

ARDUINO

아두이노란?



- 이탈리아(<http://www.arduino.cc>)에서 처음으로 개발되었고 AVR 기반의 마이크로 컨트롤러 하드웨어와 소프트웨어 개발을 쉽게 해주는 개발환경(IDE)을 합쳐서 아두이노라고 한다.
- 아두이노 소프트웨어
 - 오픈소스, cross-platform(리눅스, Windows, MAC OS) 지원
- 아두이노 활용
 - 모터제어를 통한 기초 로봇
 - 음악, 사운드 장치
 - 온도, 습도, 화염, 기울기 등 다양한 센서 인터페이스
 - 게임 분야
 - 유서, 무선 네트워크 시스템



IoT 디바이스



IoT에 사용되는 디바이스

Arduino



Raspberry Pi 2



MinnowBoard Max

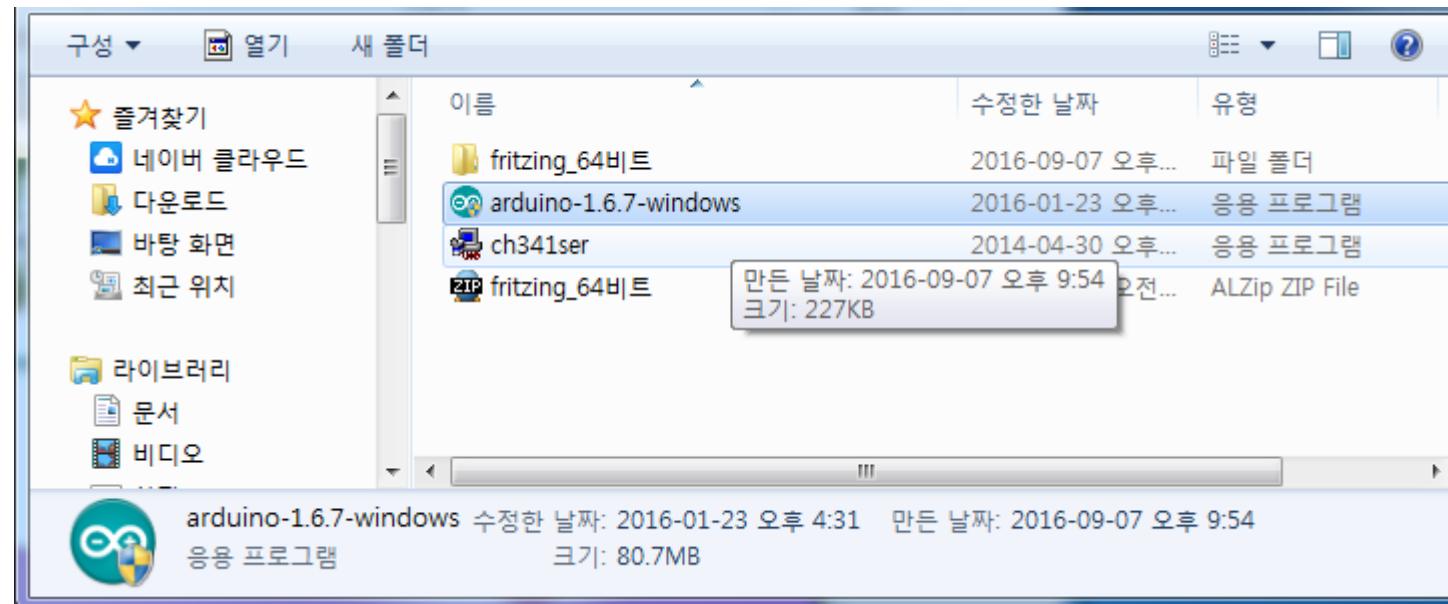


Galileo



Arduino 설치

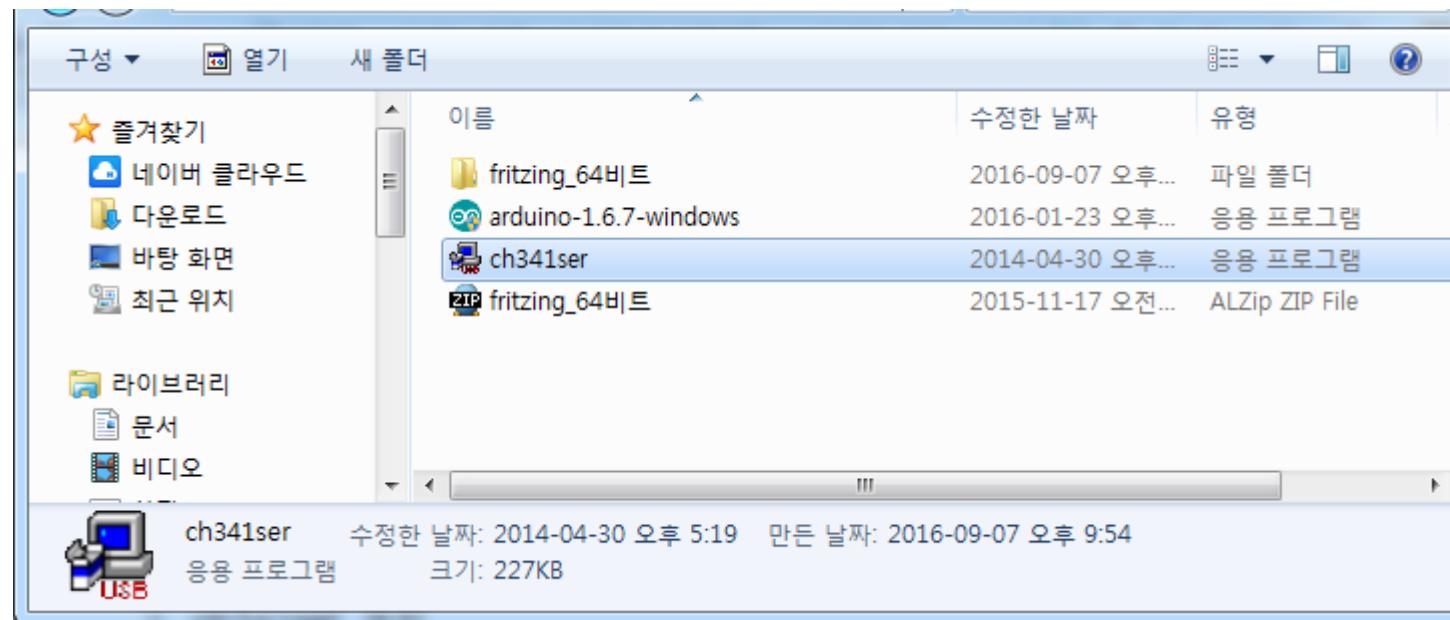
1. arduino-1.6.7-windows 설치



CH340 USB driver



ch341ser 설치

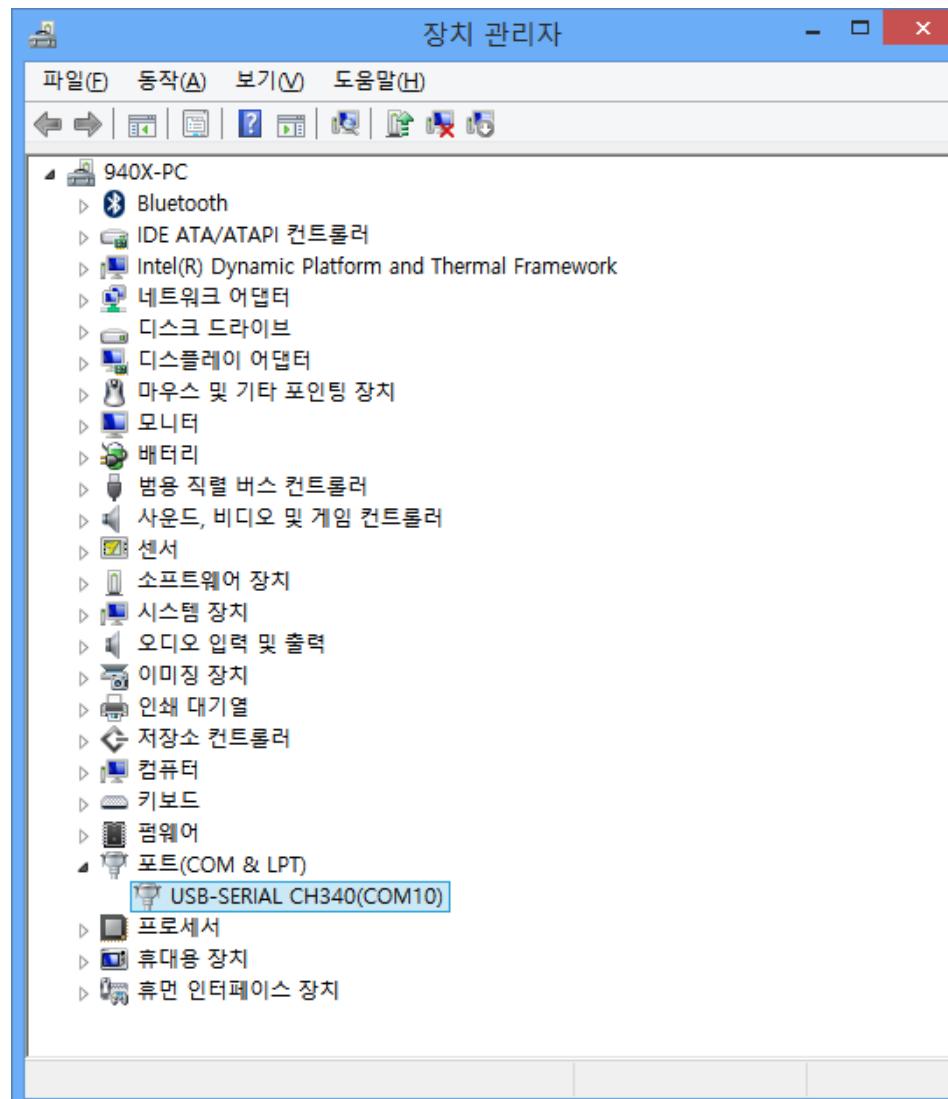


장치 연결

1. PC의 USB 포트와 UNO를 케이블로 연결
2. UNO 전원 공급 확인



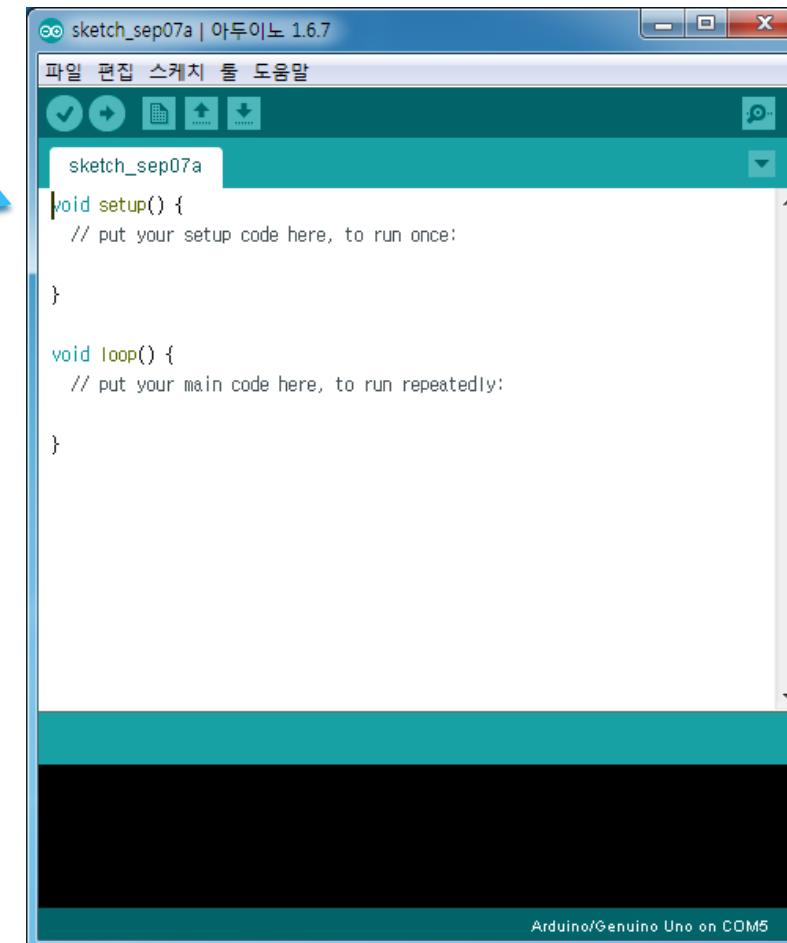
장치 연결 확인



실행



1. Arduino 프로그램 실행



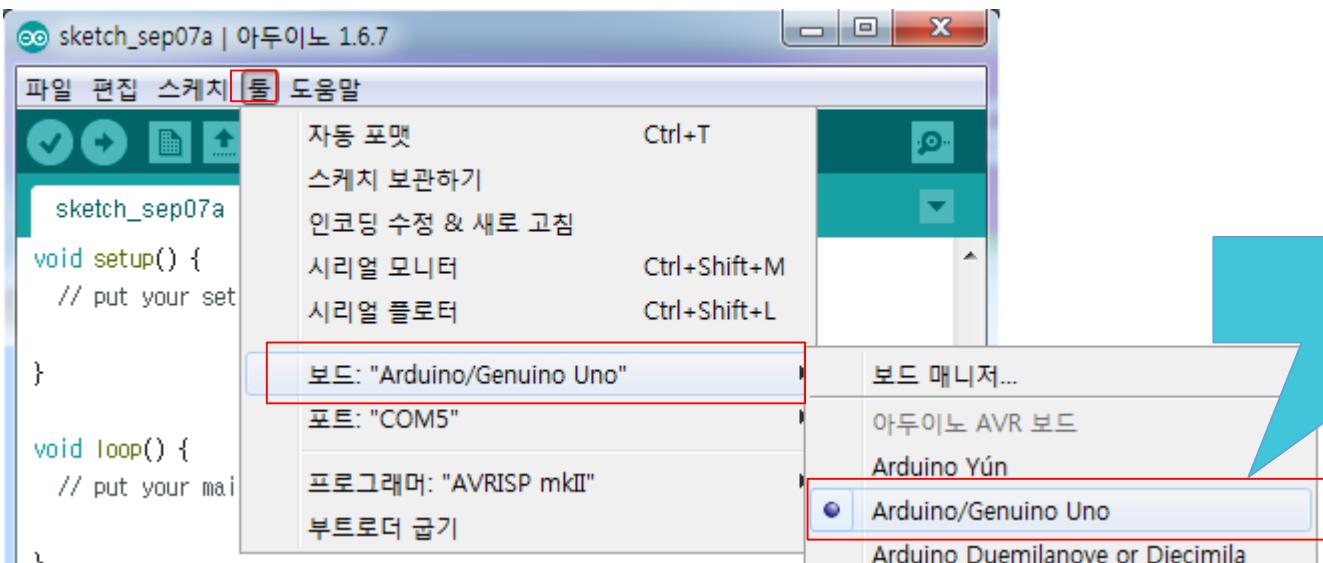
The image shows a screenshot of the Arduino IDE. The window title is "sketch_sep07a | 아두이노 1.6.7". The menu bar includes "파일", "편집", "스케치", "툴", and "도움말". The code editor contains the following code:

```
sketch_sep07a
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

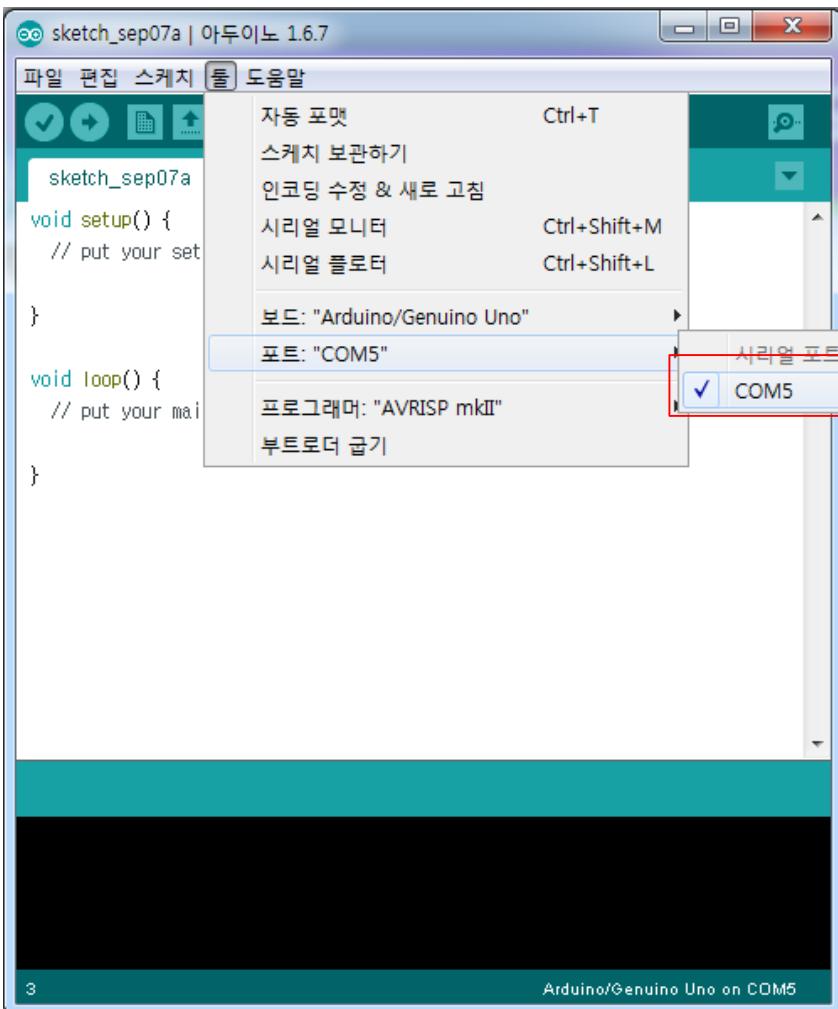
At the bottom of the IDE, a status bar displays "Arduino/Genuino Uno on COM5".

보드 세팅

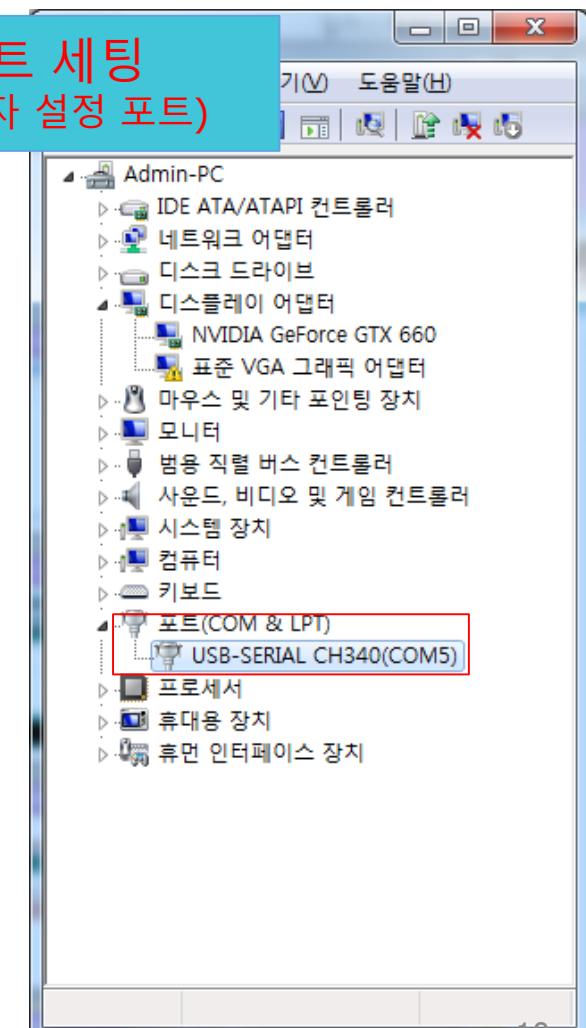


2. 보드 세팅

포트 세팅



3. 포트 세팅 (장치관리자 설정 포트)



테스트



sketch_sep07a | 아두이노 1.6.7

파일 편집 스케치 툴 도움말

sketch_sep07a\$

```
void setup() {
  Serial.begin(9600);
}

void loop() {
  Serial.println("Hello Arduino ~");
  delay(1000);
}
```

4. 코딩

4

Arduino/Genuino Uno on COM5

테스트



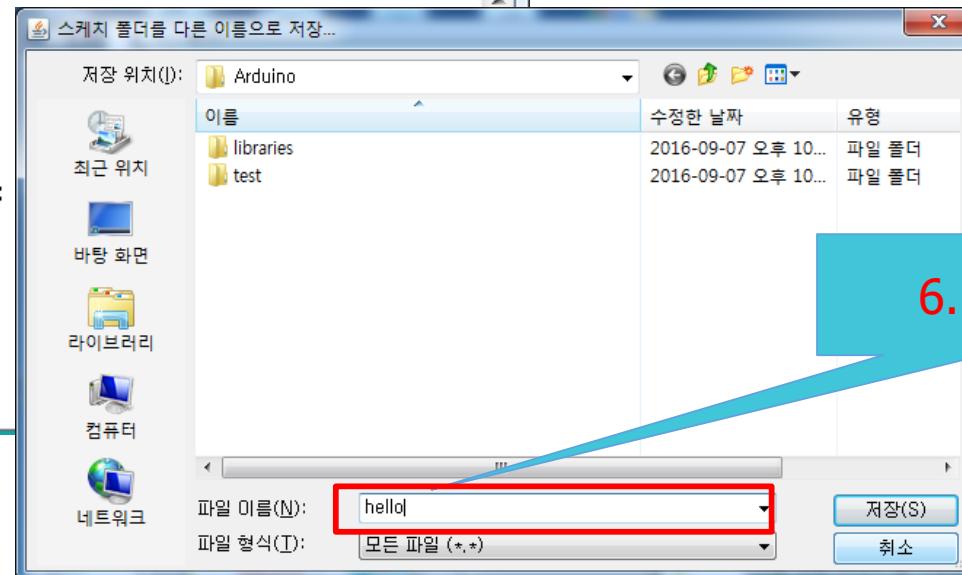
sketch_sep07a | 아두이노 1.6.7

파일 편집 스케치 둘 도움말

저장

```
sketch_sep07a$  
void setup() {  
Serial.begin(9600);  
}  
  
void loop() {  
Serial.println("Hello Arduino ~");  
delay(1000);  
}
```

5. 저장



6. "hello"

테스트

7. 확인

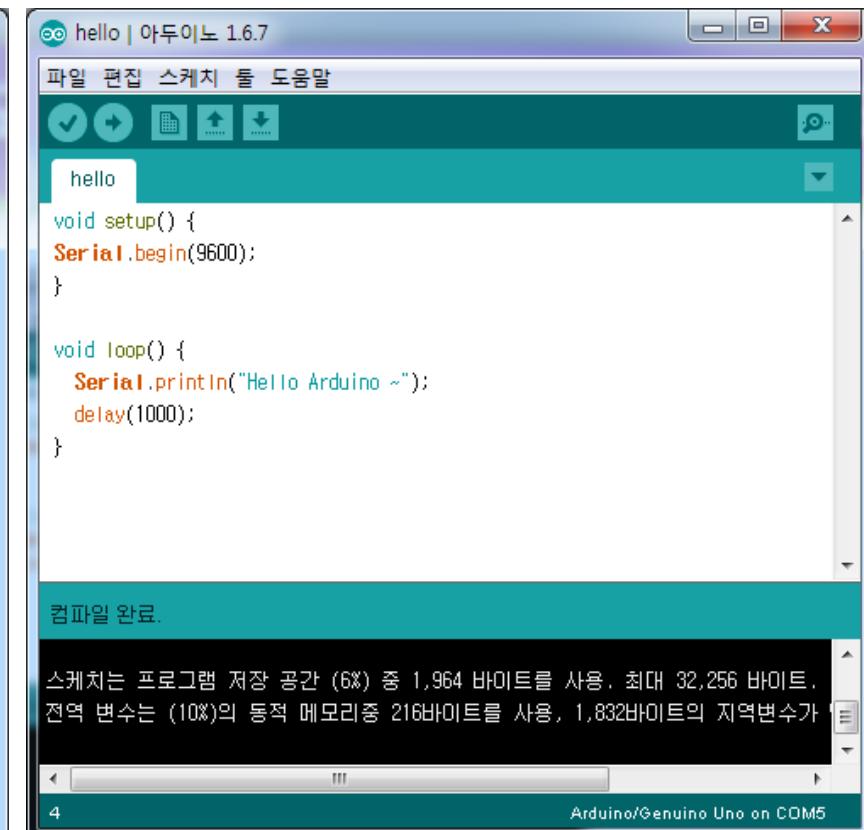


```
hello | 아두이노 1.6.7
파일 편집 스케치 툴 도움말
hello
void setup() {
Serial.begin(9600);
}

void loop() {
Serial.println("Hello Arduino ~");
delay(1000);
}

저장 완료.

4 Arduino/Genuino Uno on COM5
```



```
hello | 아두이노 1.6.7
파일 편집 스케치 툴 도움말
hello
void setup() {
Serial.begin(9600);
}

void loop() {
Serial.println("Hello Arduino ~");
delay(1000);
}

컴파일 완료.

스케치는 프로그램 저장 공간 (6%) 중 1,964 바이트를 사용. 최대 32,256 바이트.
전역 변수는 (10%)의 동적 메모리 중 216바이트를 사용, 1,832바이트의 지역변수가
4 Arduino/Genuino Uno on COM5
```

테스트



8. 업로드



hello | 아두이노 1.6.7

파일 편집 스케치 둘 도움말

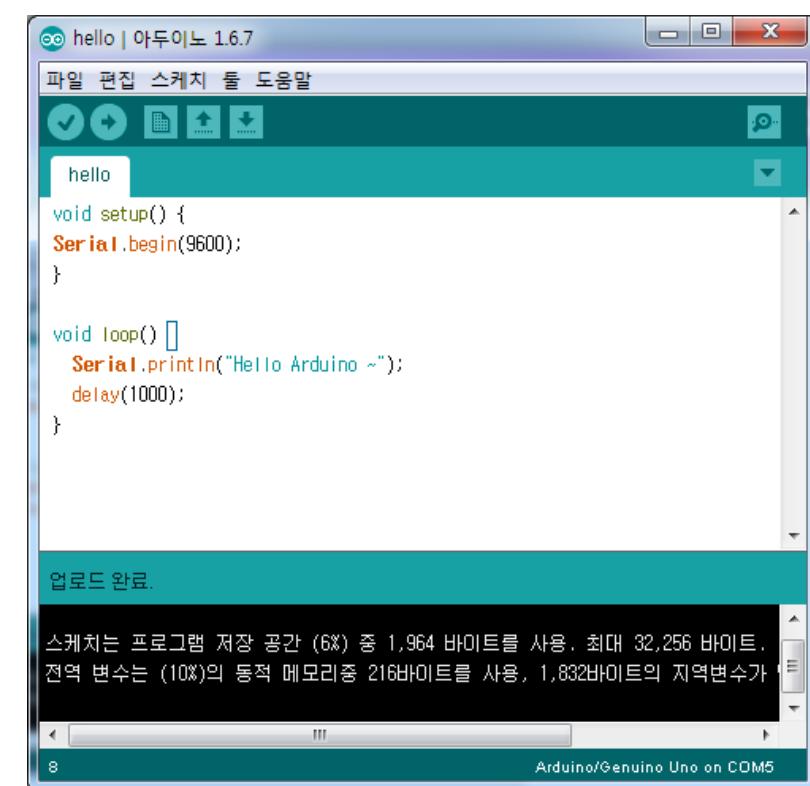
hello

```
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  Serial.println("Hello Arduino ~");  
  delay(1000);  
}
```

컴파일 완료.

스케치는 프로그램 저장 공간 (6%) 중 1,964 바이트를 사용, 최대 32,256 바이트.
전역 변수는 (10%)의 동적 메모리중 216바이트를 사용, 1,832바이트의 지역변수가

4 Arduino/Genuino Uno on COM5



hello | 아두이노 1.6.7

파일 편집 스케치 둘 도움말

hello

```
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  Serial.println("Hello Arduino ~");  
  delay(1000);  
}
```

업로드 완료.

스케치는 프로그램 저장 공간 (6%) 중 1,964 바이트를 사용, 최대 32,256 바이트.
전역 변수는 (10%)의 동적 메모리중 216바이트를 사용, 1,832바이트의 지역변수가

8 Arduino/Genuino Uno on COM5

테스트



```
hello | 아두이노 1.6.7
파일 편집 스케치 둘 도움말
hello
void setup() {
Serial.begin(9600);
}

void loop() {
Serial.println("Hello Arduino ~");
delay(1000);
}

업로드 완료.
스케치는 프로그램 저장 공간 (6%) 중 1,964 바이트를 사용. 최대 32,256 바이트.
전역 변수는 (10%)의 동적 메모리중 216바이트를 사용, 1,832바이트의 지역변수가
8
Arduino/Genuine Uno on COM5
```

1초 간격으로 출력

자동 스크롤

line ending 없음

9600 보드레이트

“9600 보드레이트”

아두이노 스케치 구조



```
hello  
  
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  Serial.println("Hello Arduino ~");  
  delay(1000);  
}
```

시리얼 포트의 통신 속도를 9600bps로 설정

문자열을 시리얼 포트로 출력

1000 밀리 초 대기



아두이노 스케치 구조

○ 아두이노 스케치 구조

```
void setup() {
```

하드웨어 초기화 함수

(코드 실행을 시작할 때 한번만 수행)

```
}
```

```
void loop() {
```

하드웨어 구동 함수

(하드웨어를 동작시키는 루틴으로 반복해서
수행)

```
}
```



구조 테스트



```
hello $  
  
void setup() {  
  Serial.begin(9600);  
  
  Serial.println("setup");  
}  
  
void loop() {  
}
```

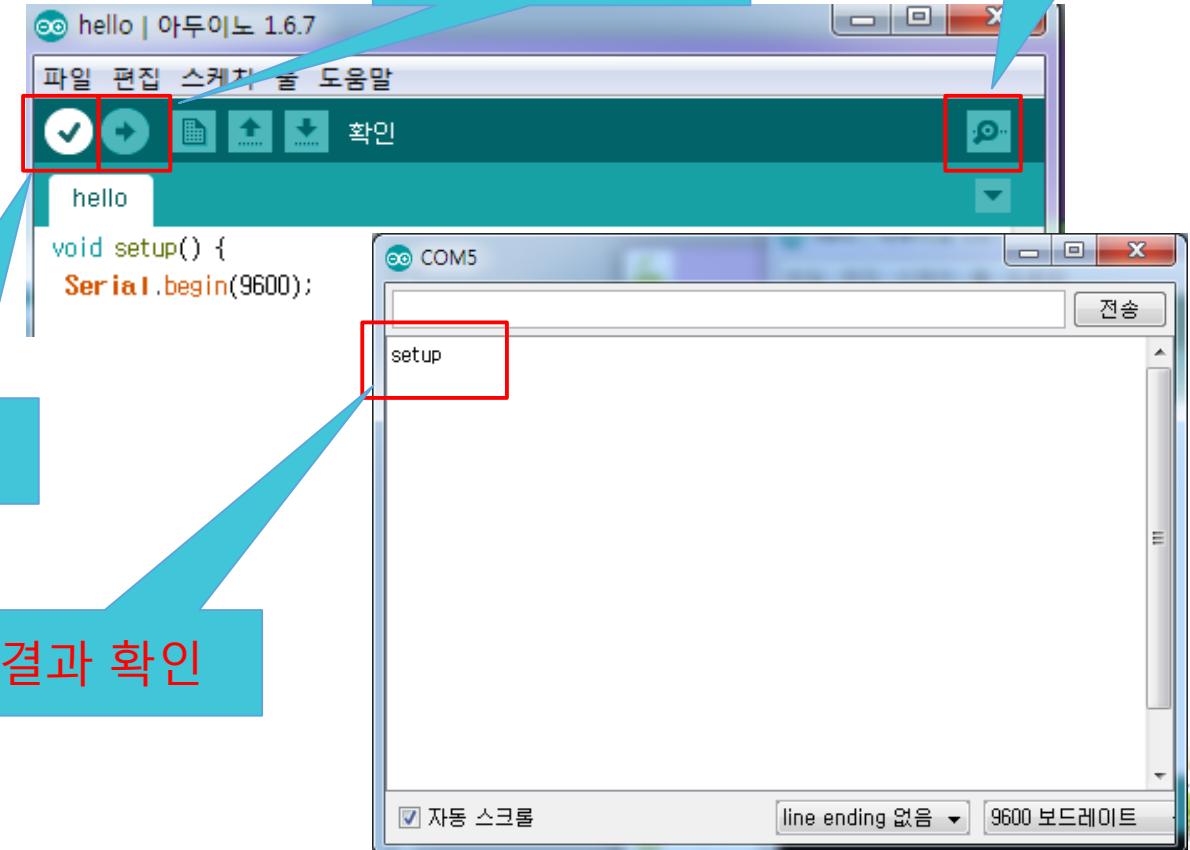
1. 코딩

4. 시리얼 모니터

3. 업로드

2. 확인

5. 결과 확인



setup() - 한번만 실행

구조 테스트

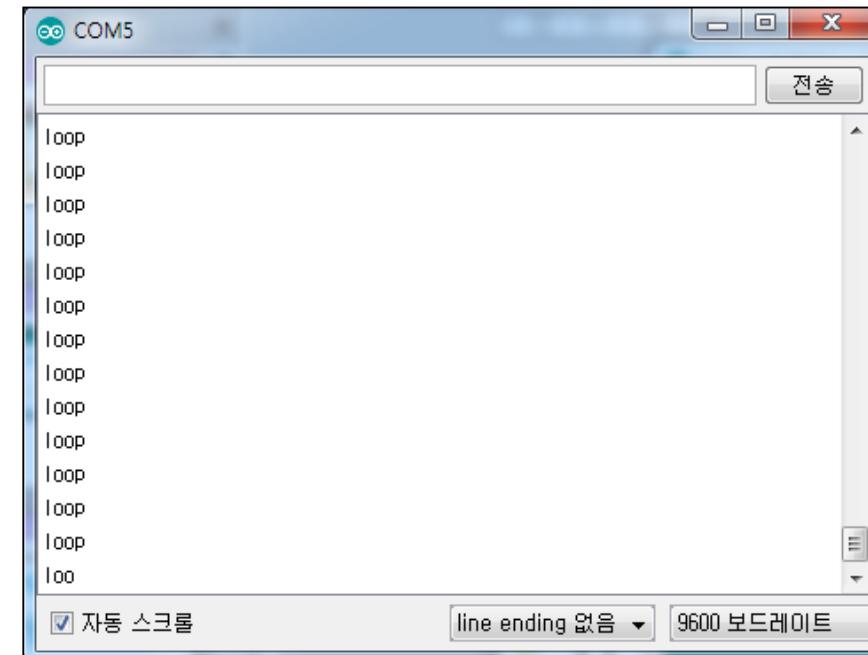


1. 코딩

```
hello  
void setup() {  
    Serial.begin(9600);  
  
    Serial.println("setup");  
}  
  
void loop() {  
    Serial.println("loop");  
}
```



2. 실행



예제

1. 코딩

```
hello

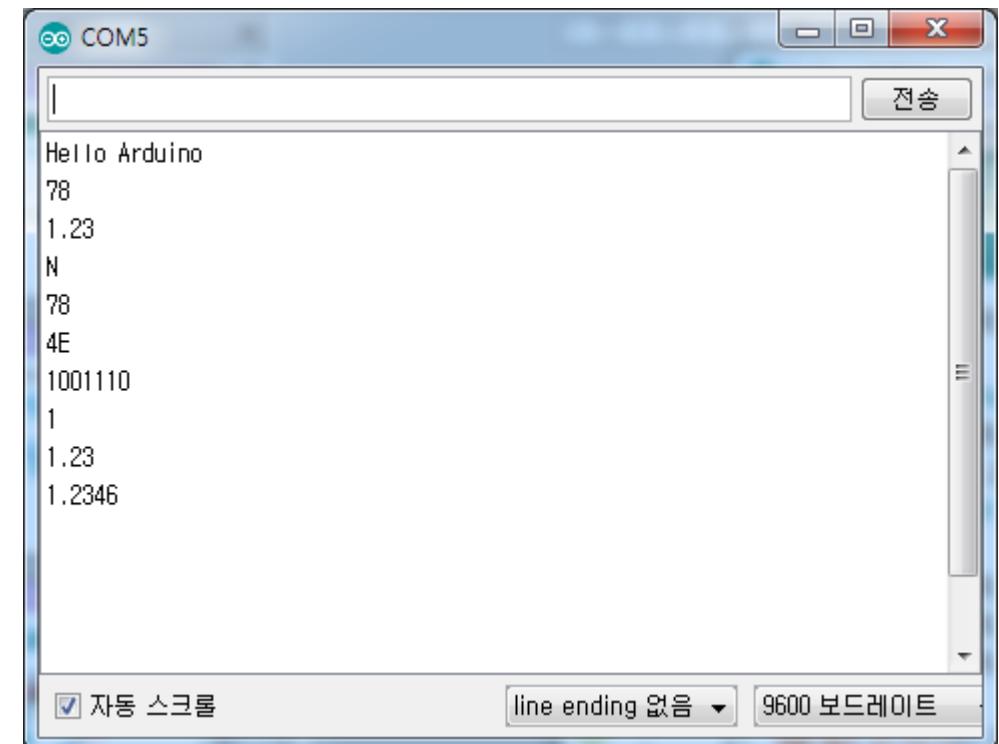
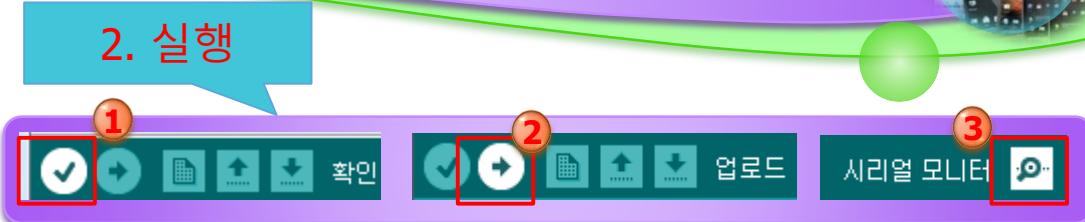
void setup() {
    Serial.begin(9600);

    Serial.println("Hello Arduino");
    Serial.println(78);
    Serial.println(1.23456);
    Serial.println('N');

    Serial.println(78, DEC);
    Serial.println(78, HEX);
    Serial.println(78, BIN);

    Serial.println(1.23456, 0);
    Serial.println(1.23456, 2);
    Serial.println(1.23456, 4);
}

void loop() {
```



```
hello  
void setup() {  
    Serial.begin(9600);  
  
    Serial.println("Hello Arduino");  
    Serial.println(78);  
    Serial.println(1.23456);  
    Serial.println('N');  
  
    Serial.println(78, DEC);  
    Serial.println(78, HEX);  
    Serial.println(78, BIN);  
  
    Serial.println(1.23456, 0);  
    Serial.println(1.23456, 2);  
    Serial.println(1.23456, 4);  
}  
  
void loop() {  
}
```

“1.23” 출력

단일 문자 'N' 출력

10진수 출력

16진수 출력

2진수 출력

소수점 이하 출력 안함

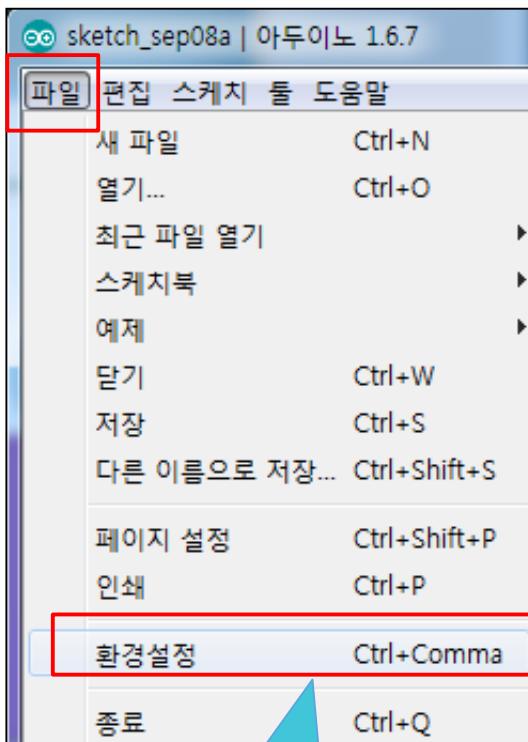
소수점 이하 두 자리 출력

소수점 이하 네 자리 출력

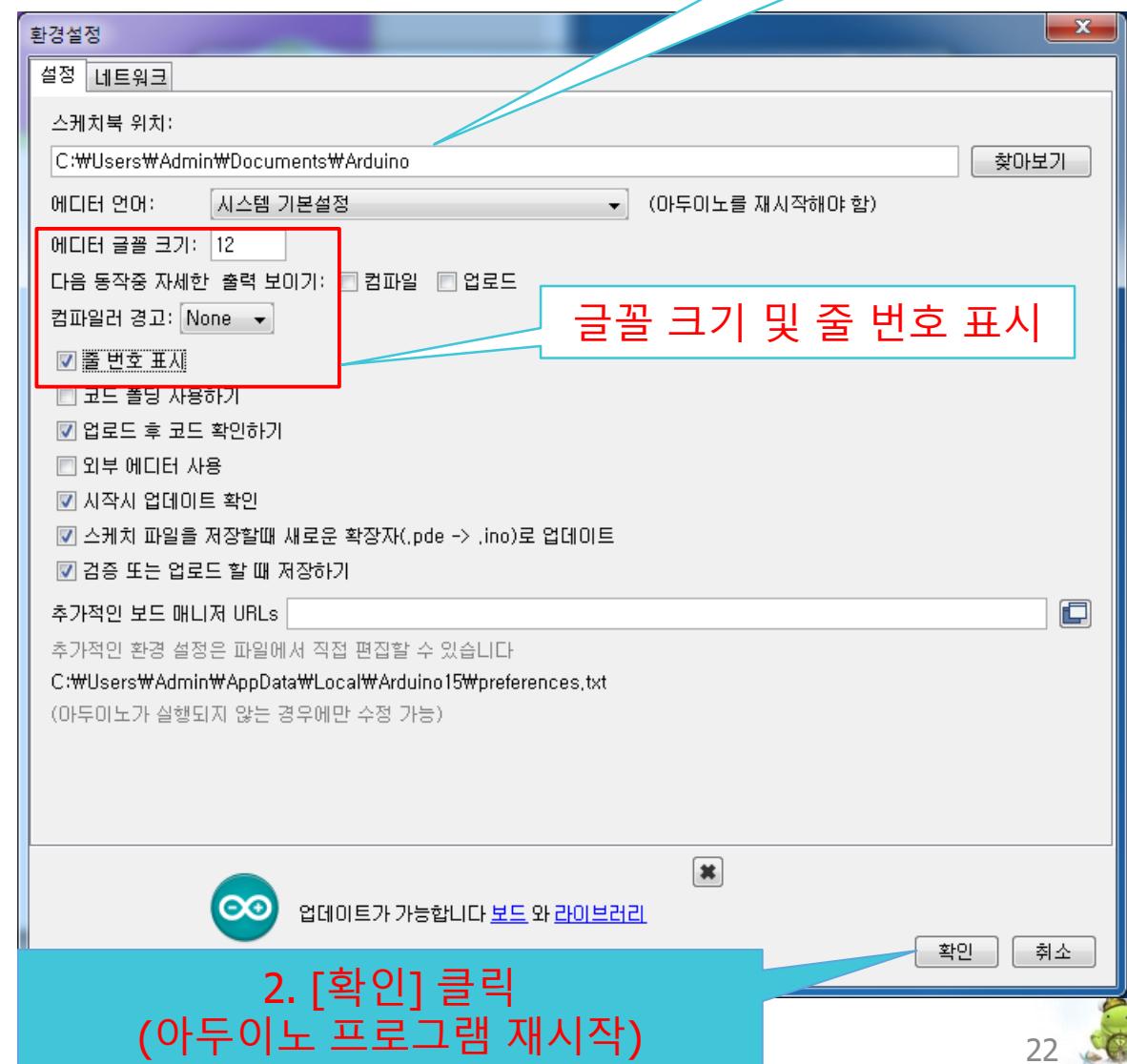
환경 설정



설치 위치

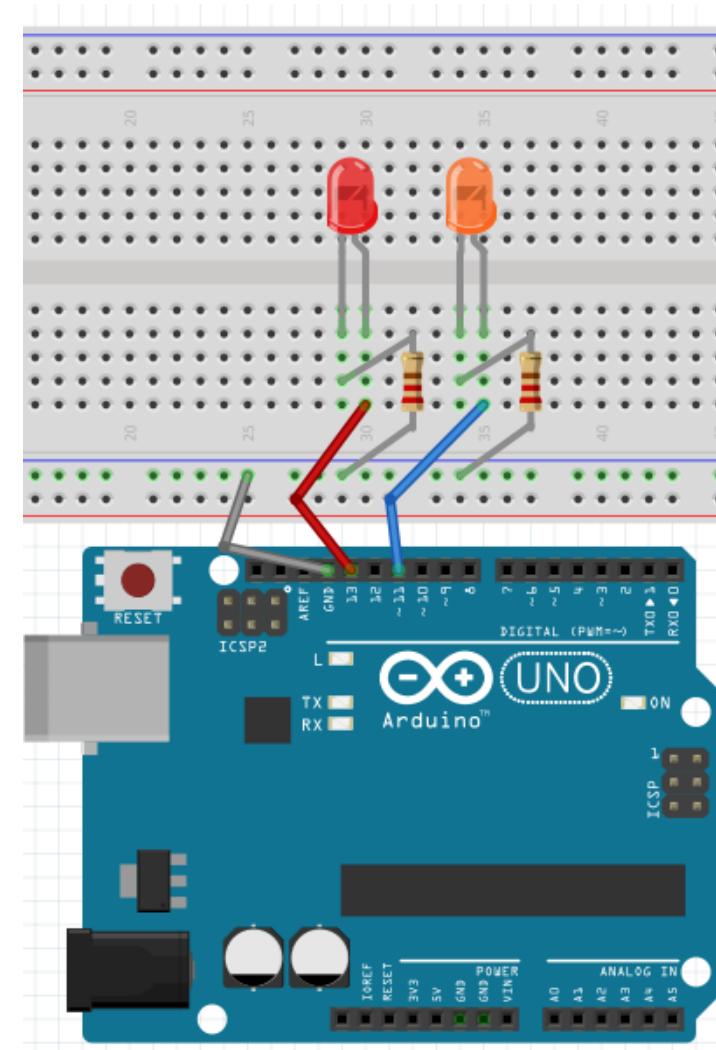


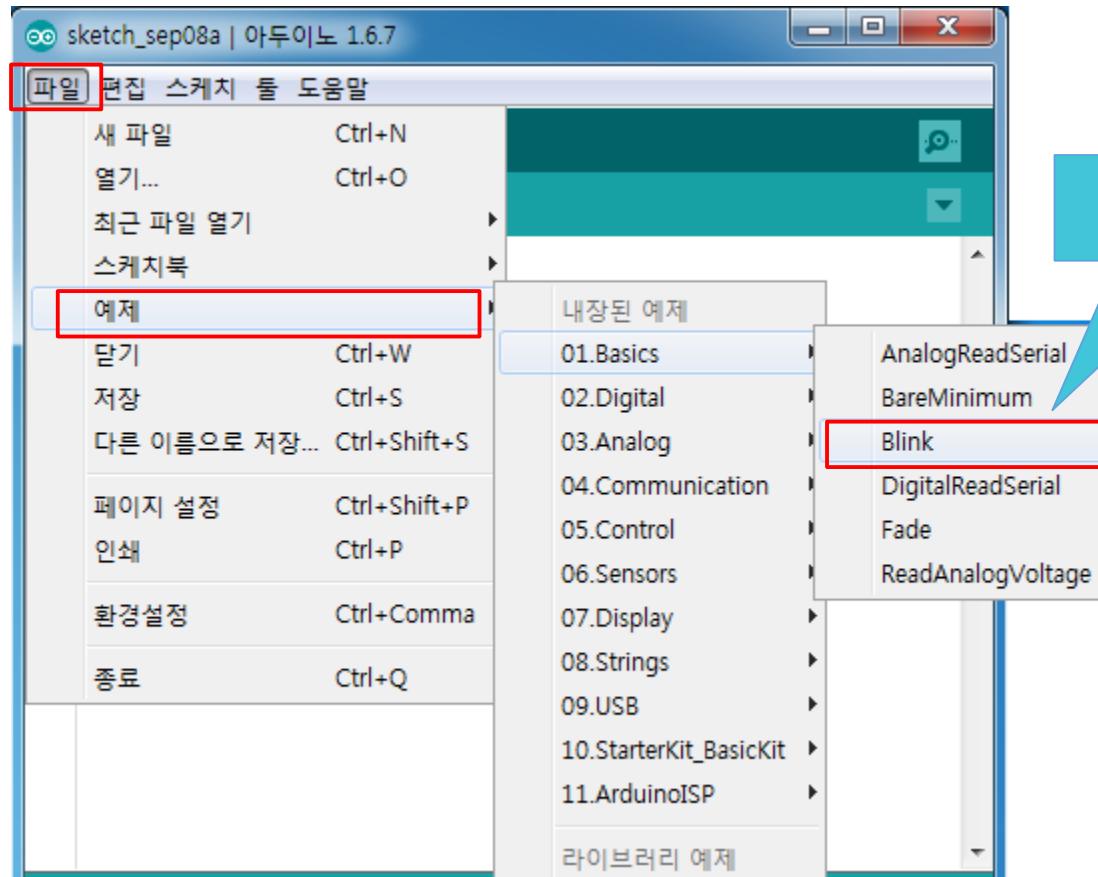
1. [환경설정]



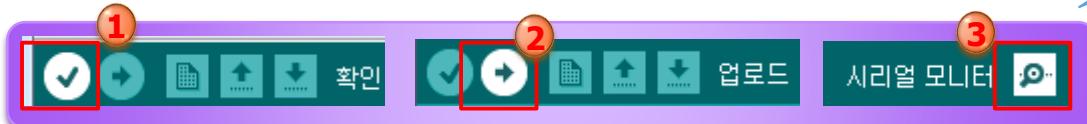
2. [확인] 클릭
(아두이노 프로그램 재시작)

세팅



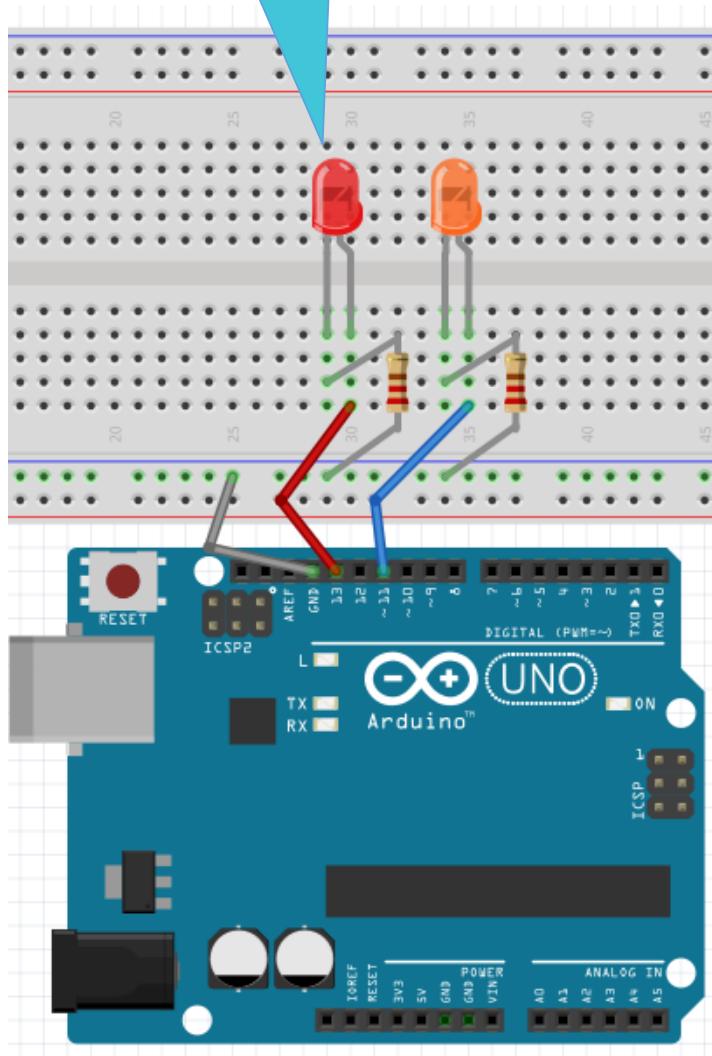


1. [Blink]



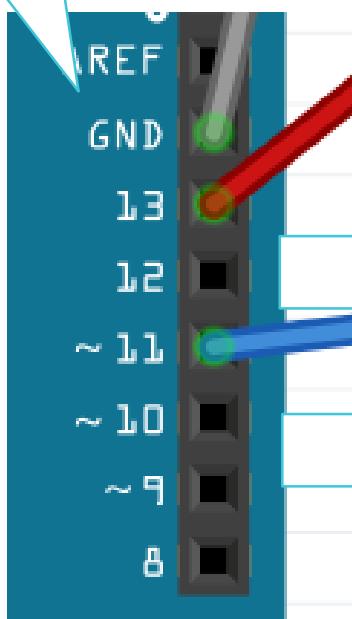
2. 실행

깜박인다.



pinMode 함수 – 디지털
13 번을 출력으로 설정

GND - 전류



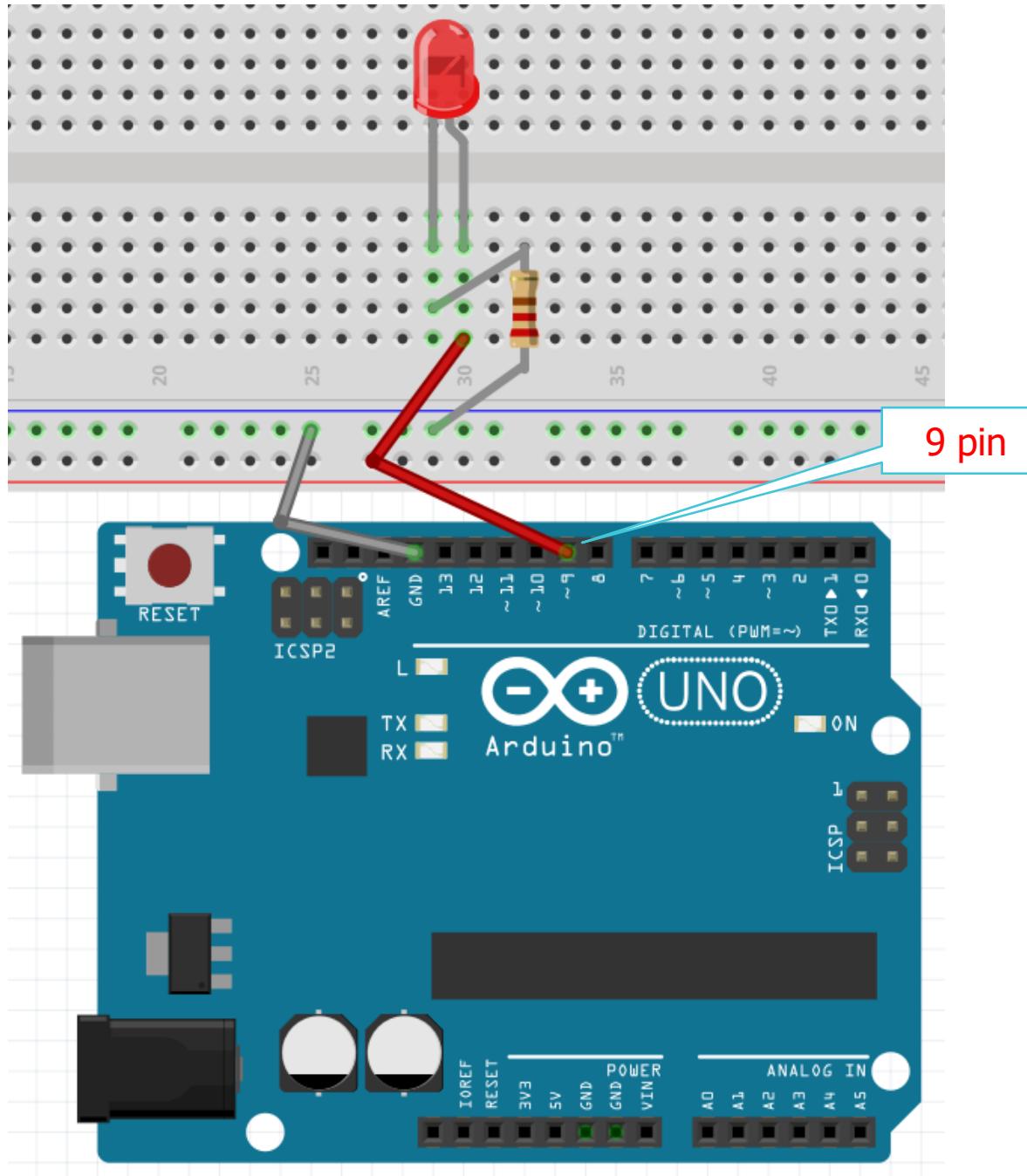
HIGH – 5v (켠다)

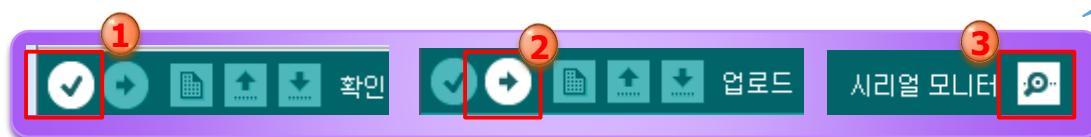
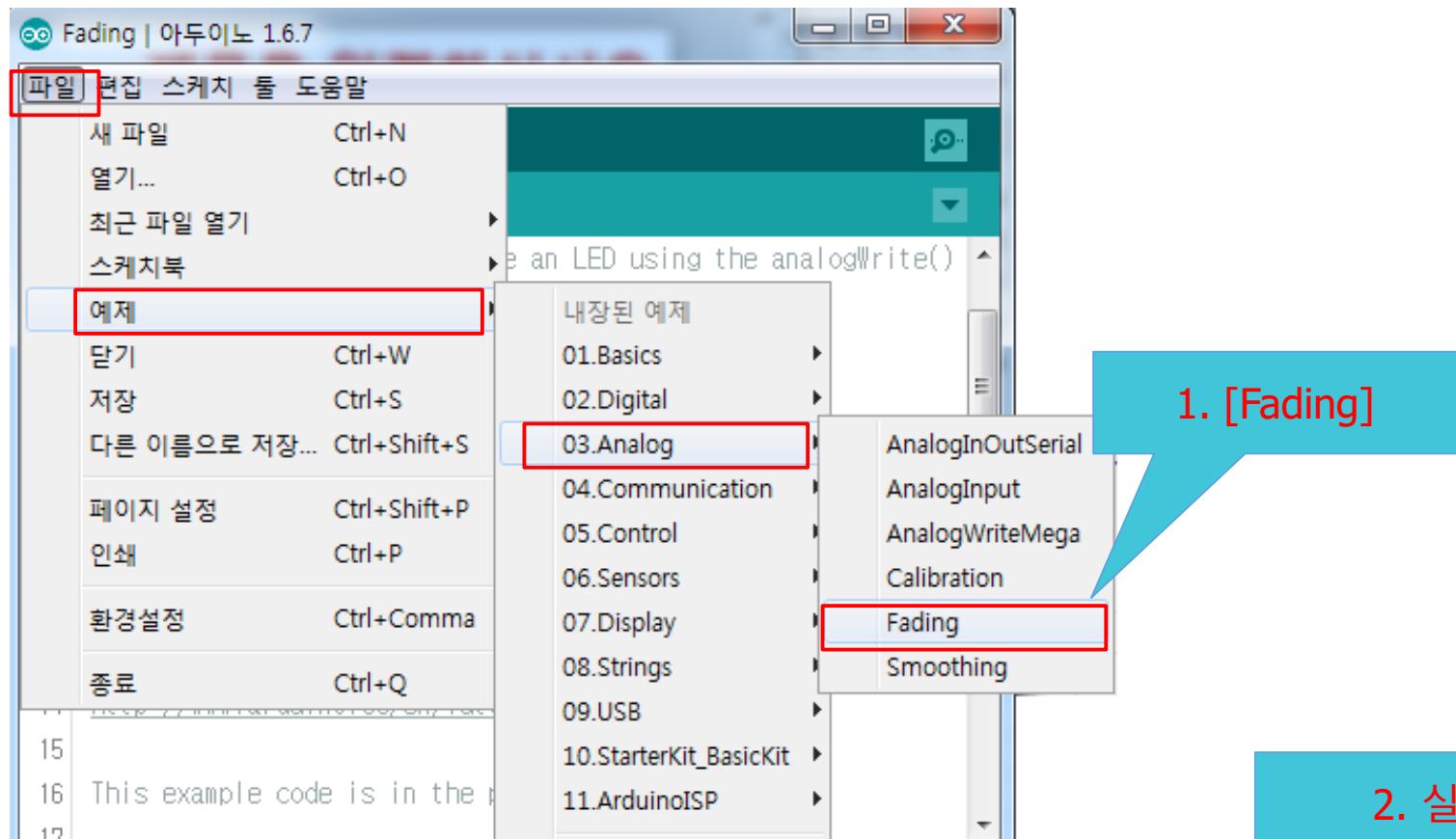
LOW – 0v (끈다)

```

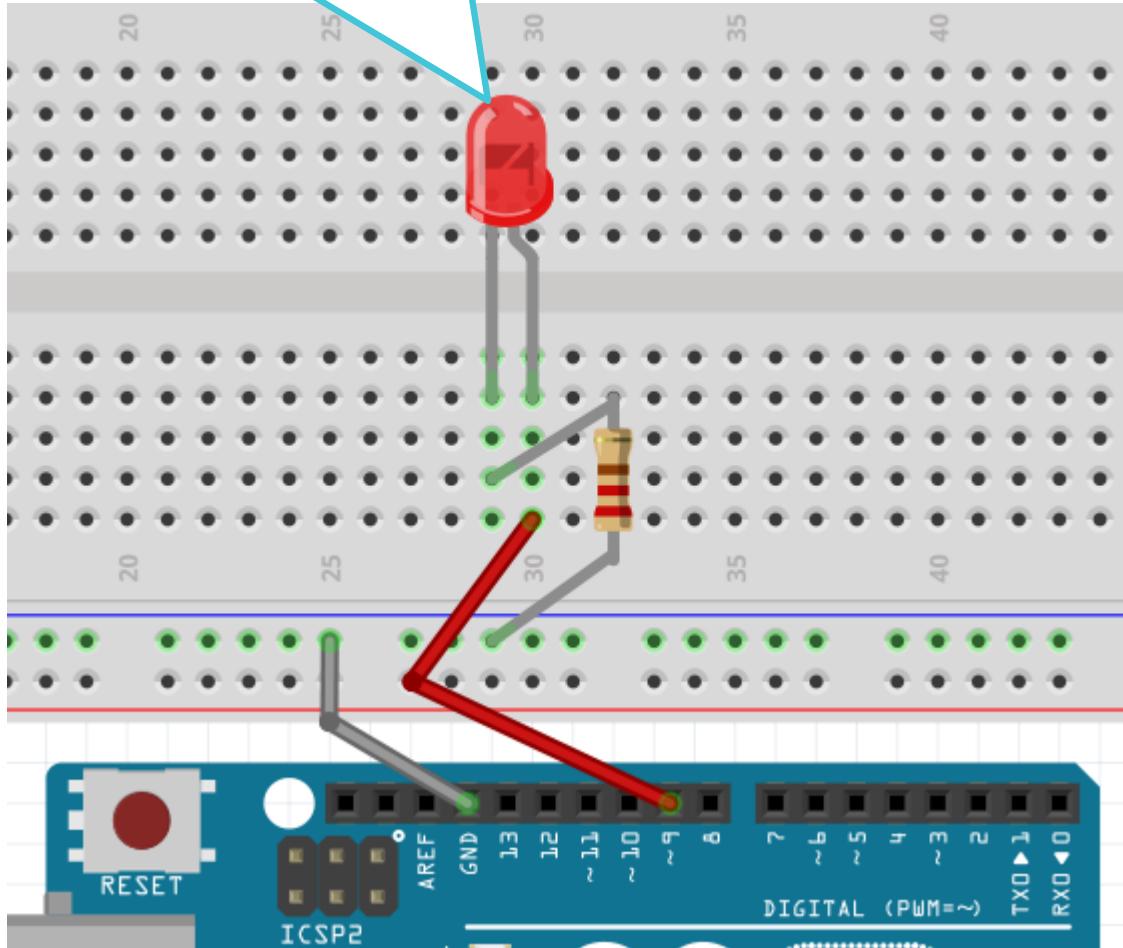
18 void setup() {
19   // initialize digital pin 13 as an output.
20   pinMode(13, OUTPUT);
21 }
22
23 // the loop function runs over and over again forever
24 void loop() {
25   digitalWrite(13, HIGH); // turn the LED on (HIGH is the
26   delay(1000); // wait for a second
27   digitalWrite(13, LOW); // turn the LED off by making t
28   delay(1000); // wait for a second
29 }

```





1.5 간격으로 LED의 밝기를 점점 밝게 ~ 점점 어둡게



Fading §

```
1
2 int ledPin = 9; // 9 번핀 연결
3
4 void setup() {
5
6
7 void loop() {
8     for (int fadeValue = 0 ; fadeValue <= 255; fadeValue += 5) {
9         analogWrite(ledPin, fadeValue);
10        delay(30);
11    }
12    for (int fadeValue = 255 ; fadeValue >= 0; fadeValue -= 5) {
13        analogWrite(ledPin, fadeValue);
14        delay(30);
15    }
16 }
```

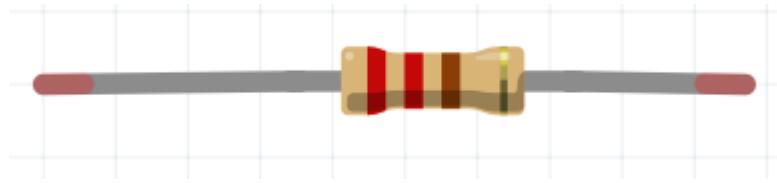
0 ~ 255 까지 5씩 증가시키면서 ledPin에 아날로그 값을
쓴다. 30밀리초 대기

0 ~ 255 까지 5씩 감소시키면서 ledPin에 아날로그 값을
쓴다. 30밀리초 대기

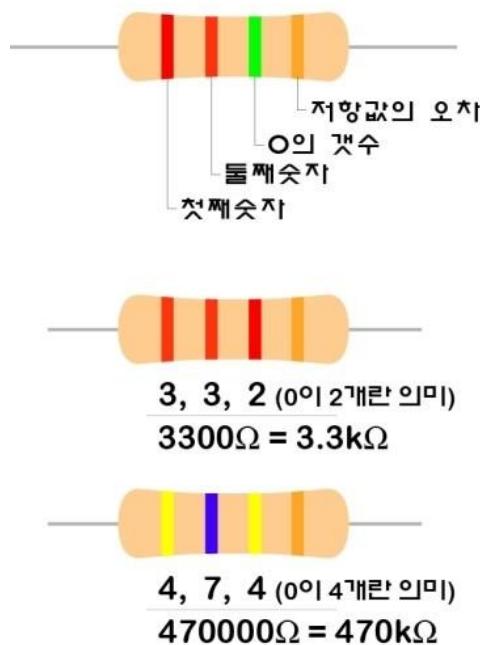


LED

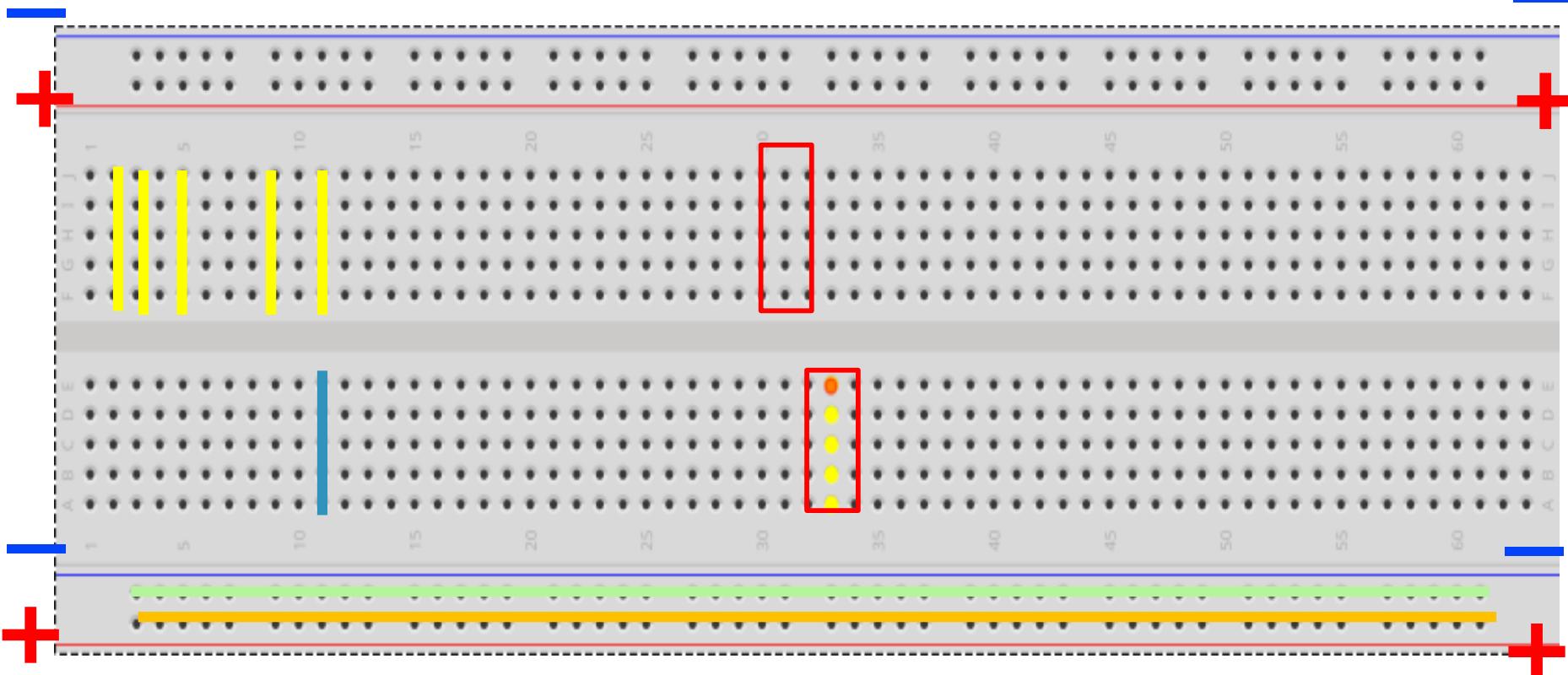
긴 핀을 전원의 양극(VCC, 5V), 짧은 핀을 음극(GND, 0V)으로 연결한다. 반대로 연결할 경우 전류가 흐르지 못해 LED 가 켜지지 않는다.



저항은 전류의 양을 조절하는 역할을 한다. 저항은 방향성이 없기 때문에 VCC와 GND(0V)에 어떤 방향으로도 연결할 수 있다.



색	값
검정색	0
갈색	1
빨강색	2
주황색	3
노란색	4
초록색	5
파란색	6
보라색	7
회색	8
하얀색	9
은색	$\pm 10\%$
금색	$\pm 5\%$
	*1
	*10
	*100
	*1000(1K)
	*10000(10K)
	*100000(100K)
	*1M
	*10M
	*100M
	*1000M
	N/A
	N/A



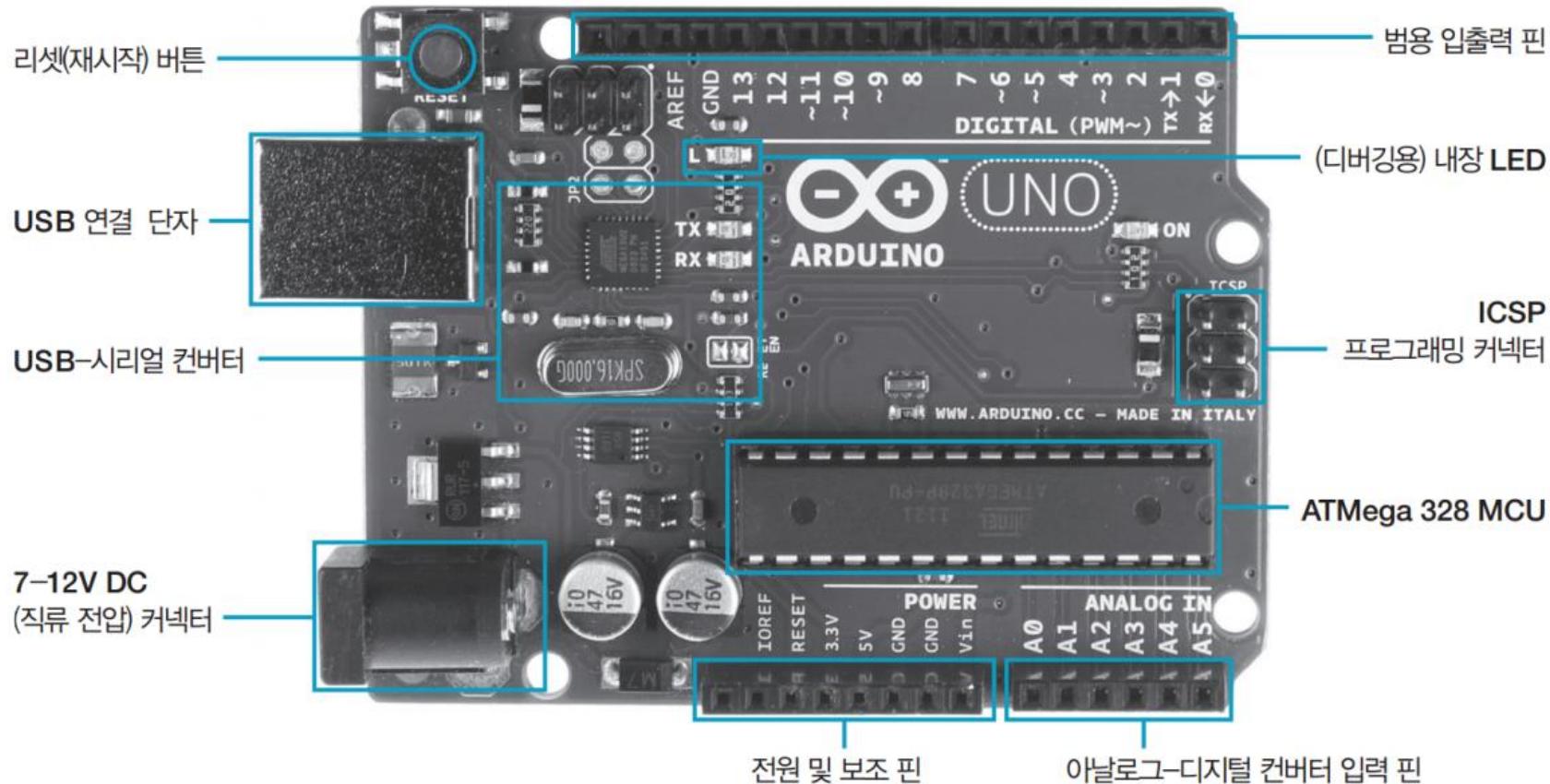
브레드 보드(일명 '빵판')

일반적으로 빨간선을 5V, 파란선을 GND(0V)에 연결



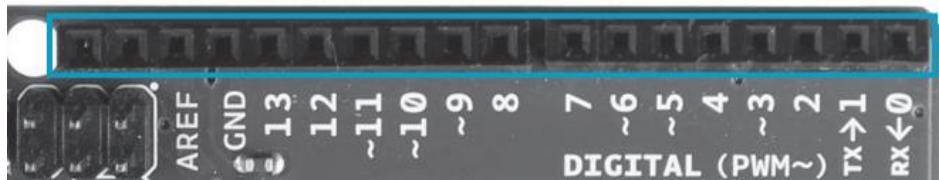
전선

브레드 보드에 부품들을 서로 연결할 때
사용하는 전선으로
‘점퍼선’이라고도 한다.

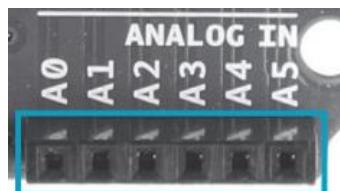


❖ 땡용 입출력(I/O) 핀과 아날로그-디지털 컨버터(ADC) 입력 핀

- 땡용 입출력 핀은 디지털 입출력에 사용하여 프로그램으로 각각의 주소를 지정할 수 있음
- 아두이노 Uno에는 총 14개의 땡용 입출력 핀이 내장



- ADC 핀은 센서에서 나오는 0~5V 사이의 전압 등 아날로그 신호를 측정하여 MCU에 디지털 신호로 입력
- 아두이노 Uno에는 총 6개의 ADC 핀이 내장



❖ Button

누른 경우 ON. 그 상태에서 떼면 OFF

❖ Switch

누른 경우 ON. 뗀 후 다시 눌렀다 떼면 OFF



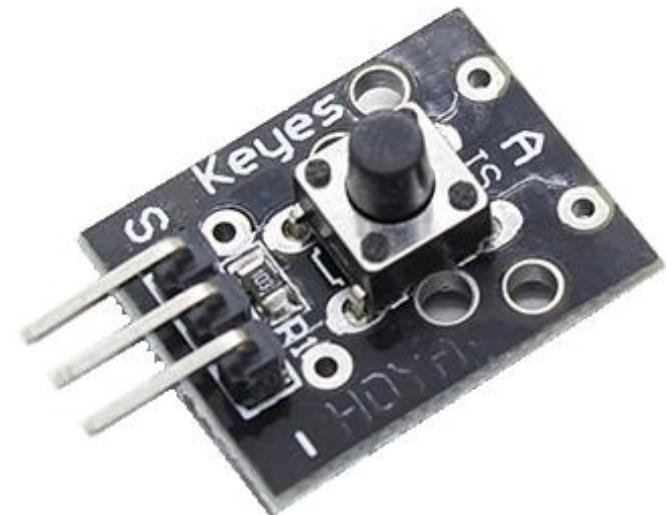
❖ Button

- 회로를 구성해야 함
- 저렴



❖ Button module

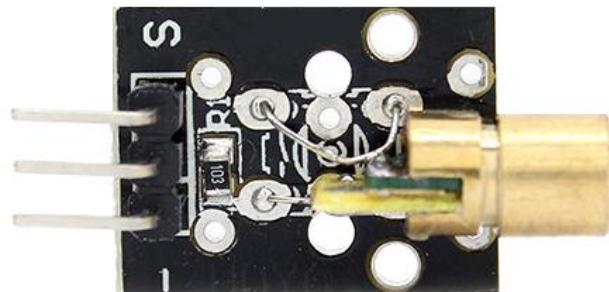
- 간단한 연결로 즉시 사용가능



- Potentiometer
- 저항의 값이 움직이면서 조절
- 돌려서 조절하는 가변저항을 “로터리저항”이라고도 함

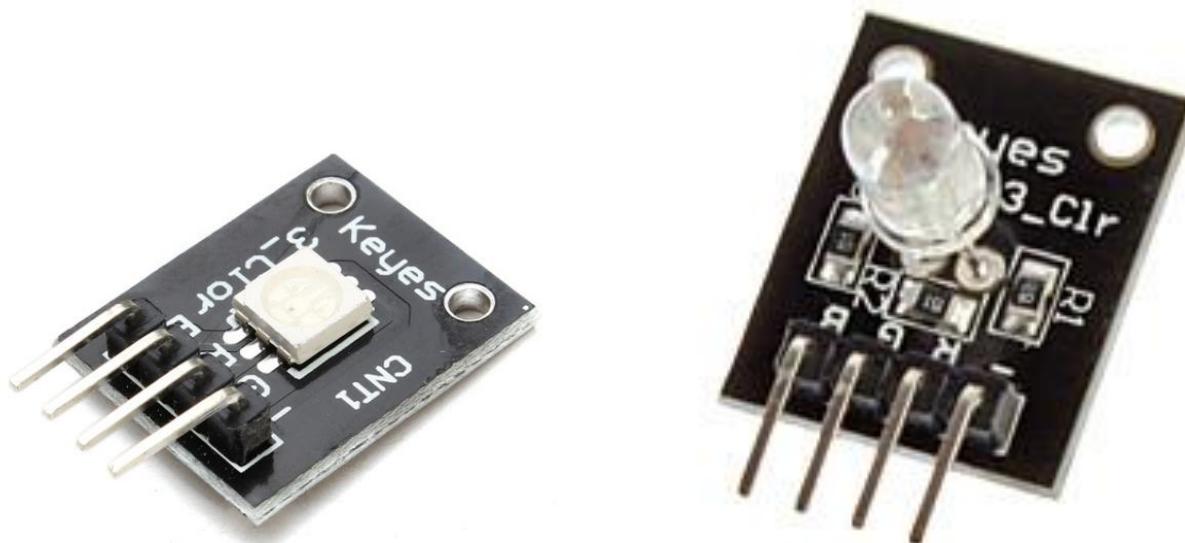


- Laser diode module

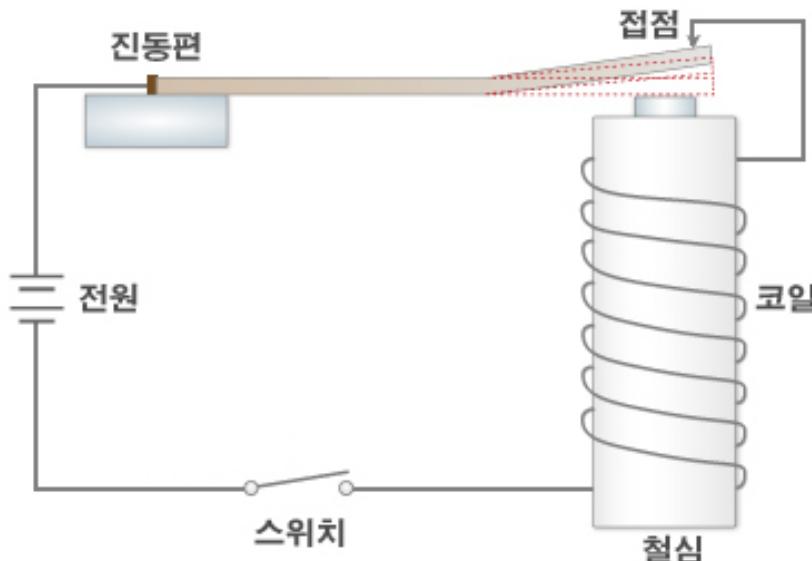


- 레이저동작을 전극을 2개 가지고 있는 반도체 레이저

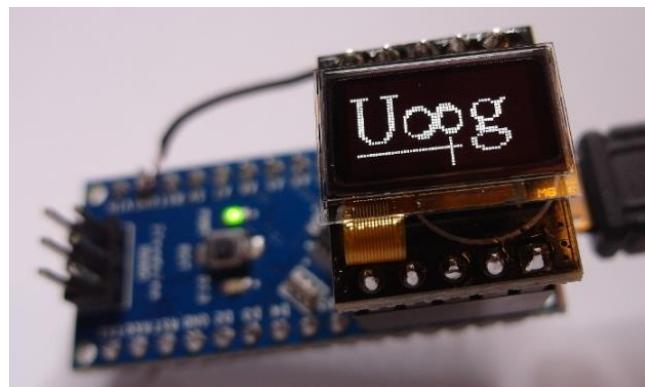
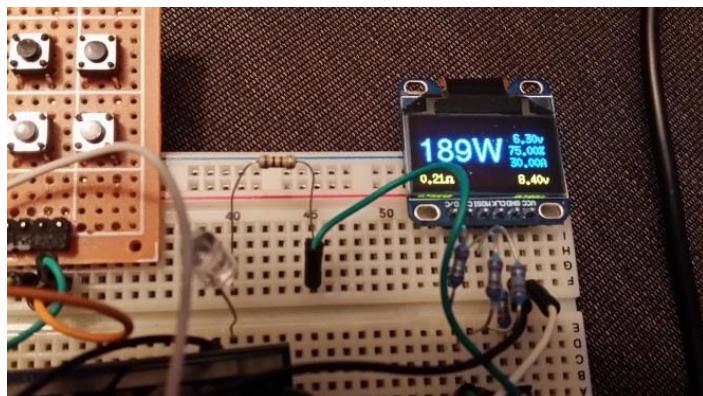
- 1개의 LED안에서 3가지 색상 출력
- 조합 - 3가지 색상을 조합하여 출력
- 개별 - 각기 색중 하나만 출력



- Buzzer module
- 전자석과 진동편의 조합으로 소리 발생
- 모듈이 아닌 부품으로 사용해도 쉬움



- 0.96 인치 흑백 OLED display
- U8glib를 이용해 코드작성
 - Universal Graphics Library for 8 Bit Embedded Systems
 - <https://code.google.com/p/u8glib/>
- 외부라이브러리 추가
 - I2Cdev.zip
 - U8glib.zip



- 디지털 출력 값
 - 0
 - 1
- 아날로그 출력 값
 - 0
 - 0 초과 값

- 출력 값
 - 0
 - 0 초과 값

- Cadmium sulphide
- 황화카드뮴
- 빛에 따라 저항 값이 달라지는 부품

