

K-Digital Training

# 신재생에너지 IoT 개발자 입문 과정

# IoT



# IoT란?

## Internet Of Things



- 사물인터넷(Internet Of Things)
- 일상적인 물건이나 기기에 인터넷을 연결하여 데이터를 주고받고, 서로 통신할 수 있도록 만드는 기술
- AIoT : IoT에 AI(인공지능)이 결합된 형태로, 단순히 데이터를 수집하는 것을 넘어 스스로 분석하고 판단하는 것

# IoT란?



- 핵심은 센서+인터넷 연결!
- 원격 제어, 모니터링 가능!

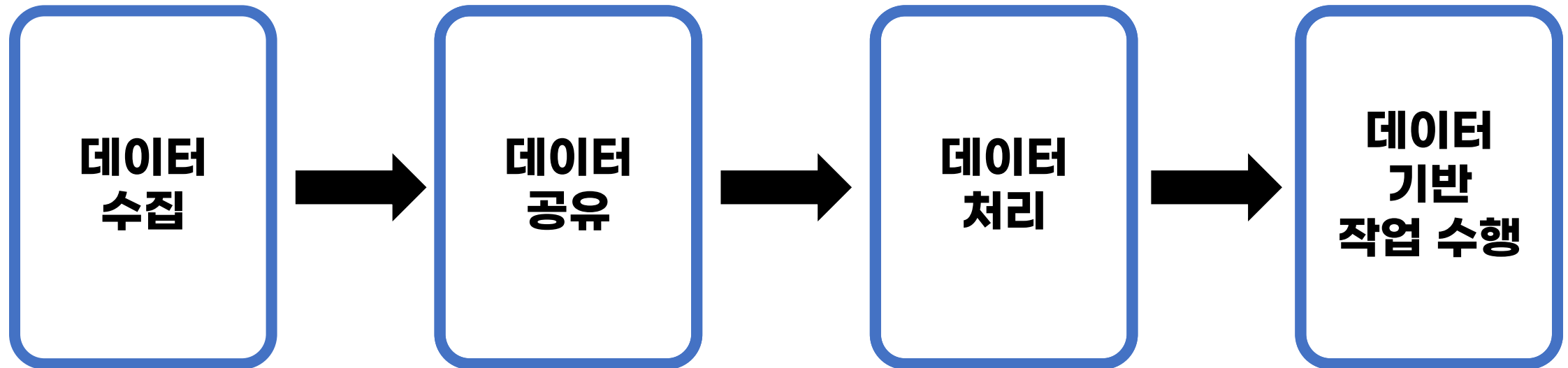
# IoT의 역사

- 1990년대 초반 IoT 개념 등장
- 1999년 케빈 애쉬튼이 자사 제품에 RFID 태그를 부착할 때, 최초로 ‘사물 인터넷’이라는 용어 등장
- 2000년대 스마트폰과 같은 모바일 기기의 보급과 함께 IoT 기술 발전

\*RFID : 사물에 고유코드가 기록된 전자태그를 부착하고 무선신호를 이용하여 해당 사물의 정보를 인식하는 기술

# IoT 작동 방식

- 수집하도록 명령 받은 모든 데이터를 수집하고, 수집된 데이터는 분석과 처리 과정을 거쳐 의사결정에 도움을 주거나 자동화하는데 활용된다.



# IoT 작동 방식

1. 데이터 수집 : 사물인터넷 기기는 센서를 통해 주변 환경 데이터를 수집한다.
2. 데이터 공유 : 수집한 데이터를 네트워크를 통해 퍼블릭, 프라이버시 클라우드 시스템에 전송한다. 또는, 다른 기기에 전송하거나 로컬에 저장한다.
3. 데이터 처리 : 전송, 저장된 데이터를 기반으로 특정한 작업을 수행하도록 처리한다.
4. 데이터 기반 작업 수행 : 네트워크 내 모든 기기에서 수집된 데이터는 분석 과정을 거치는데, 처리 장치를 통해 작업이 마무리되면 사용자에게 메시지를 발송하는 등의 여러 액션을 수행할 수 있다.

# IoT 적용 사례

**스마트홈**

**헬스케어**

**교통 및  
물류**

**공공안전  
및 환경**

**스마트팜**

**스마트  
팩토리**

**스마트  
시티**



# IoT 적용 분야 - 스마트홈

- 가정 내의 다양한 기기들을 인터넷을 통해 연결하여 스마트하게 제어하는 시스템
- 음성 명령으로 조명을 켜고, 스마트폰으로 집 안의 모든 가전제품을 원격 제어할 수 있고, 사용자의 생활 패턴을 학습해 에너지 절감까지 가능하다.
- Ex) LG 전자의 스마트 가전제품

[출처]

<https://live.lge.co.kr/2409-lg-ifa2024-ai-home/>



# IoT 적용 분야 - 스마트시티

- 도시의 교통 흐름을 실시간으로 분석하여 신호 체계를 최적화하고, 에너지 사용을 효율적으로 관리한다.
- 환경 센서를 통해 대기 질을 모니터링하고, 시민들에게 정보를 제공한다.
- Ex) 서울시의 스마트시티 프로젝트

[출처]

<https://www.smarttoday.co.kr/news/articleView.html?idxno=22303>



# IoT 적용 분야 - 산업용

- 제조업 및 산업 분야에서 사용되며, 생산 공정을 최적화하고 유지보수를 간편하게 하는 시스템을 의미한다.
- Ex) 스마트팩토리, 스마트팜, 스마트 에너지 관리 시스템 등



출처: News In 전남

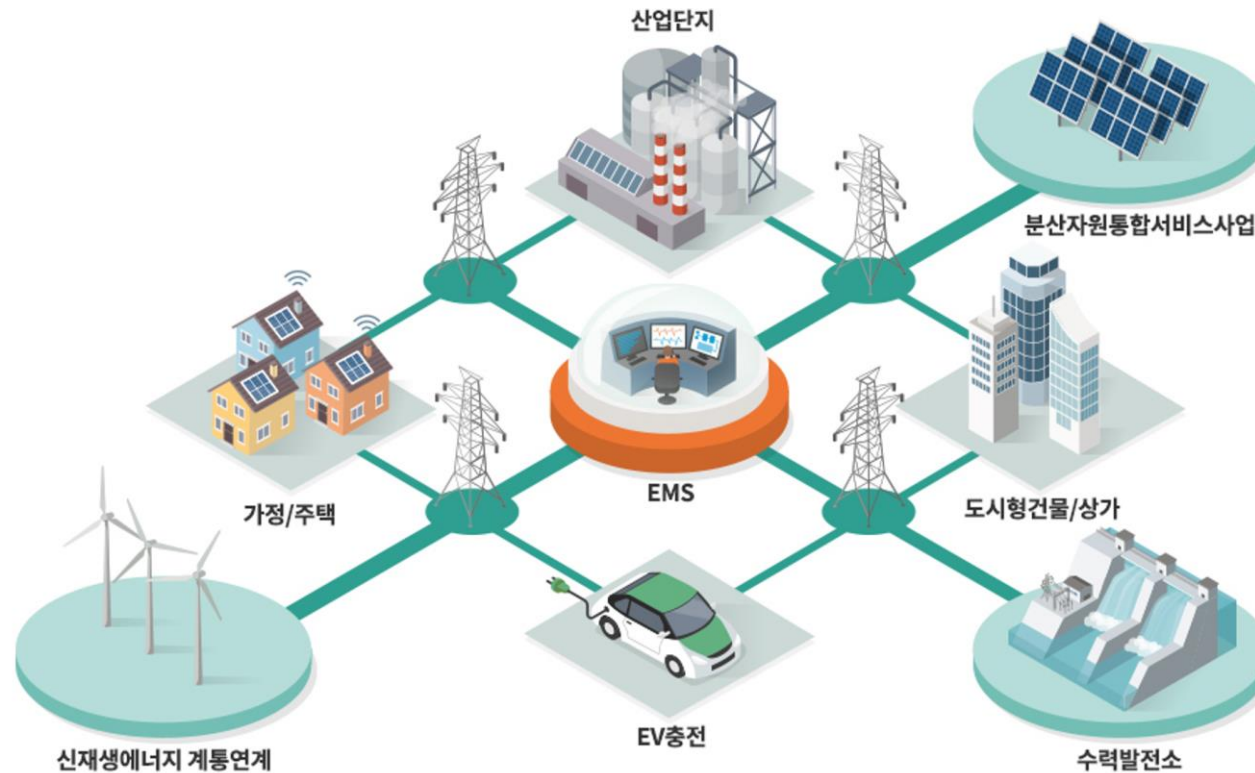
출처: SKC&C

# IoT와 신재생 에너지

- 모니터링 및 관리
  - 신재생 에너지 발전소의 운영 상태를 실시간으로 모니터링하고 관리
- 예측과 최적화
  - 센서 데이터 및 빅데이터 분석을 통해 신재생 에너지 생산의 패턴 분석과 예측 가능
- 스마트 그리드
  - 전기 및 정보 통신 기술을 활용하여 전력망을 지능화, 고도화 함으로써 고품질이 전력 서비스를 제공하고 에너지 이용 효율을 극대화하는 전력망
  - 신재생 에너지 발전소와 전력 그리드를 연결하고 통합 가능

# IoT와 신재생 에너지

지능형 전력망(스마트그리드) 개념도



출처 : 한국스마트그리드  
협회

# 아두이노



# 아두이노 란?

- 이탈리아어로 ‘절친한 친구’ 라는 뜻
- 오픈소스를 지향하는 마이크로 컨트롤러 기판
- 다양한 응용이 가능한 오픈소스 기반의 하드웨어
- 임베디드 시스템 중의 하나로 쉽게 개발할 수 있는 환경을 이용하여 장치 제어

# 마이크로 컨트롤러(MCU) 란?

- 마이크로프로세서의 일종으로 중앙 처리 장치의 기능과 메모리, 입출력 인터페이스 등을 내장한 작은 컴퓨터
- 마이크로프로세서에 비해 낮은 성능을 가지지만, 편의성을 높여 비교적 간단한 프로젝트 등에 사용
- 집약적인 제어와 저렴한 구성을 장점으로 가진다.

\*마이크로프로세서 : 컴퓨터의 중앙 처리 장치(CPU)와 같이, 산술 논리 장치, 제어 장치, 레지스터 등을 하나의 집적회로(IC)에 담은 처리 장치

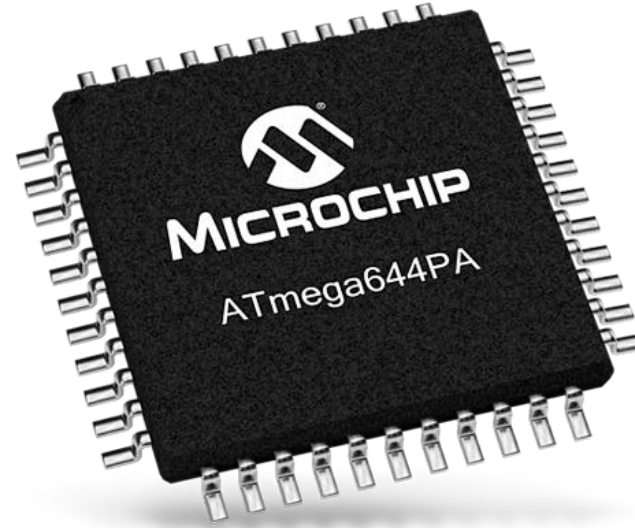


# CPU 와 MCU 차이



## CPU

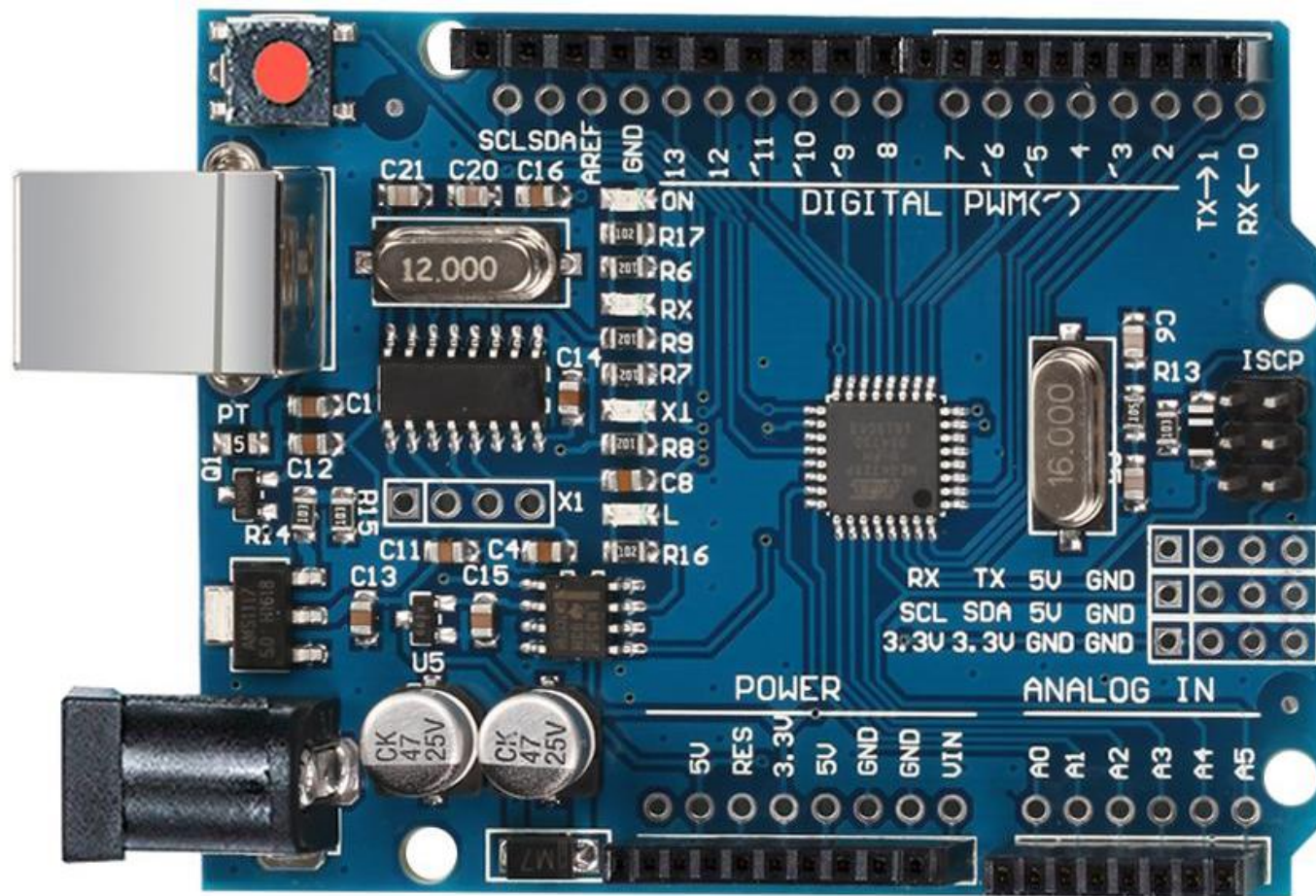
- 고성능 계산에 특화
- 여러 개의 코어로 구성
- RAM , ROM 등 부가 장치 필요



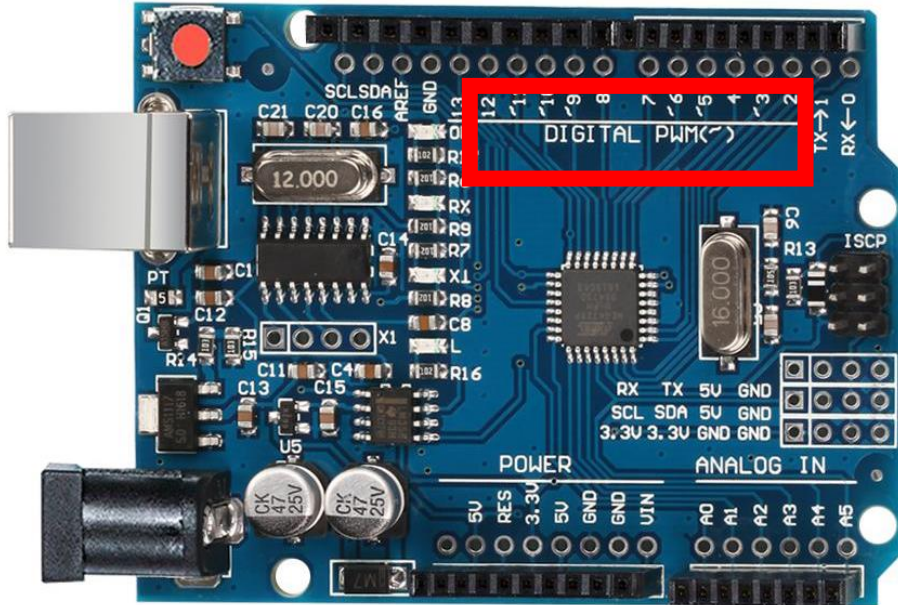
## MCU

- 대부분 단일 코어
- RAM, ROM, IO 포트 등이 내장
- SoC(System on Chip)

# 아두이노 보드



# 아두이노 보드 - 디지털 입출력

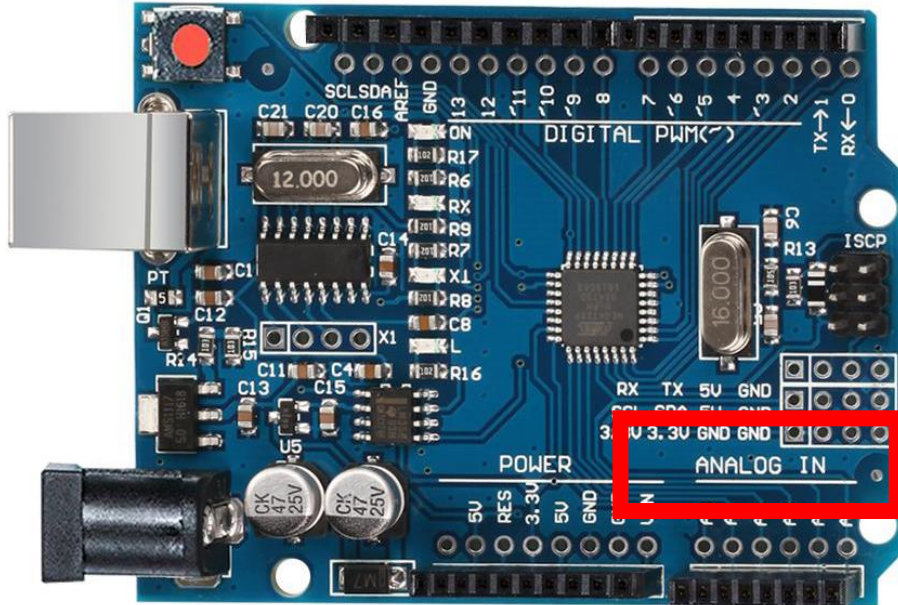


## 디지털 입출력

- 외부에서 디지털 값을 입력받거나 외부로 디지털 값을 내보내는 역할로, HIGH(5V)와 LOW(0V)로 이루어져 있다.
- Ex) PIR 센서(인체감지센서)를 디지털 핀에 연결했을 때 “감지됨/감지되지 않음” 두 가지 값이 입력되고 이 경우 디지털 입출력 핀에 센서를 연결하여 사용할 수 있다.



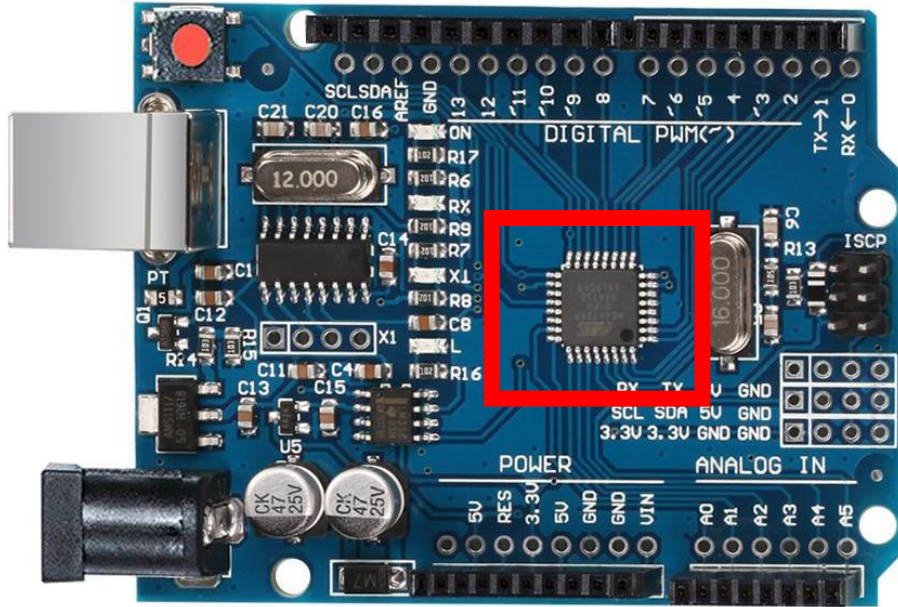
# 아두이노 보드 - 아날로그 입력



## 아날로그 입력

- 0~5V 전압값을 0~1023 사이이 값으로 변환하여 읽는 핀
- Ex) 온도 센서를 연결했을 때 10°, 15°, 20° 등 다양한 값이 입력될 수 있는데 이 경우 아날로그 핀을 사용한다.

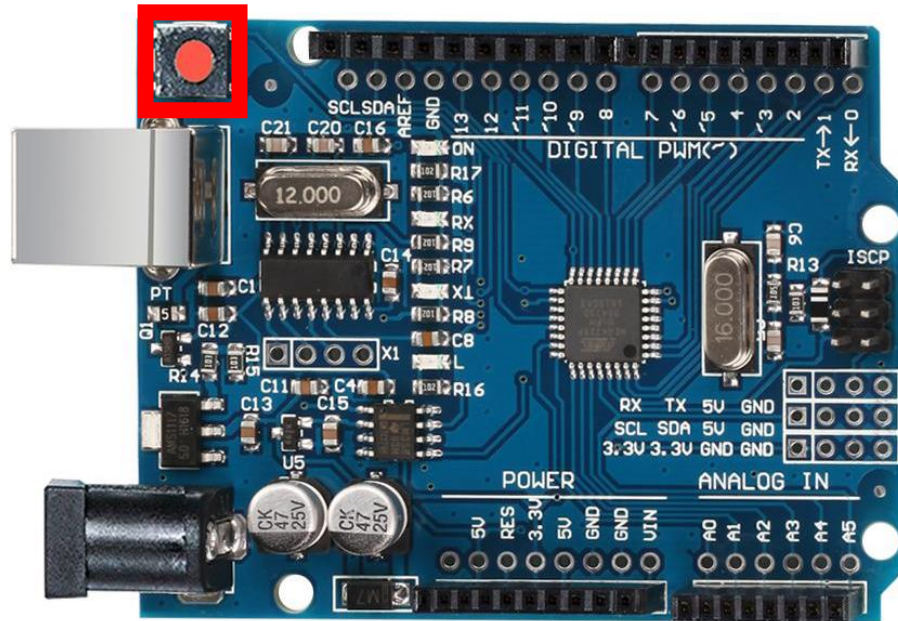
# 아두이노 보드 - 마이크로 컨트롤러



## 마이크로 컨트롤러

- 연산과 디지털 신호의 입출력, 통신, 메모리를 제어하는 기능을 담당한다.
- 디지털 입출력 핀을 통해 입력된 값을 연산하여 특정 동작을 수행하도록 명령한다.

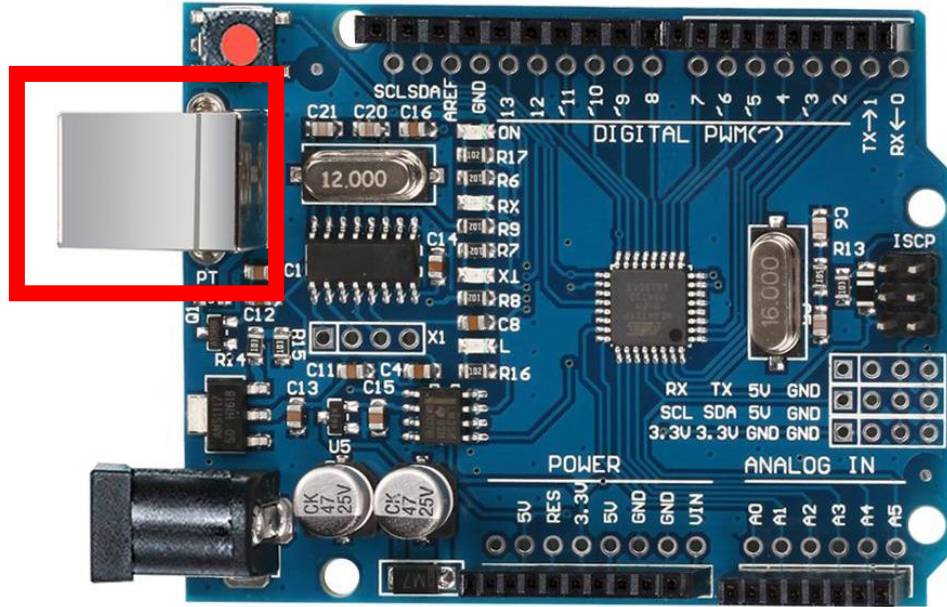
# 아두이노 보드 - 리셋 버튼



## 리셋 버튼

- 업로드한 프로그램을 리셋 즉, 아두이노를 재부팅할 때 활용한다.

# 아두이노 보드 - USB 커넥터

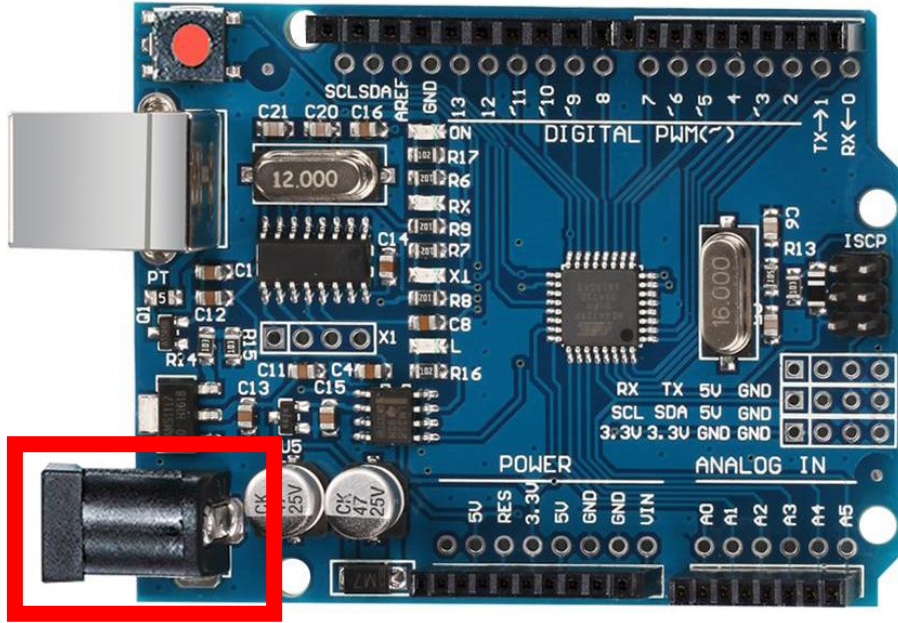


## USB 커넥터

- 컴퓨터의 USB 케이블을 사용하여 전원을 공급할 수 있고, 프로그램을 업로드할 수 있다.
- USB Type B 케이블을 USB 포트에 연결하여 사용한다.



# 아두이노 보드 - DC 전원 커넥터

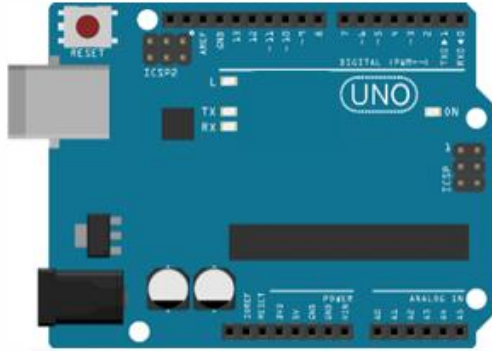


## DC 전원 커넥터

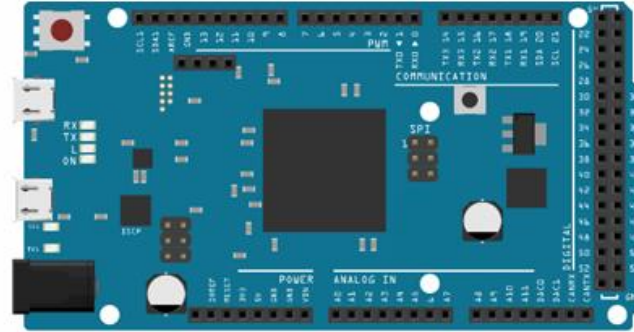
- PC와 연결하지 않고 아두이노에 전원을 공급하는 기능을 제공한다.



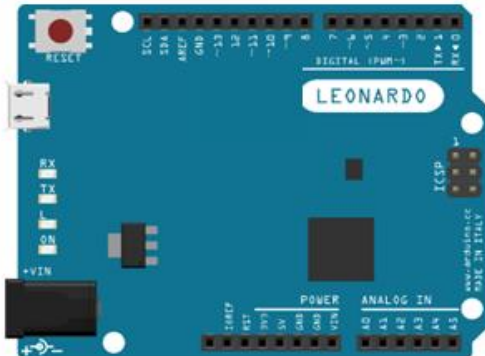
# 아두이노 보드 종류



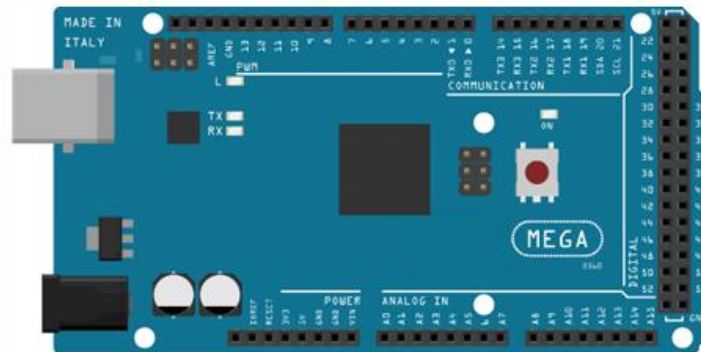
아두이노 우노(UNO) R3



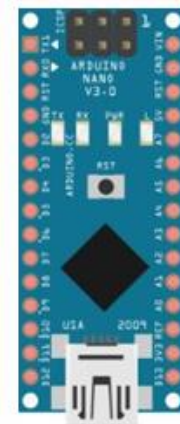
아두이노 듀에(DUE)



아두이노 레오나르도 (Leonardo)



아두이노 메가 (MEGA) 2560



아두이노 나노 (Nano)

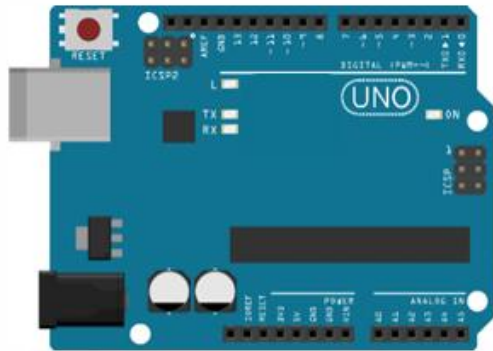
# 아두이노 보드 종류

## 1. 아두이노 우노

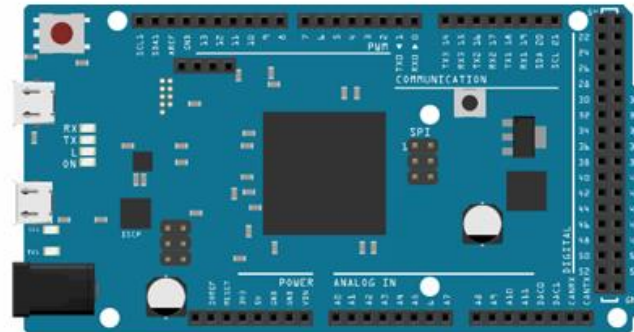
- 가장 많이 사용되는 대표적인 아두이노 보드로, 초심자 전용으로 사용된다.

## 2. 아두이노 듀에

- 기능과 성능이 높고 핀 수가 많아 전문적인 개발 및 연구에 적합하다.



아두이노 우노(UNO) R3



아두이노 듀에(DUE)

# 아두이노 보드 종류

## 3. 아두이노 레오나르도

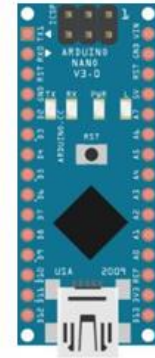
- 2개의 하드웨어 시리얼 포트를 사용한다.

## 4. 아두이노 메가

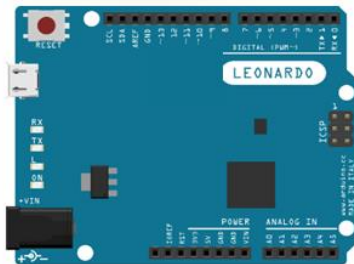
- 우노 보드보다 기능과 핀 수가 많고, IoT에 활용도가 높다.

## 5. 아두이노 나노

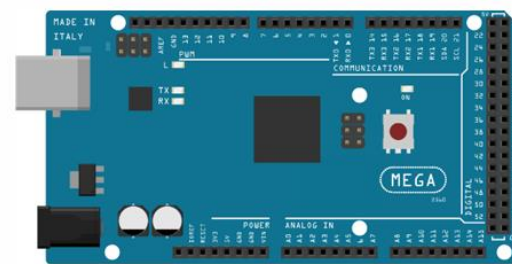
- 아두이노 우노와 비슷한 구성이며, 크기가 훨씬 작아 아두이노 완성품의 크기를 줄일 수 있다.



아두이노 나노 (Nano)



아두이노 레오나르도 (Leonardo)



아두이노 메가 (MEGA) 2560

# 점퍼 케이블



- 전자 회로에서 다양한 부품을 연결할 때 사용되는 유연한 전선이다.
- 점퍼 케이블은 흔히 '암/수'로 구별하여 사용한다.
  - 암 female : 케이블 핀이 보이지 않고 커넥터만 있다.
  - 수 male : 케이블 핀이 돌출되어 나와 있다.
- 아두이노 보드에 센서, 모듈 등을 연결하기 위한 용도로 사용되며 브레드보드와 함께 이용하는 경우가 많다.

# 점퍼 케이블의 구조



- 단자(핀)
  - 금속 핀은 수핀(Male) 또는 암핀(Female) 형태로 제작되며, 사용 용도에 따라 선택할 수 있다.
- 전선
  - 점퍼 케이블의 길이는 다양한데, 보통 10cm에서 30cm 까지 제공되며, 다양한 색상으로 구분되어 회로 구성을 더 쉽게 할 수 있다.

# 점퍼 케이블의 종류

## 1. Male to Male

- 양쪽 끝이 수핀으로 되어 있는 구조로, 브레드보드 간, 혹은 브레드보드에서 다른 장치(ex. 아두이노 보드)로 연결할 때 많이 사용된다.

## 2. Male to Female

- 한쪽 끝이 수핀, 다른 쪽 끝이 암핀으로 되어 있는 구조로, 이를 통해 브레드보드의 핀과 전자 부품의 암 커넥터를 연결할 수 있다.

## 3. Female to Female

- 양쪽 끝이 암핀으로 되어 있는 구조로, 주로 암 소켓이 있는 전자 부품 간의 연결에 사용된다.

# LED (Light Emitting Diode)

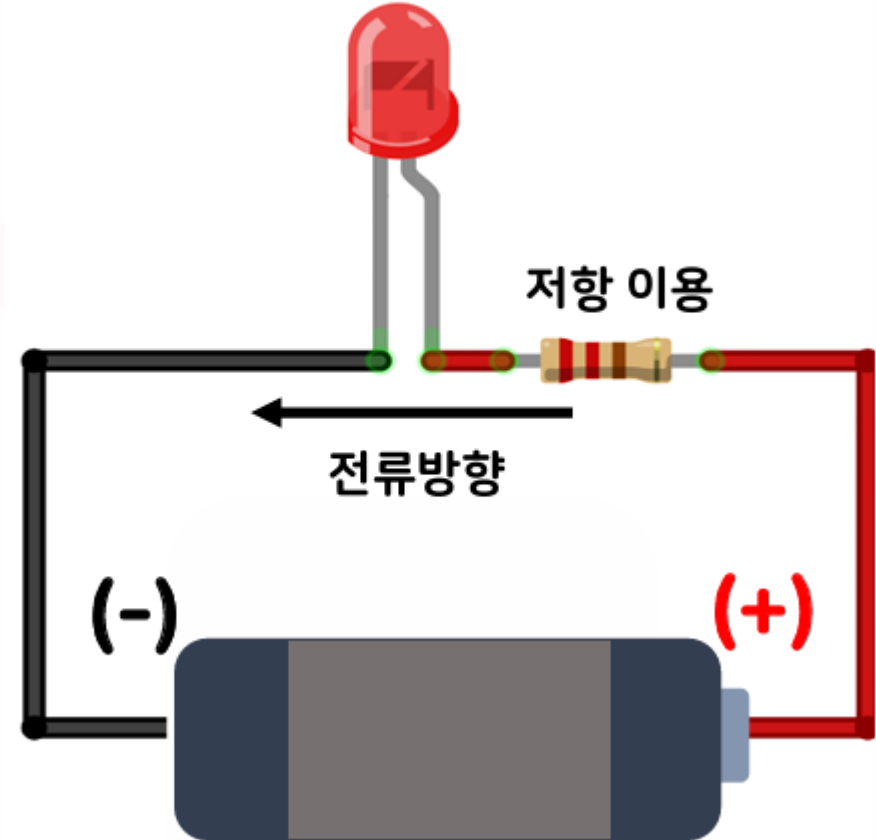


- 발광 다이오드 : 다이오드의 일종으로, 다이오드 중 전류를 빛 에너지로 변환하는 것
- 정해진 방향( (+) → (-) )으로 전류가 흐르며 빛을 내는 발광 소자

\*다이오드 : 한쪽 방향으로만 전류가 흐르는 반도체 소자

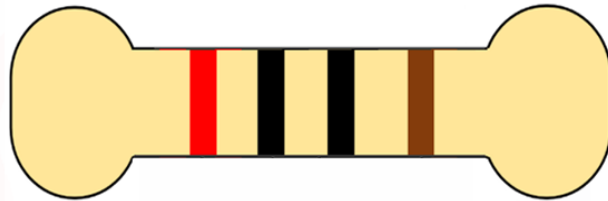
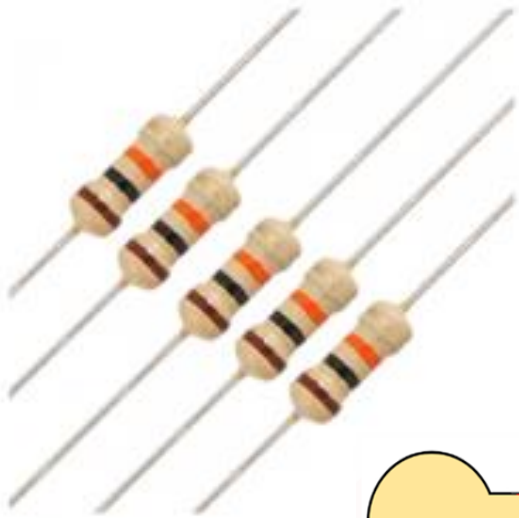


# LED (Light Emitting Diode)





# 저항 (Resistor)



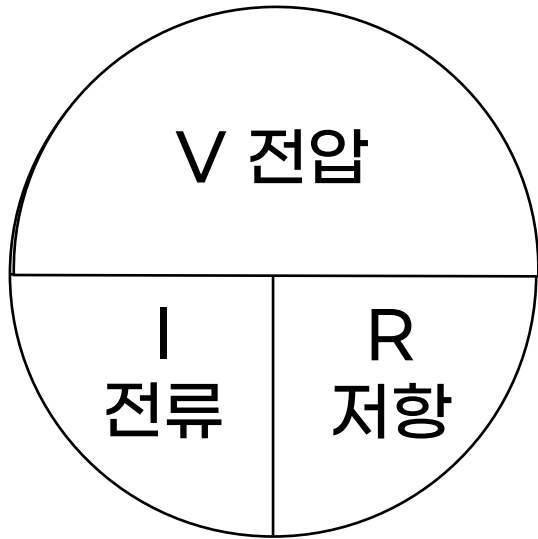
< 4색 저항 >



< 5색 저항 >

- 저항 : 운동을 방해한다.
- 물체가 전류의 흐름을 제어하는 것을 의미하며, 동일한 조건에서는 저항이 높은 물체일수록 전류가 적게 흐르고 저항이 낮은 물체일수록 전류가 많이 흐른다.
- LED를 비롯한 다양한 소자들을 과전압, 과전류로부터 보호한다.

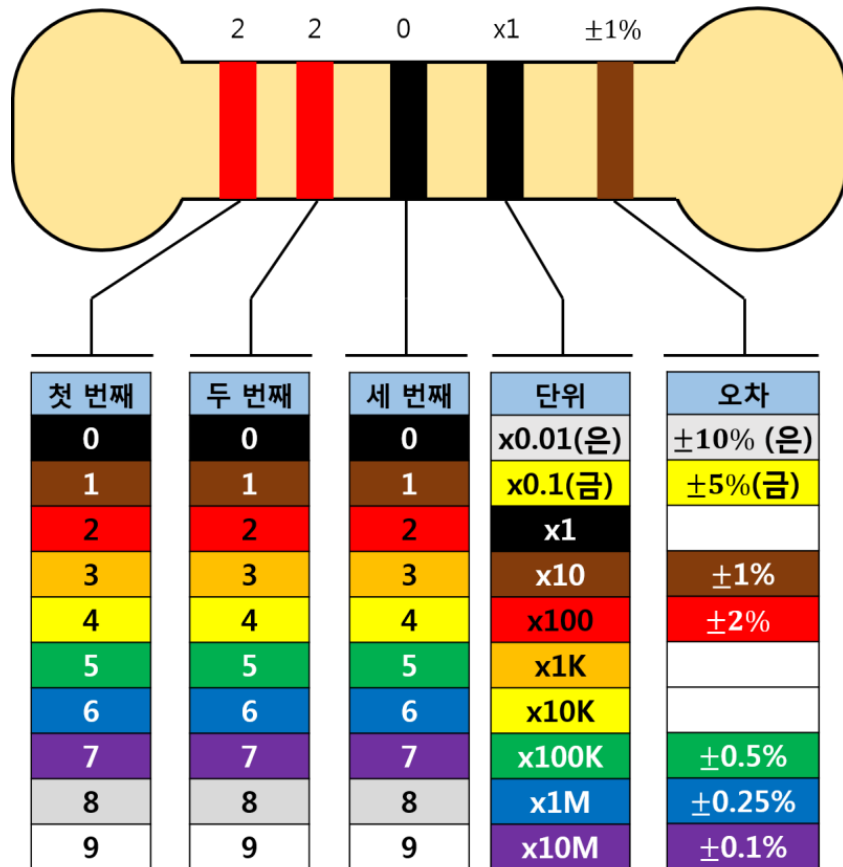
# 옴의 법칙



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

- 1826년 옴이 발견한 물리학의 기본 법칙
- 전압-전류-저항 사이의 관계를 나타내는 식
- 전기회로에서 전류(I)가 전압(V)에 비례하고 저항(R)에 반비례한다는 관계를 나타내는 법칙
- $V = I * R$  또는  $I = V / R$  이라는 수식으로 표현된다.

# 저항 (Resistor)



- 첫번째 : 빨간색 띠 - 2
- 두번째 : 빨간색 띠 - 2
- 세번째 : 검은색 - 0
- 네번째 : 배수를 의미, 검은색 띠 x1 배수
- 마지막 : ±1% (저항의 오차)

첫번째	두번째	세번째	단위	오차	저항값
빨강색	빨강색	검정색	검정색	갈색	220Ω
2	2	0	0	±1%	

# 저항 Quiz

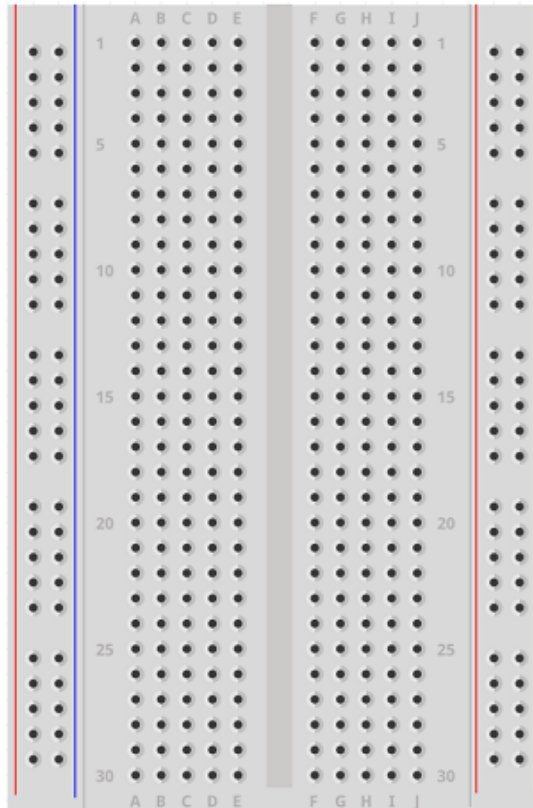


# 저항 Quiz



첫번째	두번째	단위	오차	저항값
노란색	보라색	빨강색	금색	4.7K $\Omega$
4	7	X 100	$\pm 5\%$	

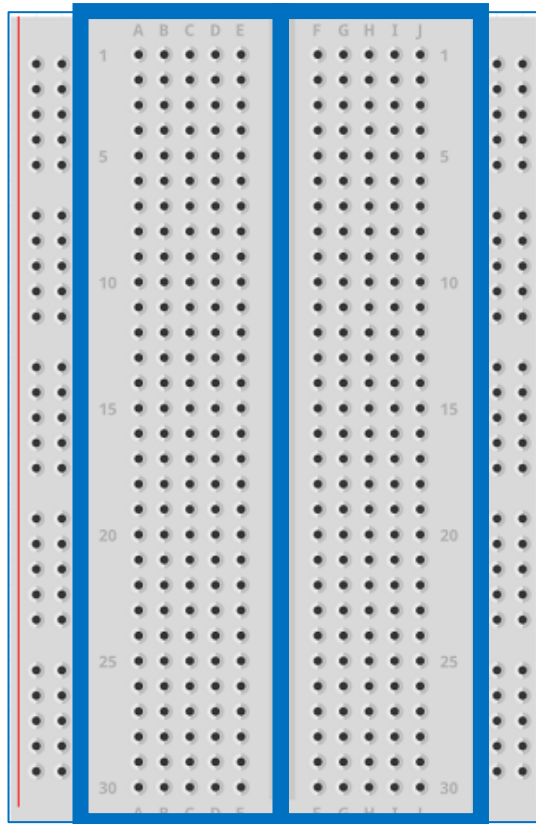
# 브레드보드 (BreadBoard)



- 일명 “빵판”으로 전자 회로를 쉽게 구성할 수 있는 기본 판
- 전자 회로 실험에서 기판에 납땜을 하지 않고도 소자들을 연결할 수 있도록, 사용하는 회로 구성 도구
- 부품들을 재사용할 수 있고, 구멍의 개수에 따라 제품이 구성된다.

# 브레드보드 (BreadBoard)

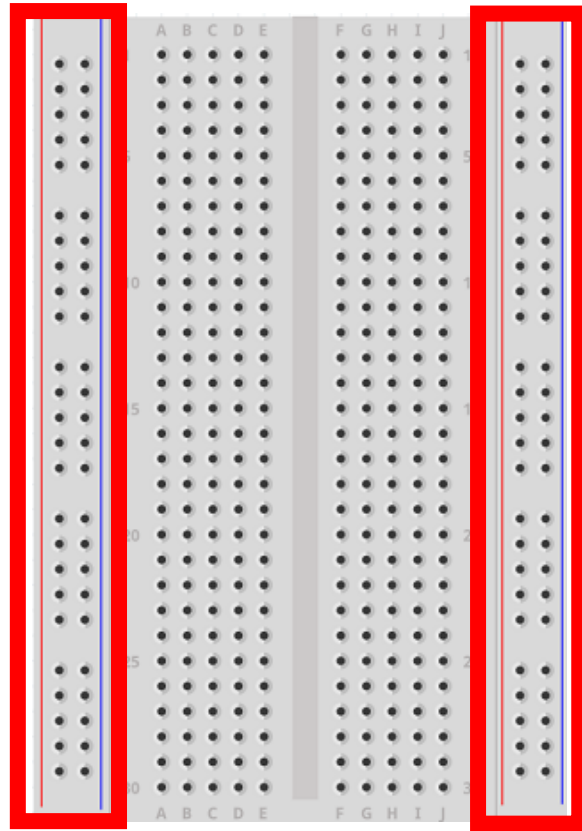
단자띠



- **단자띠** : 부품을 꽂는 영역으로 a~e, f~j까지 브레드보드 내부에서 가로로 서로 금속으로 연결되어 있다.
- 단자 띠에 꽂는 부품들은 부품이 연결 부위를 서로 다른 라인에 꽂아야 한다. 같은 라인에 꽂을 경우 합선이 일어날 수 있다.

# 브레드보드 (BreadBoard)

버스띠

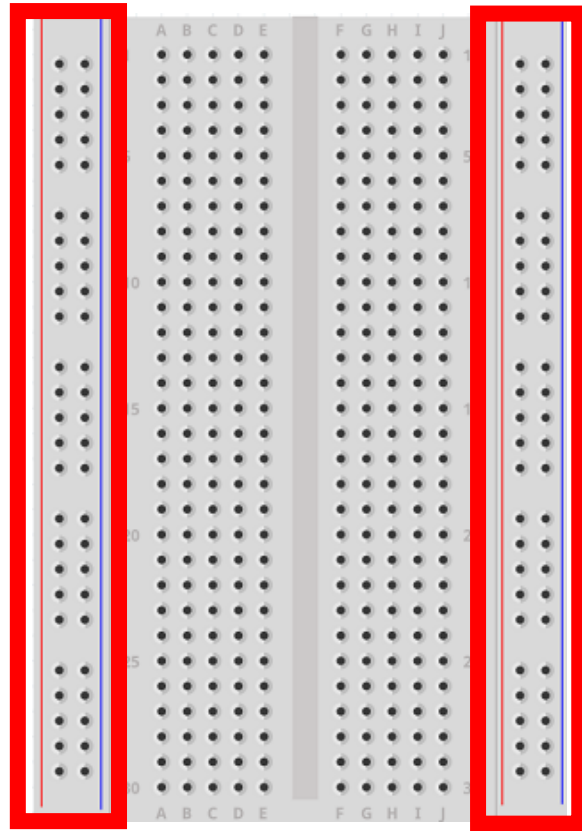


- **버스띠** : 전원 공급을 위한 띠로 + 전원선과 - 접지선이 존재한다.
- 전원선(+) : 아두이노 보드에서는 5V, 3.3V, 디지털 입출력핀과 연결하며 일반 부품들의 (+)와 연결한다. 보통 빨간색으로 표시한다.
- 접지선(-) : 아두이노 보드에서는 GND와 연결하고, 부품들의 (-)와 연결한다. 보통 파란색 또는 검정색으로 표시한다.

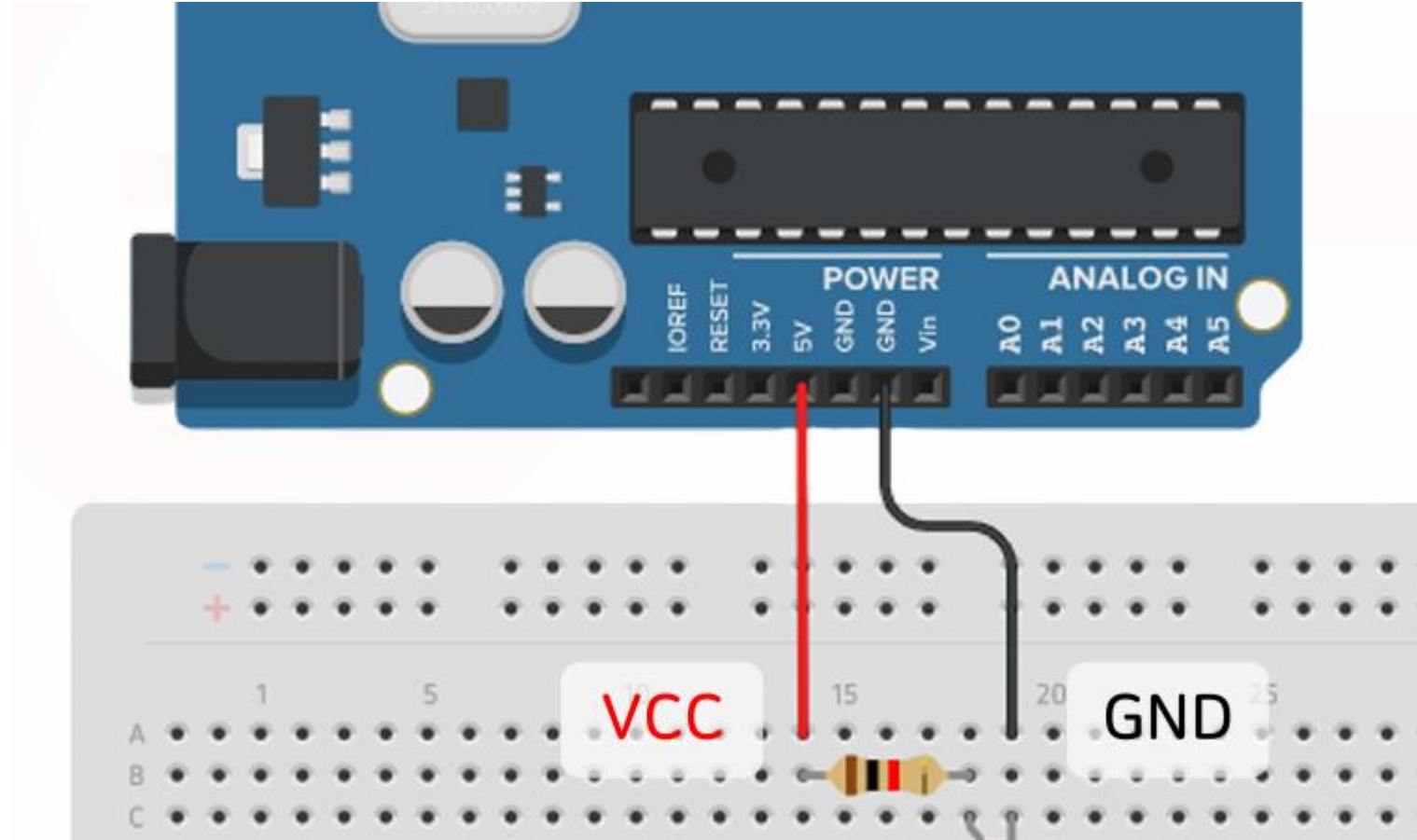


# 브레드보드 (BreadBoard)

버스띠

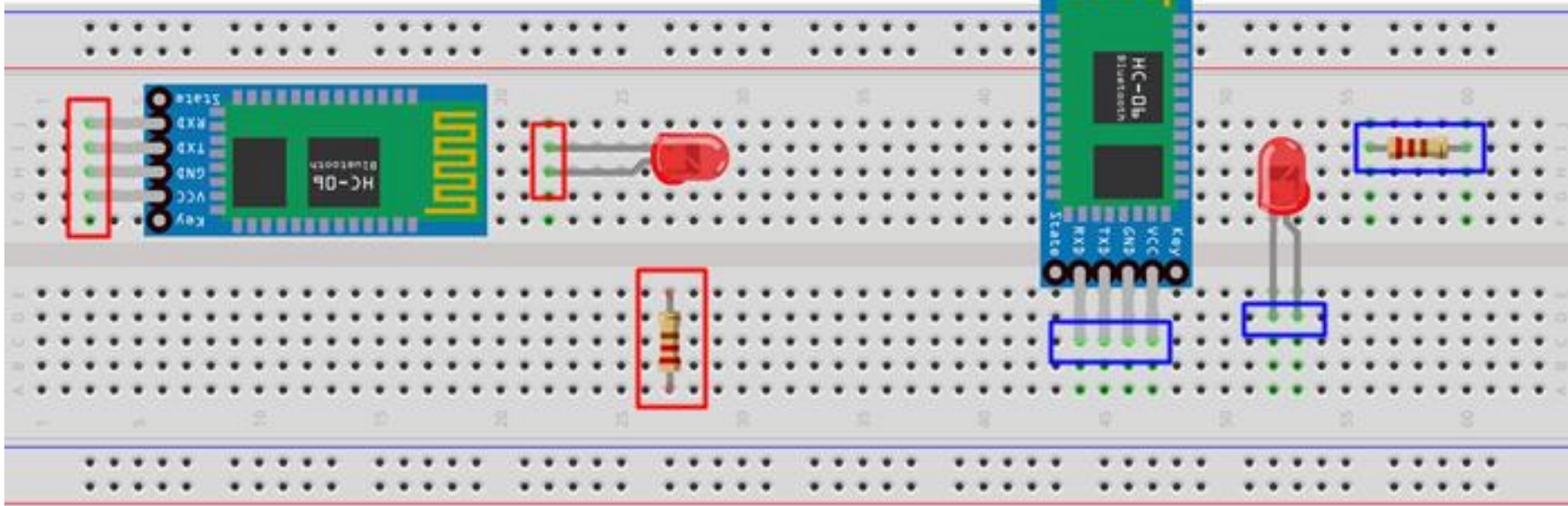


- **버스띠** : 전원 공급을 위한 띠로 + 전원선과 - 접지선이 존재한다.
- 전원선(+) : 아두이노 보드에서는 5V, 3.3V, 디지털 입출력핀과 연결하며 일반 부품들의 (+)와 연결한다. 보통 빨간색으로 표시한다.
- 접지선(-) : 아두이노 보드에서는 GND와 연결하고, 부품들의 (-)와 연결한다. 보통 파란색 또는 검정색으로 표시한다.

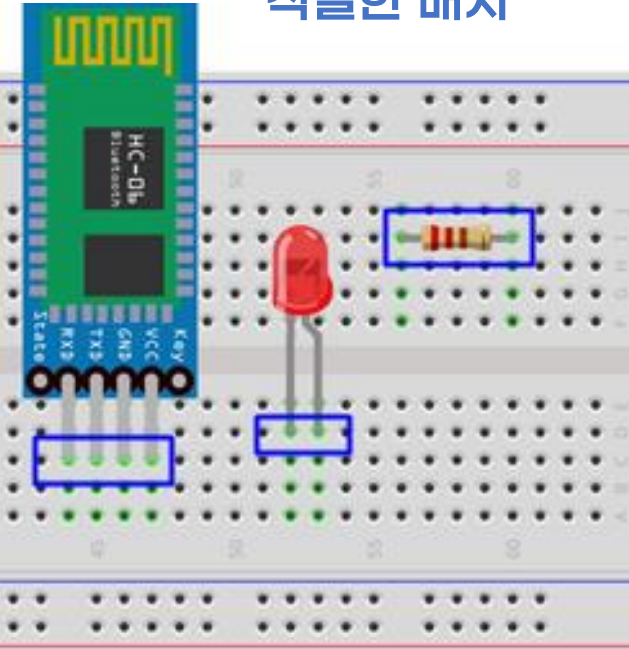


# 브레드보드 (BreadBoard)

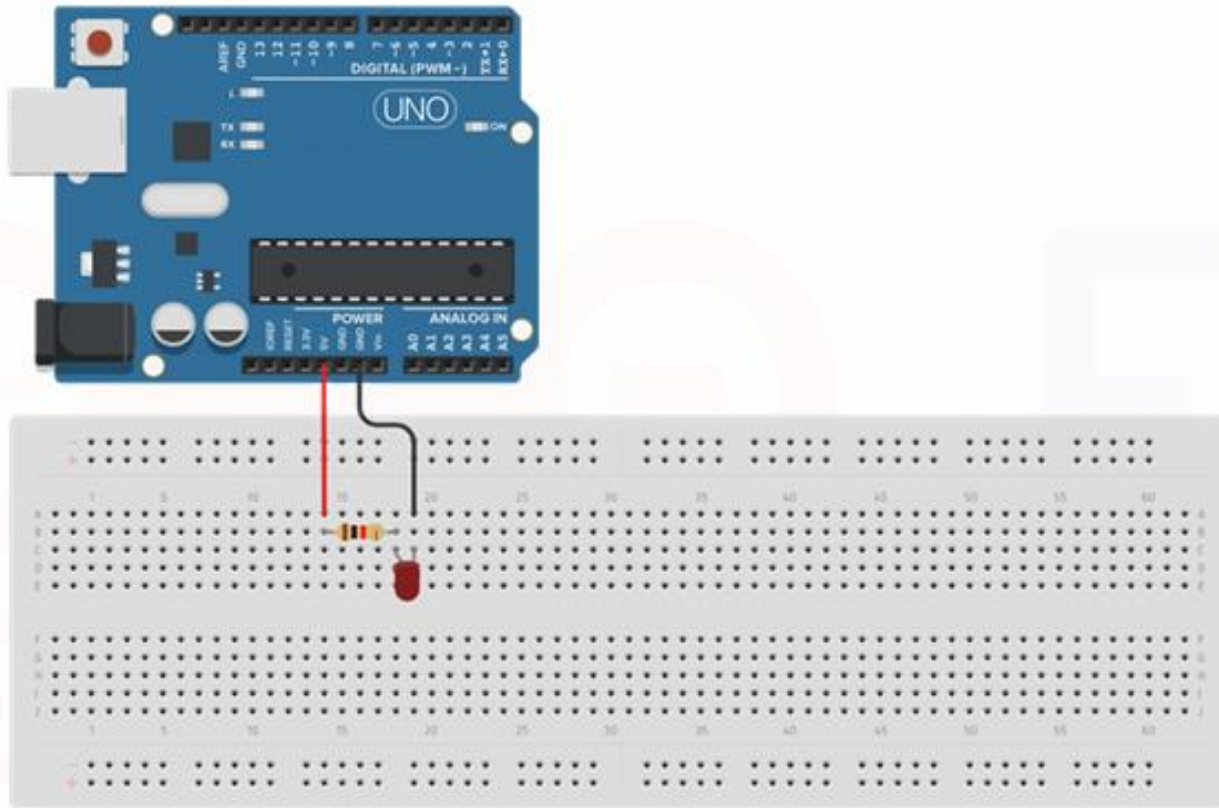
부적절한 배치



적절한 배치



# 간단 실습. LED 연결



결선 회로도

- 아두이노 보드
- 브레드보드
- USB B케이블
- 점퍼케이블(수-수) 2개
- LED 1개
- 100Ω 저항 1개
- 1KΩ 저항 1개

# 간단 실습. LED 연결

Q. 더 밝은 LED는?

100Ω 저항 vs 1KΩ 저항

# 간단 실습. LED 연결

Q. 더 밝은 LED는?

100Ω 저항 vs 1KΩ 저항

# 아두이노 스케치 설치



**Arduino IDE 2.3.6**  
[Release notes](#)

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, check the [Arduino IDE 2.0 documentation](#).

Windows Win 10 or newer (64-bit) **DOWNLOAD**

**Nightly Builds**  
Download a preview of the incoming release with the most updated features and bugfixes.

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

- <https://www.arduino.cc/en/software>

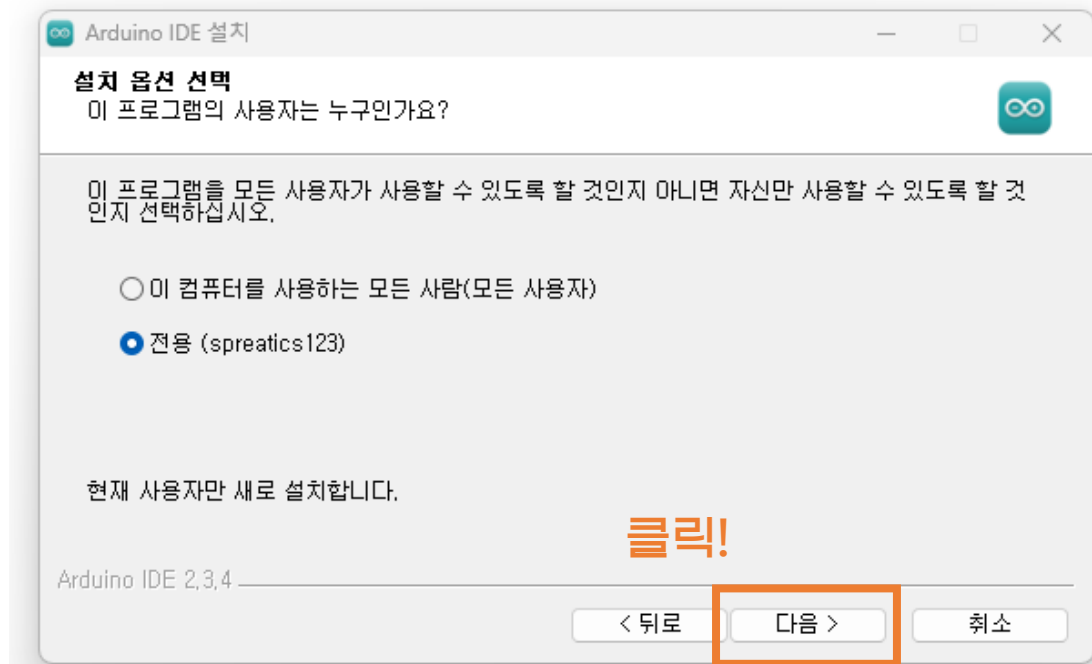
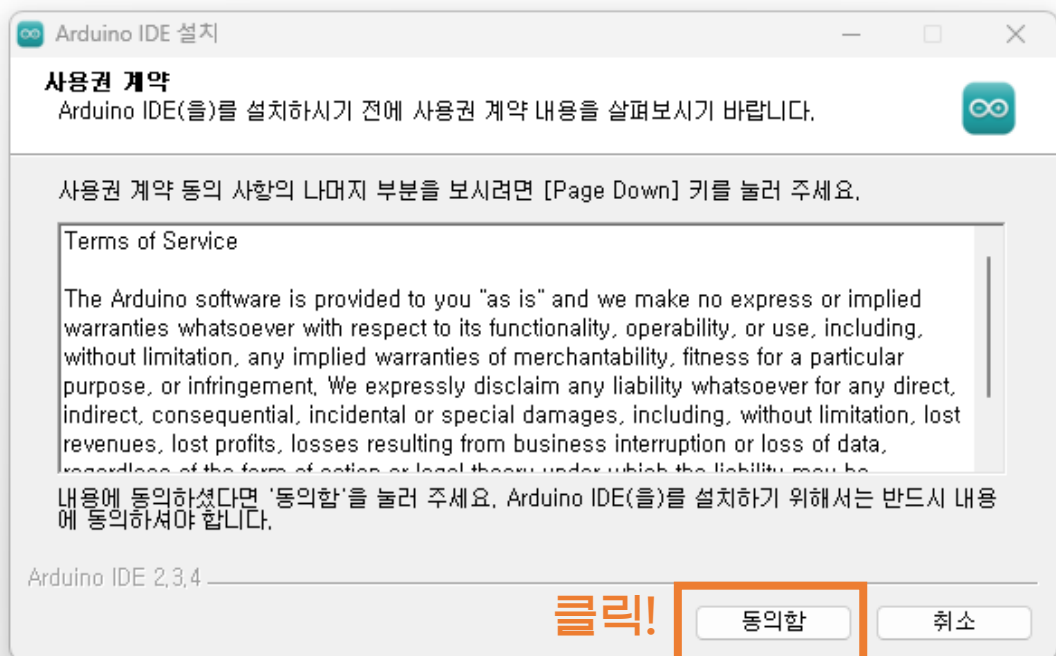


# 아두이노 스케치 설치

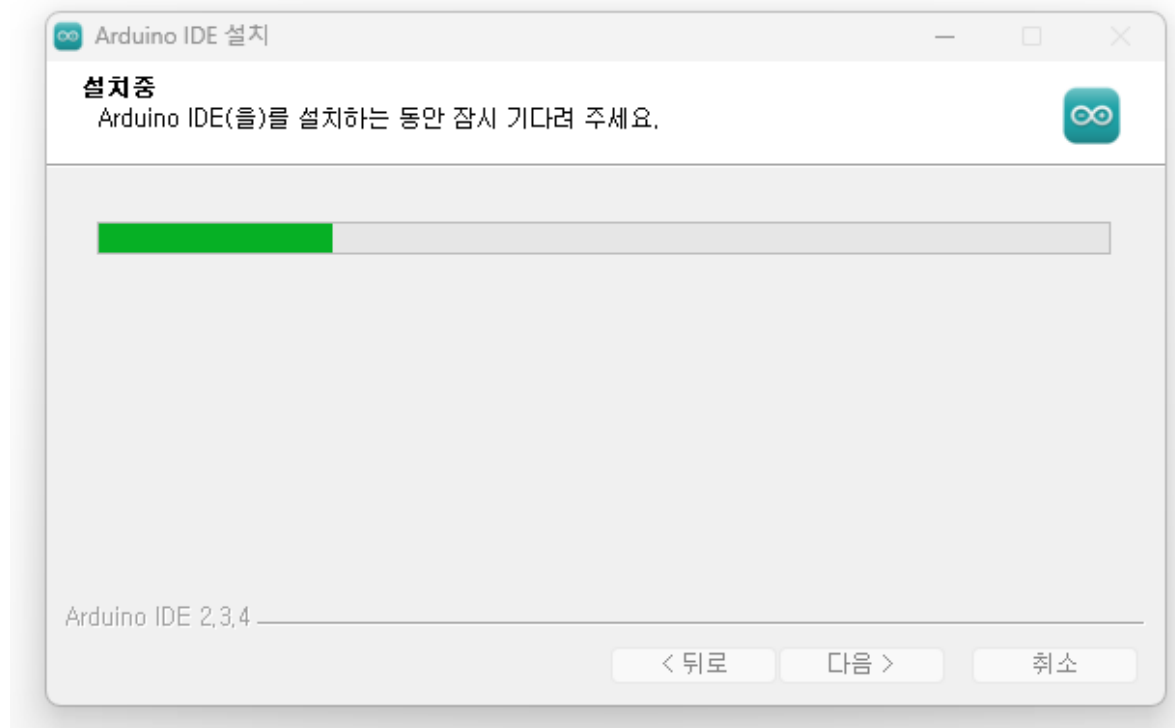
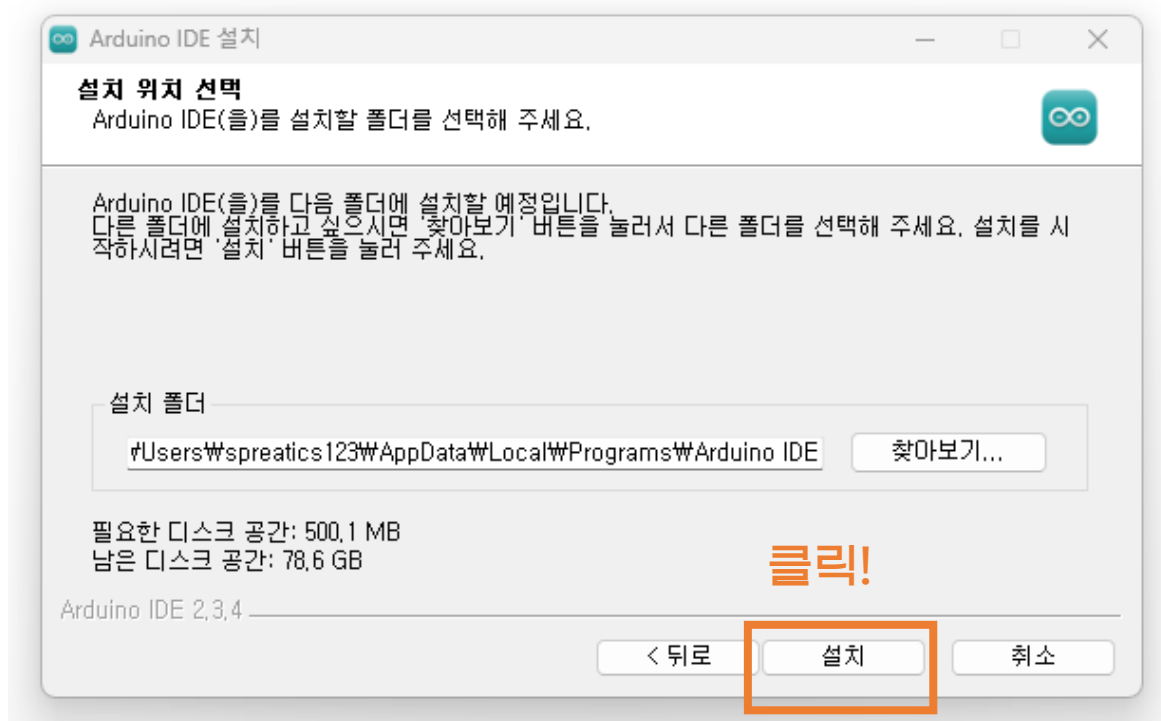


arduino-ide\_2.3.4\_Windows\_64bit.exe

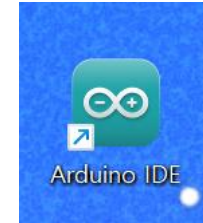
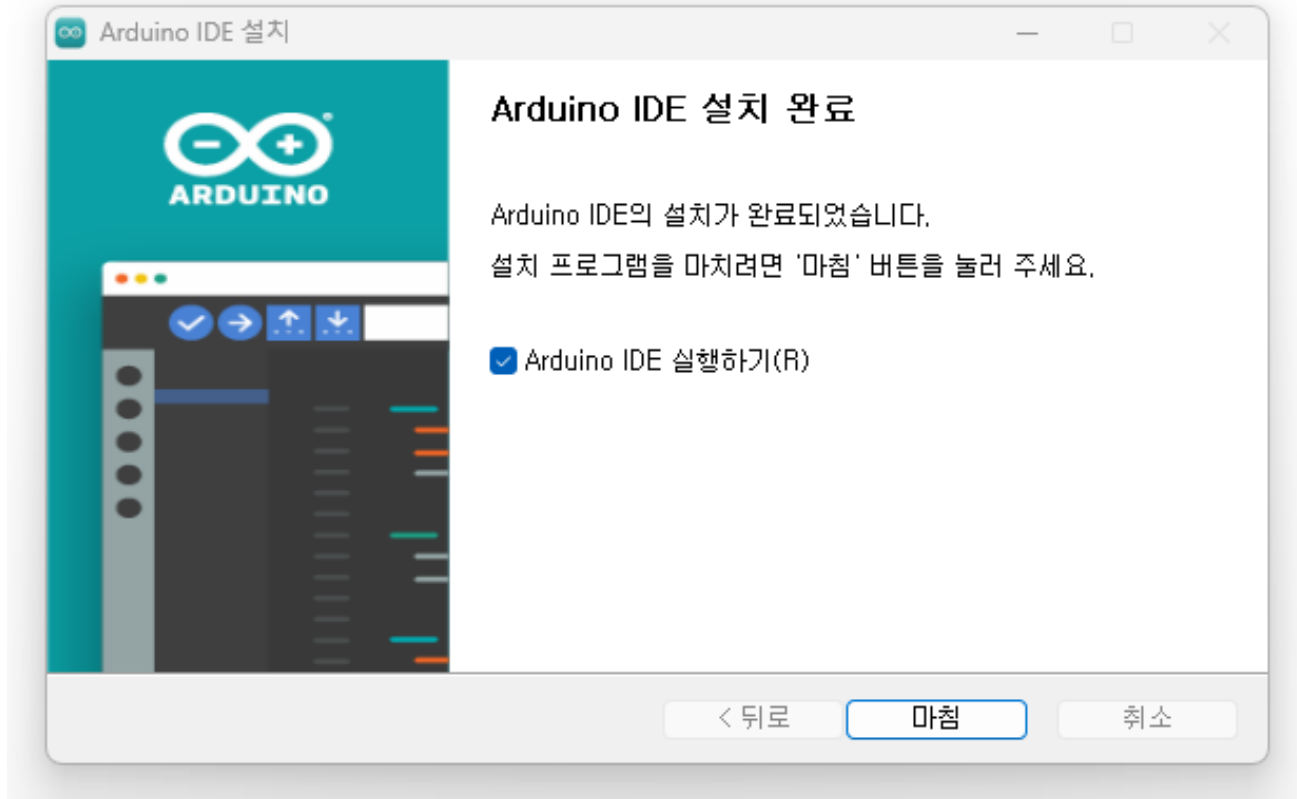
다운로드 후 아이콘 선택!



# 아두이노 스케치 설치

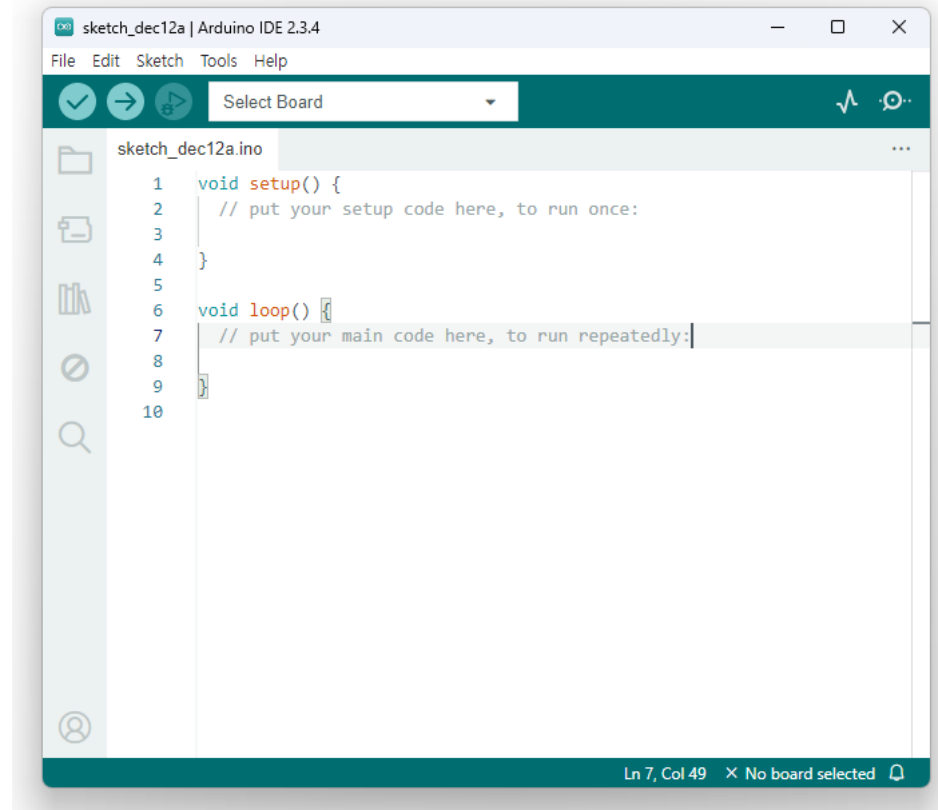


# 아두이노 스케치 설치



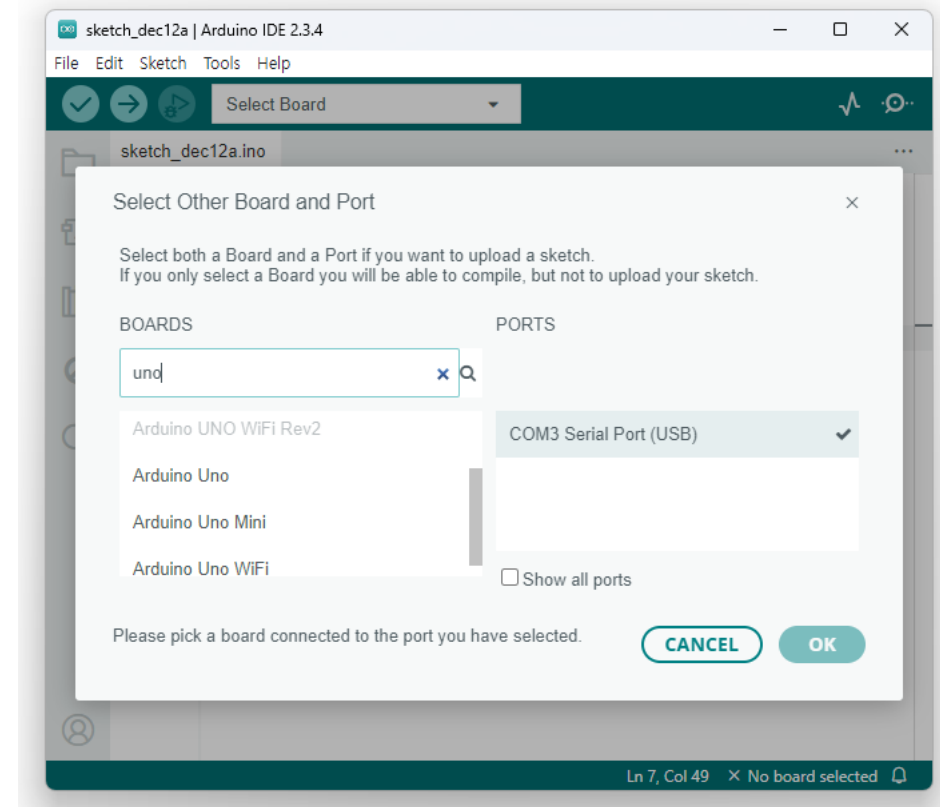
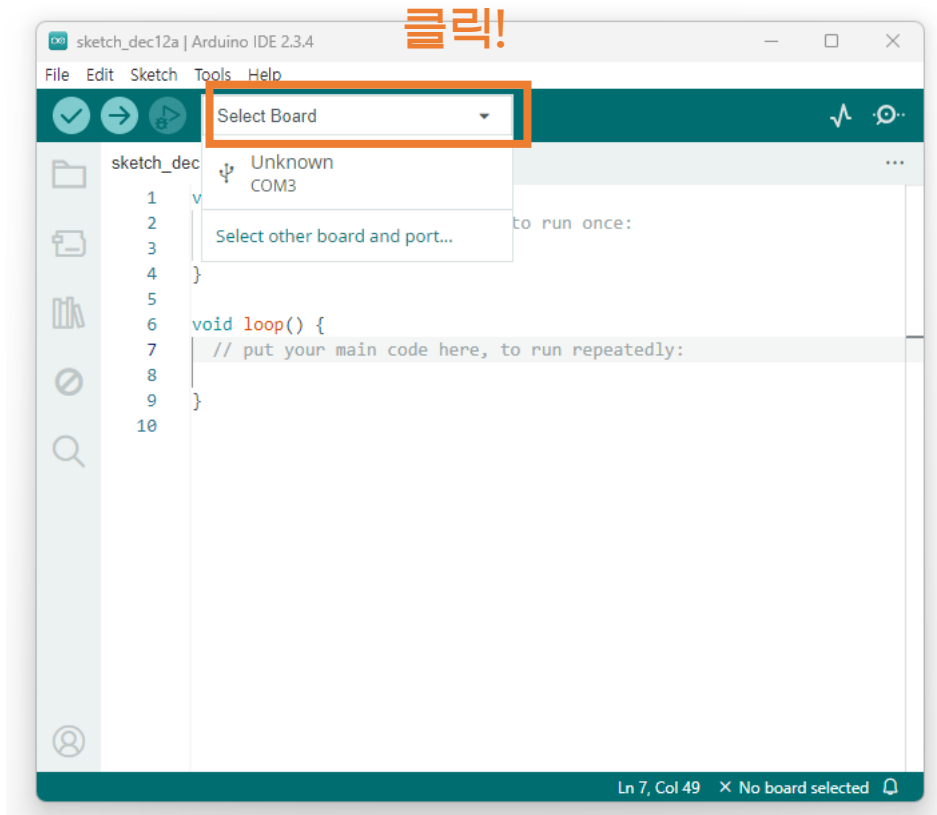
설치 후 아이콘 선택!

# 아두이노 스케치 설치

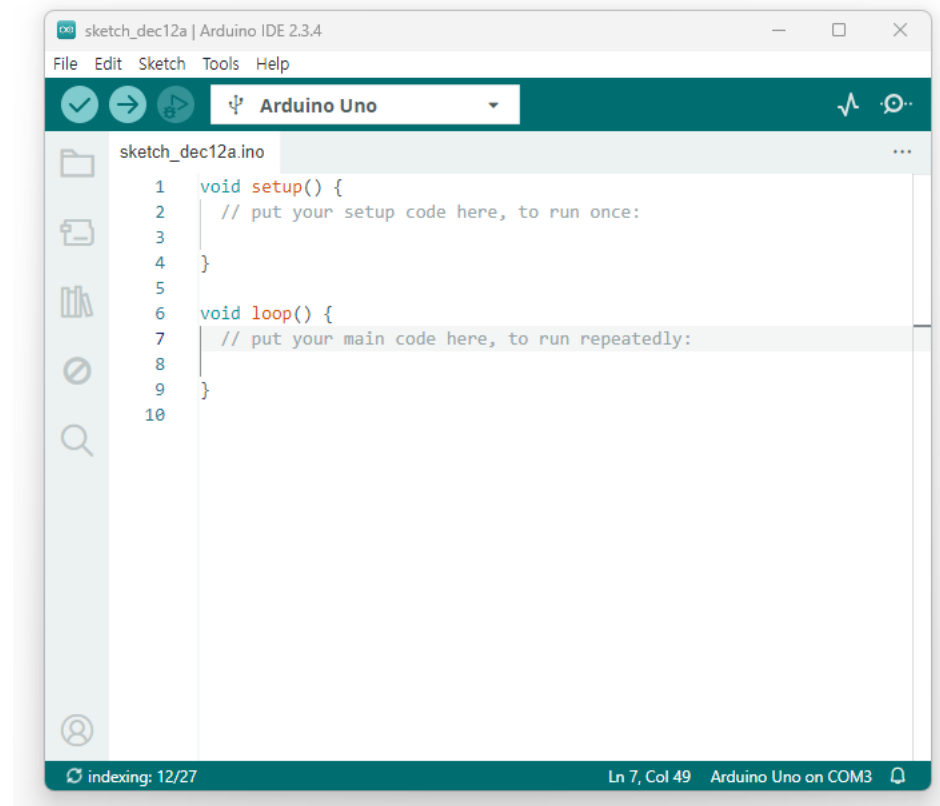
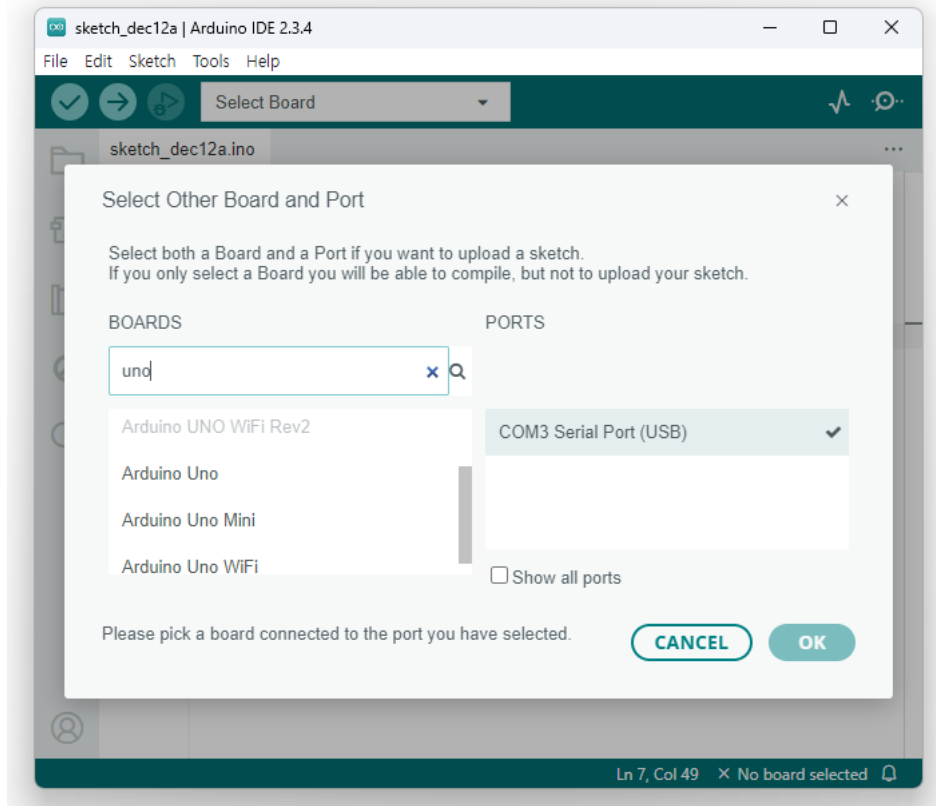


- 아두이노 스케치 첫 실행시 화면

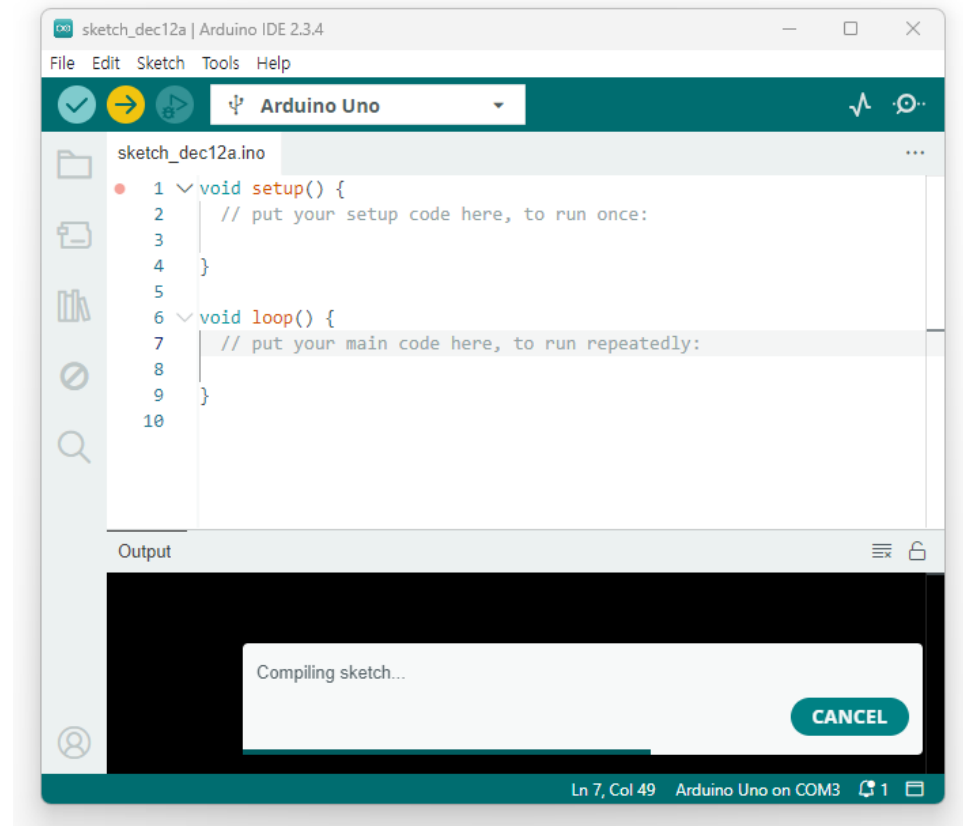
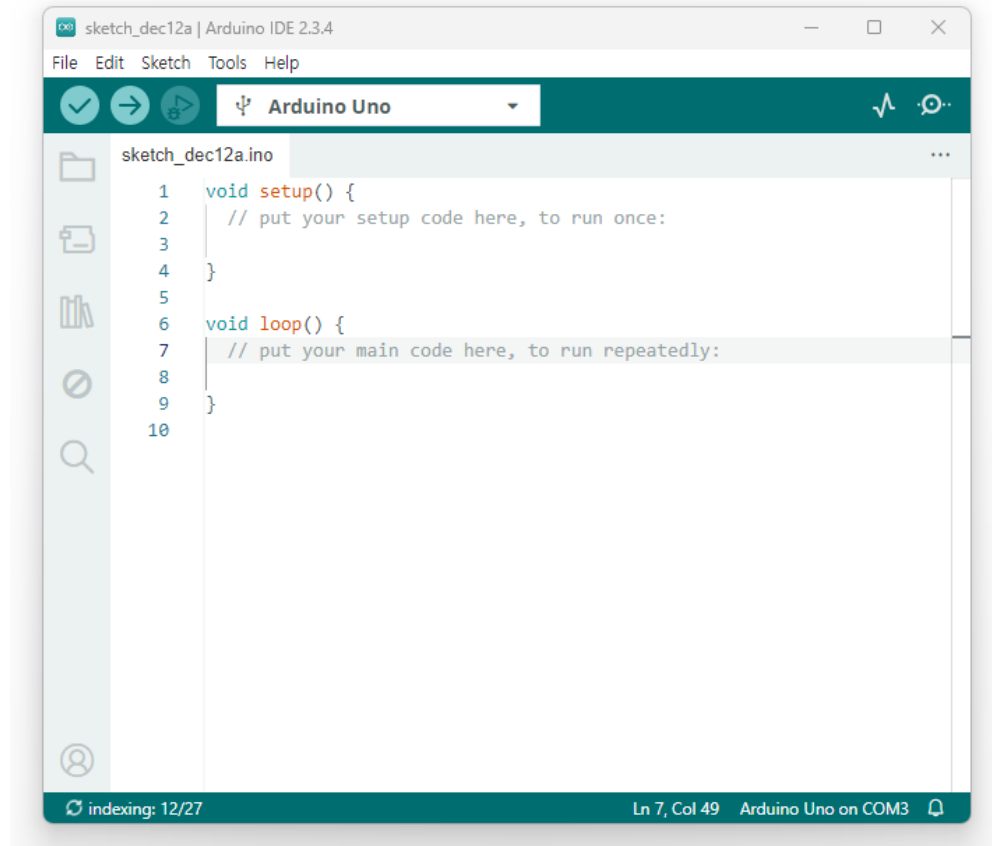
# 아두이노 스케치 설치



# 아두이노 스케치 설치

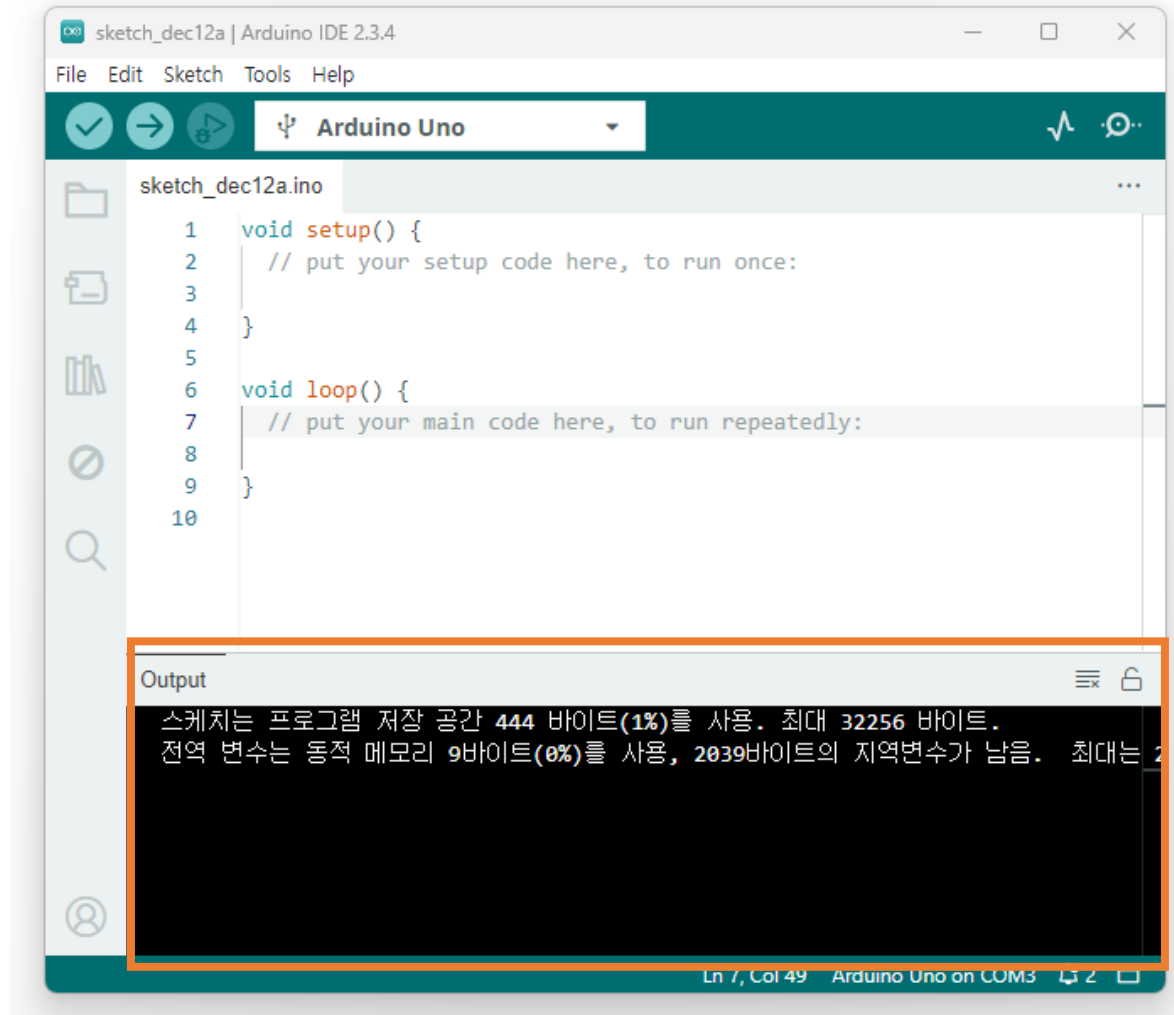


# 아두이노 스케치 설치

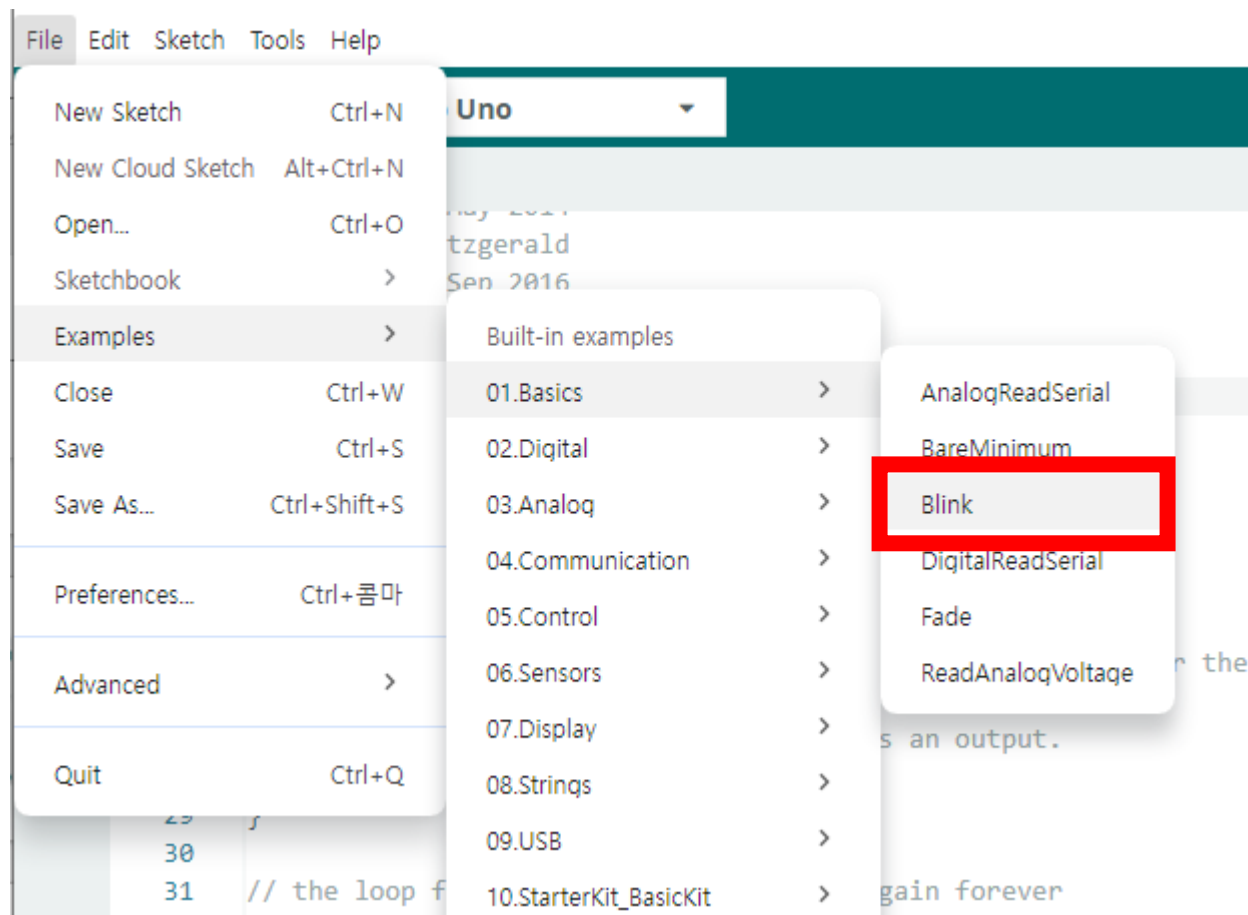




# 아두이노 스케치 설치



# 아두이노 스케치 예제



# 아두이노 스케치 예제

```
25 // the setup function runs once when you press reset or power the board
26 ✓ void setup() {
27     // initialize digital pin LED_BUILTIN as an output.
28     pinMode(LED_BUILTIN, OUTPUT);
29 }
30
31 // the loop function runs over and over again forever
32 ✓ void loop() {
33     digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
34     delay(1000);                      // wait for a second
35     digitalWrite(LED_BUILTIN, LOW);  // turn the LED off by making the voltage LOW
36     delay(1000);                      // wait for a second
37 }
38
```

# 아두이노 스케치 예제

```
25 // the setup function runs once when you press reset or power the board
26 ✓ void setup() {
27     // initialize digital pin LED_BUILTIN as an output.
28     pinMode(LED_BUILTIN, OUTPUT);
29 }
30
31 // the loop function
32 ✓ void loop() {
33     digitalWrite(LED_BL
34     delay(1000);
35     digitalWrite(LED_BL
36     delay(1000);
37 }
38
```

- setup() 함수
- 아두이노 스케치가 시작될 때 호출되는 함수
- 보드의 전원 인가 혹은 리셋 후에 한 번만 수행되며, 변수와 핀 모드 설정, 하드웨어 초기화 작업, 직렬 통신 속도 설정 등을 수행한다.

# 아두이노 스케치 예제

```
25 // the setup function runs once when you press reset or power the board
26 void setup() {
27     // initialize digital pin LED_BUILTIN as an output
28     pinMode(LED_BUILTIN, OUTPUT);
29 }
30
31 // the loop function
32 void loop() {
33     digitalWrite(LED_BUILTIN, HIGH);
34     delay(1000);
35     digitalWrite(LED_BUILTIN, LOW);
36     delay(1000);
37 }
38
```

- pinMode(pin, mode)
- Pin에 해당하는 핀의 입출력 속성을 지정하는 함수
- mode의 종류
  - INPUT(기본값), OUTPUT, PULLUP

# 아두이노 스케치 예제

- loop() 함수
- 루프를 반복하여 프로그램이 기능을 수행하고 응답할 수 있도록 하며, 아두이노 보드를 능동적으로 제어한다.

```
25 // the setup function
26 ✓ void setup() {
27     // initialize digit
28     pinMode(LED_BUILTIN, OUTPUT);
29 }
30
31 // the loop function runs over and over again forever
32 ✓ void loop() {
33     digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
34     delay(1000); // wait for a second
35     digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
36     delay(1000); // wait for a second
37 }
38
```

# 아두이노 스케치 예제

```
25 // the setup function
26 void setup() {
27     // initialize digital
28     pinMode(LED_BUILTIN, OUTPUT);
29 }
30
31 // the loop function runs over and over again forever
32 void loop() {
33     digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
34     delay(1000); // wait for a second
35     digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
36     delay(1000); // wait for a second
37 }
38
```

- digitalWrite(pin, value)
- 출력으로 지정된 디지털 핀에 “HIGH” 혹은 “LOW” 값을 보내서 해당 값을 갖도록 한다.