

사물인터넷

3주차 아두이노 기초 사용 방법.

아두이노 함수 구조

```
void setup() {  
    .....  
}
```

하드웨어 및
초기 설정

```
void loop() {  
    .....  
}
```

반복적 동작 설정

setup()

- 핀 모드 설정, 시리얼 통신 속도 설정 등 주로 하드웨어 상태 설정에 사용 되는 함수

- 장치 부팅 후 처음 한번만 동작한다

```
void setup() {  
    .....  
}
```

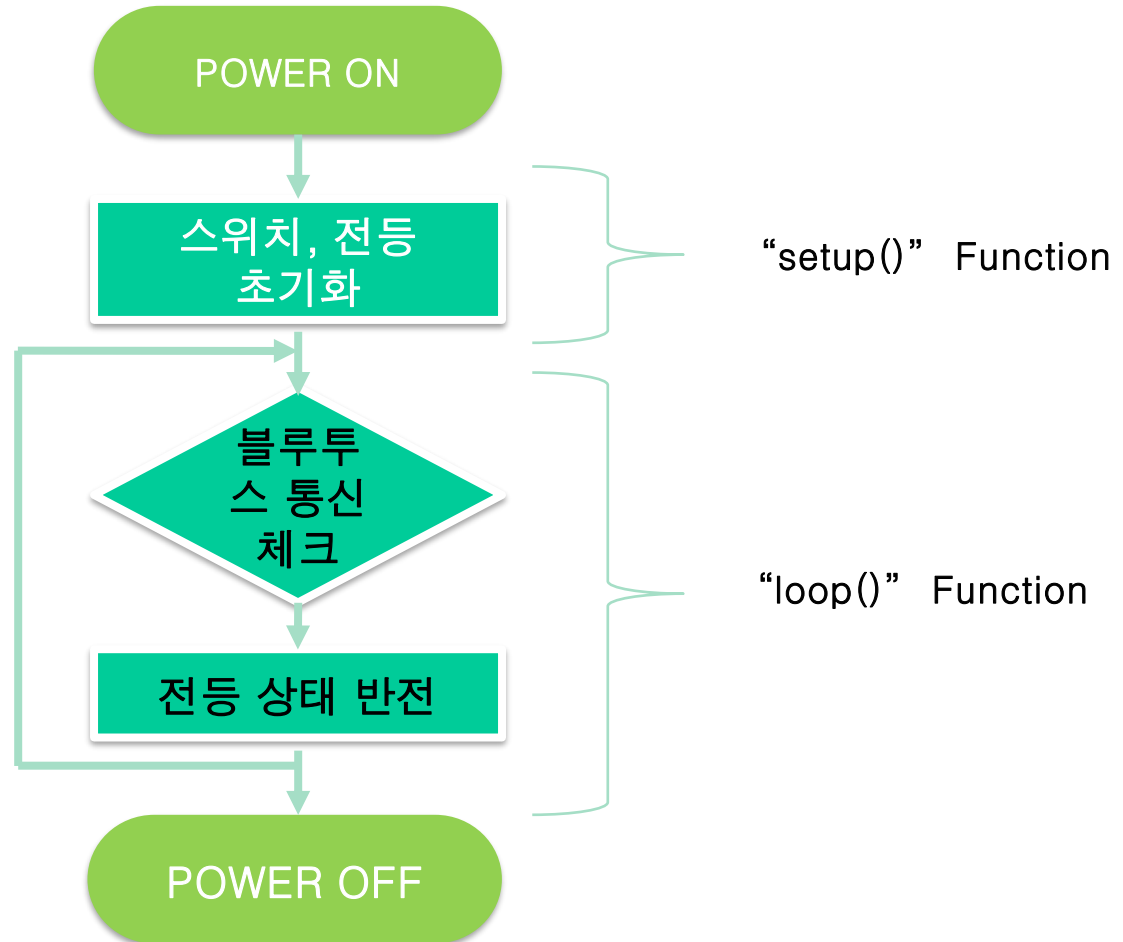
- 모든 설정이 반듯이 setup()에서만 해야 되는 것은 아니다

void loop()

- while과 같이 무한반복 되는 함수이다
- 실제 동작코드를 넣어주면 된다
- 아두이노에서 처리되어져야 할 동작을 1회를 정의하여 반복 실행되어 지도록 한다
- 대부분의 제품들은 켜진 이후로 꺼지지 않고 계속 반복적인 동작을 한다 이 반복적 동작을 정의하기 위한 함수이다

setup과 loop 함수 이해

■ 예) 스마트폰을 이용한 전등 스위치 제어



함수 소개

- pinMode(pin, mode)
- digitalWrite(pin, value)
- delay(ms)

함수 소개 – pinMode

■ pinMode(pin, mode)

❖ 핀의 입출력 모드를 설정하는 함수

❖ pin : 설정할 핀의 번호

❖ mode : 설정할 모드

- INPUT : 입력 모드
- OUTPUT : 출력 모드
- INPUT_PULLUP : 풀업 입력 모드, 잘 안 씀

❖ ex) 13번 핀의 출력을 OUTPUT으로 설정하려면 pinMode(13, OUTPUT);

함수 소개 – digitalWrite

■ digitalWrite(pin, value)

- ❖ 핀의 출력을 정하는 함수, OUTPUT 모드여야 한다
- ❖ pin : 출력 설정 할 핀의 번호
- ❖ value : 출력할 값
 - HIGH or 1 : HIGH 로 출력
 - LOW or 0 : LOW 로 출력
- ❖ ex) OUTPUT 출력인 13번 핀을 HIGH 상태로 바꿀 경우
digitalWrite(13, HIGH);

함수 소개 – delay

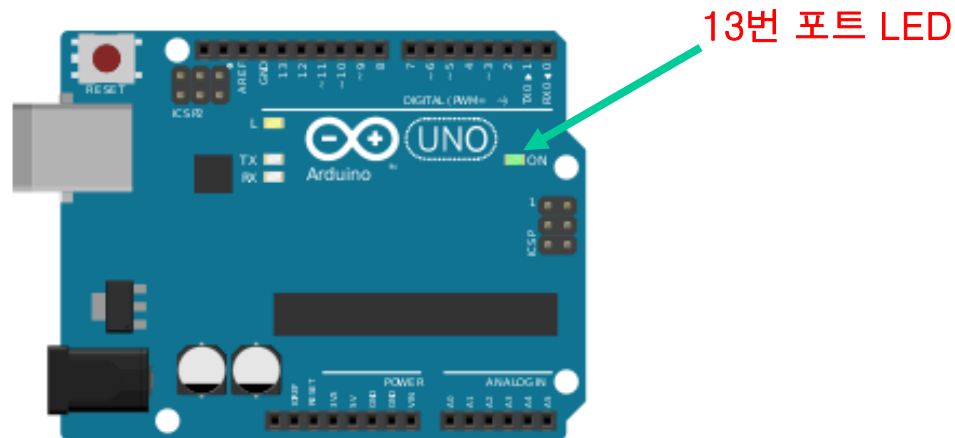
■ delay(ms)

- ❖ ms 만큼 프로그램을 일시 정지하는 함수
- ❖ ms : 일시 정지할 시간, millisecond
- ❖ ex) 1초간 프로그램 일시 정지
delay(1000);

실습

■ 목표

- ❖ `delay()` 와 `digitalWrite()` 를 1번씩만 써서 LED 가 깜빡이는 예제를 만들어보자



해답

```
int value;  
  
void setup() {  
    value = HIGH;  
    pinMode(LED_BUILTIN, OUTPUT);  
}  
  
void loop() {  
    digitalWrite(LED_BUILTIN, value);  
    value ~=value;  
    delay(1000);  
}
```

digitalWrite()는 한번
사용되어졌다. 하지만
loop()에 진입 시 상태
는 반전된다.

고찰

- `loop()`는 LED를 On/Off 동작의 반복이다. LED의 상태는 `loop()`의 호출에 따라 달라질 뿐이다
- `LED_BUILTIN`는 이미 선언되어진 값이며, port 번호 13이다. 13번 포트는 외부로 나와 있으니 사용할 때 주의하자

문제

■ value 변수를 loop() 안에 넣으면 어떻게 될까?

❖ 현상과 이유를 써주세요

■ LED 를 13번 포트가 아니라 10번 포트에 연결하고 싶다.
어디를 수정해야하는 가?

❖ 현재 코드를 쓰고 변경 코드를 적어주세요

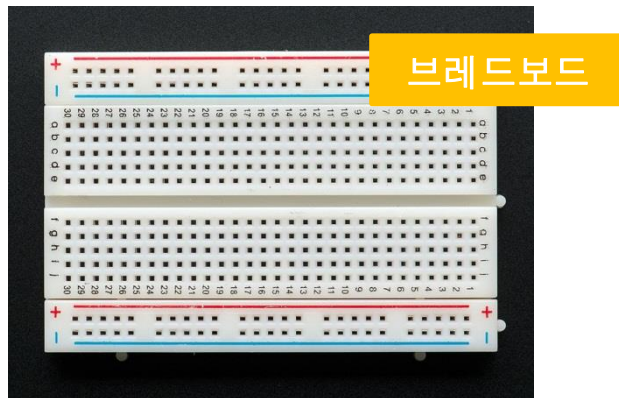
■ Hello Arduino 문자열을 출력하고 싶다. 어디에 추가를 해야 되는 가?

❖ 해당 함수와 주변 코드를 같이 적어주세요

02. Breadboard 이해

Breadboard 란?

- ‘브레드보드’ 라고 부르며 단어의 의미를 따서 ‘빵판’ 으로 부르기도 한다
- 0.25cm간격의 홀이 뚫려있어 간단한 회로나 교육용으로 사용한다
- 일반적으로 DIP타입이라는 리드가 달린 제품 전부를 사용할 수 있다
- 또 한 빵판으로 불리는 이유는 초기의 전자회로를 개발 할 때 빵을 자르는 나무 판에 제작하였기 때문이다



Breadboard 장점

■ 납땜이 필요 없다

❖ 간단한 회로의 경우 굳이 납땜을 하지 않고 구현이 가능하다

■ 부품 재활용이 가능하다

❖ 납땜을 하더라도 재활용이 가능하지만 선을 잘라내는 등의 불편함이 있다 하지만 브레드보드의 경우 전원을 차단하고 부품을 빼내는 것만으로도 재활용할 수 있다

■ 선을 따라 확인이 가능하다

❖ 회로가 작동하지 않을 경우 어디를 어떻게 연결했는지 확인 가능하다 PCB나 납땜의 경우는 확인하기가 힘들다 이에 비해서 브레드 보드는 직접 연결된 선을 따라 확인이 가능하다

Breadboard 단점

■ 회로가 복잡할수록 크기가 커진다

- ❖ 회로가 복잡해지는 경우 브레드보드에 올린 회로의 부피가 상당히 커진다 이는 PCB나 납땜에 비하여 정리가 덜 되기 때문에 크기가 커지게 된다

■ 소형화가 힘들다

- ❖ 선을 정리하여 패턴을 그리는 것이 아니기 때문에 소형화하기 힘들다 그런 이유에서 실험용으로만 쓰이며, 실제 상품화에는 적용되지 않는다.

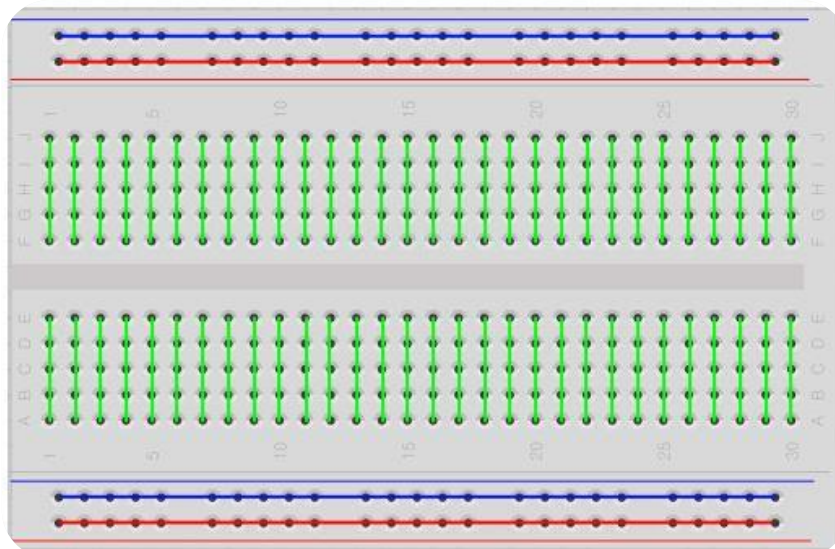
■ DIP타입 외의 제품을 사용할 수 없다

- ❖ 부품을 없어서 사용하는(실장) 타입의 제품은 변환용 기판 없이는 불가능하다.

Breadboard 연결 방식

■ 브레드보드 결선방식

- ❖ 브레드보드는 하나의 행이나 열이 모두 연결되어있다. 이는 다른 부품과 연결에 있어 편의를 제공하기 위함이다.
- ❖ 연결할 때는 Line 단위로 생각한다!!



붉은색 행과 파란색 행, 이 두개의 행은 각각 행 단위로 연결되어있다.

각 열마다 아래로 연결되어있다.

결선 시 주의 사항

■ 주의사항 - 브레드보드 결선방식

❖ 잘 못 된 결선에 의해 손상된 브레드보드



다른 극성을 서로 붙인 뒤 전원을 연결할 경우 손상될 수 있다

결선 시 주의 사항

■ 가급적 다른 색상의 점퍼선을 이용할 것

- ❖ 같은 색상의 선을 이용하는 것은 일관성을 주지만 결선이 잘 되어 있는지 판단하기 어렵게 만든다. 물론 선을 꼭 따라가면 되지만 중간에 같은 색상의 다른 선으로 혼동하여 넘어가는 경우가 있으니 주의하도록 한다.

■ 빨간색 선은 +단자, 검정색 선은 -단자로 연결한다

- ❖ 관습적으로 빨강과 검정색을 전원으로 사용하며 만약 바꿀 경우 혼돈의 여지가 있다.

부품 고정 시 주의 사항

■ 접착 물을 닦아낸다

- ❖ 리드(다리)가 달린 부품(DIP 저항, 캐패시터)은 다리의 끝부분을 접착시켜 떨어지지 않게 고정하였기 때문에 올바른 연결이라도 연결이 되지 않는 경우가 생긴다. 이 경우는 전원을 차단하고 다리 부분에 접착제가 묻어있는지 확인 후 닦아내어 연결 후 다시 결선해 본다.



포장된 리드저항

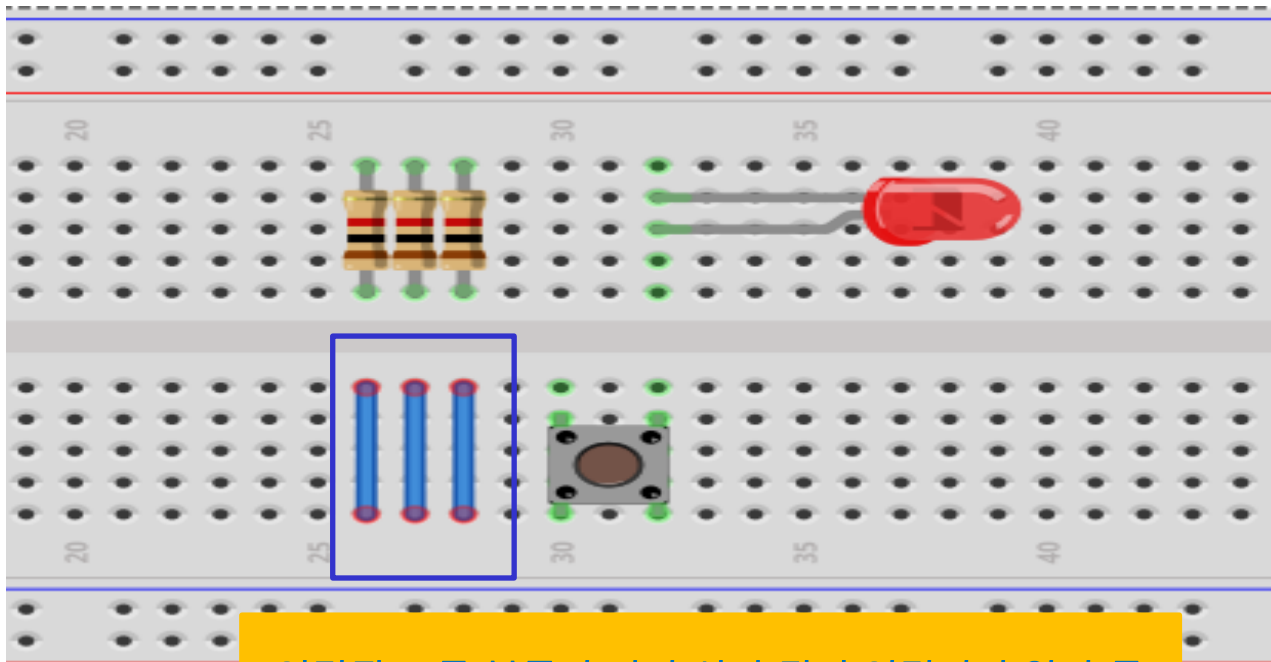


테이프 접착된 상태

부품 고정 시 주의 사항

■ 브레드보드의 같은 행과 열에 놓지 않는다

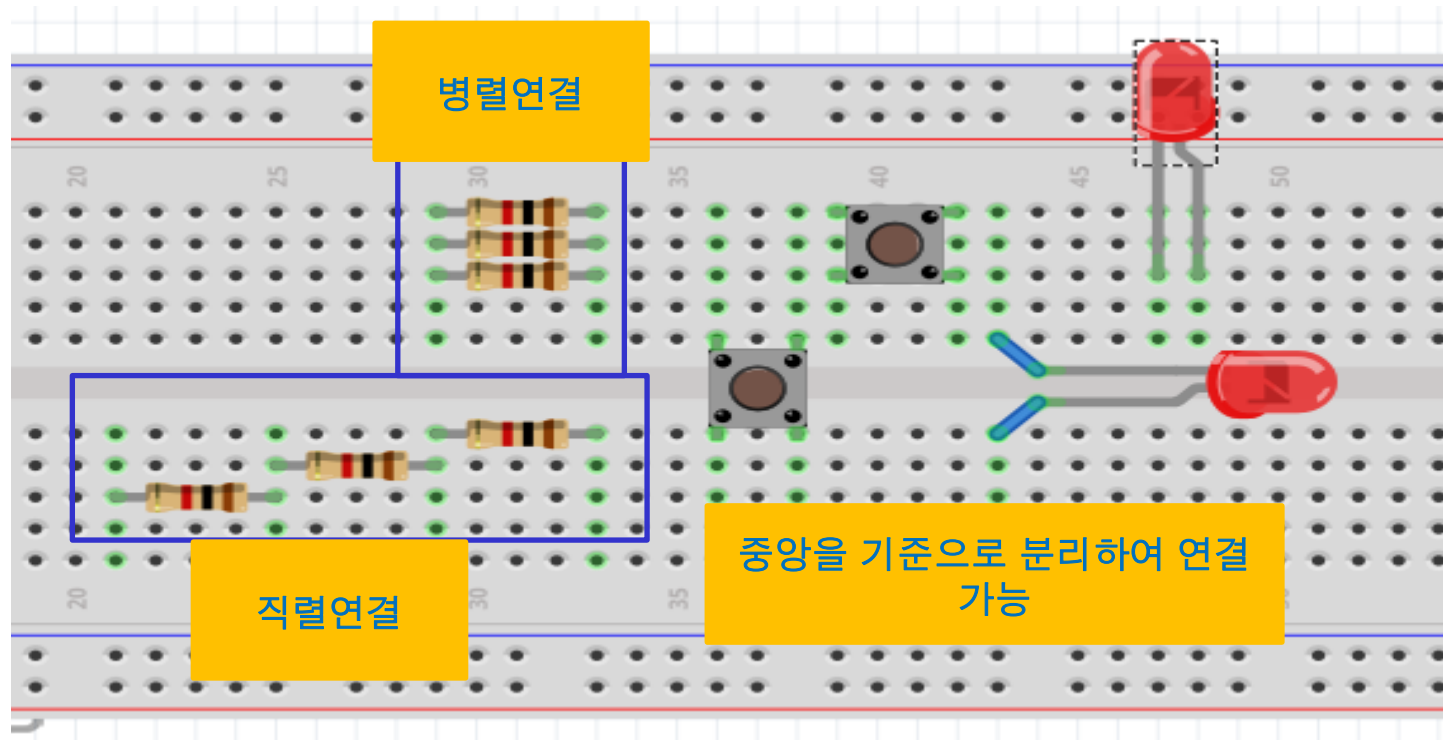
- ❖ 부품을 장착할 때 다리를 모두 같은 행이나 열에 놓게 되면 부품을 같은 위치에 놓게 되는 것이므로 이를 조심하여야 한다.



연결된 모든 부품이 파란 선과 같이 연결되어 있어 문제가 있다!!

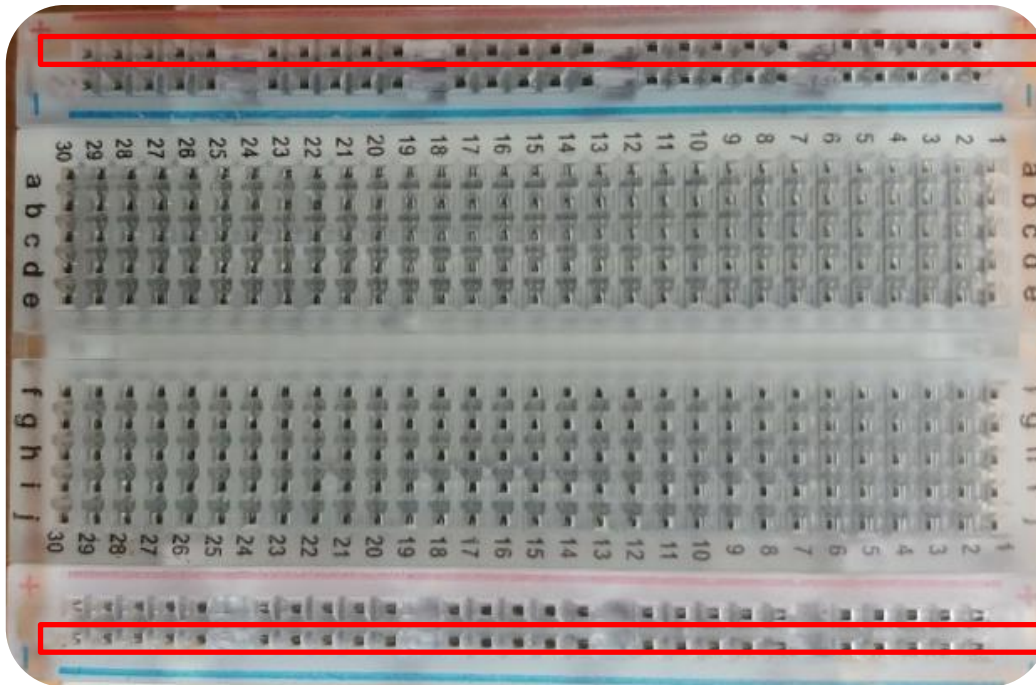
부품 연결 예시

■ 올바른 부품 장착 및 연결



전원 연결 방법

■ 전원 연결 시 위쪽을 +로 하면 아래쪽을 -로 한다.



이곳에 “+”를 연결
하였으면 바로 아래의
“-”는 가급적 사용하지 않는다

“-”를 이곳에 연결
한다

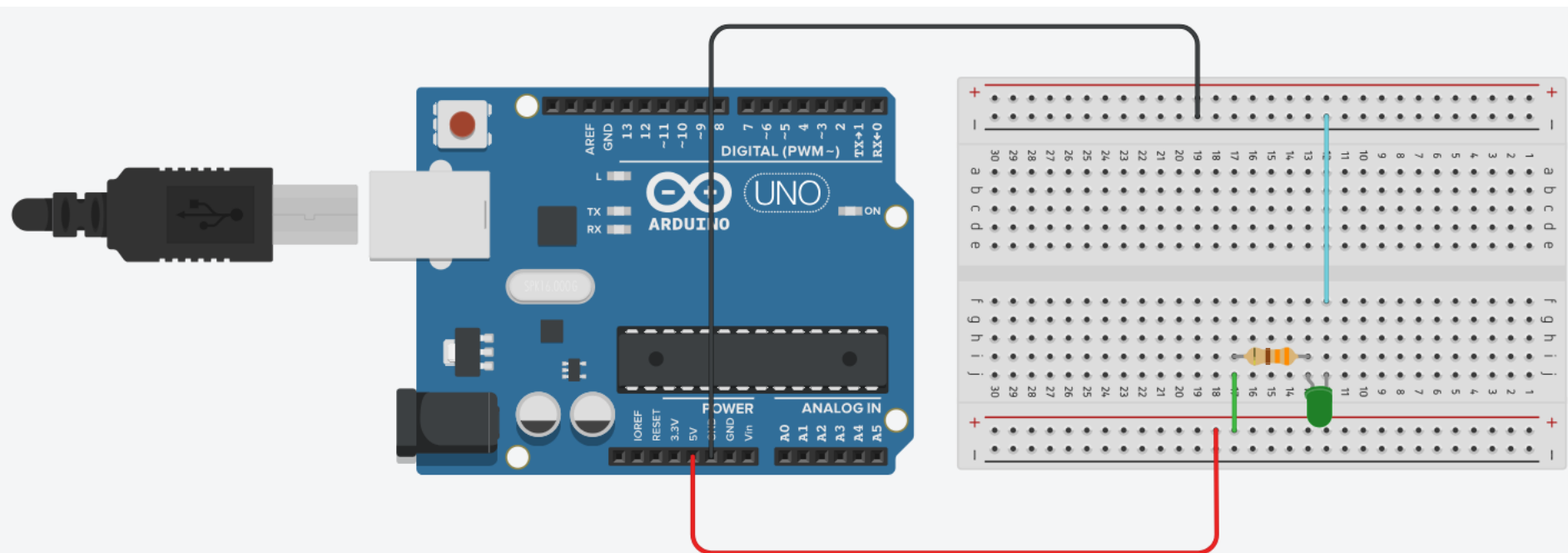
실습 - 아두이노와 연결

■ 실험 - 브레드보드 연결

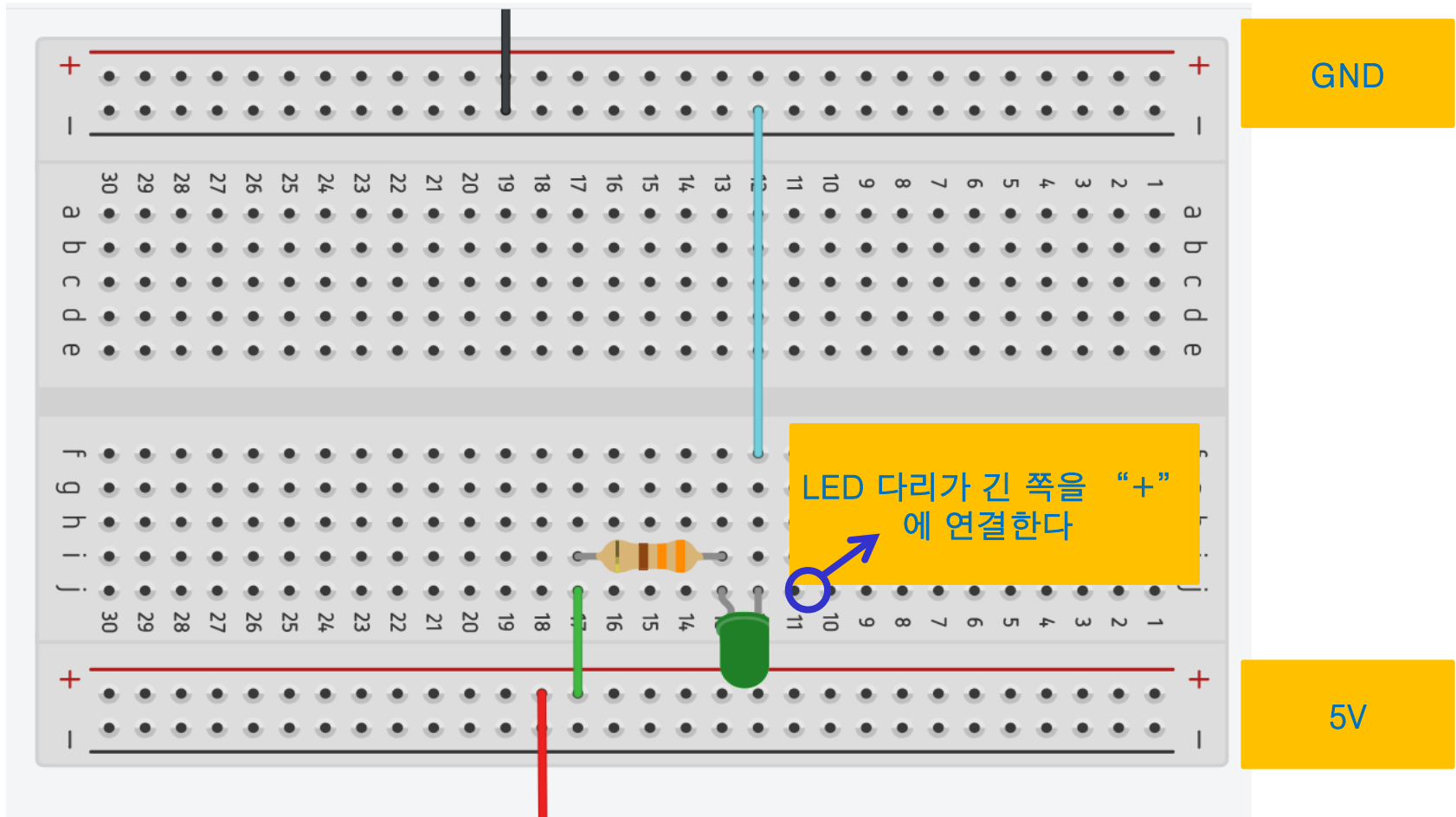
❖ 아두이노의 5V, GND와 브레드보드를 통하여 LED를 켜보도록 한다.

필요 부품 : 330옴 저항, LED,
점퍼선, 아두이노

주의사항 : LED 다리가 긴 쪽을 +에 연결한다

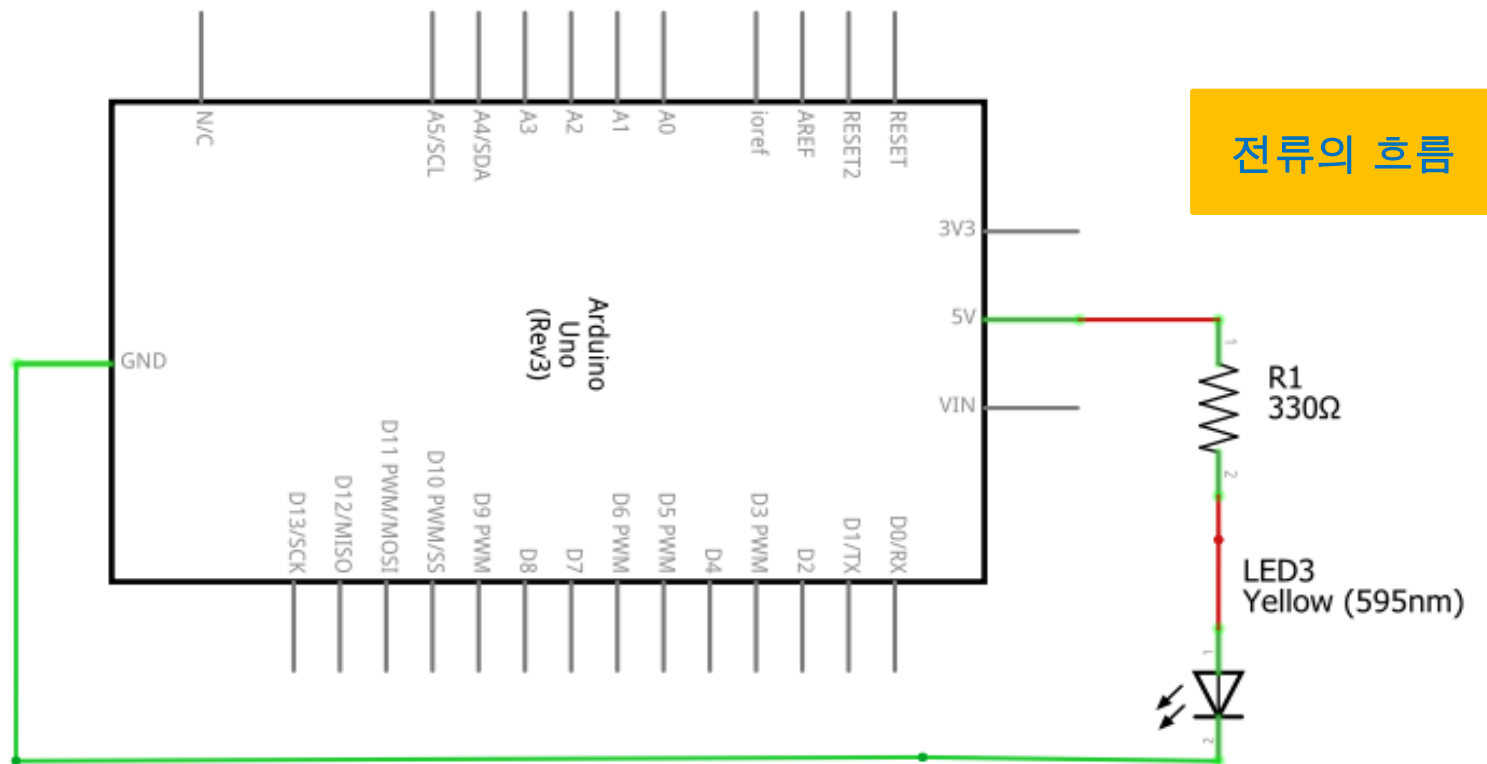


실습 - 주의 사항



실습 - 회로도로 확인하는 방법

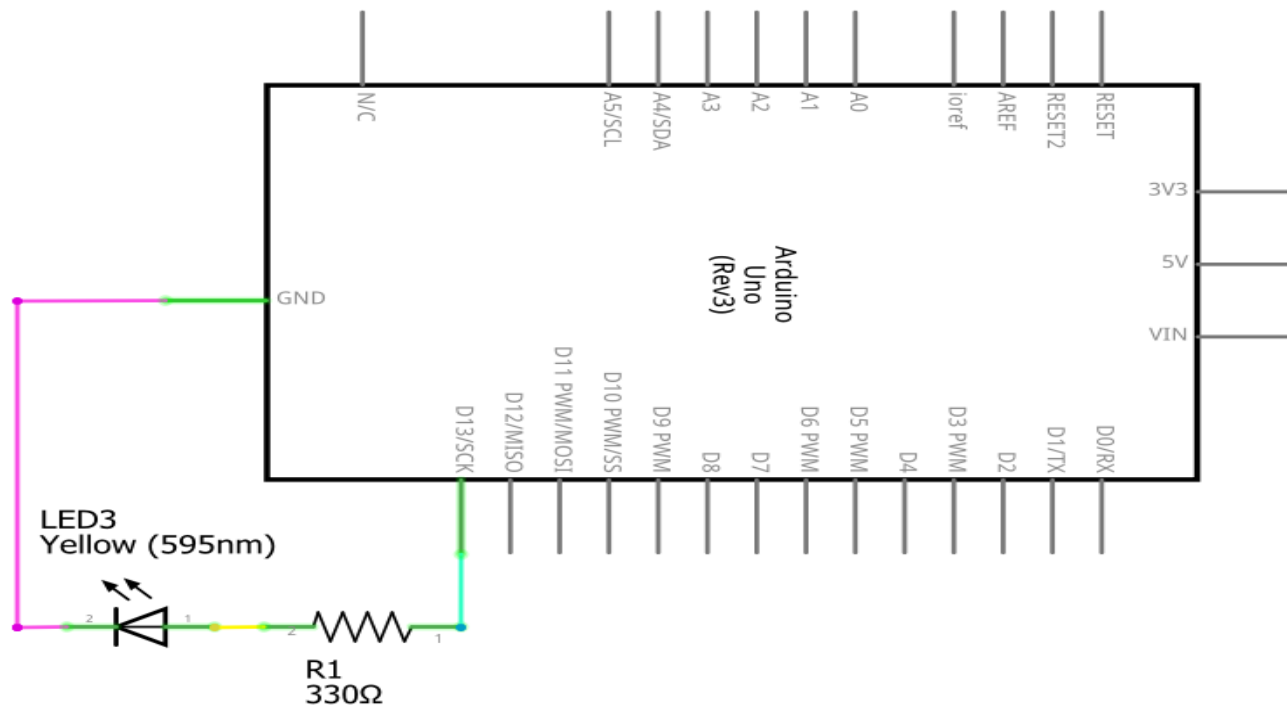
■ LED 연결 회로도



실습 - LED 결선하기

■ 회로를 브레드보드에 결선하시오.

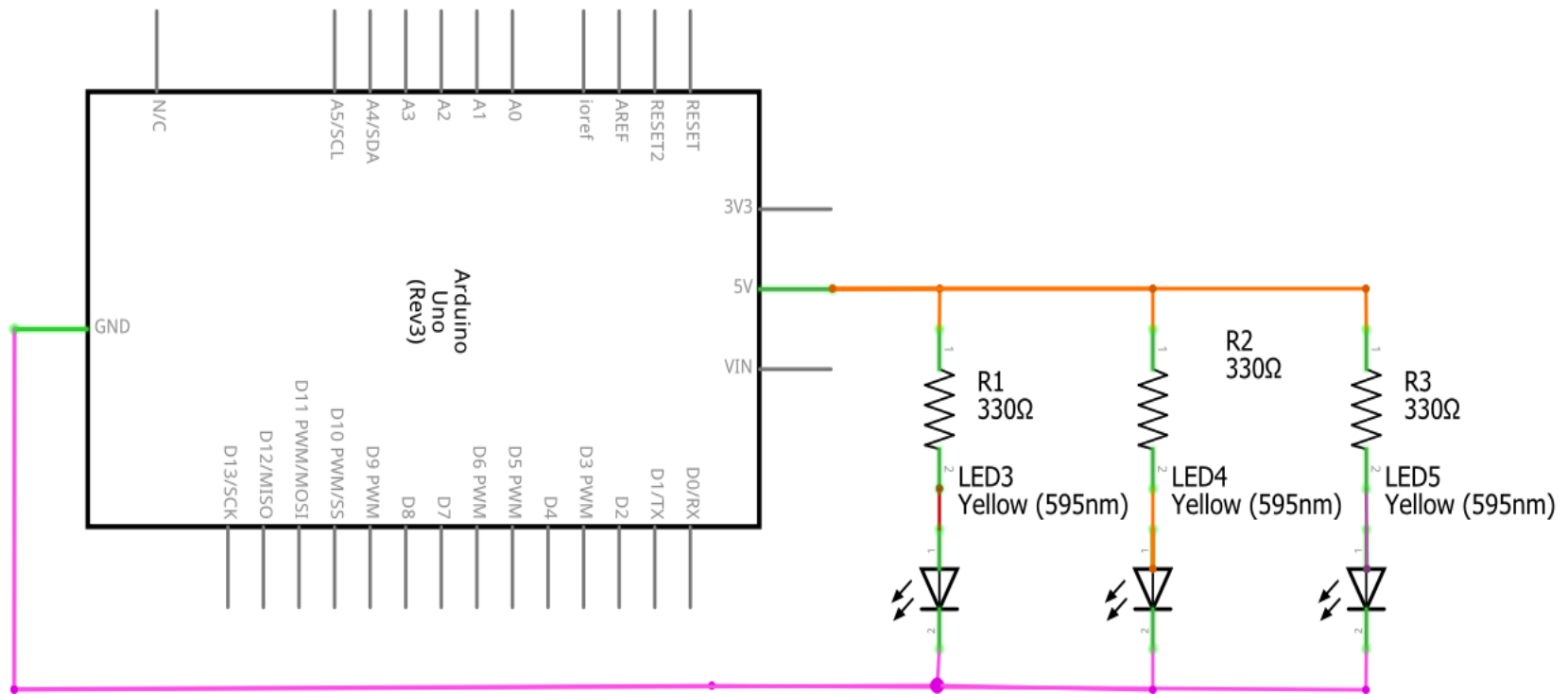
- ❖ 아두이노 13번 핀을 브레드보드 LED 에 연결하기
- ❖ Blink 예제 구동



실습 - 브레드보드를 사용하여 LED를 연결해 보자

■ 회로를 브레드보드에 결선하시오.

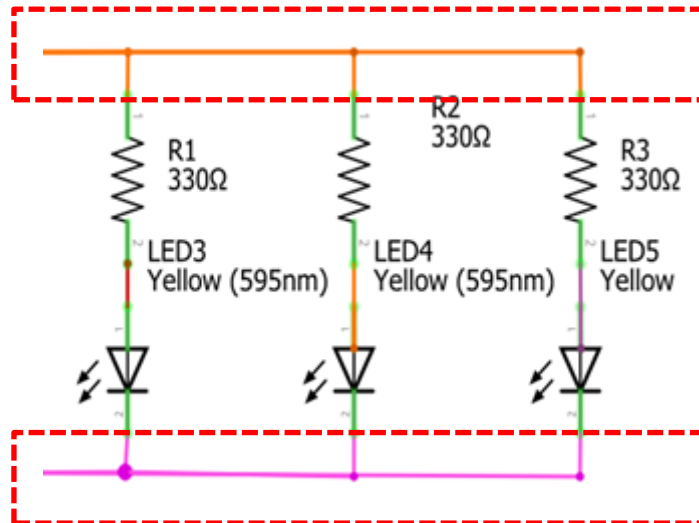
❖ LED 3개 켜기.



고찰

■ 저항 3개가 한 곳에 연결된다. 이것은 어떻게 해야 될까?

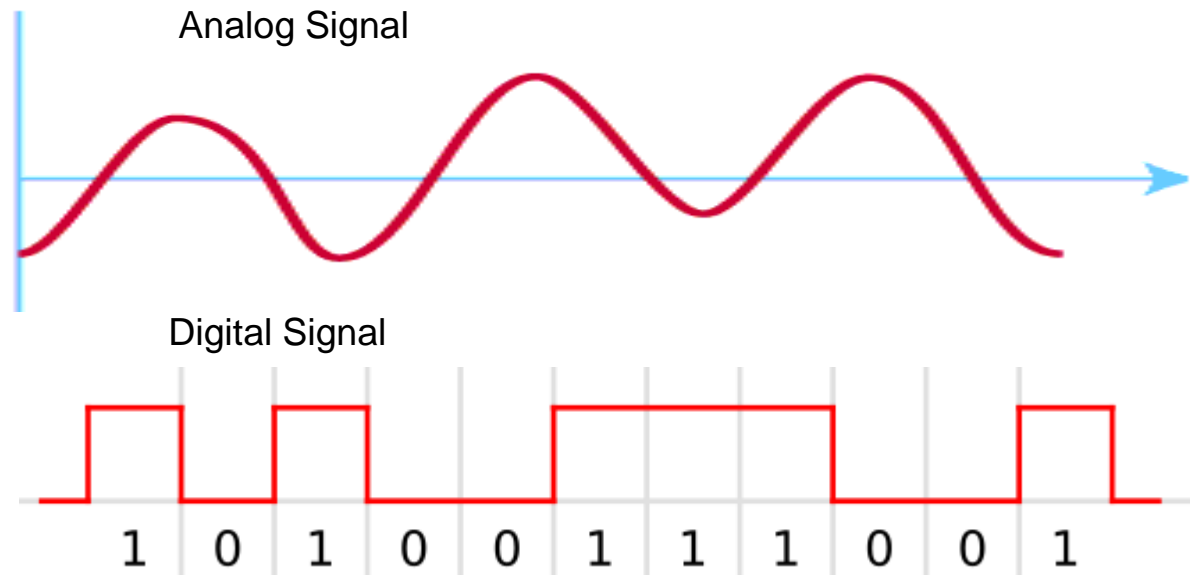
❖ LED 도 동일하다



아두이노 디지털 핀과 아날로그 핀

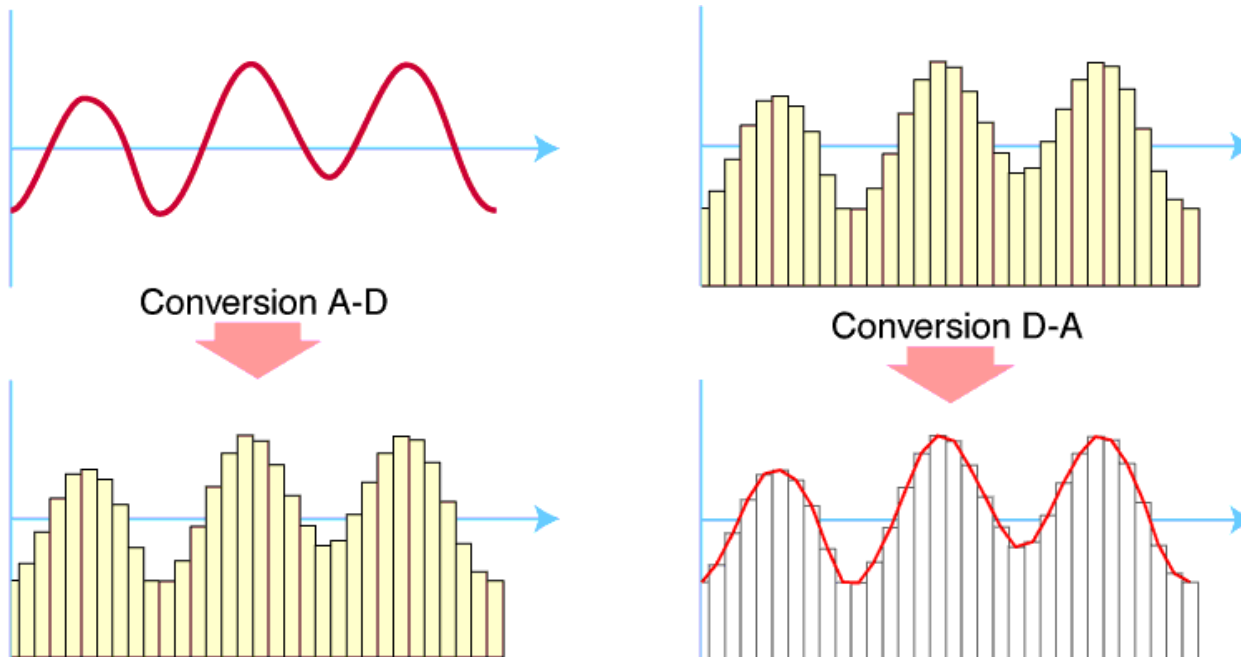
디지털 신호란

- 0과 1로만 이루어진 신호
- 기준 전압을 기준으로 1, 0을 판단
- 기준 전압 사이 값은 판단하지 못하며 있어서도 안 됨



아날로그 신호란

- 빛, 소리같이 연속적으로 변하는 신호
- 아날로그 값을 디지털 값으로 변환시키는 기능을 ADC라고 하며, 이 기능이 있는 편이나 장치 사용 필요
- 반대로 디지털을 아날로그로 변환하는 것을 DAC라고 한다

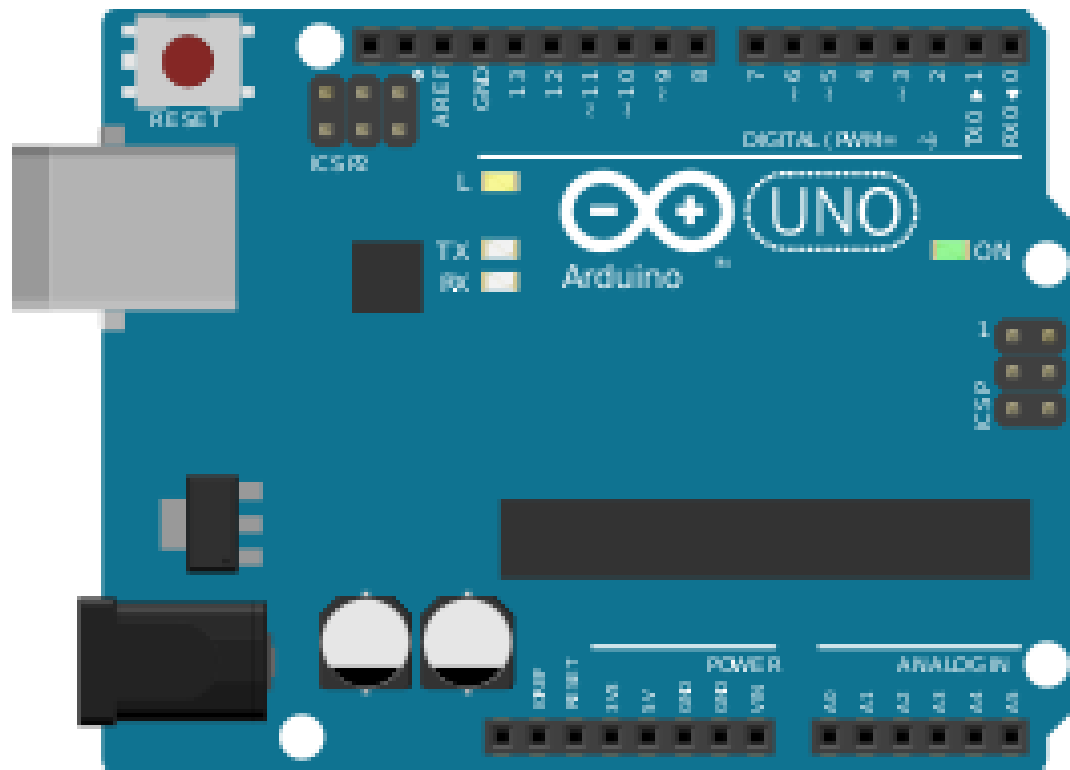


디지털 포트와 아날로그 포트

- 디지털 포트는 5V 또는 0V로 표현되지만 아날로그 포트는 0~5V로 표현을 할 수 있다
- 아날로그 포트는 ADC라는 장치를 통해서 전압을 입력 받을 수 있으며 PWM이라는 방법을 통해서 전압을 출력할 수 있다
- 아날로그 입력은 0~5V를 0~4096의 값으로 표현을 할 수 있다

아두이노 보드 핀 배열

- 아날로그 포트: A0~A6
- 디지털 포트 : 0~13 핀 (0:RXD, 1:TXD, 13:LED)



아두이노 특수 핀

■ 일반적인 디지털 입출력 외의 다른 용도를 위한 핀

상수	핀 번호	상수	핀 번호
A0	아날로그 입력0 (디지털 14)	LED_BUILTIN	온보드 LED(디지털 13)
A1	아날로그 입력0 (디지털 14)	SDA	I2C 데이터(디지털 18)
A2	아날로그 입력0 (디지털 14)	DCL	I2C 클럭(디지털 19)
A3	아날로그 입력0 (디지털 14)	SS	SPI 선택 (디지털 10)
A4	아날로그 입력0 (디지털 14)	MOSI	SPI 입력 (디지털 11)
A5	아날로그 입력0 (디지털 14)	MISO	SPI 출력 (디지털 12)
		SCL	SPI 클럭 (디지털 13)

하드웨어 통신 포트

- 아두이노는 디지털 포트를 이용하여 다른 하드웨어와 통신을 할 수 있다.
- 아두이노 우노가 UART, SPI, I2C 통신 방법을 지원한다.
- HIGH 와 LOW 를 이용하여 데이터를 표현하는 하드웨어 통신 방법이다.

디지털 입력, 출력

- 디지털 신호인 0과1을 입력 받거나 출력
- 입출력 사용시 모드를 선언해야 한다.
- 입력이란?
 - ❖ 포트에 전압을 측정하여 0인지 1인지 확인
- 출력이란?
 - ❖ 포트에서 전압을 내보내서 0과 1을 만드는 것