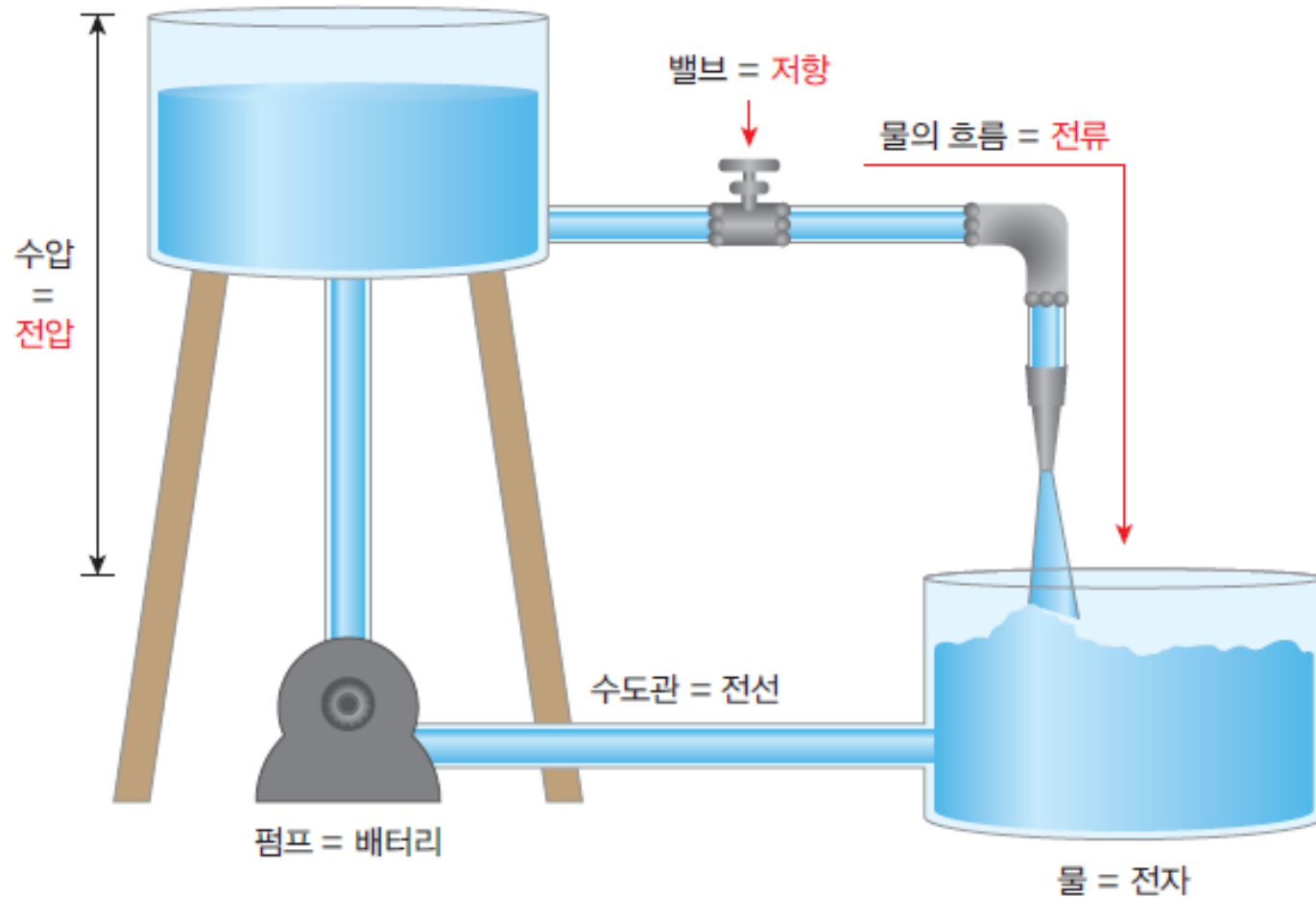
- ◆ 마이크로컨트롤러(Microcontroller): ATmega328P
- ◆ 동작 전원: 5 V
- ◆ 입력 전압(권고): 7 V ~ 12 V
- ◆ 입력 전압(제한): 6 V ~ 20 V
- ◆ 디지털 입·출력 핀: 14개 (6개의 PWM 출력포함)
- ◆ 아날로그 입력 핀: 6개
- ◆ 입·출력 핀의 DC 전류: 20 mA
- ◆ 3.3 V 핀의 DC 전류: 50 mA
- ◆ 플래쉬 메모리(Flash Memory): 32 KB (ATmega328), 이 중 부트로더 0.5 KB 사용
- ◆ SRAM: 2 KB (ATmega328P)
- ◆ EEPROM: 1 KB (ATmega328P)
- ◆ 클럭 속도(Clock Speed): 16 MHz

아두이노 나노(NANO)



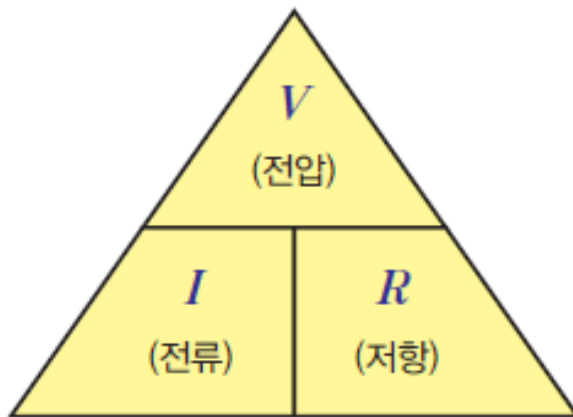
- ◆ 마이크로컨트롤러(Microcontroller):
=> ATmega328 or ATmega168
- ◆ 동작 전원: 5 V
- ◆ 입력 전압(권고): 7 V ~ 12 V
- ◆ 입력 전압(제한): 6 V ~ 20 V
- ◆ 디지털 입. 출력 핀: 14개 (6개의 PWM 출력포함)
- ◆ 아날로그 입력 핀: 8개
- ◆ 입. 출력 핀의 DC 전류: 40 mA
- ◆ 플래쉬 메모리(Flash Memory): 16KB(ATmega168), 32KB(ATmega328), 이 중 부트로더 2 KB 사용
- ◆ EEPROM: 512Byte(ATmega168), 1KB(ATmega328P)
- ◆ 클럭 속도(Clock Speed): 16 MHz

전류, 전압, 저항



- 전류(기호: I , 단위 A): 전하의 흐름
- 전압(기호: V , 단위 V): 전기장 안에서 전하가 갖는 전위의 차이
- 저항(기호: R , 단위 Ω): 도체에서 전류의 흐름을 방해하는 정도

옴의 법칙이란



$$V = IR$$

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

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

전선과 같은 도체를 통해서 전류가 흐를 때 전류의 흐름을 방해하는 것을 저항(resistance)이라고 하며, 옴(Ω)이라는 단위를 사용한다. 물탱크를 통해서 흐르는 물의 양을 줄이기 위해서는 밸브를 잠그면 될 것이다. 이때 밸브의 역할처럼 전류의 흐름은 저항으로 조절할 수 있으며 밸브 외에도 전선의 재질과 굵기 그리고 길이에 따라 저항값이 달라진다.

저항이란



- 저항은 전류의 흐름을 방해하는 성질을 가지고 있는 소자로 저항의 크기는 물질의 종류와 형태에 따라 달라진다.
- 저항의 크기는 띠의 색과 개수로 구분한다.(우측 그림 참고)
- 저항을 이용하는 목적으로는 전류 제한, 전압 분배, 로직 레벨 설정이 있다.

첫번째 숫자	두번째 숫자	자리수	오차
0	0	x 1	
1	1	x 10	±1%
2	2	x 100	±2%
3	3	x 1K	
4	4	x 10K	
5	5	x 100K	
6	6	x 1M	
7	7		
8	8	x 0.1	±5%
9	9	x 0.01	±10%

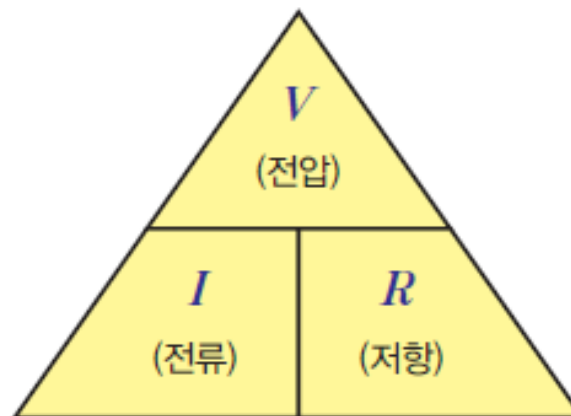
청색 LED의 필요 저항

- 청색 LED의 최소 전압 3.4V
- 핀 전류값 20mA
- 아두이노 전압 5V

LED 권장 저항값 = (입력전압 - LED를 켜기 위한 최소전압)/전류

LED 권장 저항값 = (5V - 3.4V) / 0.02A

1.6/0.02 = **80Ω**



$$V = IR$$

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

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


색상	구 분	최소전압	최대전압	전류(일반)	전류(최대)
적●	Red	1.8V	2.3V	20 mA	50 mA
등●	Orange	2.0V	2.3V	30 mA	50 mA
황●	Real Yellow	2.0V	2.8V	20 mA	50 mA
초●	emerald Green	1.8V	2.3V	20 mA	50 mA
초●	Real Green	3.0V	3.6V	20 mA	50 mA
청●	sky Blue	3.4V	3.8V	20 mA	50 mA
청●	Real Blue	3.4V	3.8V	20 mA	50 mA
자●	Pink	3.4V	3.8V	20 mA	50 mA
백○	White	3.4V	4.0V	20 mA	50 mA

Arduino IDE 설치

- 아두이노 통합개발환경(Arduino IDE)은 프로세싱을 기반으로 개발된 편집기, 컴파일러, 업로더 등이 합쳐진 소프트웨어 환경
- 홈페이지: <https://www.arduino.cc>

[HARDWARE](#) [SOFTWARE](#) [CLOUD](#) [DOCUMENTATION](#) [COMMUNITY](#) [BLOG](#) [ABOUT](#)

Downloads

 **Arduino IDE 2.0.2**

The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger.

For more details, please refer to the [Arduino IDE 2.0 documentation](#).

Nightly builds with the latest bugfixes are available through the section below.

SOURCE CODE

The Arduino IDE 2.0 is open source and its source code is hosted on [GitHub](#).

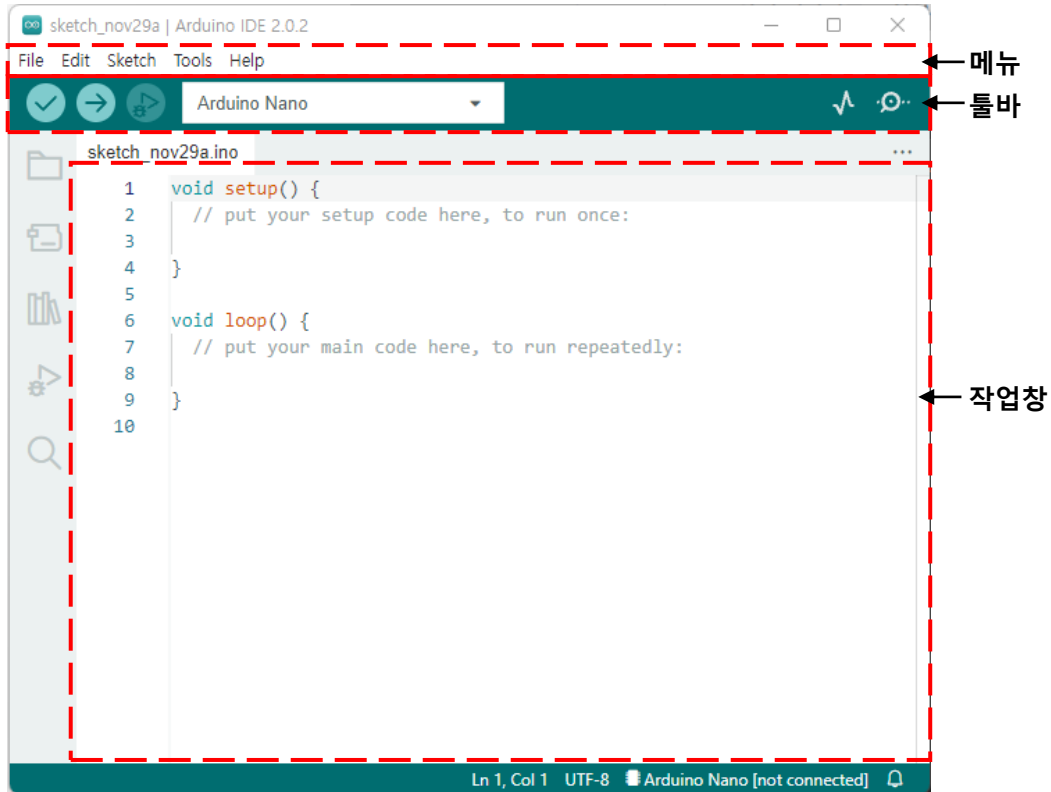
DOWNLOAD OPTIONS

Windows Win 10 and newer, 64 bits
Windows MSI installer
Windows ZIP file

Linux AppImage 64 bits (X86-64)
Linux ZIP file 64 bits (X86-64)

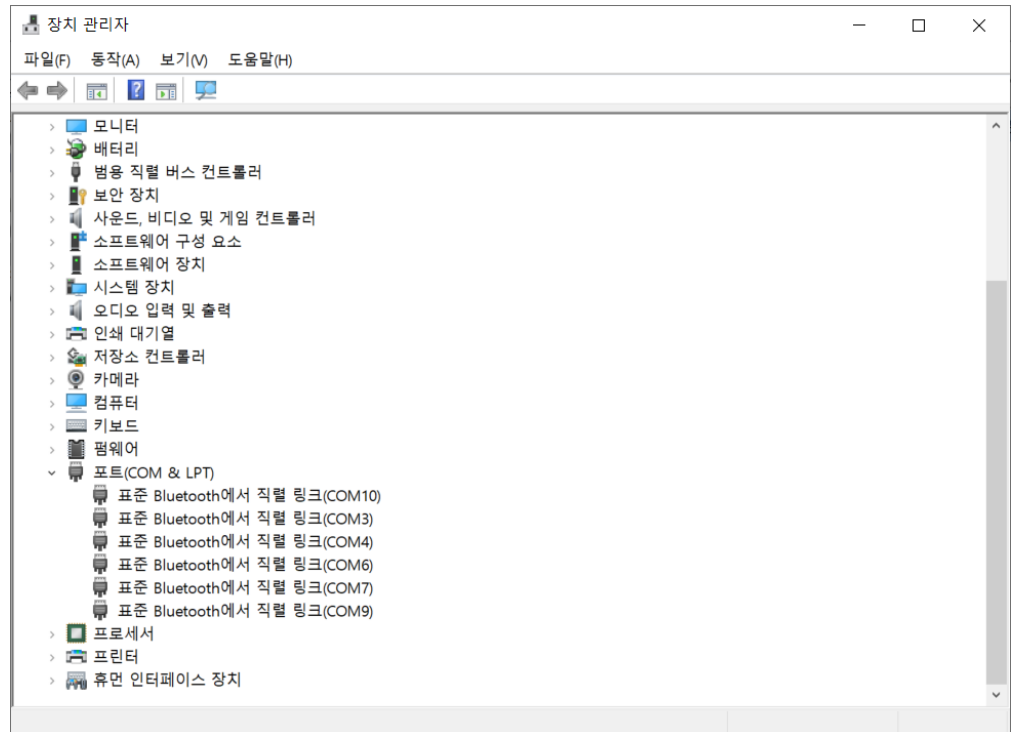
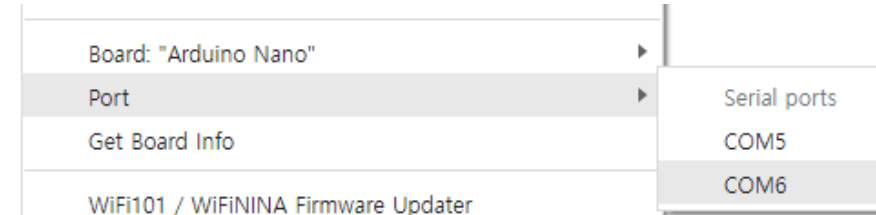
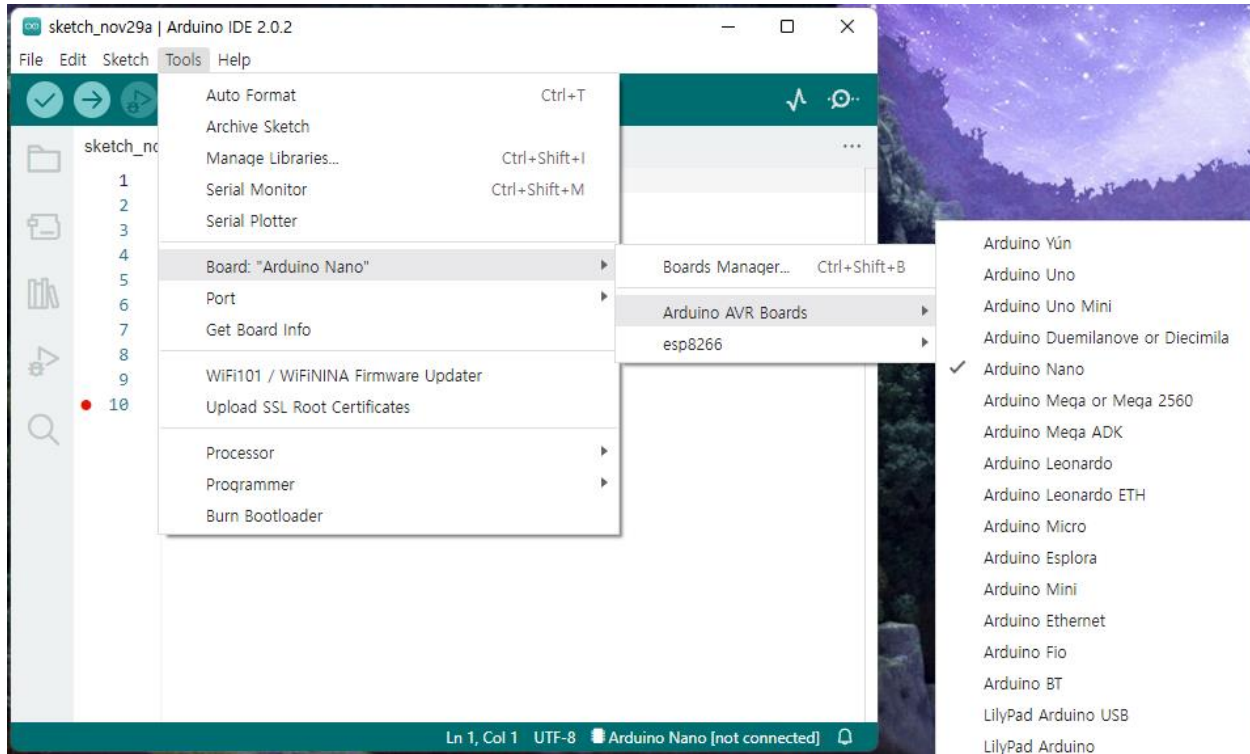
macOS 10.14: "Mojave" or newer, 64 bits

아두이노 IDE 구성 요소

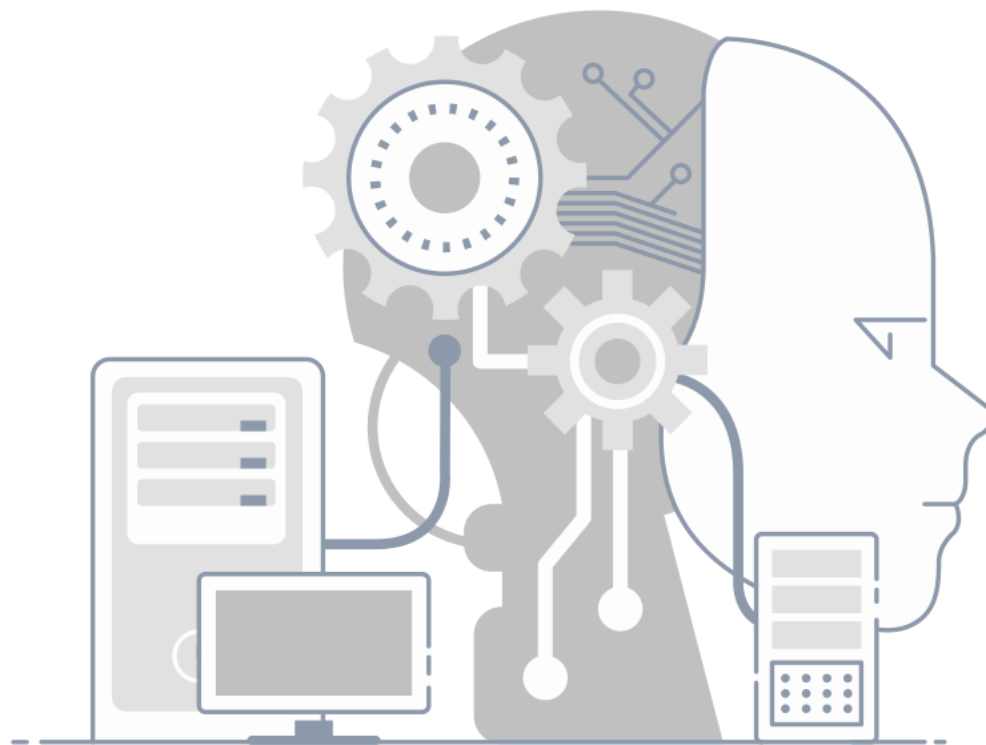


툴바에 보이는 도구는 확인, 업로드, 새파일, 열기, 저장, 시리얼 모니터가 있다.

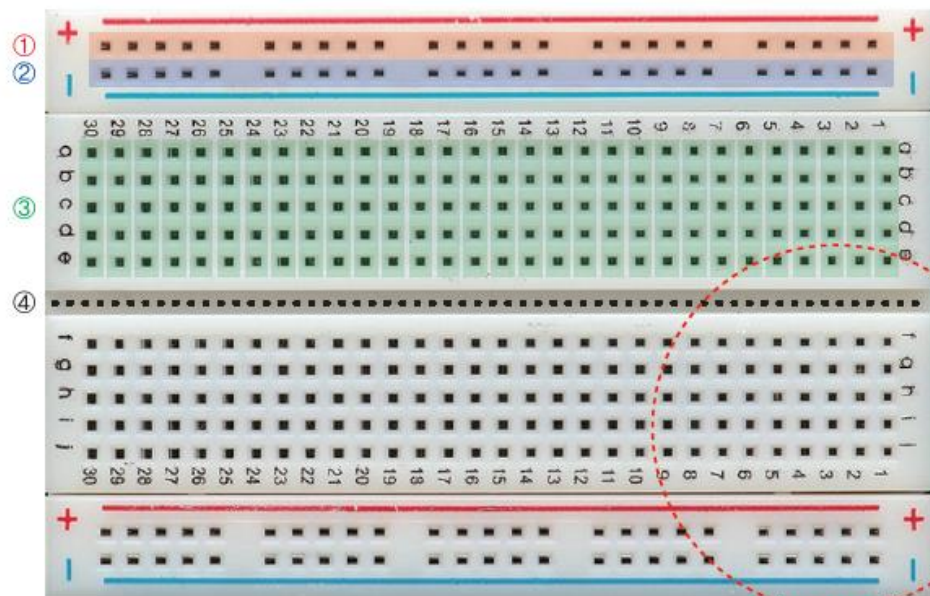
아두이노 보드 매칭 및 포트 연결



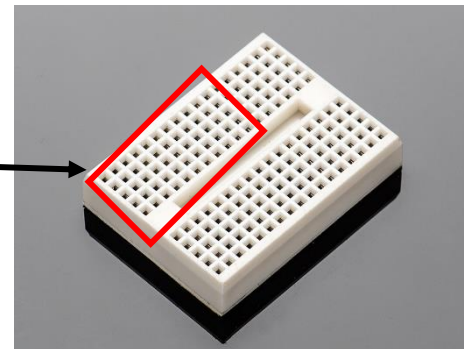
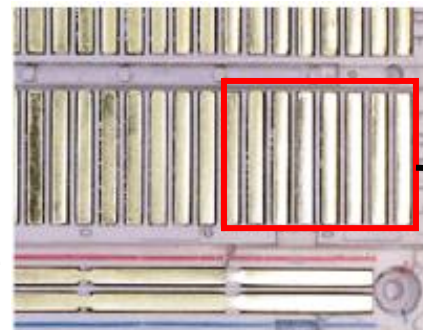
센서 활용하기



브레드보드 구조

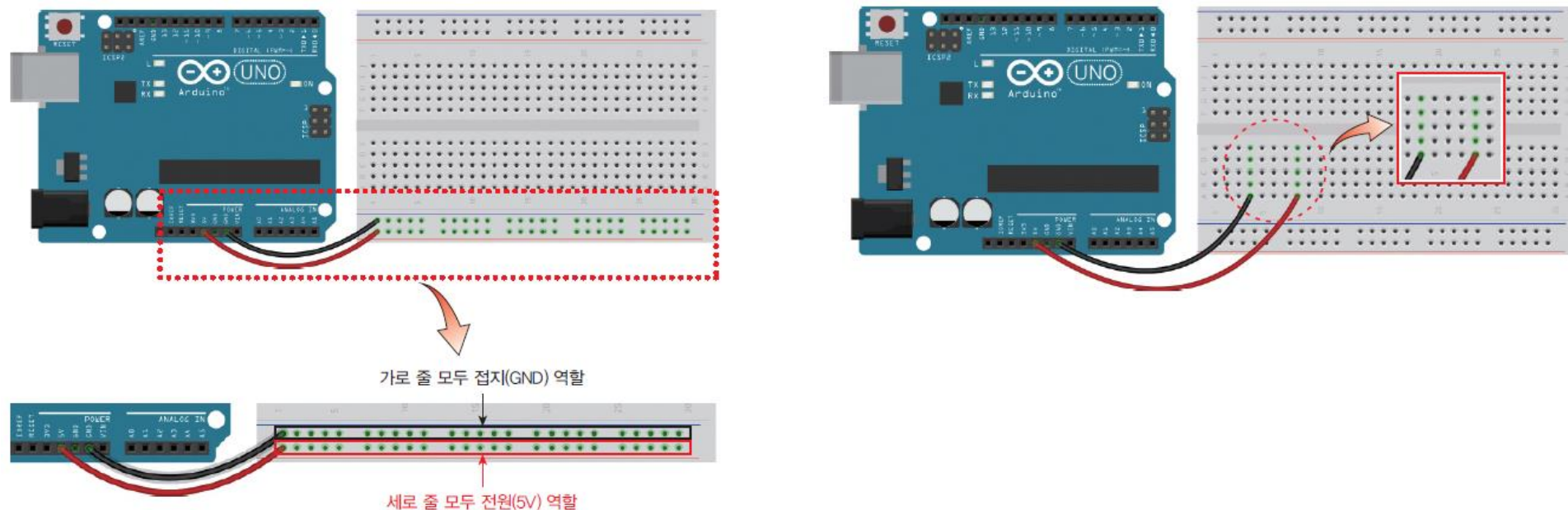


내부



- ① 가로 구멍 전체가 연결되어 있다.
- ② 가로 구멍 전체가 연결되어 있다.
- ③ 세로 구멍 5개가 연결되어 있다.
- ④ 가운데를 기준으로 위·아래는 서로 다른 구간이다.

브레드보드 사용 예



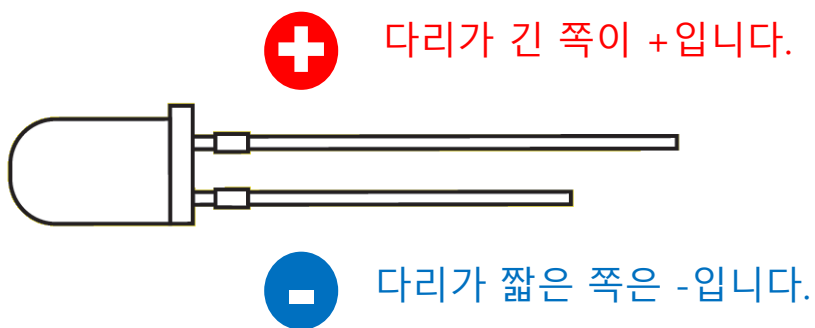
왼쪽 그림과 같이 브레드보드의 가로줄 각각에 전원과 그라운드를 연결하면 각 가로선 전체를 전원과 그라운드로 사용할 수 있다. 브레드보드의 세로줄을 오른쪽 그림처럼 활용해서 전원과 그라운드를 확장하는 방법도 가능하며 필요할 경우 다른 세로줄로 계속 연결해서 확장할 수 있다.

LED란?

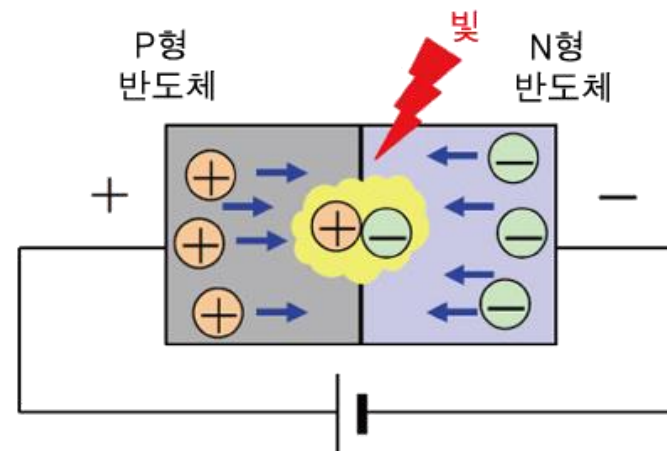


한쪽 방향으로만 전류가 흐르도록 제어하는 반도체 소자를 다이오드라고 하며, 다이오드 중 전기 에너지를 빛 에너지로 변환하는 것을 LED(Light Emitting Diode)라고 한다.
LED는 다양한 색깔의 빛을 낼 수 있으며, 에너지 효율이 높고 직진성이 좋아 멀리서도 잘 보인다는 장점이 있다.

LED 구조 및 원리

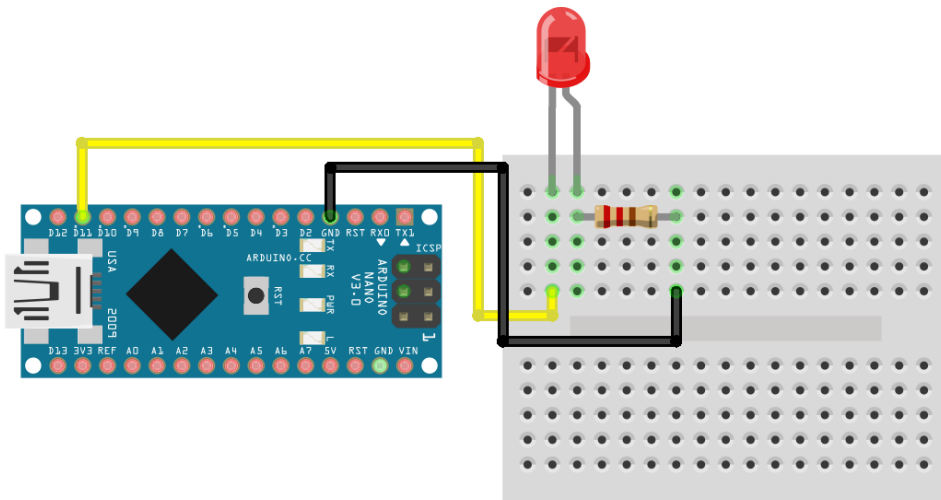


LED는 2개의 전극 단자로 구성되어 있으며, 긴 단자는 애노드(Anode), 짧은 단자는 캐소드(Cathode)라 부른다. 애노드에 (+)전극을 캐소드에 (-)전극을 연결하면 LED가 켜지게 된다.



LED는 발광 다이오드라고 하며, Ga(갈륨), P(인), As(비소)를 재료로 하여 반도체. 다이오드의 특성을 가지고 있으며, 전류를 흐르게 하면 각종 색으로 빛을 발한다. LED에 전류를 흐르면 각 반도체에 있는 전자와 정공이 결합하게 되는데 이 때 생기는 에너지가 빛이 된다.

아두이노 보드 디지털 핀에 LED와 저항 연결



```
// setup 함수는 보드에 전원 공급 또는 리셋 버튼을 눌렀을 때 한 번 실행된다.  
void setup()  
{  
  pinMode (11, OUTPUT);  // 11번 디지털 핀을 출력으로 설정  
}  
  
// loop 함수는 전원이 켜져 있는 동안 무한 반복해서 실행된다.  
void loop()  
{  
  digitalWrite (11, HIGH);  // 11번 핀에 HIGH 설정. LED 켜짐  
  delay (1000);             // 1000 ms(1초)동안 대기(시간 지연)  
  digitalWrite (11, LOW);   // 11번 핀에 LOW 설정. LED 꺼짐  
  delay (1000);             // 1000 ms(1초)동안 대기(시간 지연)  
}
```

330 Ω 의 저항과 LED의 긴 쪽을 디지털 핀 11번 핀에 연결하고 짧은 쪽을 그라운드 즉 GND(Ground) 핀에 연결한다. 아두이노 보드에 내장된 LED에 이미 저항이 연결되어 있어서 이것과 같은 선으로 연결된 11번 핀의 경우 저항 없이 회로를 구성해도 문제 되지 않는다.

함수 설명

```
void setup()
```

```
{
```

```
    // put your setup code here, to run once:
```

```
}
```

아두이노가 전원이 인가될 때 단 한번

실행이 되는 함수

함수 설명

```
void loop()
```

```
{
```

```
//put your main code here, to run repeatedly:
```

```
}
```

setup 함수 실행 후 아두이노 전원이 꺼

질 때까지 반복적으로 실행되는 함수

함수 설명

pinMode(사용할 핀의 번호 , 입력(INPUT)
또는 출력(OUTPUT));

핀의 상태를 입력 또는 출력으로 설정

* 디지털 입/출력시 사용

함수 설명

`delay(대기할 시간(밀리초 / ms 단위));`

입력한 시간만큼 프로그램 일시 중지(대기, 유지)

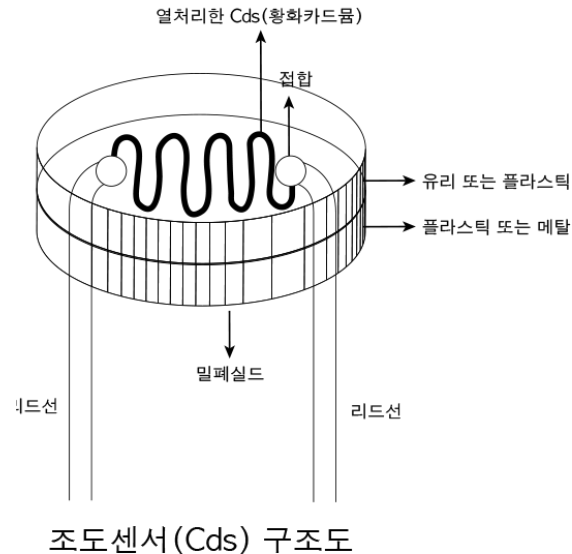
$* 1\text{ms} = 1/1000\text{s}$

함수 설명

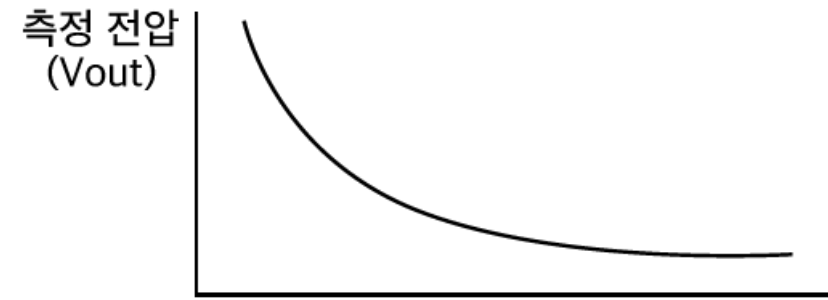
`digitalWrite(사용할 핀의 번호 , ON(HIGH, 1) 또는 OFF(LOW, 0));`

pinMode에서 출력으로 설정한 핀의 상태를 제어(**ON**, **OFF**)

광센서(CdS 센서) 동작 원리



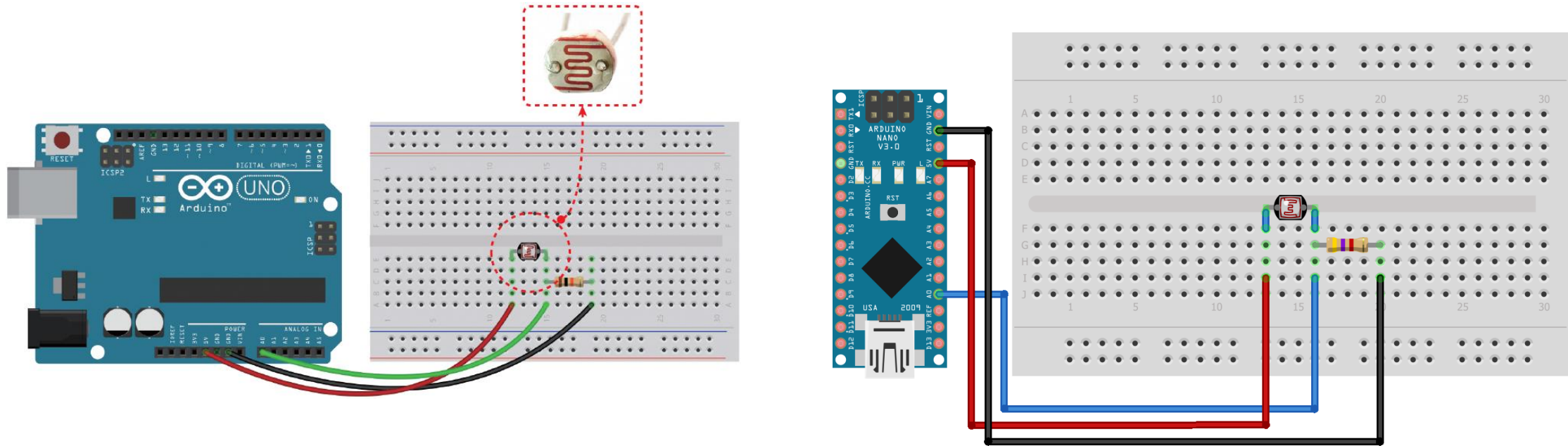
빛을 받으면 그 에너지의 영향으로 내부에 자유 전자가 발생하여, 전도율이 변하는 이른바 광전효과를 가진 소자로 구성되어 있다. 황화카드뮴(CdS)을 소자로 사용하는 경우, CdS 센서라고 부른다.



풀업 저항 사용시 밝기에 대한 측정 전압

조도 센서는 빛의 양에 따라 전도율이 변한다. 빛의 양이 많아질수록 전도율이 증가해 저항이 감소하고, 빛의 양이 적어질수록 전도율이 감소해 저항이 증가하는 원리이다.

광센서 연결



CdS 리드선의 한쪽에는 5 V를 연결하고 나머지 다른 쪽 리드선은 10 K Ω 저항과 아두이노 보드의 아날로그 A0번 핀에 연결한다. 회로에 사용된 저항의 크기에 따라 CdS 센서에 가해지는 전압이 변경된다. 13번 핀을 사용하게 되면 감지된 Cds 센서값에 따라 LED 밝기 조절은 되지 않는다. 따라서 analogWrite 함수를 통해서 LED 밝기가 조절되는 것을 확인하려면 PWM 출력이 되는 핀에 LED를 연결해야 한다.

광센서 연결

광센서

```
const int sensorPin = A0; // 광센서 값을 측정하기 위한 핀. A0에 연결

void setup()
{
  Serial.begin(9600); //시리얼 통신 초기화
}

void loop()
{
  int readValue = analogRead (sensorPin); // 센서값 읽음
  Serial.println(readValue);
  delay (2000);
}
```

광센서 + LED

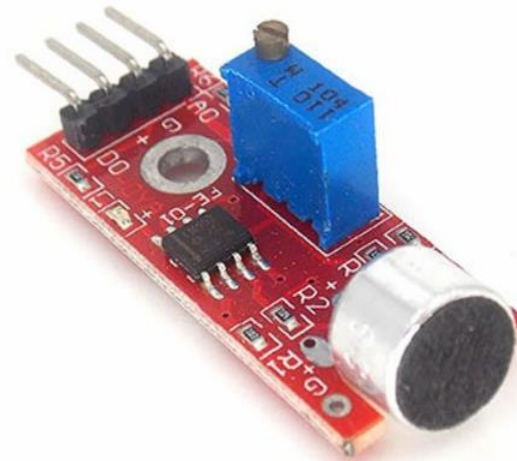
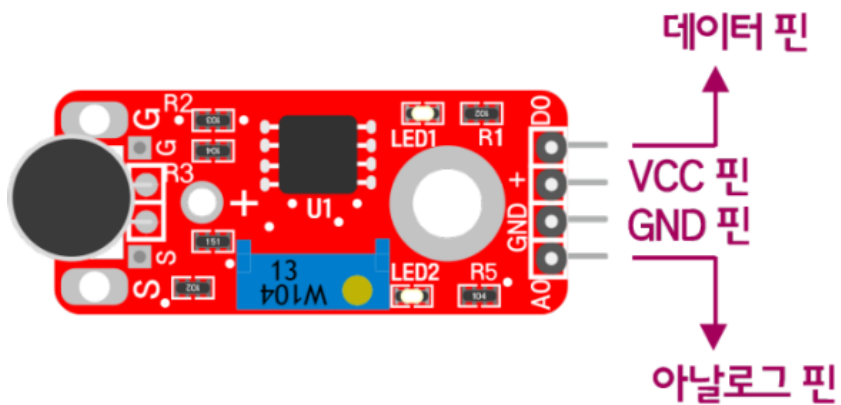
```
const int sensorPin = A0; // 광센서 값을 측정하기 위한 핀. A0에 연결

void setup()
{
  Serial.begin(9600); //시리얼 통신 초기화
  pinMode(9, OUTPUT);
}

void loop()
{
  int readValue = analogRead (sensorPin); // 센서값 읽음
  Serial.println(readValue);

  if( readValue < 250 )
    digitalWrite( 9, HIGH );
  else
    digitalWrite( 9, LOW );
}
```

고감도 사운드 센서(ky-038) 동작 원리



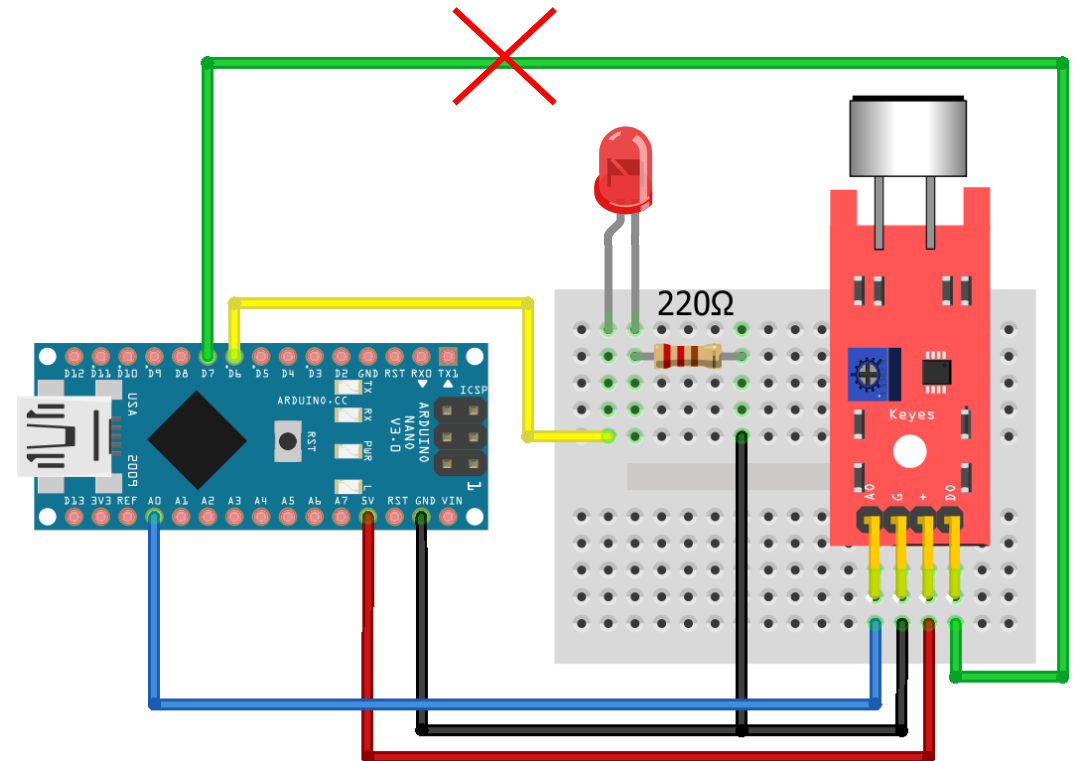
고감도 사운드 센서는 소리를 감지하는 센서로 센서 주변에서 발생하는 소리를 마이크로 모아, 소리의 크기를 증폭시켜 출력한다. 센서는 총 4개의 핀으로 구성되어 있으며, 'A0핀, GND핀, +핀, D0핀'로 표기된 핀은 아두이노 보드의 '아날로그 입력핀, GND핀, 5V핀, 디지털 핀'에 연결할 수 있다.

고감도 사운드 센서(ky-038) 연결

```
int led = 6;
int sensor = A0; //고감도 사운드 센서가 연결된 A0(아날로그 핀)을
sensor 변수에 저장
int val = 1;

void setup()
{
  Serial.begin(9600); //시리얼 통신 초기화
  pinMode(led, OUTPUT); //led 출력으로 설정
  pinMode(sensor, INPUT); // sensor 입력으로 설정
}

void loop()
{
  val = analogRead (sensor); // sensor의 아날로그 입력값 val 변수에 저장
  int light = map(val, 0, 1023, 0, 255);
  analogWrite (led, light);
}
```



함수 설명

`map(변환할 변수 , 현재 변수의 가장 낮은 값 , 현재 변수의 가장 높은 값 , 변환되는 변수의 가장 낮은 값 , 변환되는 변수의 가장 높은 값);`

특정 변수의 값의 범위를 변경해야 하는 경우 사용

감사합니다!

