

INDEX



마이크로컨트롤러



아두이노



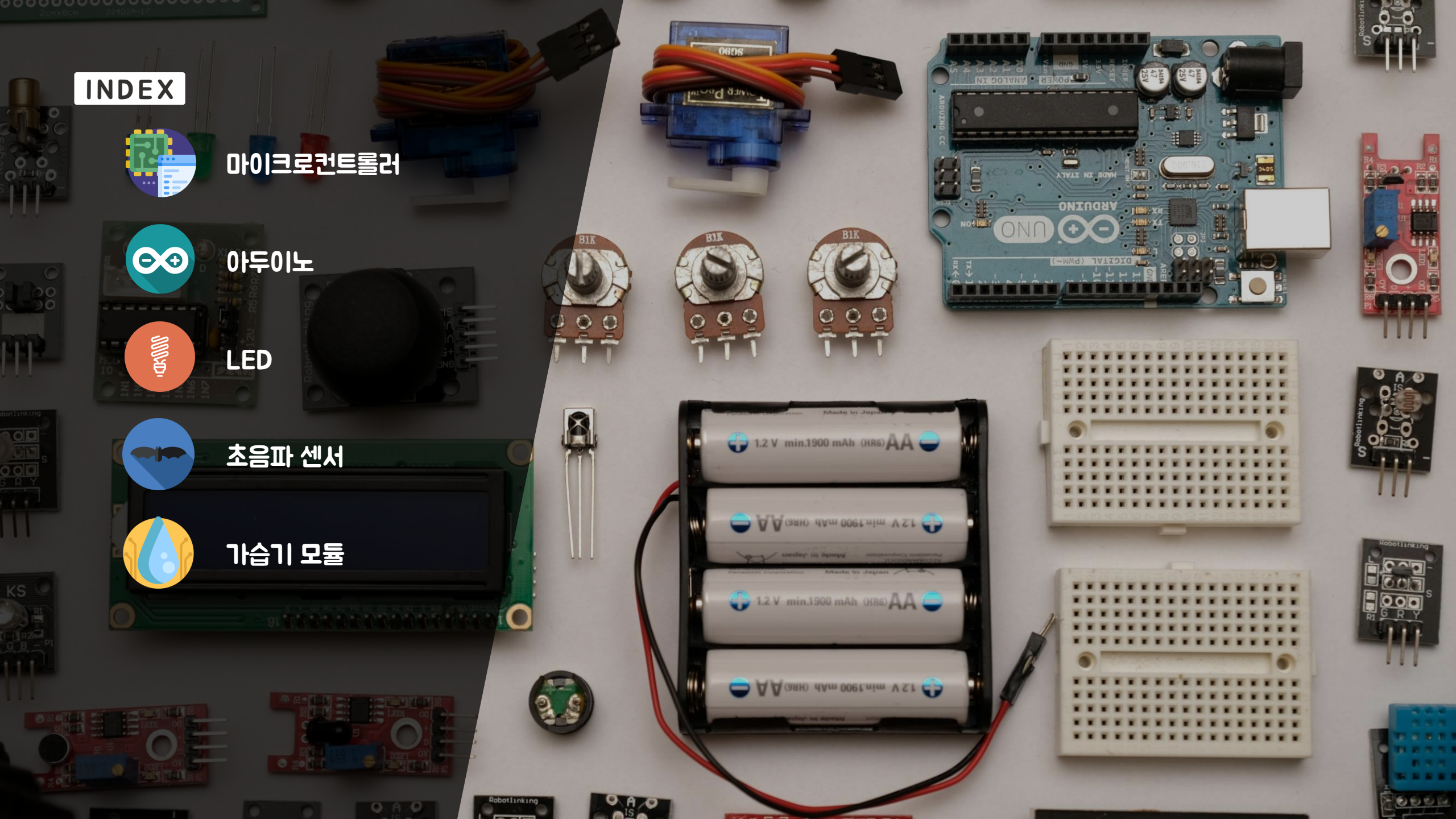
LED



초음파 센서



가습기 모듈





마이크로컨트롤러

MCU, Micro Controller Unit

제어 장치 제작을 목적으로 중앙 처리 장치에 입출력 및 메모리 장치를 포함하여

하나의 칩으로 구현한 마이크로 프로세서의 일종



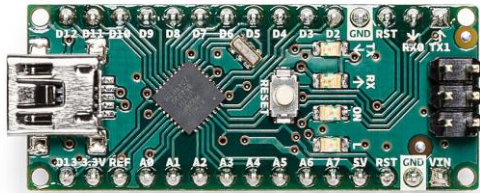
아두이노
Arduino

마이크로 컨트롤러 보드와 소프트웨어 개발환경까지 함께 이르는 말





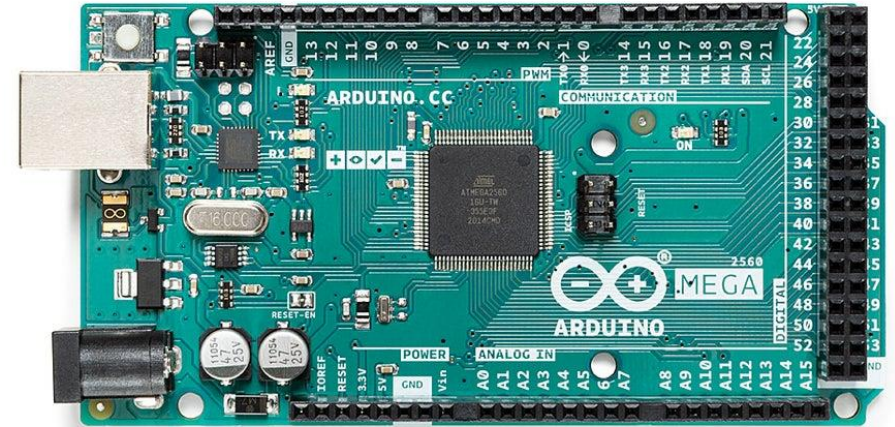
아두이노 Arduino



아두이노 나노



아두이노 우노

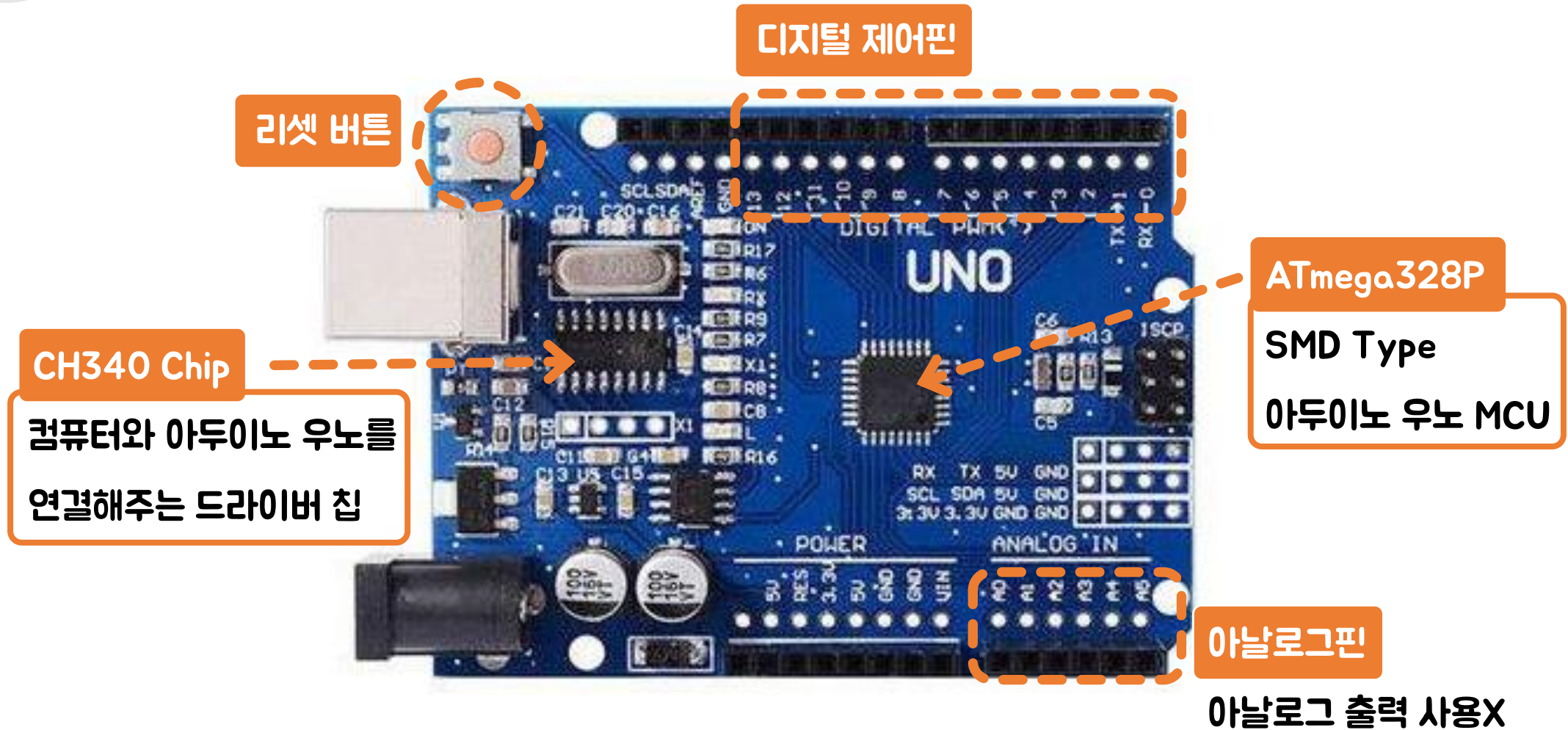


아두이노 메가

이 외에도 아두이노 프로, 아두이노 레오나르도, 아두이노 듀에 등 다양하게 존재하며
사용 목적에 맞게 선택하여 작업하면 됩니다.



아두이노 우노 Arduino Uno





아두이노 우노

Arduino Uno 전력 공급 방법

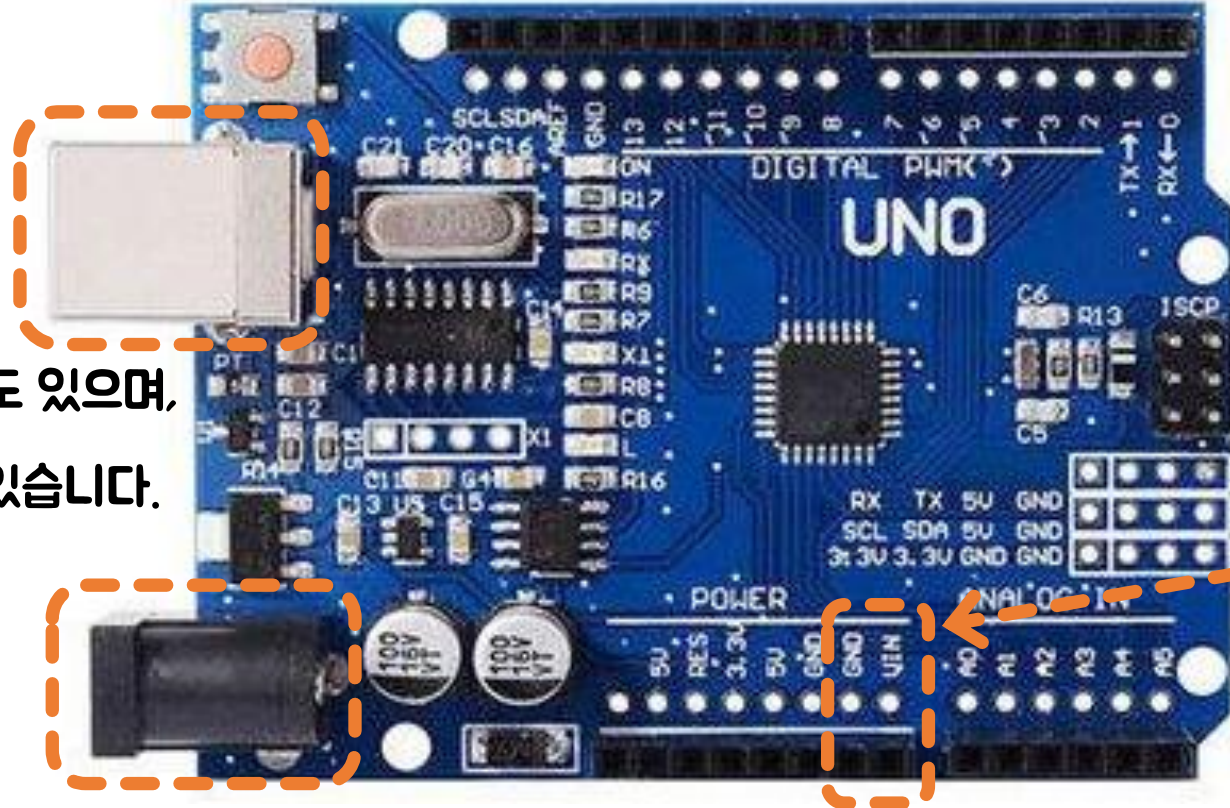
1. USB

소스를 업로드 할 수도 있으며,
전원을 공급해줄 수 있습니다.

2. DC 배럴잭

7~12V DC 입력

레귤레이터를 통해 5V 변환되어 아두이노 전력 공급



역전압이 생길 수 있으니
전원 공급 방법은 하나를 택일 하여
사용해주세요.

3. VIN

VIN에 +극 연결

GND에 -극 연결 전력 공급



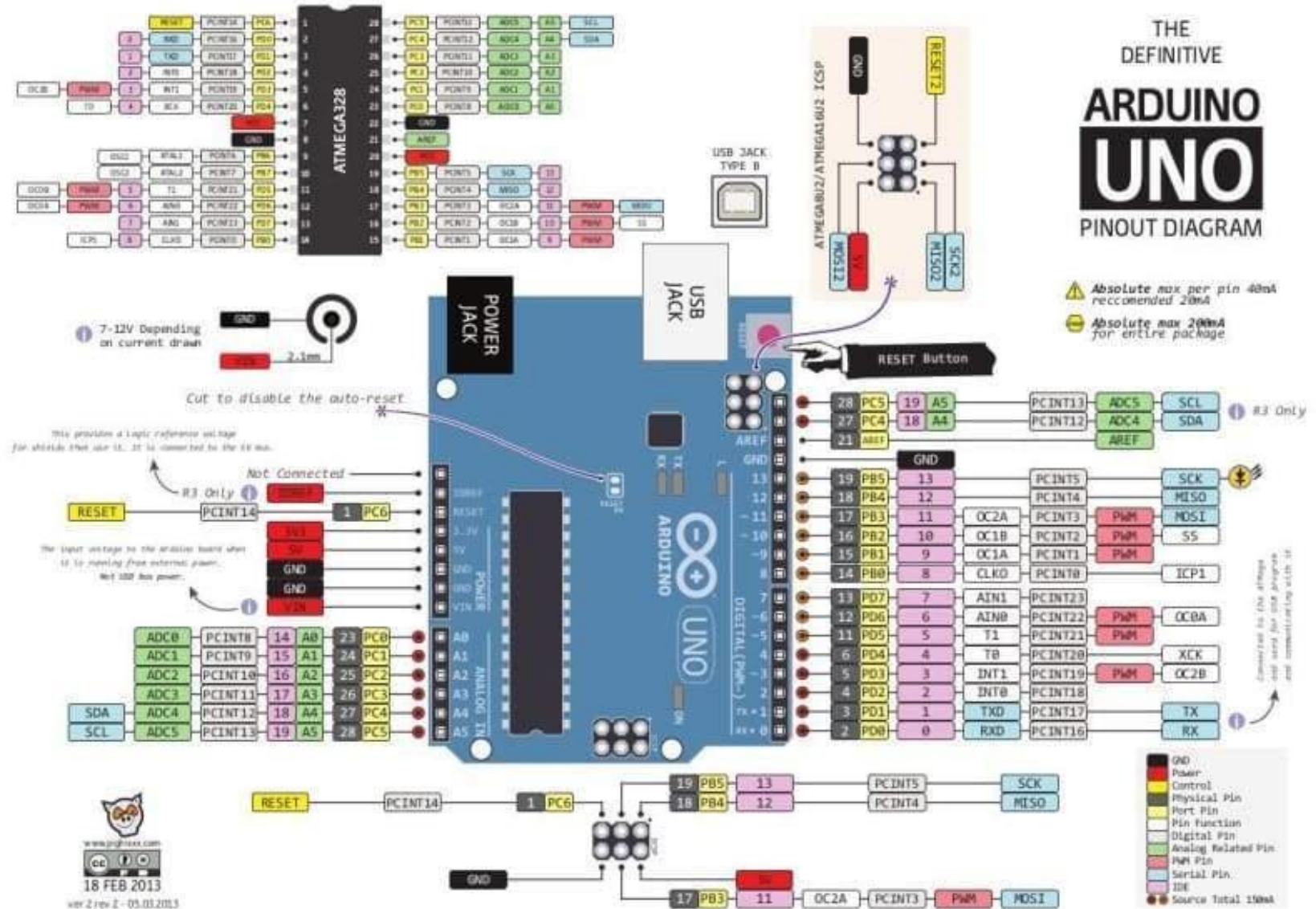
아두이노 우노

Arduino Uno

항목	내용	비고
마이크로컨트롤러	ATmega328	
동작 전압	5V	
입력 전압	7~12V	추천 입력 범위
디지털 입출력 핀	14개	6개 PWM 출력 핀
아날로그 입력 핀	6개	
플래시 메모리	32KB	ATmega328, 부트로더 0.5KB
SRAM	2KB	ATmega328
EEPROM	1KB	ATmega328
클록 주파수	16MHz	



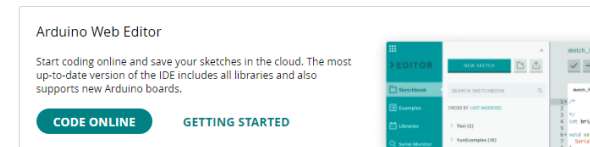
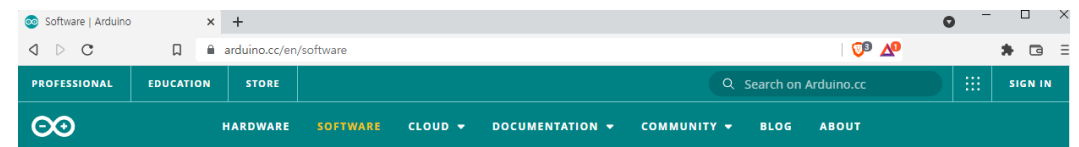
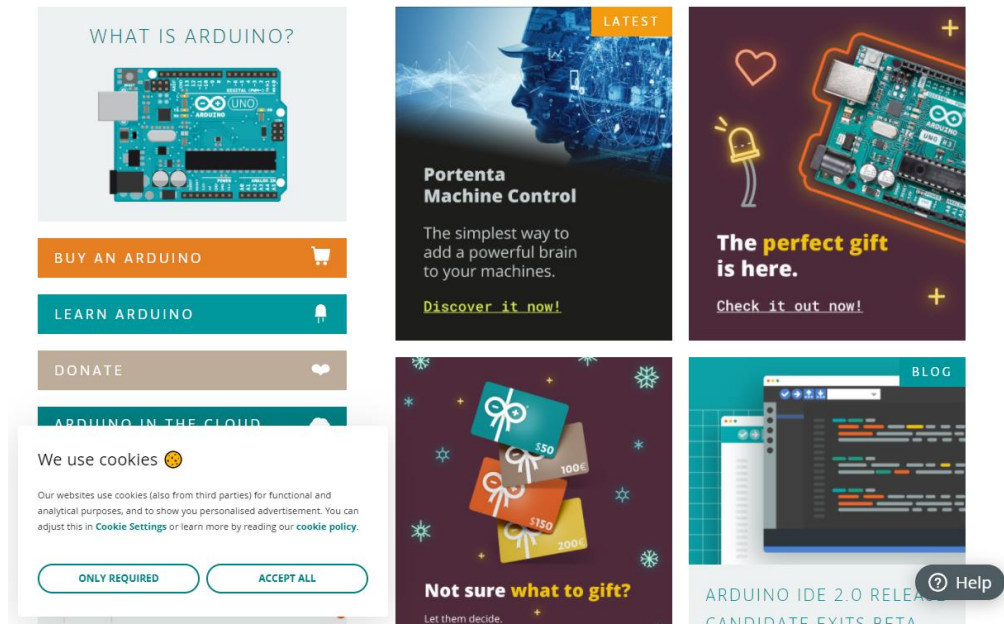
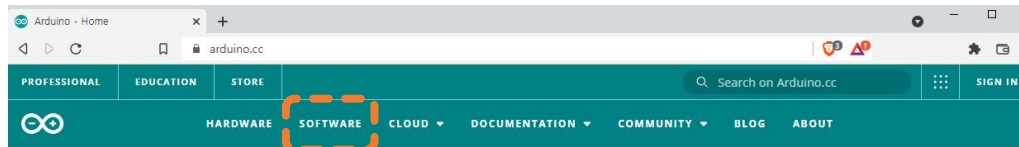
아두이노 우노 Arduino Uno





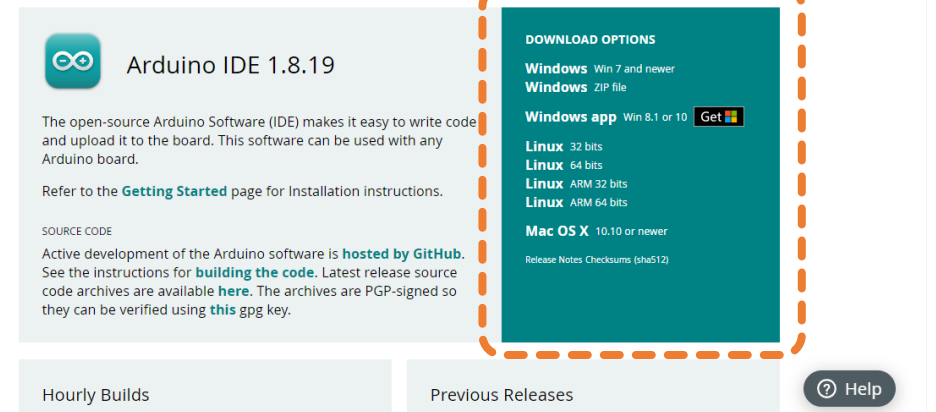
아두이노 IDE 설치 Install Arduino IDE

<https://arduino.cc>



사용하시는 OS에 맞게
다운로드 해주시면 됩니다.

Downloads

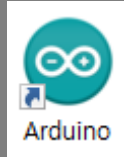


설명에는 Windows Win7 and newer로 진행하겠습니다.



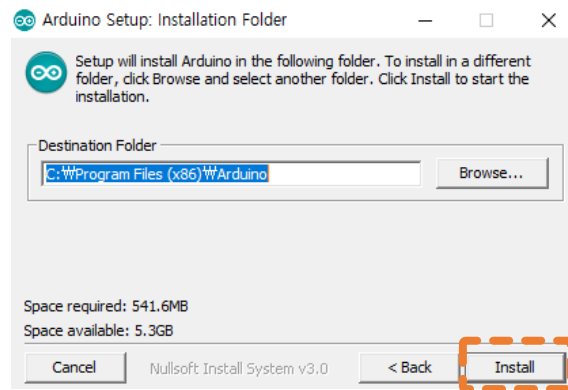
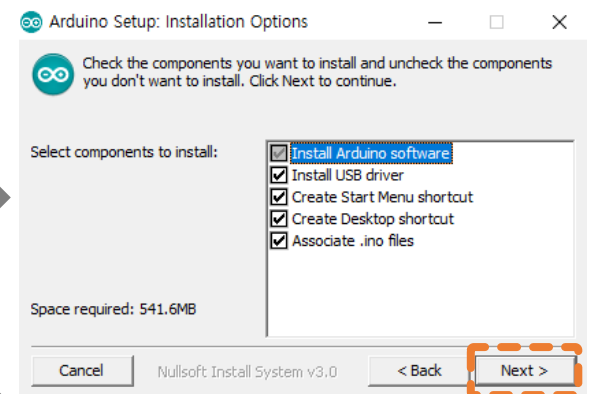
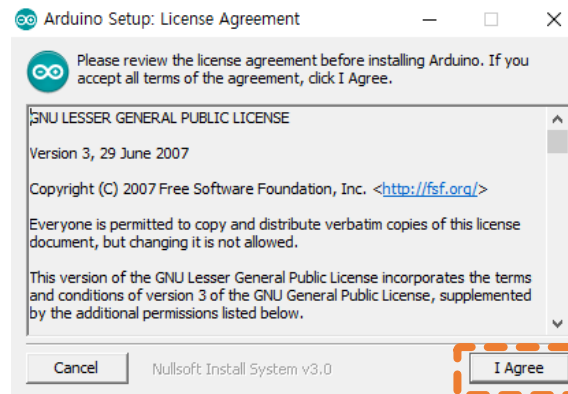
아두이노 IDE 설치 Install Arduino IDE

클릭시 해당 그림처럼 나오게 됩니다.



내용을 동일하게 실행하면
바탕화면에 왼쪽과 같은 아이콘 생성 됩니다.
해당 아이콘을 실행하면 아두이노 IDE를 실행할 수 있습니다.

다운 받아진 파일을 실행하고 보안경고가 나오면 '확인'



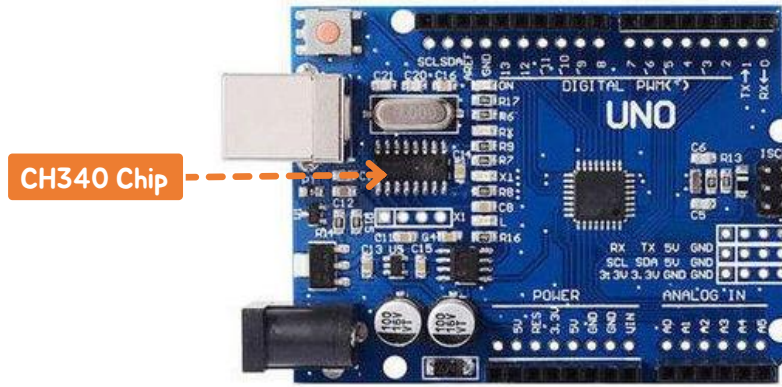


드라이버 설치 Install Driver

우리가 사용하는 드라이버 칩은 CH340 Chip이기에
해당 드라이버를 설치해주도록 하겠습니다.
다른 드라이버 칩이라면 다른 드라이버를 설치하셔야 합니다.

<https://blog.naver.com/neo3ds/221428425034>

네오3D 솔루션



아두이노 대포가습기를 만든 네오3D솔루션 블로그에
CH340 알집 파일을 제공하고 있습니다. (제공하는 파일은 윈도우용)
해당 파일을 다운받아 주세요.

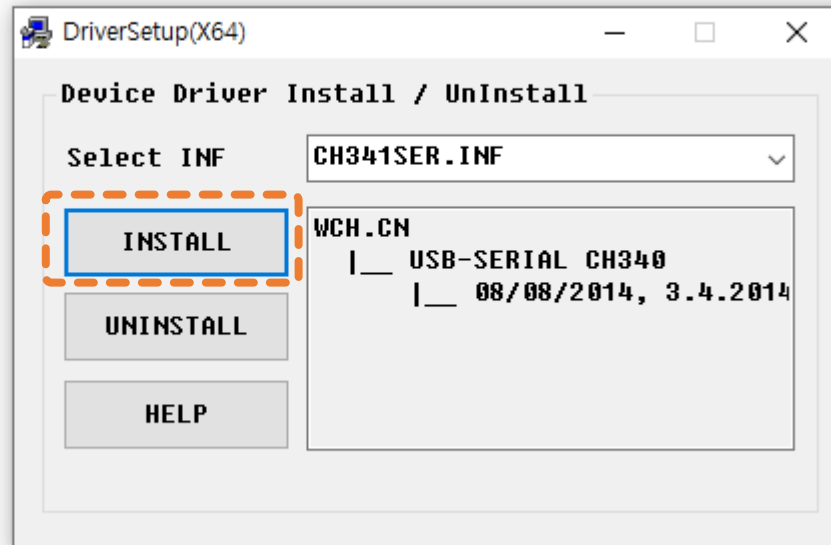
다운이 안되신다면 <https://sparks.gogo.co.nz/ch340.html> 해당 사이트를
이용 해주셔도 됩니다.



드라이버 설치 Install Driver

압축 파일을 다운 받으셨다면, 압축을 풀어주세요.

CH34x_Install_Windows_v3_4.EXE 2017-01-24 오후 1:17 응용 프로그램 238KB



압축 푸신 파일을 실행하게 되시면
하나의 프로그램이 실행되게 됩니다.
INSTALL을 클릭하여 주세요.

만약 Fail이 뜨게 된다면 UNINSTALL을 누르고
다시 INSTALL을 실행해주세요.

INSTALL이 성공했다면 이 창을 껐주셔도 됩니다.



Comport 확인 Check Comport

장치와 연결이 되었다면 Comport 번호를 알아야 시리얼 통신을 할 수 있습니다.



+



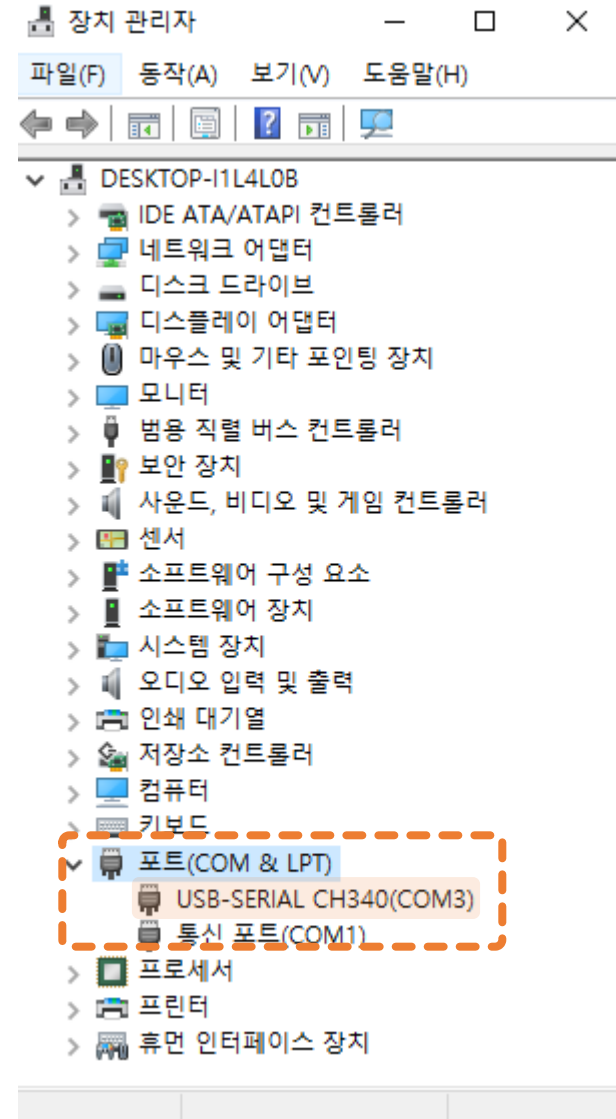
키를 동시에 누르면 검색창이 나오게 되는데 **장치 관리자**를 치고
해당 제어판을 들어가 주세요.



장치 관리자
제어판

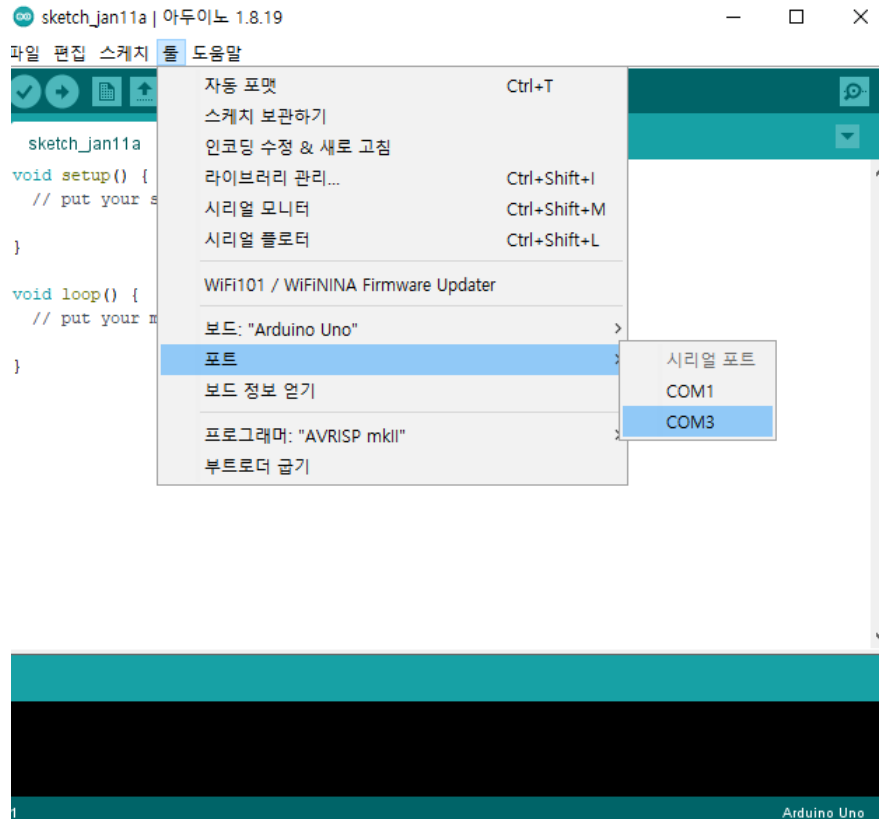
포트 내용 중에 **USB-SERIAL CH340(COM번호)**

여기에 COM번호를 통해서 시리얼 통신을 진행할 것입니다.





IDE 세팅 Setting IDE



아두이노 우노를 업로드할 준비가 되었는지 확인하기 위해
세팅을 확인하면서 안되어 있다면 변경해주도록 하겠습니다.
내용은 아래와 같습니다.

메뉴 부분에 `툴`

- 보드 Arduino Uno
- 프로그래머: AVRISP mkII
- 포트: 연결할 컴포트

보드는 IDE의 버전에 따라 상이할 수 있으나,
Arduino Uno와 관련되기만 하면 됩니다.

연결할 컴포트는 장치 관리자를 통해 알아낸
USB-SERIAL CH340의 Com숫자 번호입니다.
해당 번호를 클릭해주시고,
툴을 다시 클릭해보면 비워 있던 포트 내용이 채워집니다.



IDE 세팅

Setting IDE

환경설정

설정 네트워크

스케치북 위치:
C:\Users\Wjune\Documents\Arduino [찾아보기]

에디터 언어: System Default (아두이노를 재시작해야 함)

에디터 글꼴 크기: 12

Interface scale: ☒ 자동 100% (아두이노를 재시작해야 함)

테마: 디폴트 테마 (아두이노를 재시작해야 함)

다음 동작중 자세한 출력 보이기: ☐ 컴파일 ☐ 업로드

컴파일러 경고: None

☐ 줄 번호 표시 ☐ 코드 폴딩 사용하기

☒ 업로드 후 코드 확인하기 ☐ 외부 에디터 사용

☒ 시작시 업데이트 확인 ☒ 검증 또는 업로드 할 때 저장하기

☐ Use accessibility features

추가적인 보드 매니저 URLs [Browse]

추가적인 환경 설정은 파일에서 직접 편집할 수 있습니다
C:\Users\Wjune\AppData\Local\Arduino15\preferences.txt
(아두이노가 실행되지 않는 경우에만 수정 가능)

확인 취소

메뉴 '파일' → 환경설정

환경설정에 들어가서 글꼴 크기를 바꾼다거나,

코드의 줄 번호를 볼 수 있게 설정한다거나

환경을 변경할 수 있습니다.



발광 다이오드

LED, Light-Emitting Diode



LED + / - 구분하는 방법

1. 다리 길이 구별법

길이가 긴 쪽 +
짧은 쪽 -

2. 금속판 폭 구별법

폭이 좁은 곳 +
폭이 넓은 곳 -

금속판 폭을 통해 구별하는 것이 확실한 방법입니다.



발광 다이오드

LED, Light-Emitting Diode



왼쪽의 사진에 예시처럼
LED는 우리의 일상 속에서 흔하게 볼 수 있습니다.
그러면 LED를 사용하는 이유는 무엇일까요?





발광 다이오드

LED, Light-Emitting Diode

일반 조명과 LED 조명 비교



자료: 지식경제부

LED을 사용하는 이유는 다음과 같습니다.

1. 에너지 효율이 좋다
2. 친환경적이다
3. 다루기 쉽다

다루기 쉽다라고 되어 있는데 아두이노 우노를 통해 얼마나 다루기 쉬운지 알아보도록 하겠습니다.



발광 다이오드

LED, Light-Emitting Diode

PIN MAP

UNO

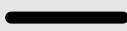
LED

D6

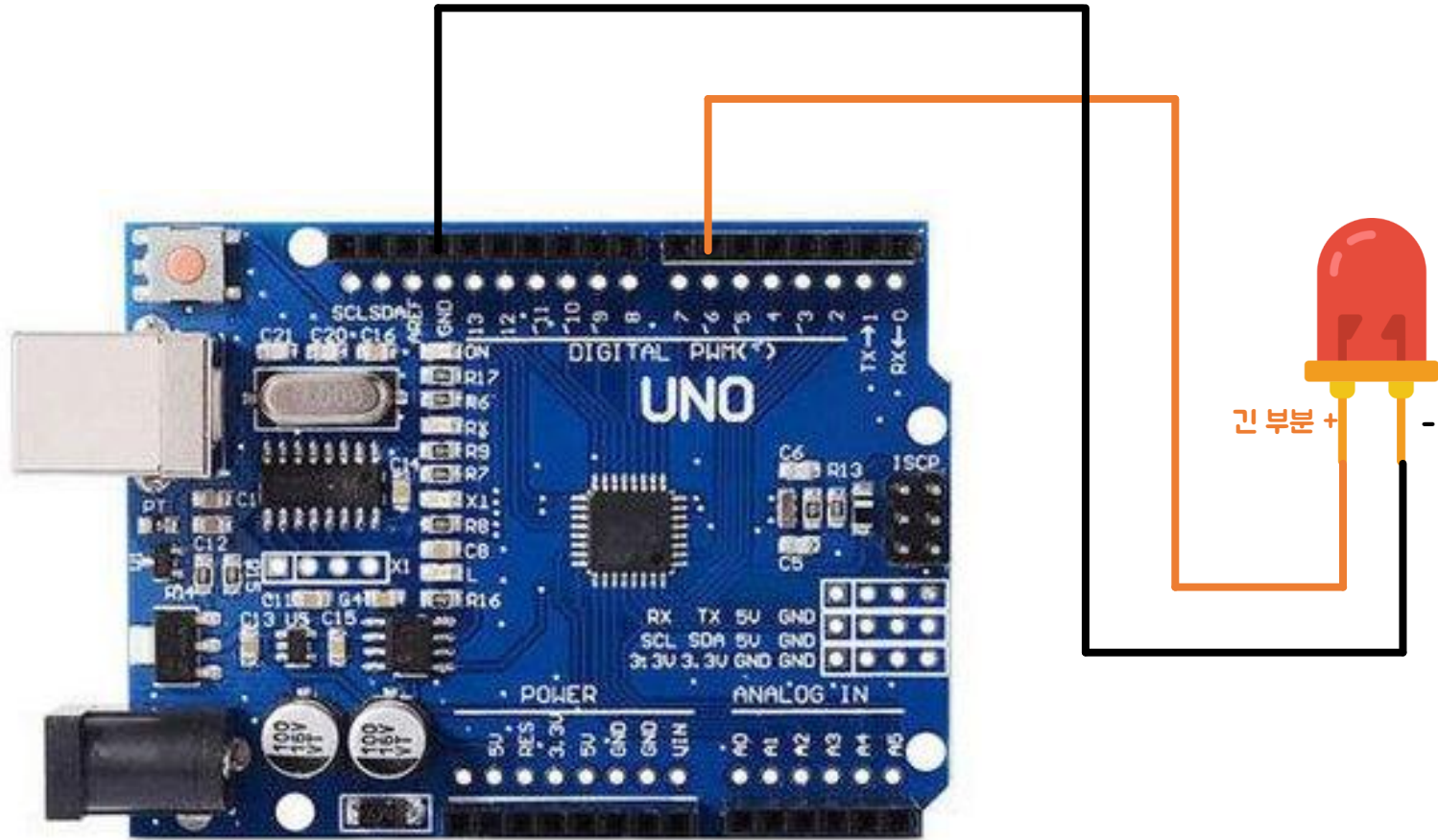


+

GND



-





아두이노 문법

Arduino

setup()

아두이노 보드에 전원이 켜지고
초기에 한번 실행하는 영역(함수)
cf) 보통 설정하는 코드를 작성합니다.

ex) `pinMode()`, `Serial.begin()` 등

loop()

아두이노 보드에 전원이 켜지고
지속적으로 실행하는 영역(함수)



아두이노 문법

Arduino

pinMode()

pinMode(핀번호, 입출력 설정);

핀번호

입출력 설정: OUTPUT / INPUT

- OUTPUT: 출력
- INPUT: 입력

digitalWrite()

digitalWrite(핀번호, 전압 설정);

핀번호

전압 설정: HIGH / LOW

- HIGH: 5V 전압 인가
- LOW: 0V 전압 (=GND 사용 가능)

digitalRead()

digitalRead(핀번호);

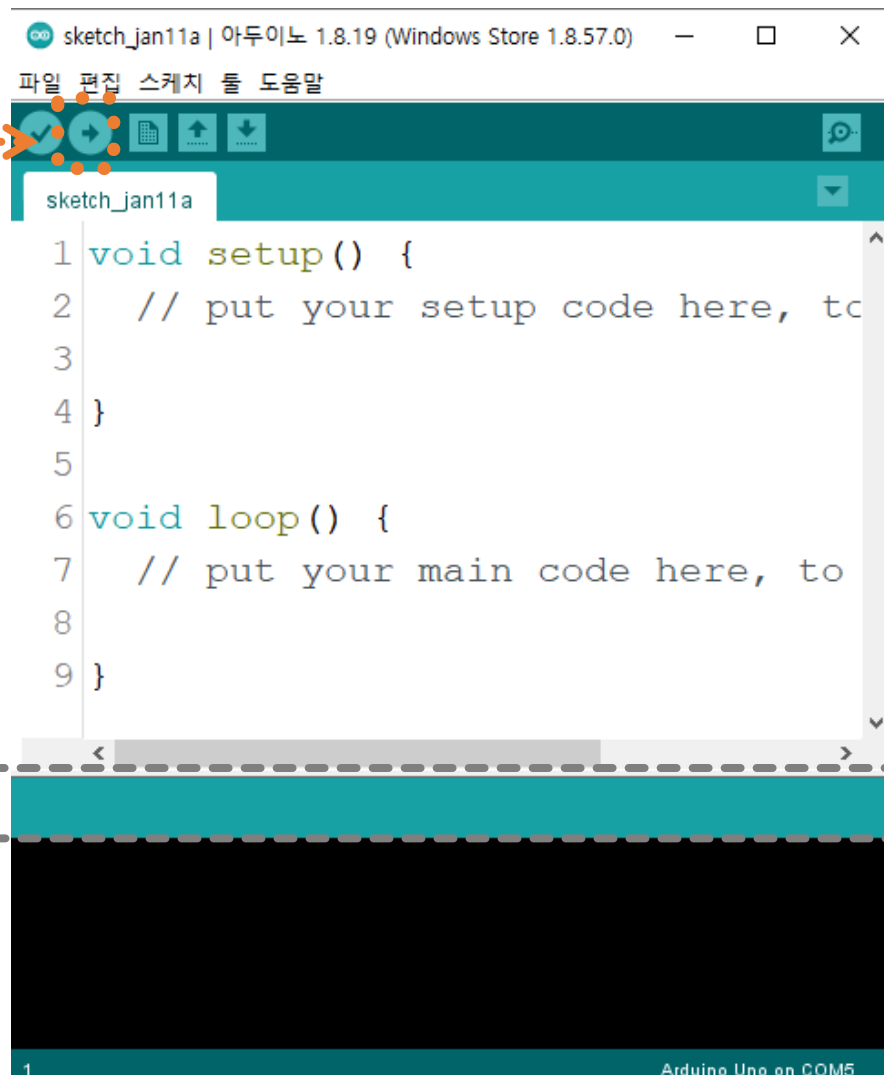
해당 핀의 센서값(0~1)을 가져옵니다.

핀번호



업로드 Upload

업로드 버튼



업로드 버튼을 누르면

오른편에 진행상태바가 나오고 완료가 되면

왼쪽편에 '업로드 완료'라고 나오게 됩니다.



발광 다이오드

LED, Light-Emitting Diode

- LED ON

LED를 켜주세요.



발광 다이오드

LED, Light-Emitting Diode

- LED ON

LED를 켜주세요.

```
1 void setup() {  
2   // put your setup code here, to run once:  
3   pinMode(6, OUTPUT);  
4   digitalWrite(6, HIGH);  
5 }  
6  
7 void loop() {  
8   // put your main code here, to run repeatedly:  
9  
10 }
```

// 이 앞에 있으면 한 줄 주석입니다.

주석은 설명을 작성할 때 사용합니다.

코드 실행시에 작동되는 문법은 아닙니다.



아두이노 문법 Arduino

`delay()`

`delay(ms);`

프로그램을 ms 시간만큼 멈추게 합니다.

ms: milliseconds

ex) 1000(ms) -> 1초

Delay를 쓰면 여러가지 동작을 하면서
지속적인 센서의 값을 받거나 얻어야 할 때 문제가 생길 수 있습니다.
이러한 방법을 해결하기 위해서는
millis나 다른 방법을 사용해야 합니다.
대포 가습기 완성에 이러한 부분까지 필요 없으니 언급만 하고
넘어가도록 하겠습니다.



발광 다이오드

LED, Light-Emitting Diode

- LED Blink

LED를 1초 간격으로 켜졌다가 꺼졌다가를 반복하게 해주세요.



발광 다이오드

LED, Light-Emitting Diode

- LED Blink

LED를 1초 간격으로 켜졌다가 꺼졌다가를 반복하게 해주세요.

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



PWM이란

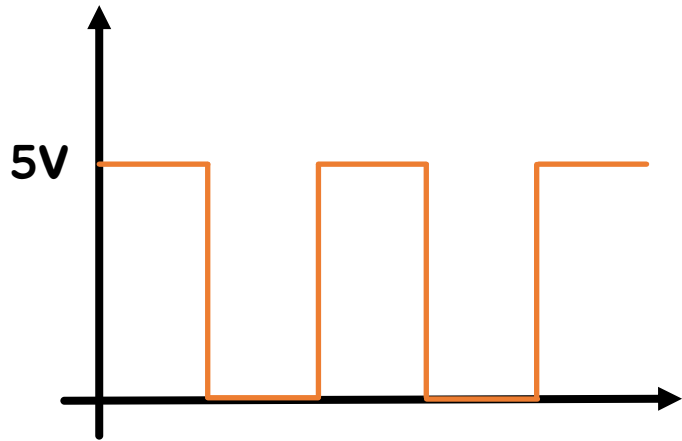
디지털 신호 -> 아날로그 신호

D: 0%



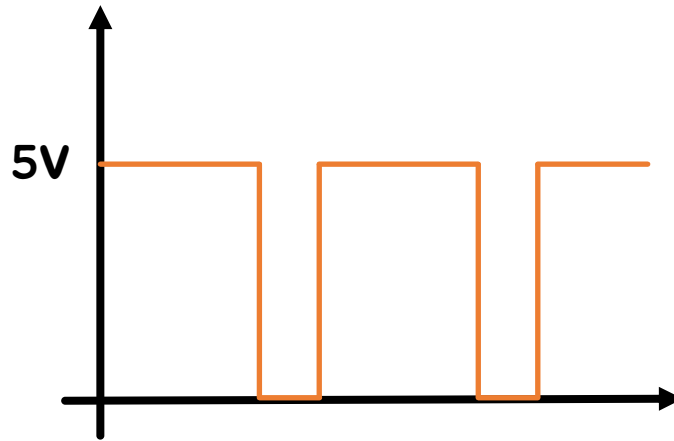
PWM이란

디지털 신호 -> 아날로그 신호



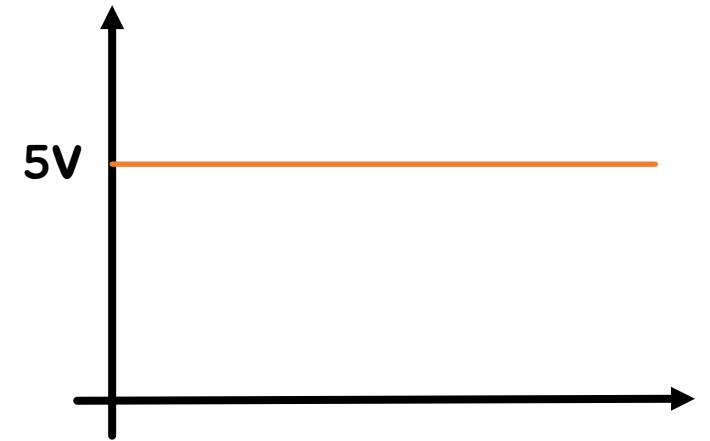
50% Duty Cycle

2.5V



75% Duty Cycle

3.75V



100% Duty Cycle

5V



아두이노 문법 Arduino

analogWrite()

analogWrite(핀번호, 세기값);

핀번호

세기값: 0~255

analogRead()

analogRead(핀번호);

해당 핀의 센서 값(0~1023)을 가져옵니다.

핀번호

analogWrite나 analogRead를 사용할 때는 pinMode를 사용하지 않습니다.

코드 가독성을 위해서 사용하기도 하니 이점은 이해하시고 넘어가시기만 하면 됩니다.

cf) 아날로그 핀을 디지털 입/출력으로 사용하기 위해서는 pinMode을 사용해주셔야 합니다.



발광 다이오드

LED, Light-Emitting Diode

- 세기 조정

LED의 밝기 세기를 나타내주세요.



발광 다이오드

LED, Light-Emitting Diode

- 세기 조정

LED의 밝기 세기를 나타내주세요.

```
1 void setup() {  
2   analogWrite(6, 30);  
3   delay(500);  
4   analogWrite(6, 100);  
5   delay(500);  
6   analogWrite(6, 255);  
7 }  
8  
9 void loop() {  
10  
11 }
```



변수 variable

변수

변수는 '변하는 수' 라는 의미로

프로그래밍에서는 숫자나 문자 같은 데이터를 저장하는 공간을 의미합니다.

$x = x + 1$ 이라 하면 수학적으로는 모순이지만,

프로그래밍에서는 x 라는 값에 1을 더한 값을 다시 x 에 넣는 식입니다.

쉬운 예시로, 우리의 나이가 1년이 지나면 한살 먹는 것과 유사하다고 보시면 됩니다.

$x = x + 5$ 라면 x 라는 값에 5을 더한 값을 다시 x 에 넣어주는 것이겠죠?!



변수 variable

변수 명명규칙

1. 변수의 이름은 영문자(대소문자), 숫자, 언더스코어(_)로만 구성
2. 변수의 이름은 숫자로 시작될 수 없음
3. 변수의 이름 사이에는 공백을 포함할 수 없음
4. 변수의 이름으로 C언어에서 미리 정의된 키워드(keyword) 사용할 수 없음

auto, break, case, char, const, continue, default,
do, double, else, enum, extern, float, for,
goto, if, int, long, register, return, short,
signed, sizeof, static, struct, switch, typedef, union
unsigned, void, volatile, while



변수 variable

Camel Case

서로 다른 단어가 합쳐질 때

뒤에 첫 단어의 맨 앞에 스펠링이 대문자로 쓰이는 작성 방법

Example)

digital + write = digitalWrite

analog + read = analogRead

pin + mode = pinMode





상수 Constant

상수

변하지 않는 값

상수를 변수명으로 정한다면 모든 스펠링을 대문자로 작성

Example)

HIGH, LOW, OUTPUT, INPUT 등



반복문 for문

for문

```
for (초기값; 조건식; 스텝) {  
    수행문장  
}
```

초기값: 시작값

조건식: 참/거짓 값이 나와야하며, 참일 때만 반복 진행

스텝: 반복할때 마다 증감할 정도

ex) 변수 num에 0~100의 값을 할당

```
int num = 0;  
for (int val = 0; val < 100; val++) {  
    num = val  
}
```



발광 다이오드

LED, Light-Emitting Diode

- 점점 밝아지게

LED 세기가 0에서 1씩 커져서 밝아지게 해주세요. (시간 간격: 0.02초)



발광 다이오드

LED, Light-Emitting Diode

- 점점 밝아지게

LED 세기가 0에서 1씩 커져서 밝아지게 해주세요. (시간 간격: 0.02초)

```
1 int led = 6;
2
3 void setup() {
4     for (int power = 0; power < 255; power++) {
5         analogWrite(led, power);
6         delay(20);
7     }
8 }
9
10 void loop() {
11 }
```



발광 다이오드

LED, Light-Emitting Diode

실습1. 점점 어두워지게

LED 세기가 최대치에서 1씩 줄어들어 꺼지게 해주세요. (시간 간격: 0.02초)



발광 다이오드

LED, Light-Emitting Diode

실습1. 점점 어두워지게

LED 세기가 최대치에서 1씩 줄어들어 꺼지게 해주세요. (시간 간격: 0.02초)

```
1 int led = 6;  
2  
3 void setup() {  
4     for (int power = 255; power >= 0; power--) {  
5         analogWrite(led, power);  
6         delay(20);  
7     }  
8 }  
9  
10 void loop() {  
11 }
```



오버플로우/언더플로우

Overflow/Underflow

analogWrite의 범위가 0~255라고 하였는데 세기값에 범위 외의 값을 넣으면 어떻게 될까?

-10이나 256을 넣고 확인해보자.



오버플로우/언더플로우

Overflow/Underflow

-1을 넣으면 255의 세기와 동일하게 켜지게 될 것이고,
256을 넣으면 0의 세기와 같게 꺼지게 될 것이다.

이러한 현상을 오버플로우/언더플로우 라고 합니다.

오버플로우는 설정값을 넘어선 경우, 언더플로우는 설정값 미만인 경우 발생합니다.

대표적인 예로 싸이 강남스타일의 유튜브 조회수를 볼 수 있습니다.

<https://www.youtube.com/watch?v=oDUdh9ToeJ4>

즉, 범위를 넘어서서 세기값이 이뤄지는 것은 문제가 되는 프로그래밍입니다.

아두이노에서 잡아주기에 오류 없이 실행이 되는 것이지 잡아주지 않는다면 원하지 않는 결과를 얻기 때문입니다.

이와 함께 각 자료형의 타입은 범위를 가지고 있습니다.

구분	자료형	범위	바이트
정수형	char	-128 ~ 127	1(8)
	unsigned char	0 ~ 255	1(8)
	short	-32768 ~ 32767	2(16)
	int	-2,147,483,648 ~ 2,147,483,647	4(32)
	long	-2,147,483,648 ~ 2,147,483,647	4(32)
	unsigned short	0~65535	2(16)
실수형	float	$8.4 \times 10^{-37} \sim 3.4 \times 10^{38}$	4(32)
	double	$2.2 \times 10^{-308} \sim 1.8 \times 10^{308}$	8(64)
나열형	enum	정수를 대신하여 사용하는 별명, int형의 크기	
무치형	void	실제 자료는 없음을 명시적으로 선언	

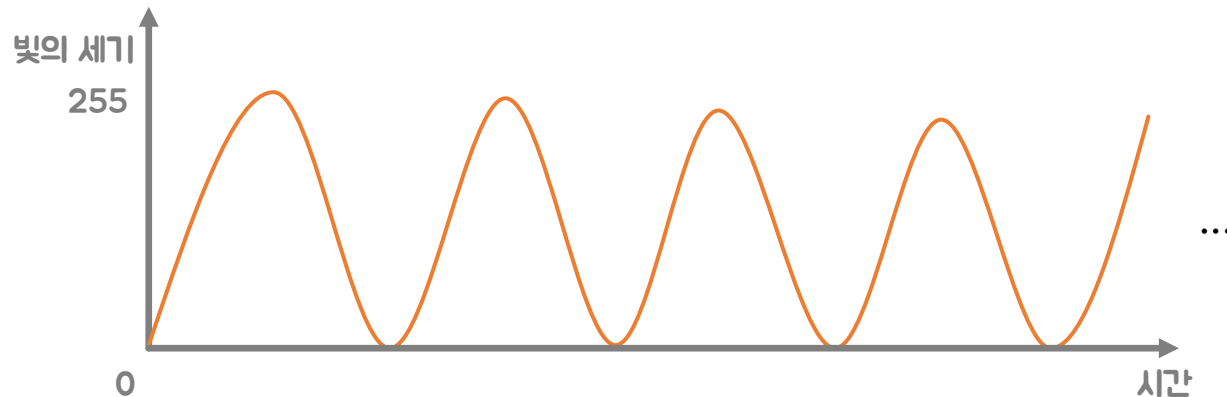


발광 다이오드

LED, Light-Emitting Diode

실습2. 점점 밝아졌다 어두워지게 반복

LED 세기가 1씩 밝아졌다가 최대치 세기가 되면 점점 1씩 어두워지게
0이 되어 꺼지면 다시 점점 밝아지게 이 로직을 반복하여 계속 점점 밝아졌다가
어두워졌다가를 반복하는 프로그램을 작성해주세요. (시간 간격: 0.01초)





발광 다이오드

LED, Light-Emitting Diode

실습2. 점점 밝아졌다 어두워지게 반복

쉬운 방식으로 '점점 밝아지게'와
'점점 어두워지게'를 합쳐서 loop문에 두어
계속 반복하게 작성한 코드입니다.

보다 효율적이지만 어려운 방식으로는
시간 복잡도라는 개념을 고려해 loop문에
반복문을 사용하지 않고 가정문으로
처리하는 방식이 있습니다.

```
1 int led = 6;
2
3 void setup() {
4 }
5
6 void loop() {
7   for (int power = 0; power < 255; power++) {
8     analogWrite(led, power);
9     delay(10);
10  }
11  for (int power = 255; power >= 0; power--) {
12    analogWrite(led, power);
13    delay(10);
14  }
15 }
```



가정문 if문

if문

```
if (조건식) {  
    수행문장;  
}
```

조건식: 참/거짓 값이 나와야하며, 참일 때만 중괄호 안에 수행문장 실행



가정문 if문

If~else문

```
if (조건식) {  
    수행문장A;  
} else {  
    수행문장B;  
}
```

조건이 참일 때는 수행문장A 실행

조건이 거짓일 경우는 else문에 있는 수행문장B 실행

If~else if~else문

```
if (조건식A) {  
    수행문장A;  
} else if(조건식B) {  
    수행문장B;  
} else {  
    수행문장C;  
}
```

조건A가 참일 때는 수행문장A 실행

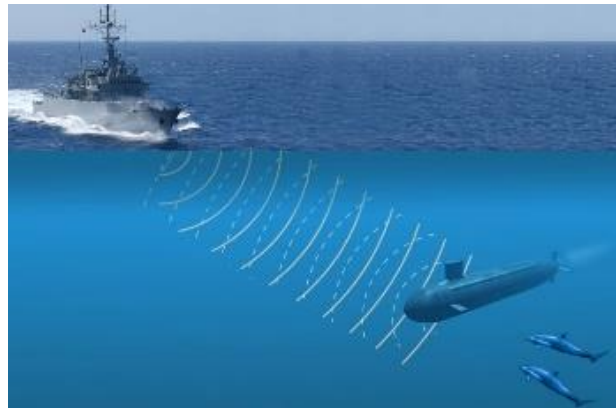
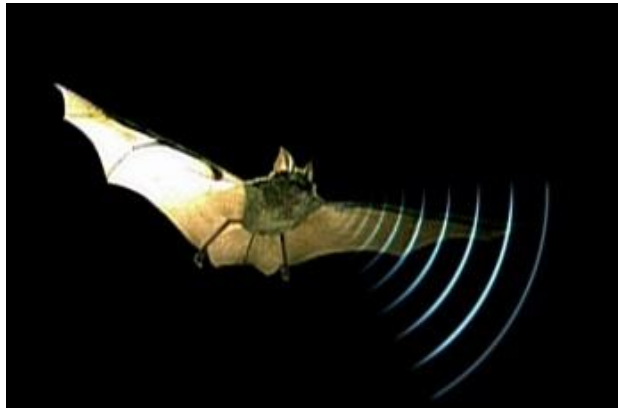
조건A가 거짓일 경우 조건B 참/거짓 판단

조건B가 참이면 수행문장B 실행

조건B도 거짓이면 else문에 있는 수행문장C 실행



초음파 센서 Ultrasonic Sensor



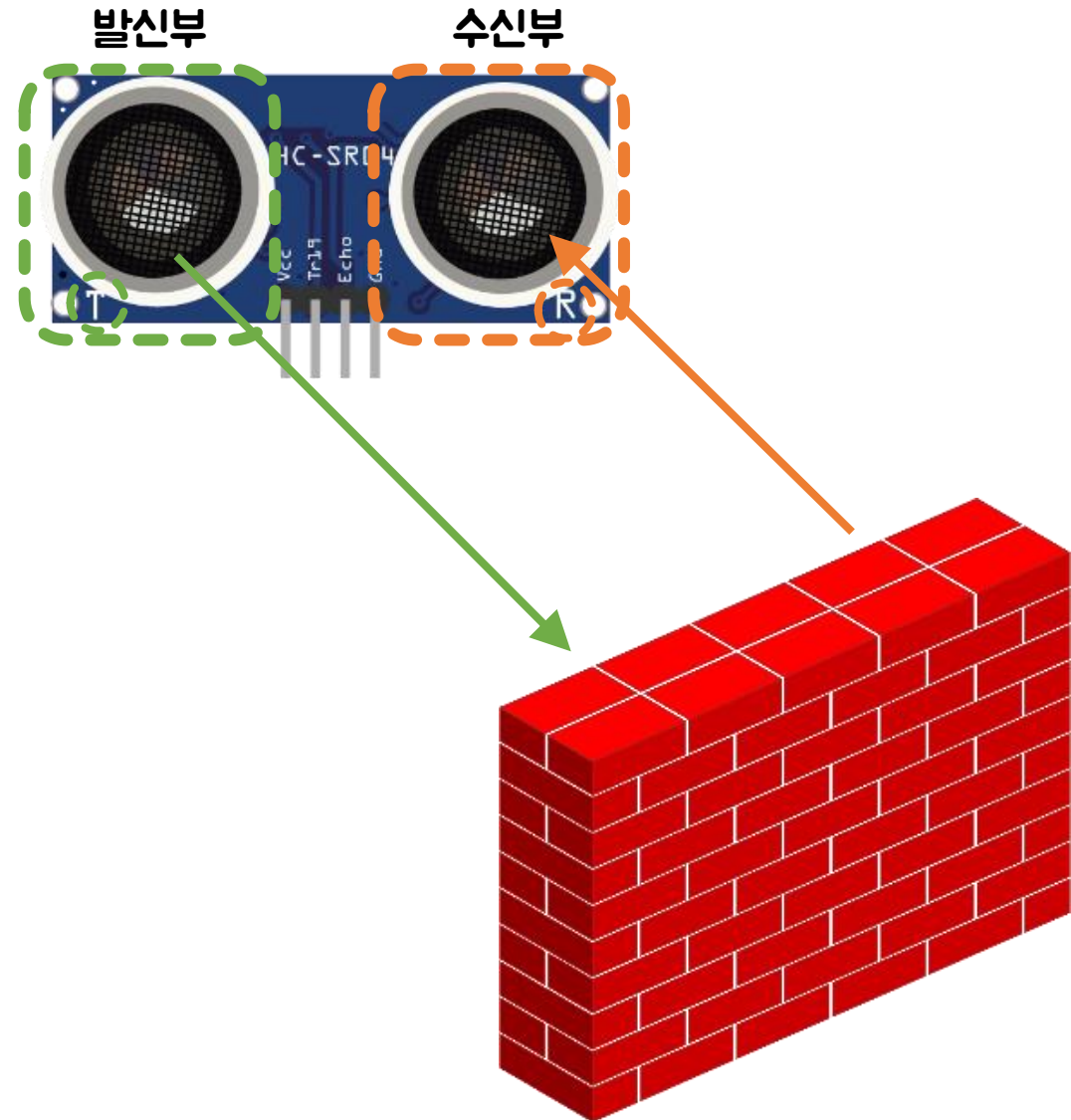
박쥐나 고래와 같은 동물이 초음파를 보내서 거리를 탐지하여 장애물을 피하기도 하고,
이러한 원리를 이용해서 초음파로 잠수함이 수심이 깊은 곳에서 장애물을 탐지하거나
우리의 몸의 건강 검진을 하는데 사용하고 있습니다. 이러한 예시 말고도 더 다양하게 사용되고 있습니다.

우리는 초음파 센서를 통해서 거리를 확인하고
그에 따라서 특정한 작업이 되게 처리되는 일을 해볼 것입니다.



초음파 센서 Ultrasonic Sensor

초음파가 발신부 나간 후에, 장애물에 맞고 반사되어 수신부로 돌아오는 시간을 초음파 속도와 곱해서 거리가 나옵니다. 이러한 특성으로 인하여 센서에서 2cm 정도는 장애물이 떨어져 있어야 거리를 인식할 수 있습니다. 최대 거리는 4m까지 인식할 수 있습니다.





초음파 센서 Ultrasonic Sensor

PIN MAP

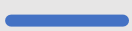
UNO

5V

D12

D13

GND



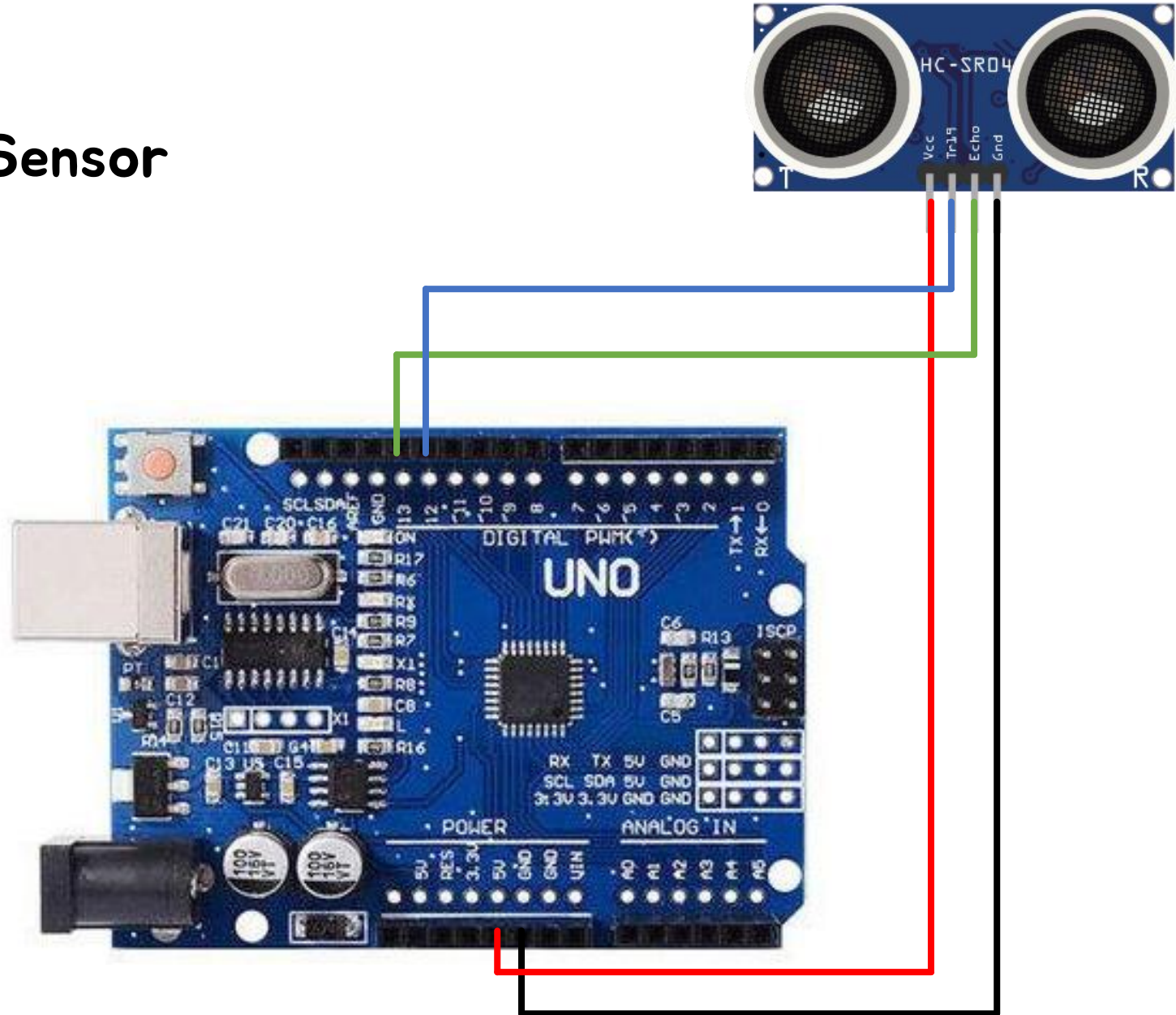
초음파
센서

VCC

TRIG

ECHO

GND





초음파 센서

Ultrasonic Sensor

```
float duration = pulseIn(PIN_ECHO, HIGH);  
float distance = duration / 29.0 / 2;
```

$1 / 29.0 = 0.0344827...$

초음파 센서의 상온 속도 (단위=cm)

pulseIn()

[Advanced I/O]

설명

핀에서 펄스(HIGH 또는 LOW)를 읽습니다. 예를 들어, value 가 HIGH 이면, pulseIn() 은 핀이 HIGH 가 될 때까지 기다리고, 타이머를 시작하고, 핀이 LOW 가 될 때까지 기다리고 타이머를 멈춥니다. 펄스의 길이를 마이크로초 단위로 반환합니다. 정해진 timeout 안에 펄스가 시작되지 않으면 0을 반환합니다. 이 함수의 타이머는 경험적으로 결정되고, 긴 펄스에 대해 오류를 낼 수도 있습니다. 10 마이크로초부터 3분까지의 길이의 펄스에 대해 동작합니다.



시리얼 통신

Serial Communication

시리얼 통신을 통해 컴퓨터와 아두이노 보드간
데이터를 송수신 할 수 있습니다.
해당 통신을 위해서는 기본적인 설정이 필요합니다.
해당 소스는 다음과 같습니다.

1byte 데이터를 주고 받는 직렬 통신입니다.

1byte = 8bit

초음파 센서 거리값 확인 CODE

```
1 int trig = 12;
2 int echo = 13;
3
4 void setup() {
5     Serial.begin(9600);
6     pinMode(trig, OUTPUT);
7     pinMode(echo, INPUT);
8 }
9
10 void loop() {
11     digitalWrite(trig, LOW);
12     delayMicroseconds(2);
13     digitalWrite(trig, HIGH);
14     delayMicroseconds(10);
15     digitalWrite(trig, LOW);
16     float duration = pulseIn(echo, HIGH);
17     float distance = duration / 29 / 2;
18
19     Serial.print("거리: ");
20     Serial.print(distance);
21     Serial.println(" cm");
22 }
```

시리얼 통신 속도 지정
보드레이트 → 1초에 9600bit 송수신

print → 출력
println → 출력후 개행(엔터)
('\n' 은 소문자 'L')



시리얼 모니터 Serial Monitor

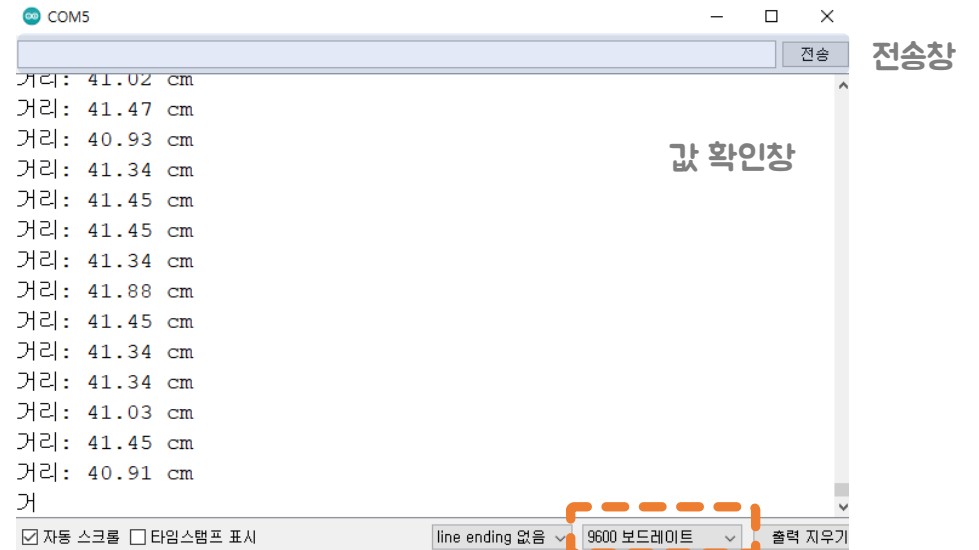
```
ultrasonic_serial | 아두이노 1.8.19
파일 편집 스케치 툴 도움말
ultrasonic_serial
1 int trig = 12;
2 int echo = 13;
3
4 void setup() {
5   Serial.begin(9600);
6   pinMode(trig, OUTPUT);
7   pinMode(echo, INPUT);
8 }
9
10 void loop() {
11   digitalWrite(trig, LOW);
12   delayMicroseconds(2);
13   digitalWrite(trig, HIGH);
14   delayMicroseconds(10);
15   digitalWrite(trig, LOW);
16   float duration = pulseIn(echo, HIGH);
17   float distance = duration / 29 / 2;
18
19   Serial.print("거리: ");
20   Serial.print(distance);
21   Serial.println(" cm");
22 }
```

업로드 완료

스케치는 프로그램 저장 공간 3788 바이트(11%)를 사용. 최대 32256 바이트 사용 가능.
전역 변수는 동적 메모리 212바이트(10%)를 사용, 1836바이트의 지역변.

시리얼 모니터

시리얼 모니터 창



보드레이트 설정

Serial.begin(값)과 동일해야 합니다.



시리얼 통신

Serial Communication

- 시리얼 통신을 통해 LED 제어

시리얼 통신을 통해서 센서, 모듈들을 제어할 수도 있습니다.

시리얼 모니터에 0이라는 문자를 작성하면 LED가 꺼지고,

1이라는 문자를 작성하면 LED가 켜지게 해보겠습니다.



시리얼 통신

Serial Communication

- 시리얼 통신을 통해 LED 제어

시리얼 통신을 통해서 센서, 모듈들을 제어할 수도 있습니다.
시리얼 모니터에 0이라는 문자를 작성하면 LED가 꺼지고,
1이라는 문자를 작성하면 LED가 켜지게 해보겠습니다.

자동 형변환

: 기본 타입에서 자신보다 큰 타입이면 컴파일러가

자동으로 큰 타입으로 맞춰주는 것을 자동 형변환이라고 합니다.

현재 Serial.read()의 타입은 byte 형태인데 char형태로

자동 형변환 되어 문자형태로 받아드릴 수 있습니다.

```
1 int led = 10;
2
3 void setup() {
4     Serial.begin(9600);
5     pinMode(led, OUTPUT);
6 }
7
8 void loop() {
9     if (Serial.available()) {
10        char cmd = Serial.read();
11        if (cmd == '0') digitalWrite(led, LOW);
12        else if (cmd == '1') digitalWrite(led, HIGH);
13    }
14 }
```



초음파 센서 Ultrasonic Sensor

- 거리 확인하기

시리얼 모니터를 통해서 거리를 확인해보겠습니다.

전체 소스

```
1 int led = 6;
2 int trig = 12;
3 int echo = 13;
4
5 void setup() {
6   Serial.begin(9600);
7   pinMode(led, OUTPUT);
8   pinMode(trig, OUTPUT);
9   pinMode(echo, INPUT);
10 }
11
```

```
12 void loop() {
13   digitalWrite(trig, LOW);
14   delayMicroseconds(2);
15   digitalWrite(trig, HIGH);
16   delayMicroseconds(10);
17   digitalWrite(trig, LOW);
18   float duration = pulseIn(echo, HIGH);
19   float distance = duration / 29 / 2;
20   Serial.print(distance);
21   Serial.println(" cm");
22 }
```




초음파 센서 Ultrasonic Sensor

- 거리 확인하기

시리얼 모니터를 통해서 거리를 확인해보겠습니다.

```
1 int led = 6;   핀 번호
2 int trig = 12;
3 int echo = 13;
4
5 void setup() {
6   Serial.begin(9600);
7   pinMode(led, OUTPUT);
8   pinMode(trig, OUTPUT);
9   pinMode(echo, INPUT);
10 }
11
```

→ 시리얼 통신 속도 1초 9600bit 설정

핀 입/출력 설정



초음파 센서 Ultrasonic Sensor

- 거리 확인하기

시리얼 모니터를 통해서 거리를 확인해보겠습니다.

```
12 void loop() {  
13   digitalWrite(trig, LOW);  
14   delayMicroseconds(2);  
15   digitalWrite(trig, HIGH);  
16   delayMicroseconds(10);  
17   digitalWrite(trig, LOW);  
18   float duration = pulseIn(echo, HIGH);  
19   float distance = duration / 29 / 2;  
20   Serial.print(distance);  
21   Serial.println(" cm");  
22 }
```

초음파 발신

시리얼 모니터 내용 작성

(초음파 수신)

해당 핀이 LOW → HIGH로 변하는 순간부터
다시 LOW로 바뀌는 시간을 us(마이크로 초)로 반환

상온에서 초음파의 속도는 대략 340m/s 입니다.

cm 단위로 환산하면 34,000cm/s

us(마이크로 초)로 환산하면 34,000 / 1000 / 1000

→ 0.034cm/us → 대략 1/29

거리 = 속도 x 시간 이고, 해당 시간은 왕복의 시간이기에

거리 = 시간 / 29 / 2



초음파 센서 Ultrasonic Sensor

실습1. 스마트 감지 조명

20cm 근처에 물체/사람이 감지되면 3초간 안전을 위해 조명을 켜주는
프로그램을 만들어주세요.

(20cm 이내에 감지가 계속 되면 지속적으로 LED가 ON)



초음파 센서 Ultrasonic Sensor

실습1. 스마트 감지 조명

20cm 근처에 물체/사람이 감지되면 3초간 안전을 위해 조명을 켜주는 프로그램을 만들어주세요.

(20cm 이내에 감지가 계속 되면 지속적으로 LED가 ON)

```
1 int led = 6;
2 int trig = 12;
3 int echo = 13;
4
5 float getDistance() {
6     digitalWrite(trig, LOW);
7     delayMicroseconds(2);
8     digitalWrite(trig, HIGH);
9     delayMicroseconds(10);
10    digitalWrite(trig, LOW);
11    float duration = pulseIn(echo, HIGH);
12    return duration / 29 / 2;
13 }
14
15 void setup() {
16     pinMode(led, OUTPUT);
17     pinMode(trig, OUTPUT);
18     pinMode(echo, INPUT);
19 }
20
21 void loop() {
22     float distance = getDistance();
23
24     if (distance < 20) {
25         digitalWrite(led, HIGH);
26         delay(3000);
27     } else {
28         digitalWrite(led, LOW);
29     }
30 }
```

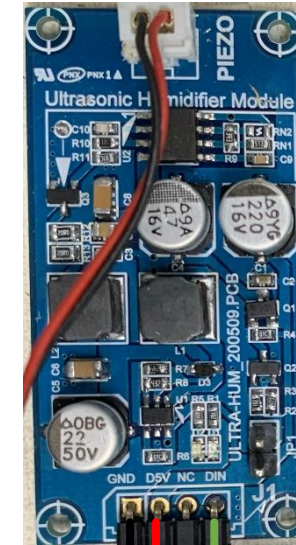


가습기 센서 Humidifier Sensor

PIN MAP

UNO		가습기 센서
GND	—	GND
5V	—	D5V
		NC
D5	—	DIN

사용방법은 LED를 다루던 것과 동일합니다.





NC

No Connection

NC는 No Connection의 약자로 아무것도 연결할 필요가 없는 핀입니다.



가습기 센서 Humidifier Sensor

최종. 대포 가습기

현재까지 다뤄본 센서들을 통하여 대포 가습기를 만들어 보세요.

(권장 → 20cm 이내: LED&가습기 모듈 ON / 20cm 외: LED & 가습기 모듈 OFF)



가습기 센서

Humidifier Sensor

최종. 대포 가습기

전체 소스

```
1 int led = 6;
2 int trig = 12;
3 int echo = 13;
4 int humidify = 5;
5
6 void setup() {
7   Serial.begin(9600);
8   pinMode(led, OUTPUT);
9   pinMode(trig, OUTPUT);
10  pinMode(echo, INPUT);
11  pinMode(humidify, OUTPUT);
12 }
```

```
13
14 void loop() {
15   digitalWrite(trig, LOW);
16   delayMicroseconds(2);
17   digitalWrite(trig, HIGH);
18   delayMicroseconds(10);
19   digitalWrite(trig, LOW);
20   float duration = pulseIn(echo, HIGH);
21   float distance = duration / 29 / 2;
22   Serial.println(distance);
23
24   if (distance < 20) {
25     digitalWrite(led, HIGH);
26     digitalWrite(humidify, HIGH);
27   } else {
28     digitalWrite(led, LOW);
29     digitalWrite(humidify, LOW);
30   }
31 }
```




가습기 센서

Humidifier Sensor

최종. 대포 가습기

```
1 int led = 6;
2 int trig = 12;
3 int echo = 13;
4 int humidify = 5;
5
6 void setup() {
7     Serial.begin(9600);
8     pinMode(led, OUTPUT);
9     pinMode(trig, OUTPUT);
10    pinMode(echo, INPUT);
11    pinMode(humidify, OUTPUT);
12 }
```



가습기 센서 Humidifier Sensor

최종. 대포 가습기

거리가 20cm 보다 작으면 LED와 가습기 모듈 ON
이 외에는(20cm와 같거나 클 경우) LED와 가습기 모듈 OFF

```
13
14 void loop() {
15     digitalWrite(trig, LOW);
16     delayMicroseconds(2);
17     digitalWrite(trig, HIGH);
18     delayMicroseconds(10);
19     digitalWrite(trig, LOW);
20     float duration = pulseIn(echo, HIGH);
21     float distance = duration / 29 / 2;
22     Serial.println(distance);
23
24     if (distance < 20) {
25         digitalWrite(led, HIGH);
26         digitalWrite(humidify, HIGH);
27     } else {
28         digitalWrite(led, LOW);
29         digitalWrite(humidify, LOW);
30     }
31 }
```