

사물인터넷(IoT)

3. LED 응용 프로그램 제작하기, 시리얼 모니터 확인하기











목차

- 아두이노 기초 이론
- Fritzing 소개/연습
- LED 응용 프로그램 제작하기
 - 1개 LED 제어하기
 - 3개 LED 제어하기
 - 7개 LED 제어하기
- 시리얼 모니터 확인하기

아두이노 부품 회로 구성



디지털 입출력 장치

디지털 입력				디지털 출력			
 Pushbutton	 Slideswitch	 PIR Sensor	 DIP Switch DPST	 LED	 LED RGB	 DC 모터	 Hobby Gearmotor
				 7 Segment Display	 LCD 16 x 2		

아날로그 입출력 장치

아날로그 입력				아날로그 출력			
 Potentiometer	 Photoresistor	 Ultrasonic Distance Sensor	 Ultrasonic Distance Sensor	 LED RGB	 DC 모터	 Micro Servo	 Hobby Gearmotor
 Temperature Sensor [TMP36]				 Piezo	 LCD 16 x 2		

아날로그, 디지털

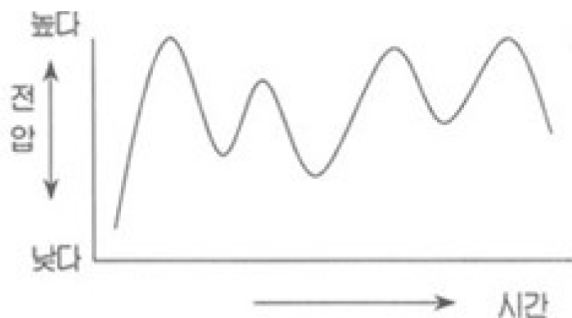
- **아날로그**

- 어떤 양 또는 데이터를 연속적으로 변환하는 물리량(전압, 전류 등)으로 표현
- 아날로그 신호는 전류의 주파수나 진폭 등 연속적으로 변화하는 형태로 전류를 전달

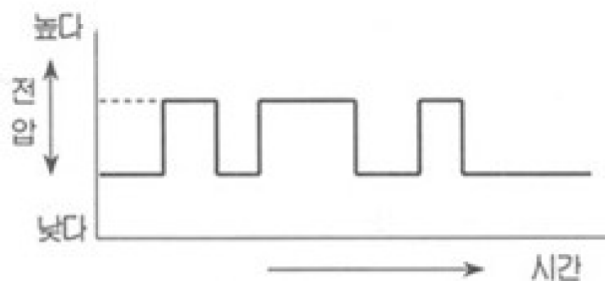
- **디지털**

- 어떤 양 또는 데이터를 2진수로 표현
- 디지털 신호는 전류가 흐르는 상태(1)와 흐르지 않는 상태(0)의 2가지 조합으로 전달

아날로그, 디지털



(a) 아날로그 신호의 파형



(b) 디지털 신호의 파형

이렇게면, 디지털은
0부터 1사이는 0,
1부터 2사이는 1,
이런식으로 표시,
아날로그는 0.3은 0.3,
0.327은 0.327로 그대로 표시

아날로그, 디지털 입출력

- **디지털**

- LOW(0V, 0볼트, GND, GROUND), HIGH(5V, 5볼트)의 2가지 상태를 입/출력한다.
- digitalWrite(), digitalRead()

- **아날로그**

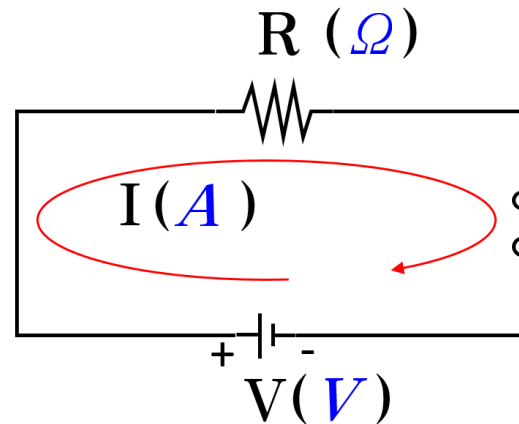
- 0 ~ 255 또는 0 ~ 1023 사이의 값을 입/출력 한다.
- analogWrite(), analogRead()

옴의 법칙 (Ohm's Law)

- 전기회로에 흐르는 전류는 전압에 비례하고 저항에 반비례

$$V = IR$$

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



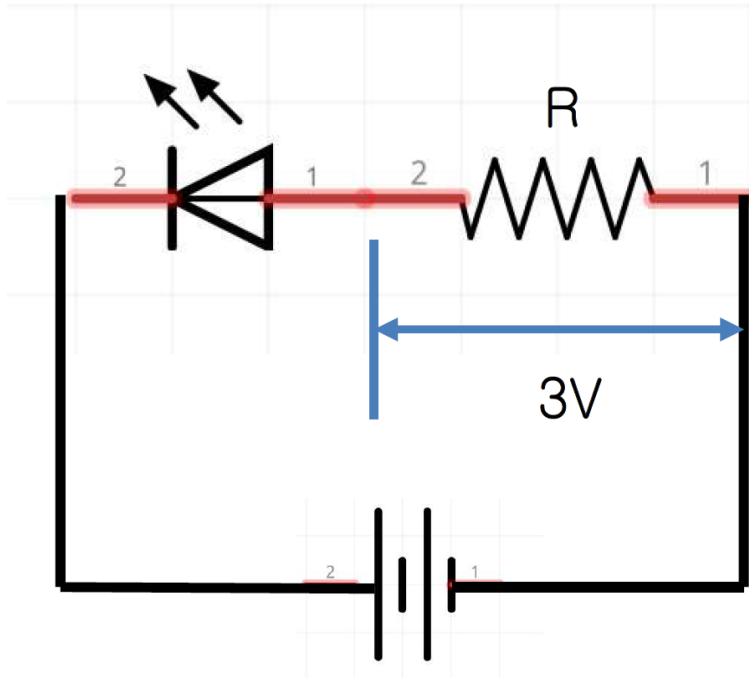
전압(V : Volt): 전류를 흐르게 하는 전기적인 압력, 단위 볼트 [V]

전류(I : Intensity of Current): 단위 시간에 통과하는 전하의 양, 단위 암페어 [A]

저항(R : Resistance): 전류의 흐름을 방해하는 성질, 단위 옴 [Ω]

LED가 202 Ω 저항을 사용하는 이유?

LED의 최대 전류값은 20mA이며, 사용 전압은 약2V이다.



$$R = V/I$$

$$R = 3V/0.02A$$

$$R = 150 \Omega$$

150 Ω 이상인 저항을 사용하면 된다.

보통 LED회로에서는 220 Ω 을 많이 사용한다.
구하기 쉽고 LED를 충분히 밝게 한다.

저항값 읽기

4색 코드 저항



5, 0, 2(50×100)
 $50 \times 100 = 5,000\Omega$
 $5,000\Omega = 5\text{ K}\Omega$ 오차 $\pm 5\%$

5색 코드 저항



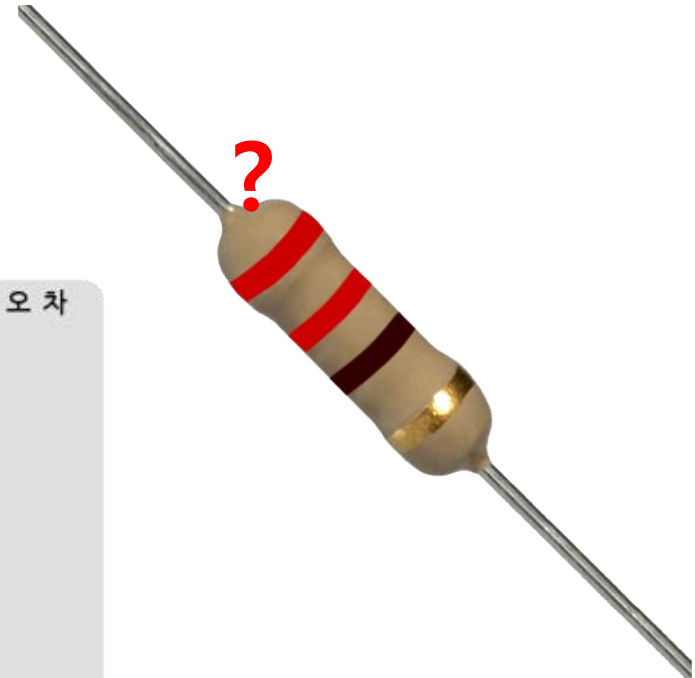
2, 6, 0, 3(260×1000)
 $260 \times 1000 = 260,000\Omega$
 $260,000\Omega = 260\text{ K}\Omega$ 오차 $\pm 10\%$

흑
갈
적
등
황
녹
청
자
회
백

검정색
갈 색
빨강색
주황색
노란색
초록색
파란색
보라색
회 색
하얀색
금 색
은 색
없음(무)

숫 자	오 단위
0	1
1	10
2	100
3	1K
4	10K
5	100K
6	1M
7	10M
8	100M
9	1G

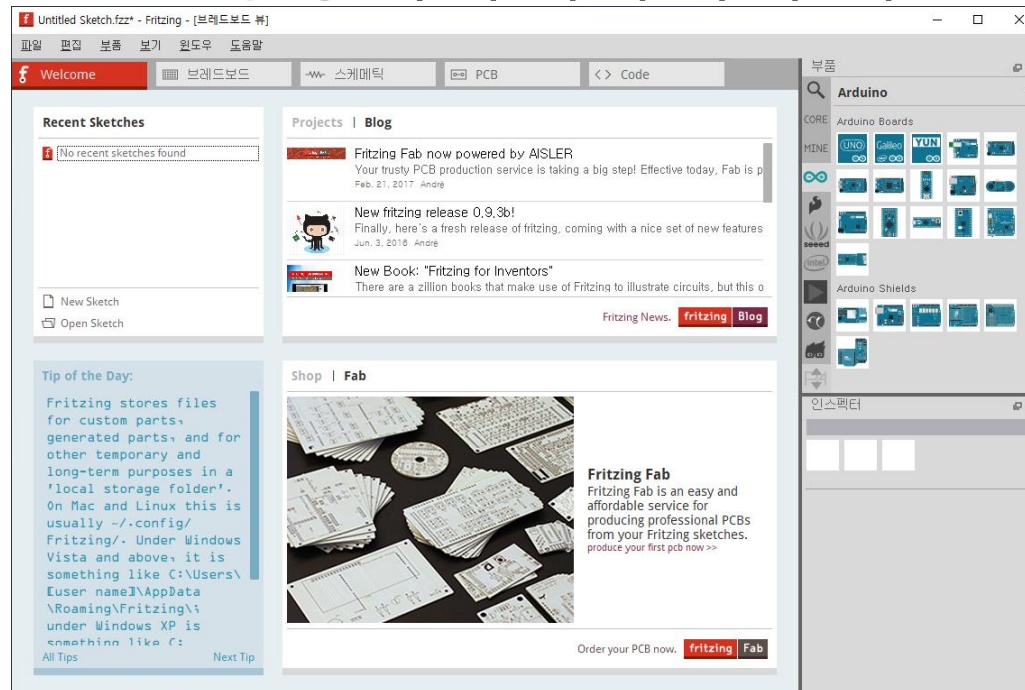
오 차
5%
10%
20%



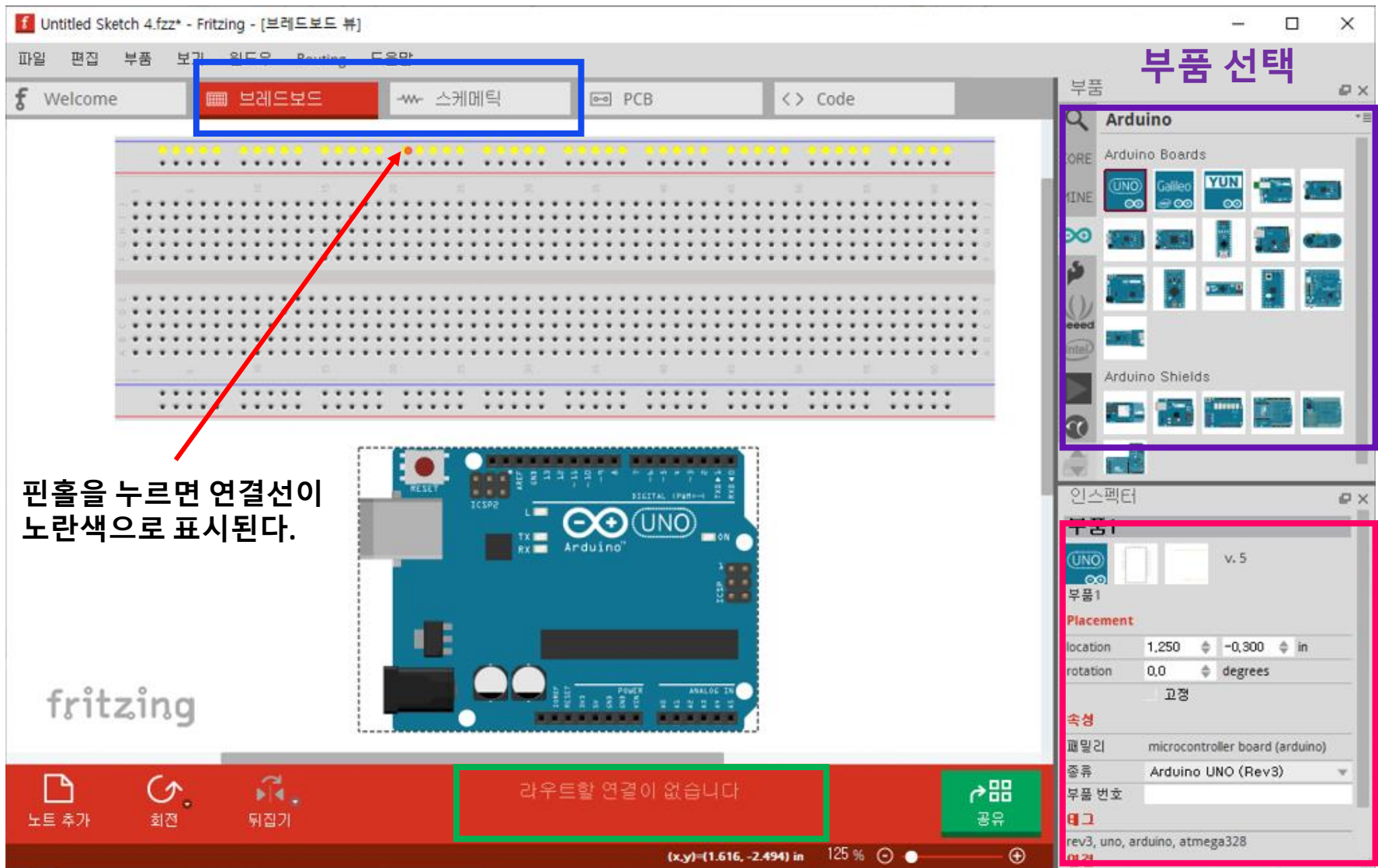
재미있는 아두이노 DIY

Fritzing

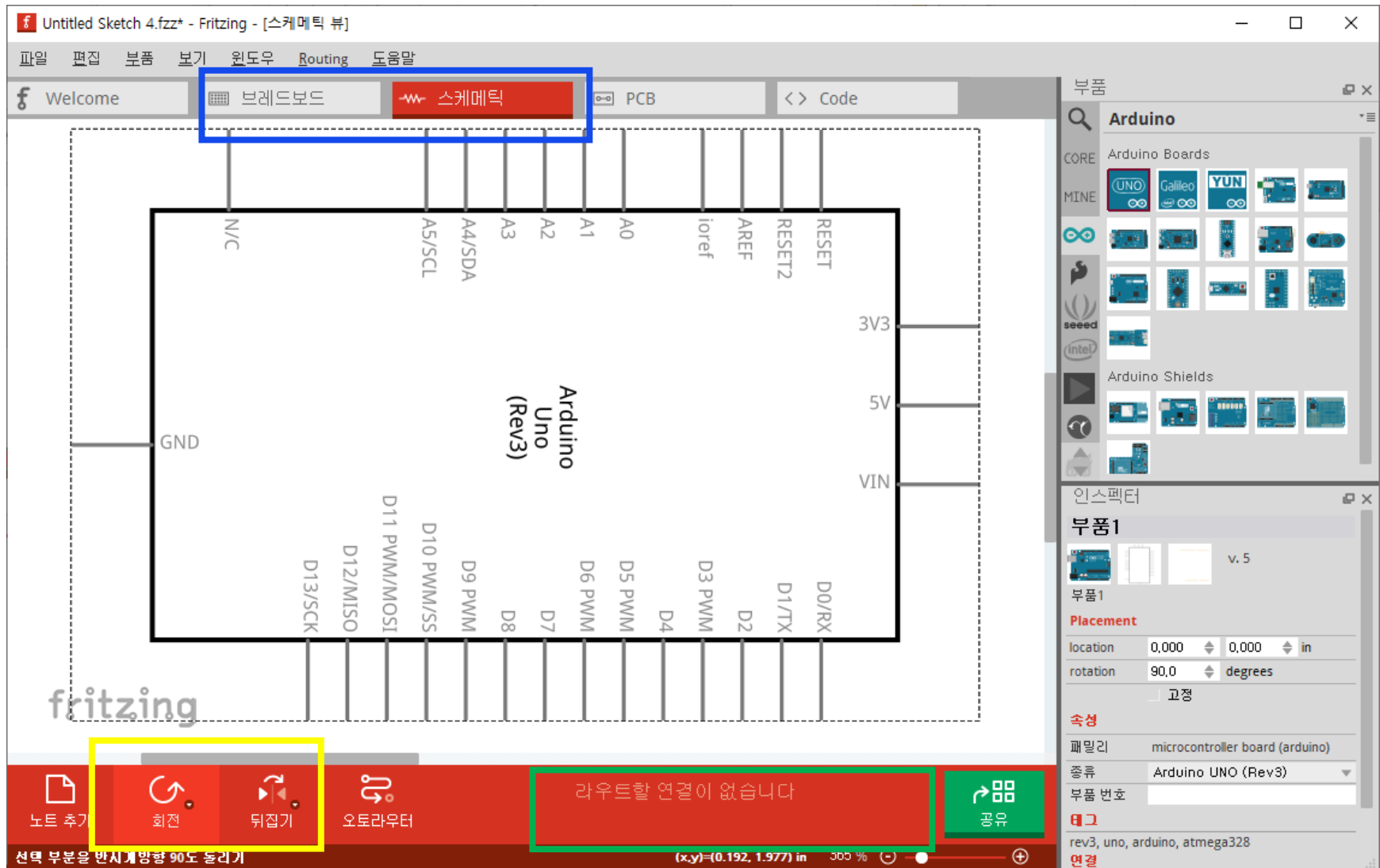
- 전자 하드웨어 설계 오픈 소스 소프트웨어
- 프로토타입에서 제품으로 쉽게 이동할 수 있게 하는 설계 도구
- <http://www.fritzing.org>
- 독일 포츠담 응용과학대학에서 개발



- 브레드보드/스케메틱에서 회로 구성
 - 브레드보드 회로



• 스케메틱 회로



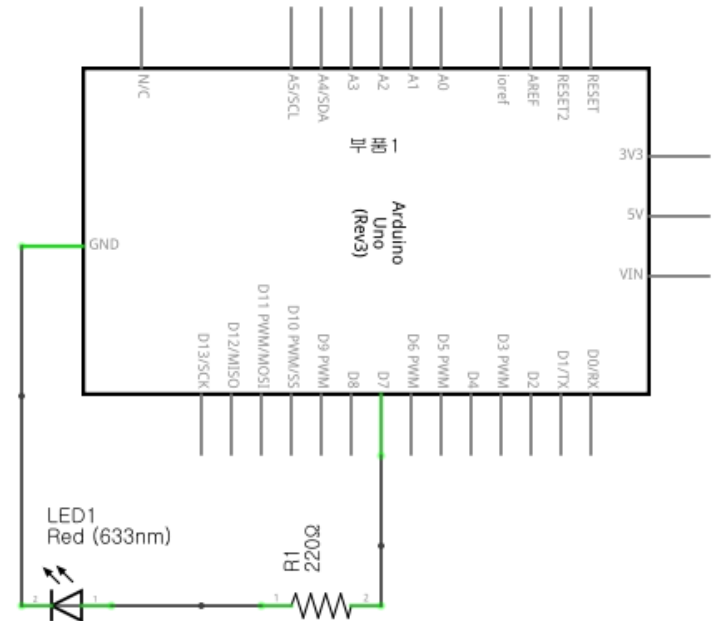
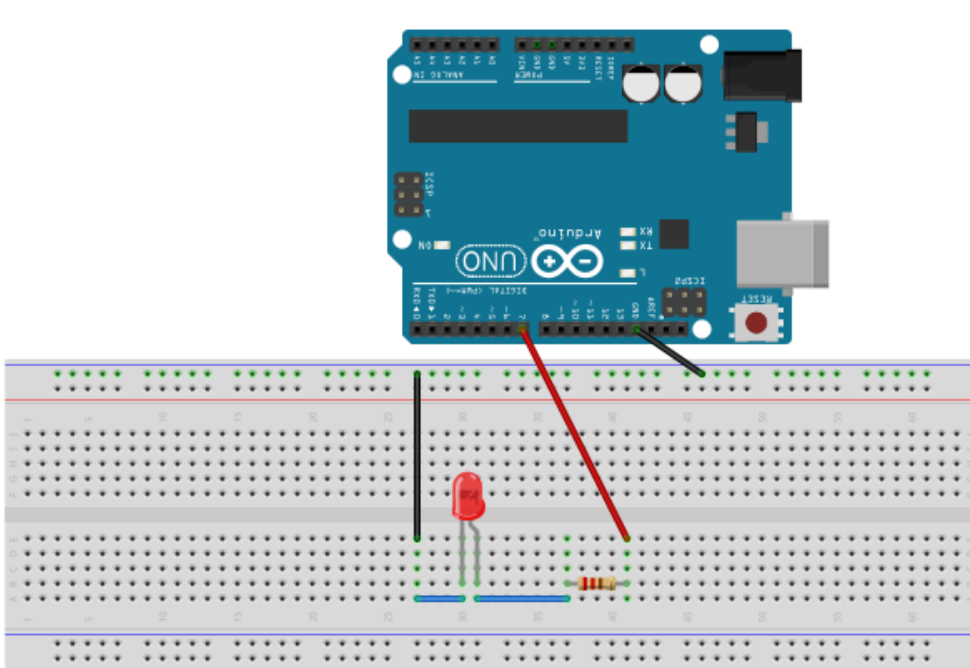
적절히 사용한다.

회로 구성 결과 확인

1개 LED 제어하기

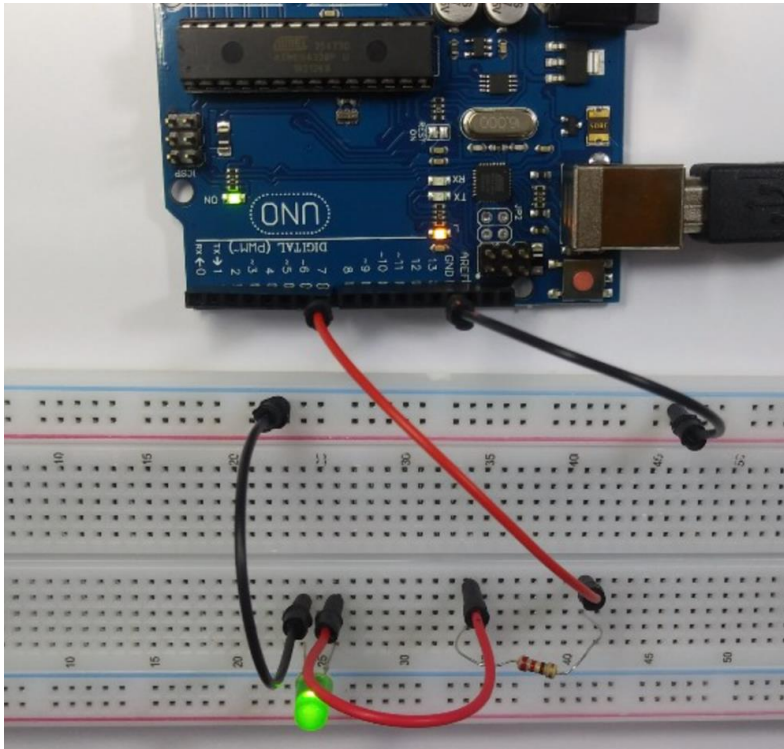
- 한 개의 LED를 0.1초 간격으로 켜고 끄도록 함

1. 회로 설계하기(with fritzing)



2. 실제 회로 구성

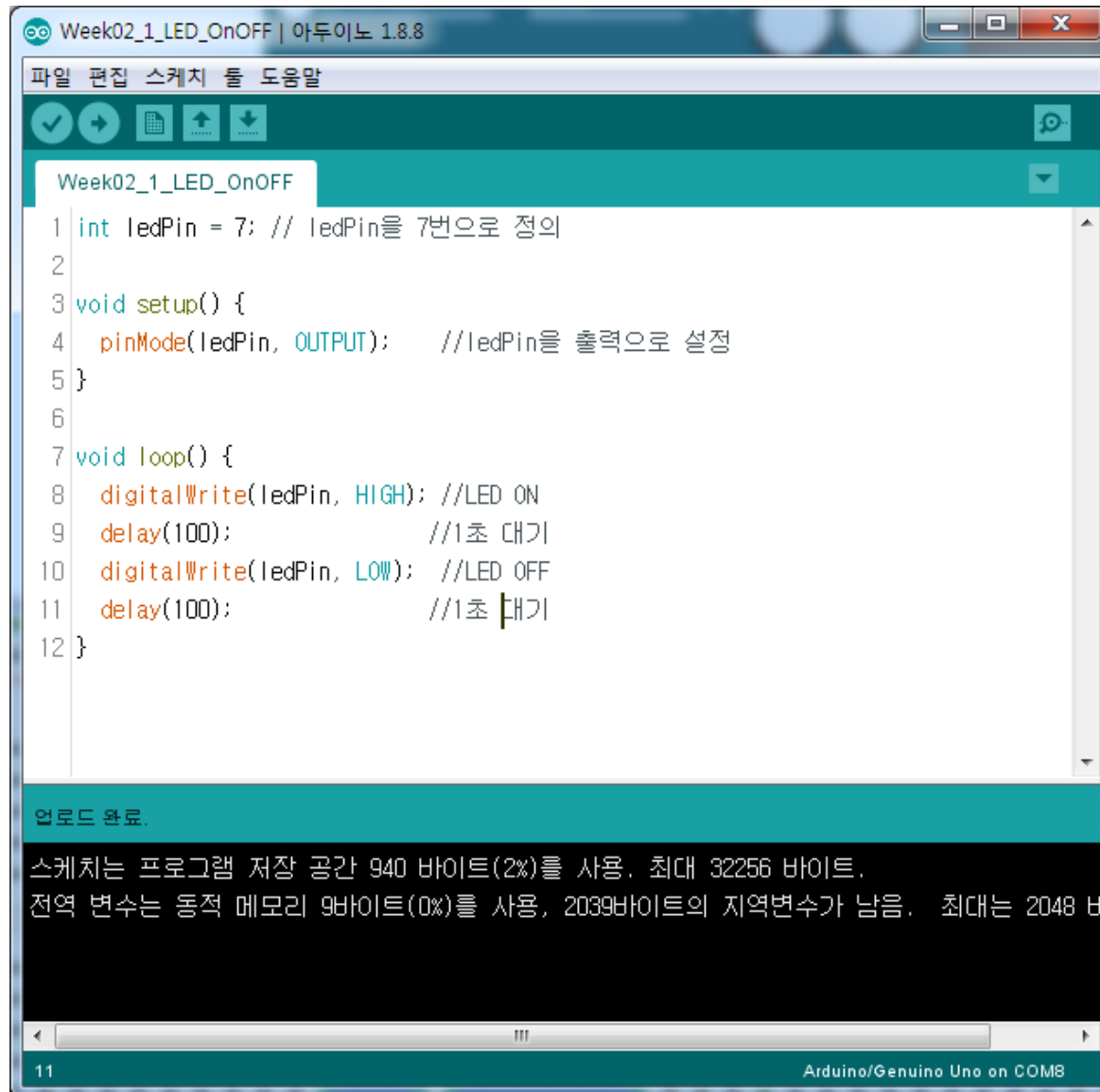
- 준비물 : 브레드보드, 아두이노, 저항 220 Ω 1개, LED 1개



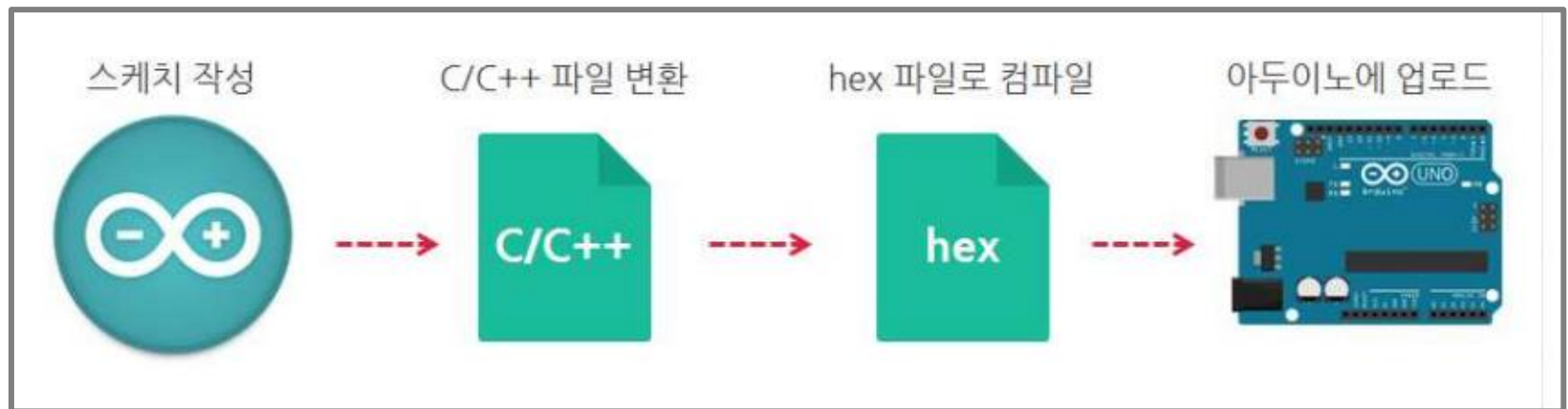
주의사항

- 아두이노 회로 구성 시 반드시 PC 연결 USB 제거할 것
- 업로드 실패 시 COM포트 확인할 것

3. 스케치 프로그래밍



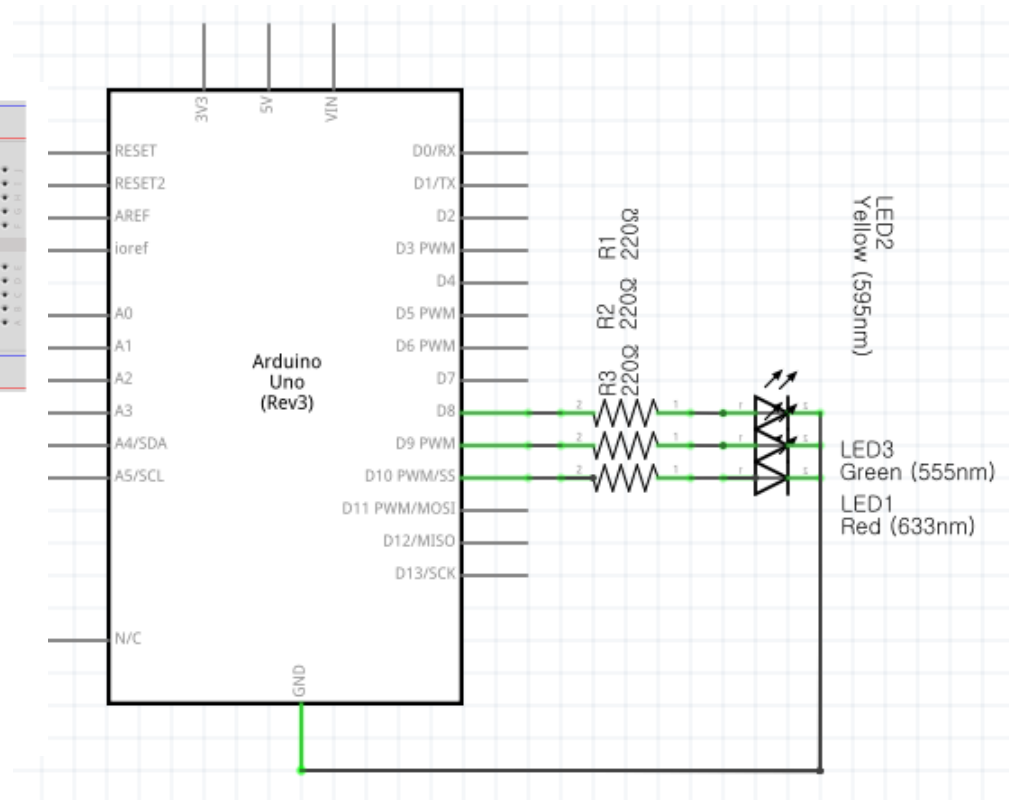
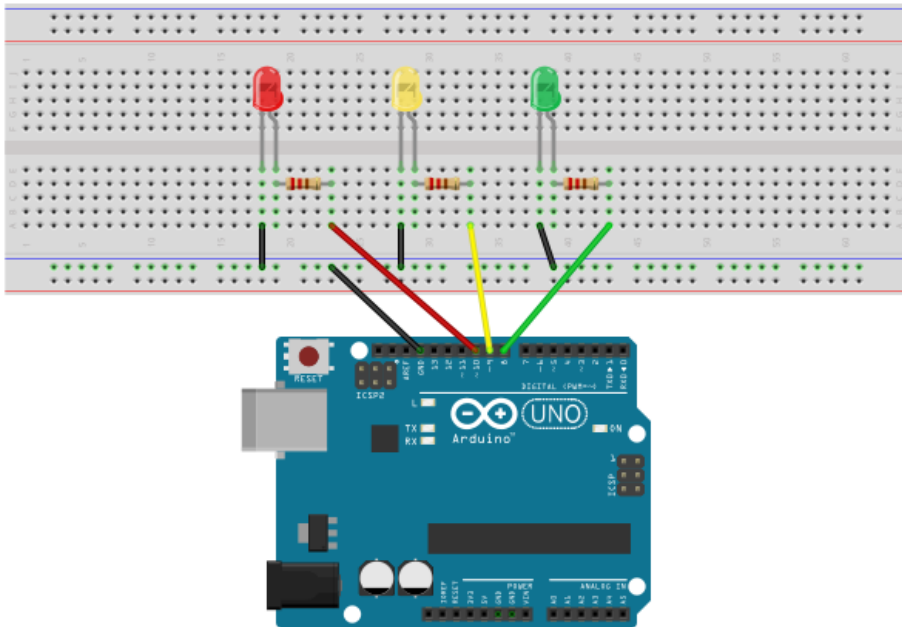
- 스케치로 작성한 프로그램의 동작 과정



3개 LED 제어하기 – 신호등

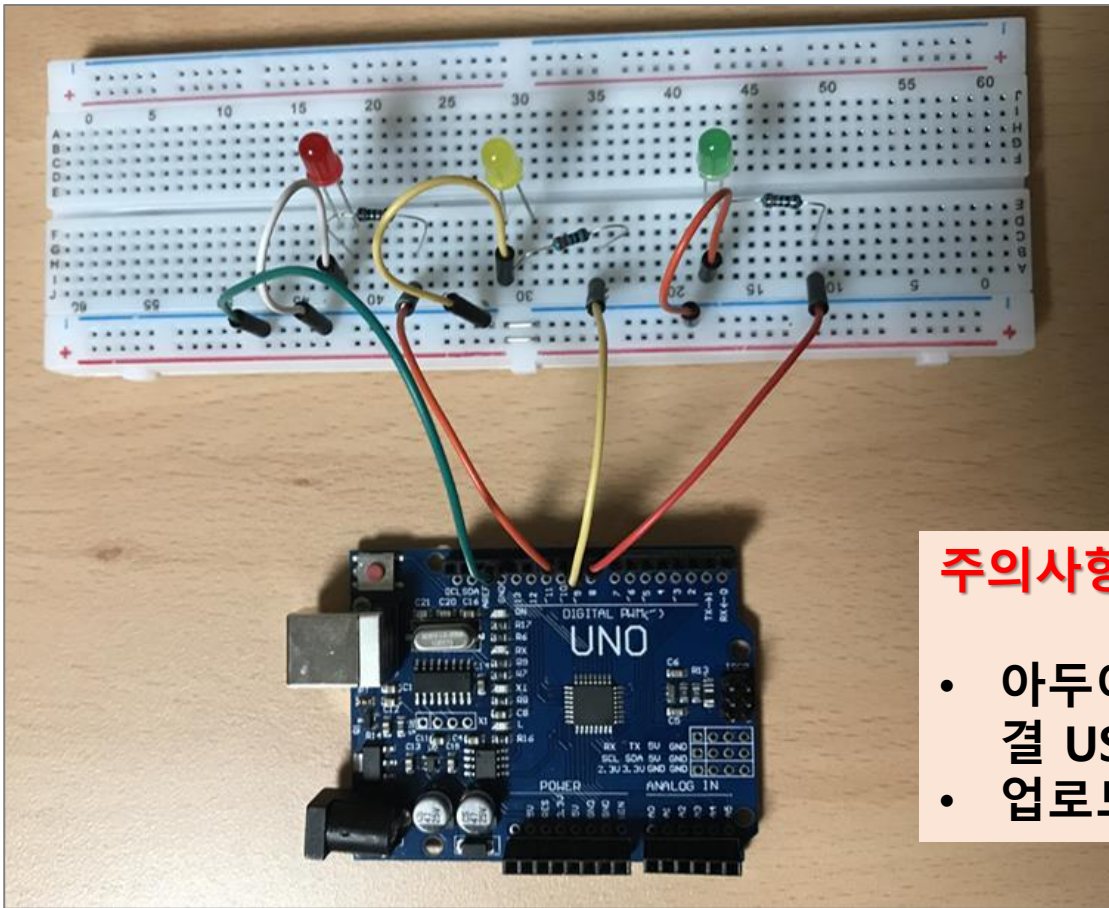
- 빨강, 노랑, 초록의 LED를 1초 간격으로 켜고 끄도록 함
1. 회로 설계하기(with fritzing)
 - 출력핀 – 10 : 빨강, 9 : 노랑, 8 : 초록
 2. 실제 회로 구성하기
 - 준비물 준비하기
 3. 스케치 프로그래밍
 4. 실행

1. 회로 설계하기(with fritzing)



2. 실제 회로 구성하기

- 준비물 : 브레드보드, 아두이노, 저항 220 Ω 3개, LED 3개



주의사항

- 아두이노 회로 구성 시 반드시 PC 연결 USB 제거할 것
- 업로드 실패 시 COM포트 확인할 것

3. 스케치 프로그래밍



```
sketch_mar16a $
int red = 10;
int yellow = 9;
int green = 8;
int delaytime=1000;

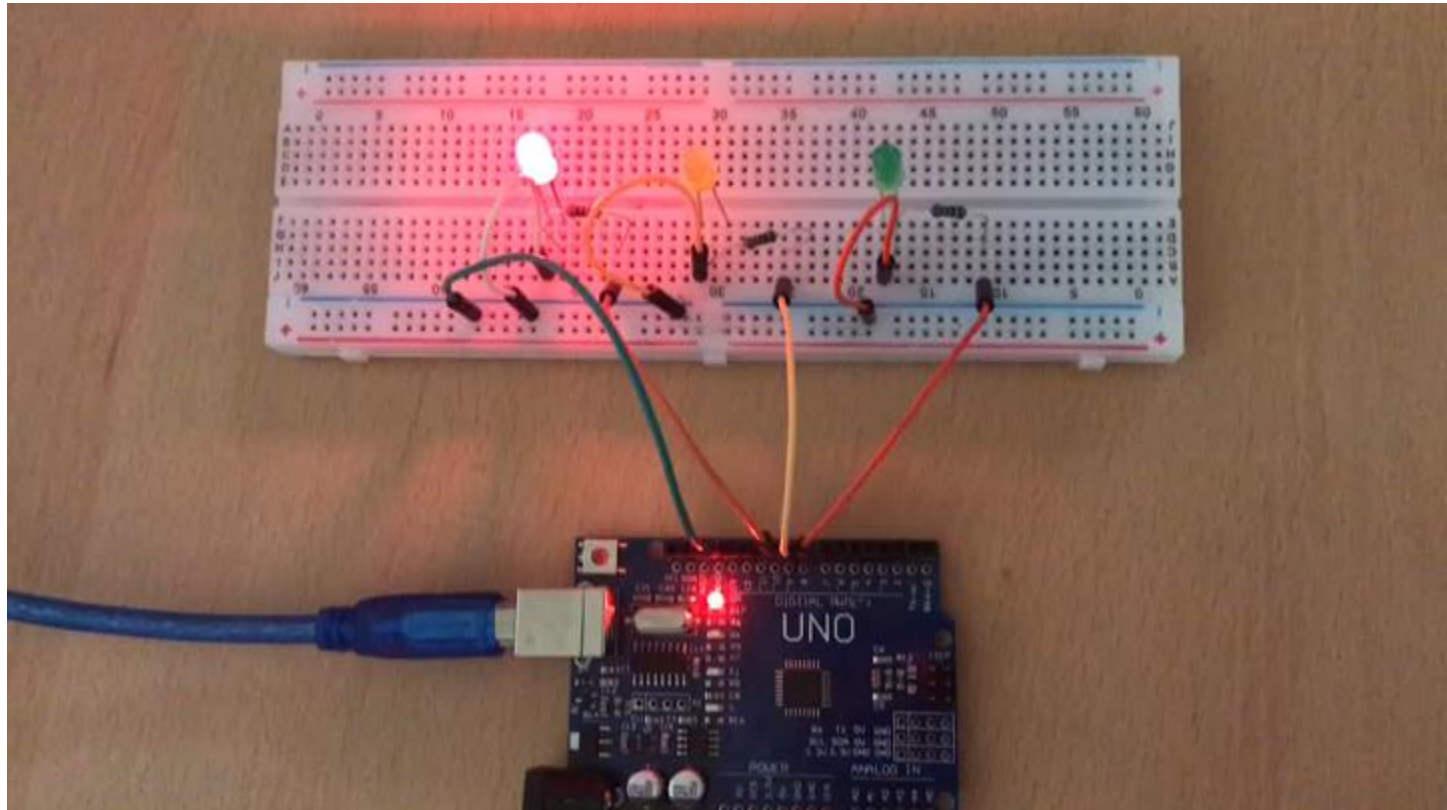
void setup() {
  // 8, 9, 10핀을 출력핀으로 설정
  pinMode(red,OUTPUT);
  pinMode(yellow,OUTPUT);
  pinMode(green,OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(red, HIGH);
  delay(delaytime);
  digitalWrite(red, LOW);
  delay(delaytime);

  digitalWrite(yellow, HIGH);
  delay(delaytime);
  digitalWrite(yellow, LOW);
  delay(delaytime);

  digitalWrite(green, HIGH);
  delay(delaytime);
  digitalWrite(green, LOW);
  delay(delaytime);
}
```

4. 실행

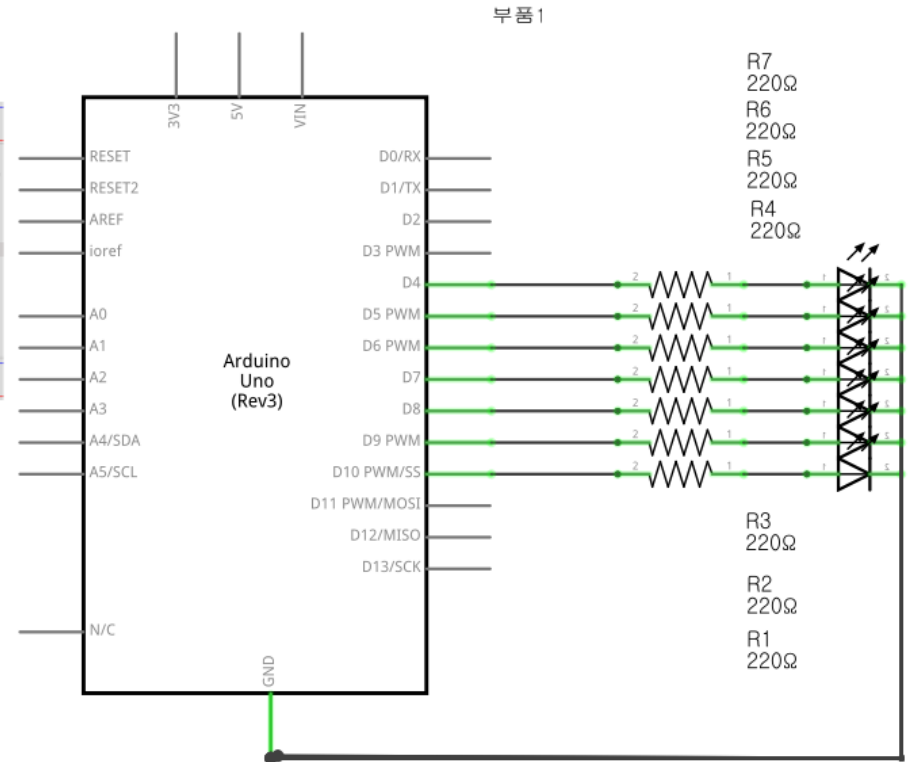
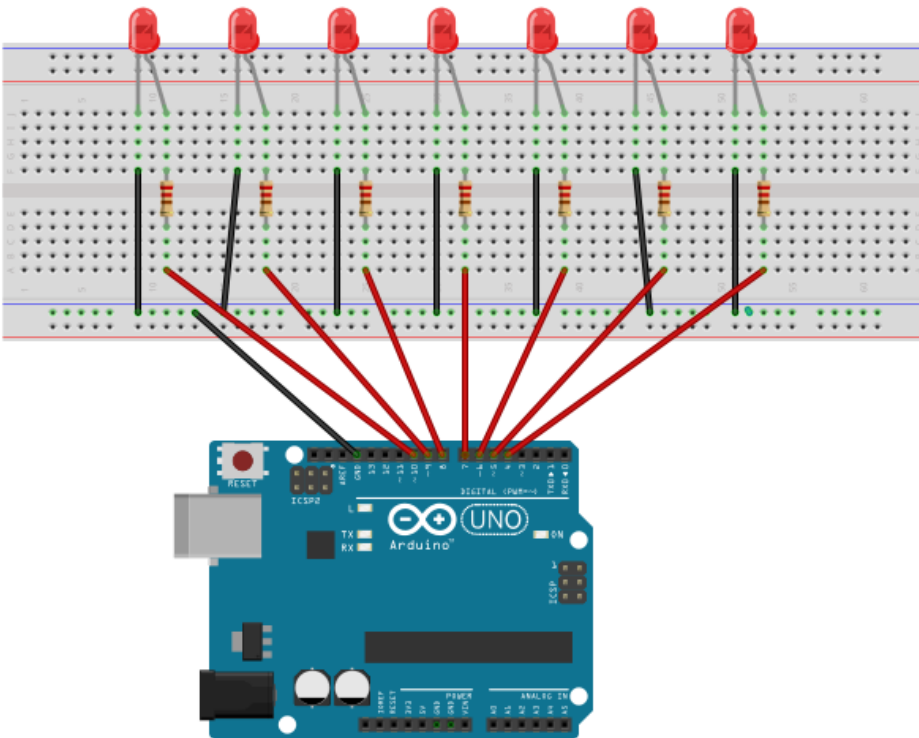


7LED 제어하기-위험 안내 LED

- 빨강색 LED 7개가 1초에 한번씩 점멸하는 표지판 제작하기
 1. 회로 설계하기(with fritzing)
 - 출력핀 - 4~10까지 7개
 2. 실제 회로 구성하기 - 준비물
 3. 스케치 프로그래밍
 4. 실행

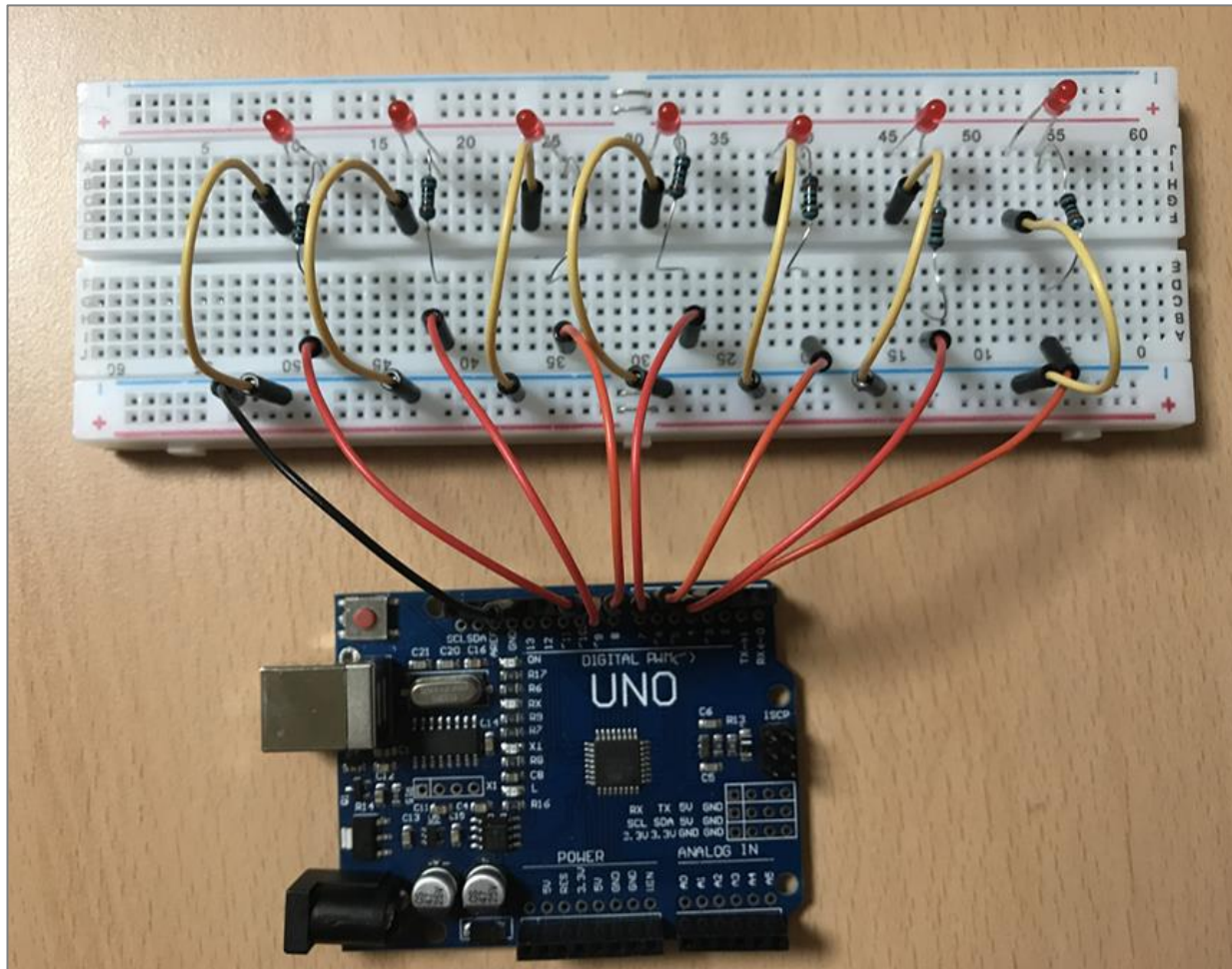


1. 회로 설계하기(with fritzing)



2. 실제 회로 구성

- 준비물 : 브레드보드, 아두이노, 저항 220 Ω 7개, LED 7개



3. 스케치 프로그래밍



The screenshot shows the Arduino IDE interface with a sketch titled "04-7LED \$". The code is written in C++ and is designed to control 7 LEDs connected to pins 4 through 10. The setup function initializes the serial port at 9600 baud and sets all pins from 4 to 10 as OUTPUT. The loop function turns all LEDs on (HIGH) and waits for 1000ms, then turns all LEDs off (LOW) and waits for another 1000ms, creating a blinking effect.

```
04-7LED | 아두이노 1.8.8
파일 편집 스케치 툴 도움말

04-7LED $

void setup() {
  Serial.begin(9600); //아두이노와 PC간의 통신 속도 설정
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(8, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(10, OUTPUT);
}

void loop() {
  //4~10까지의 핀에 연결된 LED를 켜다.
  digitalWrite(4, HIGH);
  digitalWrite(5, HIGH);
  digitalWrite(6, HIGH);
  digitalWrite(7, HIGH);
  digitalWrite(8, HIGH);
  digitalWrite(9, HIGH);
  digitalWrite(10, HIGH);

  delay(1000); //1초 켜다.

  //4~10까지의 핀에 연결된 LED를 끈다.
  digitalWrite(4, LOW);
  digitalWrite(5, LOW);
  digitalWrite(6, LOW);
  digitalWrite(7, LOW);
  digitalWrite(8, LOW);
  digitalWrite(9, LOW);
  digitalWrite(10, LOW);

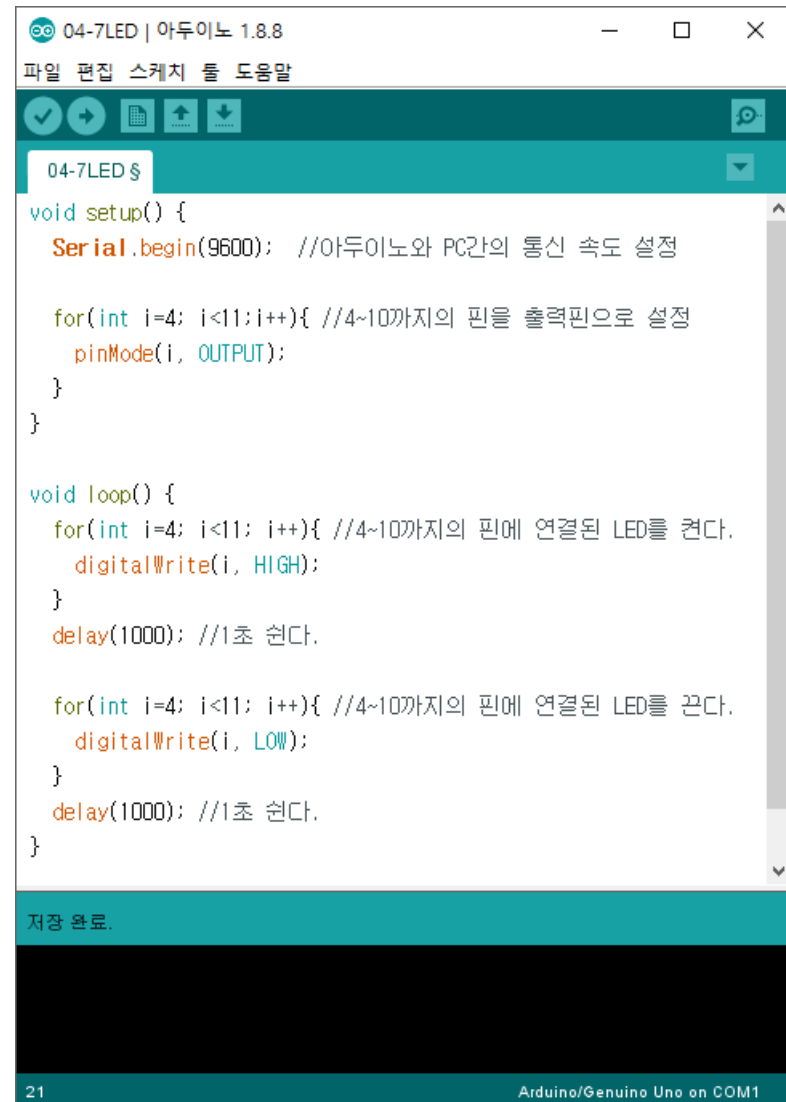
  delay(1000); //1초 켜다.
}
```

3. 스케치 프로그래밍

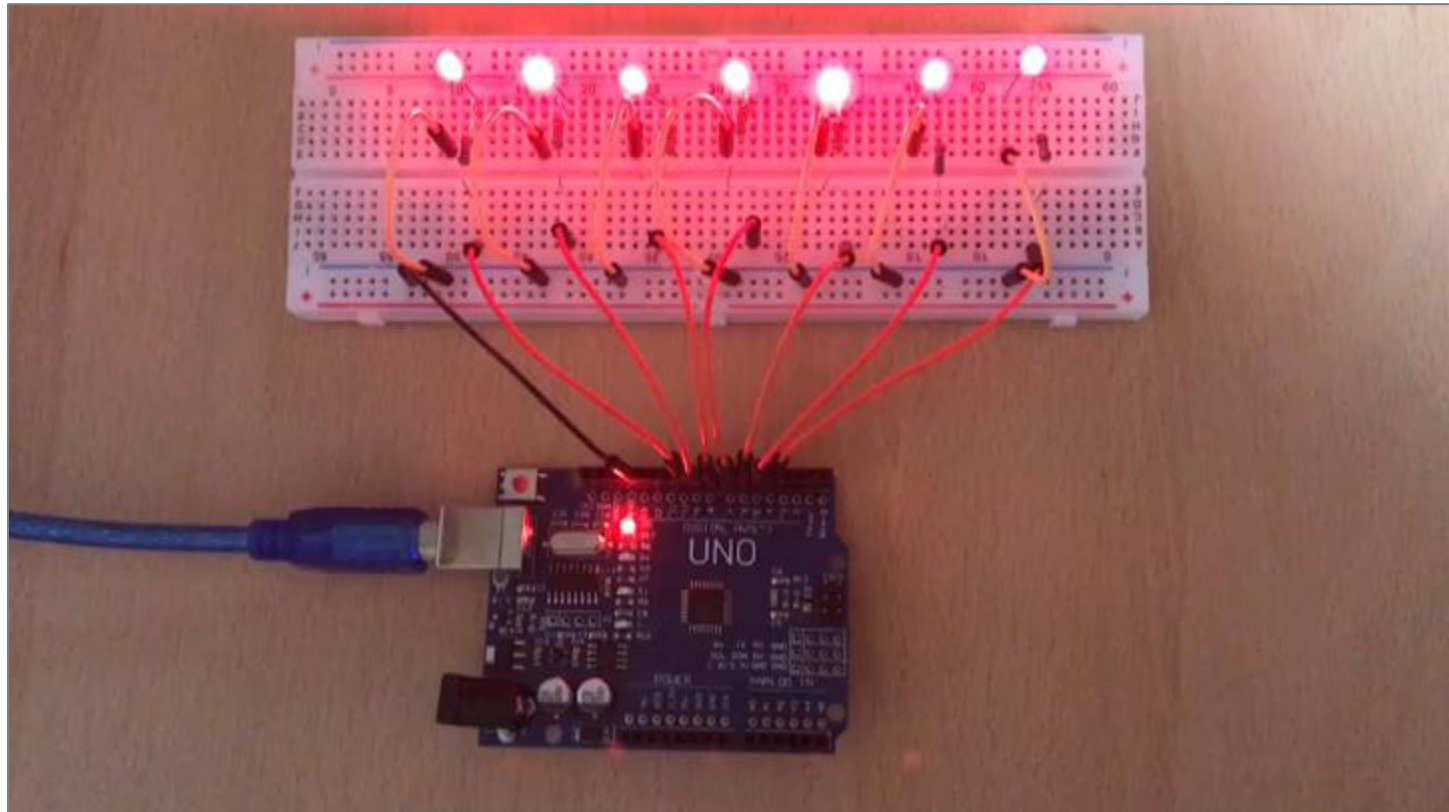
- For문 사용하여 프로그래밍
- For문의 문법

```
for(초기화; 조건; 증감값) {  
    // 명령문 블록  
}
```

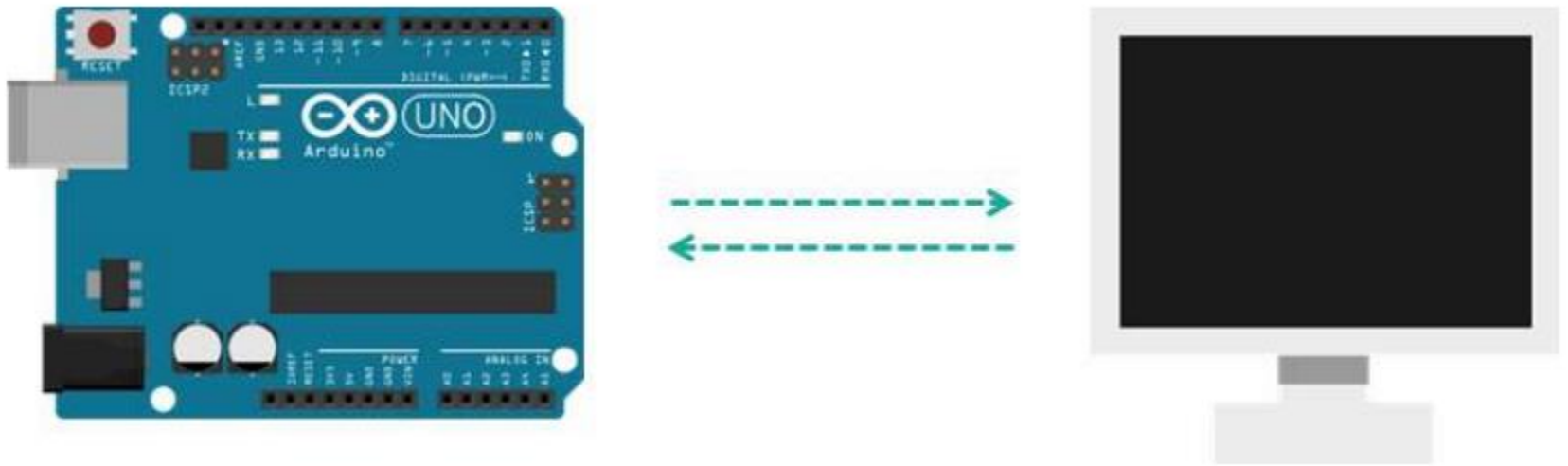
- 초기화는 제 정확히 한 번만 발생
- 루프를 통과 할 때마다 조건 테스트
- 조건 결과가 참(true)이면 증감값과 명령문 블록이 실행
- 조건 결과가 거짓(false)이면 루프 종료



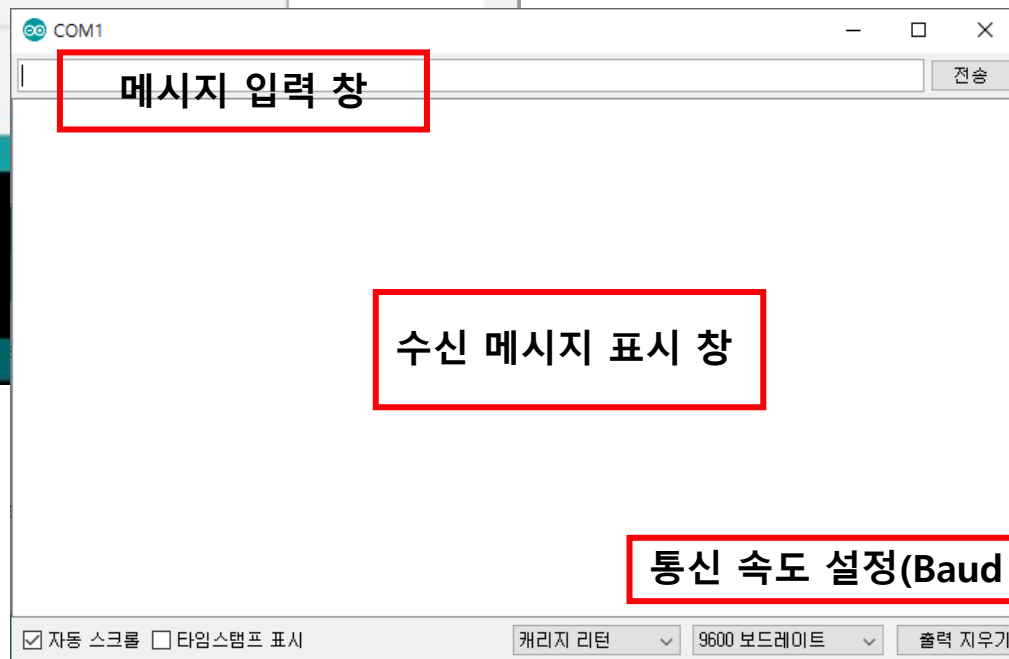
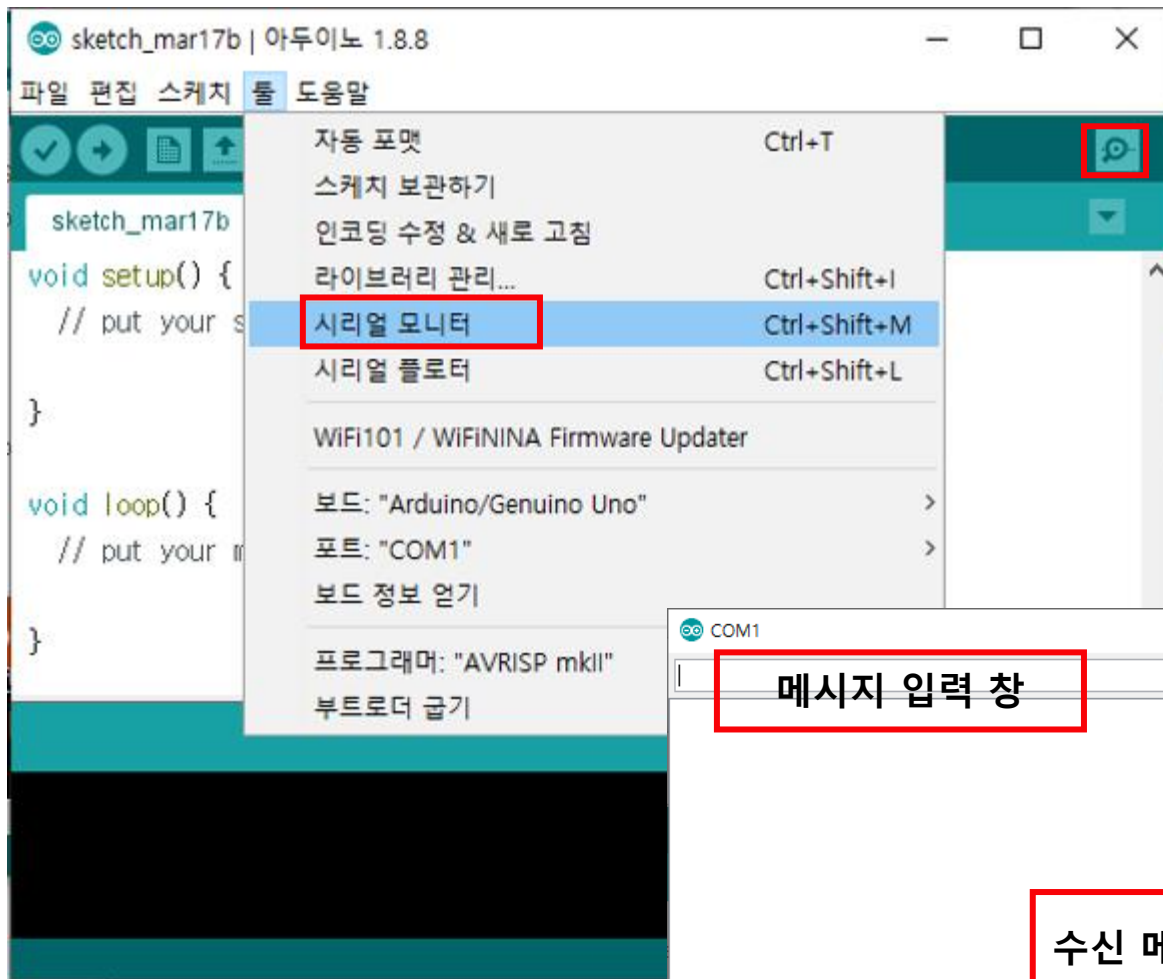
4. 실행



시리얼 모니터



- 컴퓨터와 아두이노 간의 **통신**을 위해 사용하는 툴
- 아두이노 **디버깅**, 데이터 입력을 위해 사용
- 시리얼 모니터를 이용한 디버깅 메시지를 보는 것이 거의 유일한 디버깅 방법



Serial.begin(speed)

- PC와 아두이노가 직렬 데이터 송수신을 위한 속도 설정
- Baud Rate (통신 속도)
 - 시리얼 라인으로 전송되는 데이터 속도 단위
 - Bits-per-second (bps) 단위로 표시

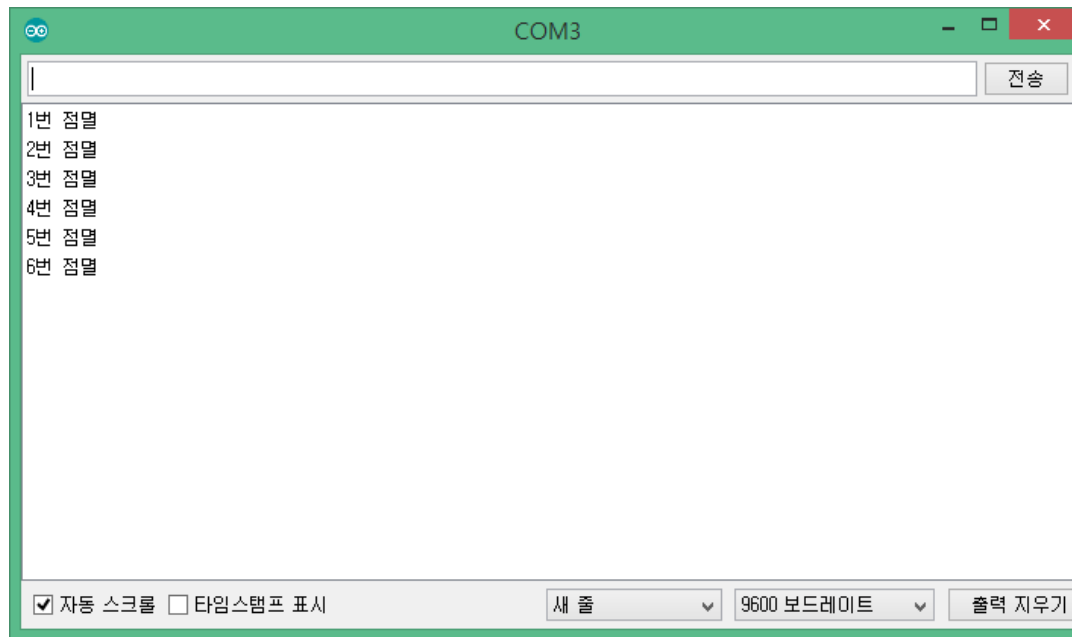
```
void setup() {  
    Serial.begin(9600);  
    //아두이노와 PC의 통신속도를 9600으로 지정  
}
```


Serial.println(value)

- 아두이노에서 PC로 메시지를 보내는 명령어
- **Serial.println(value)**는 메시지 끝에 줄바꿈 기호가 붙어서 보내지고 **Serial.print(value)**는 없이 보내짐
- Serial.begin("Hello, World");

```
void loop() {  
    Serial.println("Hello, World!");  
    //아두이노에서 PC로 문자 보내기  
}
```

- 위험 안내 LED 실습에서 아두이노 LED가 점멸 횟수를 PC에서 다음과 같이 표시하도록 프로그램을 변경 하시오.



```
void setup() {  
    Serial.begin(9600); //아두이노와 PC간의 통신 속도 설정  
  
    for(int i=4; i<11; i++){ //4~10까지의 핀을 출력핀으로 설정  
        pinMode(i, OUTPUT);  
    }  
}  
  
void loop() {  
    for(int i=4; i<11; i++){ //4~10까지의 핀에 연결된 LED를 켜다.  
        digitalWrite(i, HIGH);  
    }  
    delay(1000); //1초 켜다.  
  
    for(int i=4; i<11; i++){ //4~10까지의 핀에 연결된 LED를 끈다.  
        digitalWrite(i, LOW);  
    }  
    delay(1000); //1초 켜다.  
  
    count+=1; //점멸횟수 변수를 증가시킨다.  
    Serial.print(count); //시리얼모니터에 출력한다.  
    Serial.println("번 점멸");  
}
```

<참고> 단위 크기 접두어

인 자	접 두 어	기 호	인 자	접 두 어	기 호
10^1	데 카	da	10^{-1}	데 시	d
10^2	헥 토	h	10^{-2}	센 티	c
10^3	킬 로	k	10^{-3}	밀 리	m
10^6	메 가	M	10^{-6}	마이크로	μ
10^9	기 가	G	10^{-9}	나 노	n
10^{12}	테 라	T	10^{-12}	피 코	p
10^{15}	페 타	P	10^{-15}	펨 토	f
10^{18}	엑 사	E	10^{-18}	아 토	a
10^{21}	제 타	Z	10^{-21}	zepto	z
10^{24}	요 타	Y	10^{-24}	욕 토	y