

3주차

- 아두이노 기초 이론
- Fritzing을 사용한 LED 회로 제작, 소스 코드 보기
 - 내부 LED ON/OFF
 - 1개의 LED ON/OFF
 - 2개의 LED ON/OFF 순환
 - 3개의 LED ON/OFF 순환
 - 함수(function) 사용하기
 - 3개의 LED 랜덤으로 ON/OFF하기
- 하드웨어 구성 및 소스 코드 실행

NodeMCU 실습 키트

- NodeMCU v2 CP212 1개
- 브래드 보드 1개
- LED : R, G, B, W
- 온습도 센서(DHT11) 1개
- 저항 220 Ω
- LCD(16x16, I2C) 1개
- 점퍼 케이블
- 부품 보관함

아두이노 부품 회로 구성



디지털 입출력 장치

디지털 입력	디지털 출력
<div data-bbox="114 544 387 875"><p>Pushbutton</p></div> <div data-bbox="402 544 675 875"><p>Slideswitch</p></div> <div data-bbox="690 544 963 875"><p>PIR Sensor</p></div> <div data-bbox="978 544 1251 875"><p>DIP Switch DPST</p></div>	<div data-bbox="1284 544 1556 875"><p>LED</p></div> <div data-bbox="1572 544 1844 875"><p>LED RGB</p></div> <div data-bbox="1860 544 2132 875"><p>DC 모터</p></div> <div data-bbox="2147 544 2420 875"><p>Hobby Gearmotor</p></div>
	<div data-bbox="1284 919 1556 1258"><p>7 Segment Display</p></div> <div data-bbox="1572 919 1844 1258"><p>LCD 16 x 2</p></div>

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

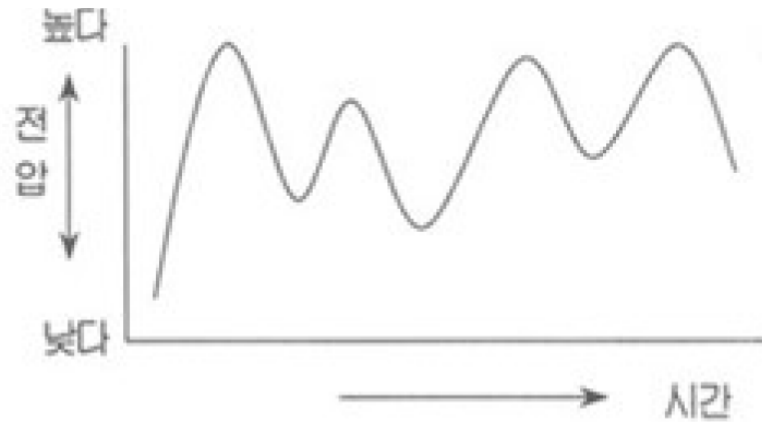
- **아날로그**

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

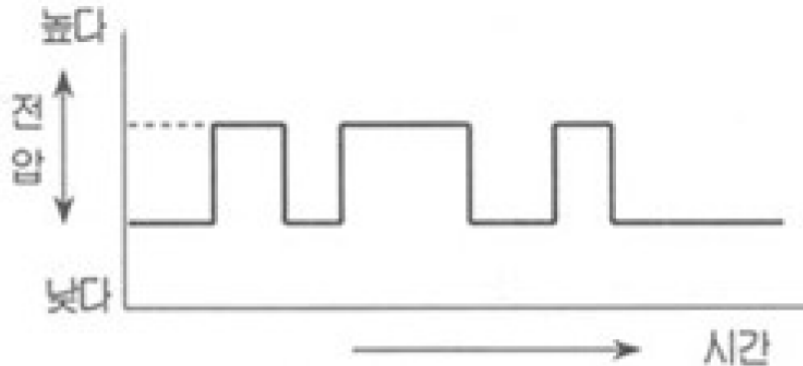
- **디지털**

- 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

이렇게 하면, 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**

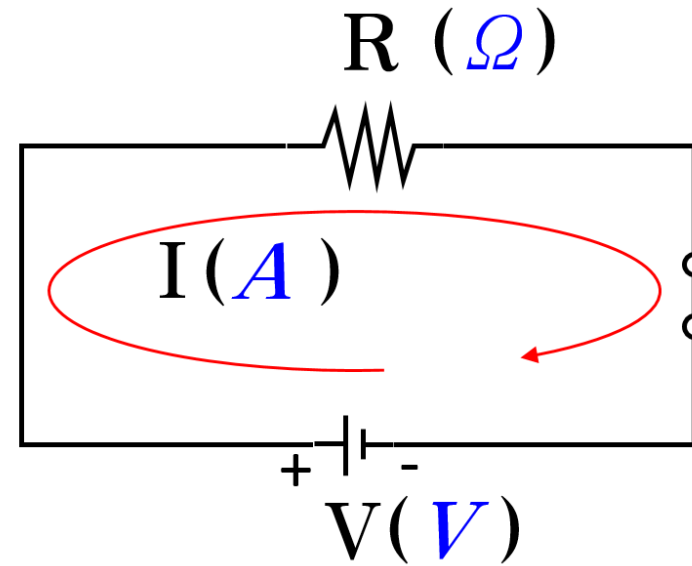
- 0 ~ 255 또는 0 ~ 1023 사이의 값을 입/출력 한다.
- `analogWrite()`, `analogRead()`

옴의 법칙 (Ohm's Law)

- 전기회로에 흐르는 전류는 전압에 비례하고 저항에 반비례

$$V = IR$$

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



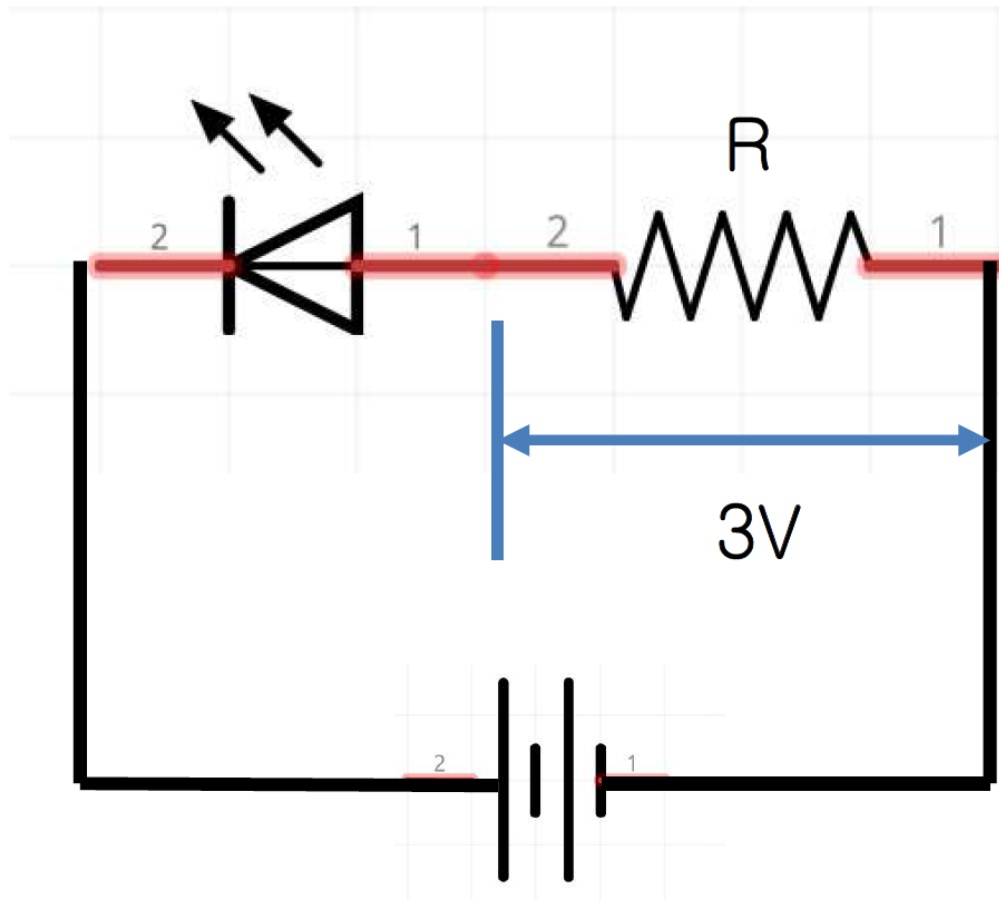
전압(V : Volt): 전류를 흐르게 하는 전기적인 압력, 단위 **볼트 [V]**

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 Ω이상인 저항을
사용하면 된다.

보통 LED회로에서는 220
Ω을 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 $\pm 5\%$

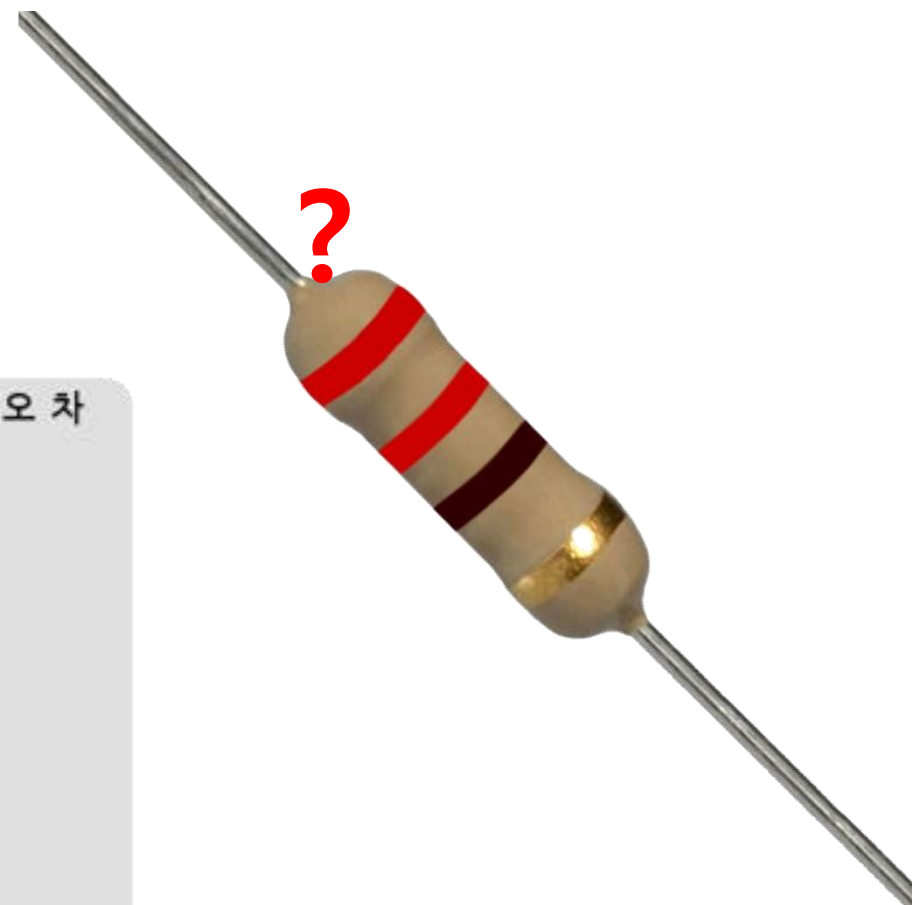
5색 코드 저항



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

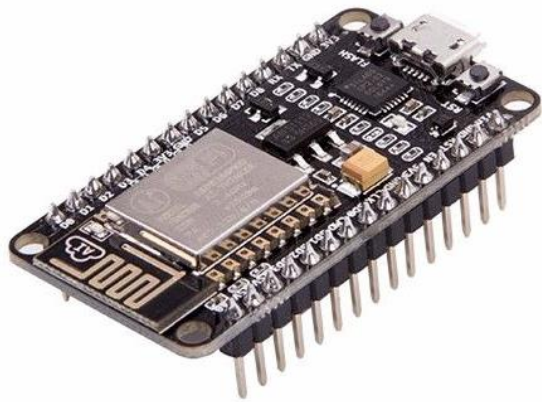
흑갈적등황녹청자회백

	숫 자	요단위	오 차
검정색	0	1	
갈 색	1	10	
빨강색	2	100	
주황색	3	1K	
노란색	4	10K	
초록색	5	100K	
파란색	6	1M	
보라색	7	10M	
회 색	8	100M	
하얀색	9	1G	
금 색			5%
은 색			10%
없음(무)			20%



NodeMCU LED 제어 기초

- 준비물



NodeMCU



점프 케이블

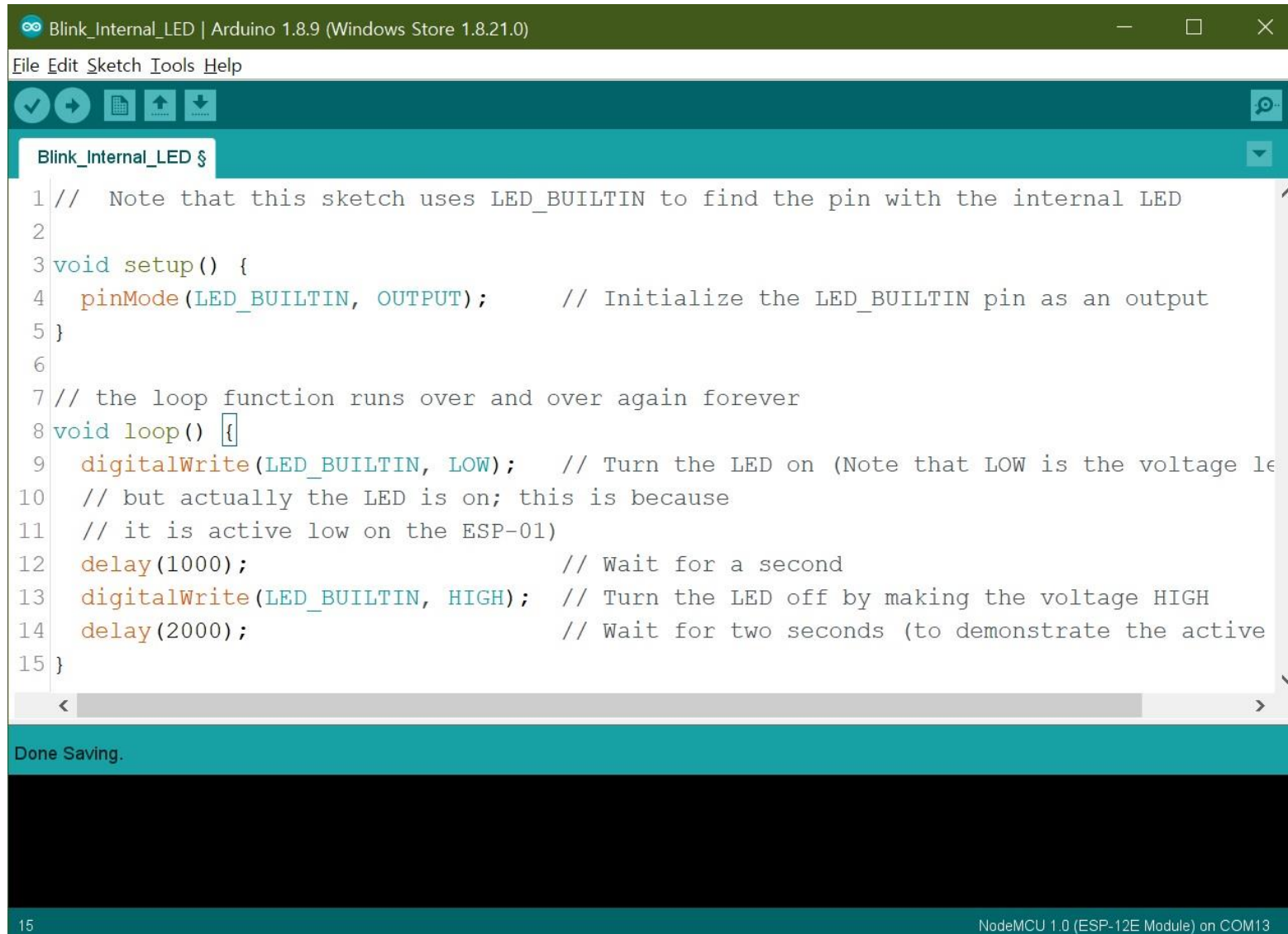


LED



저항(220 Ω)

내부 LED ON/OFF : LED_BUILTIN (D0)



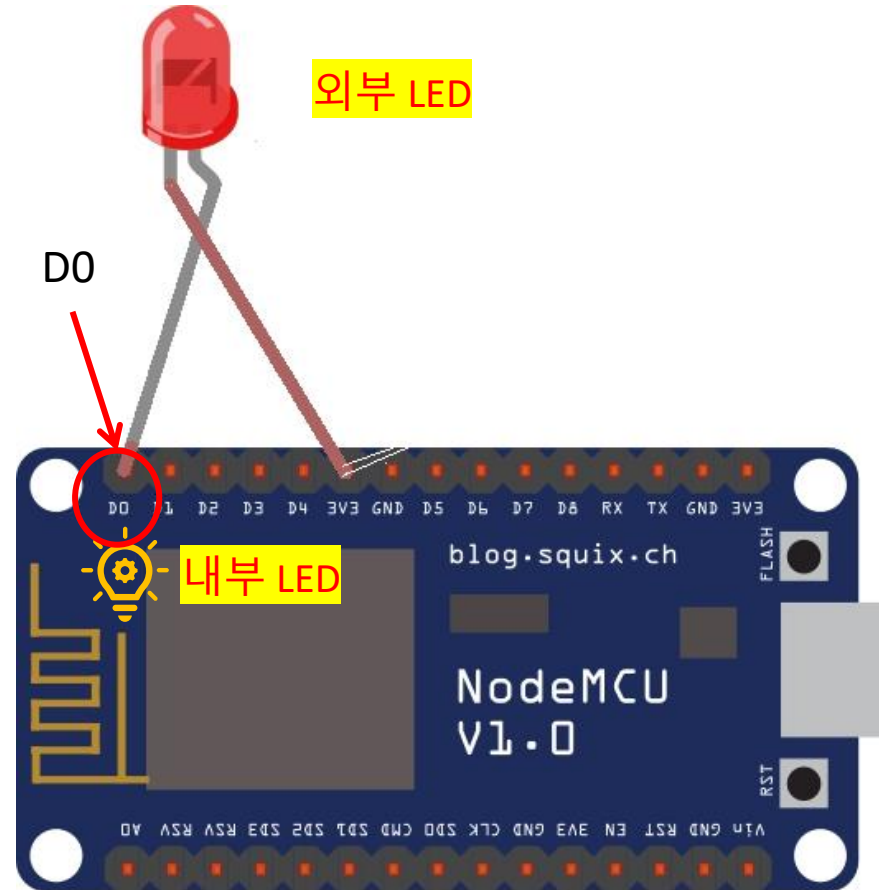
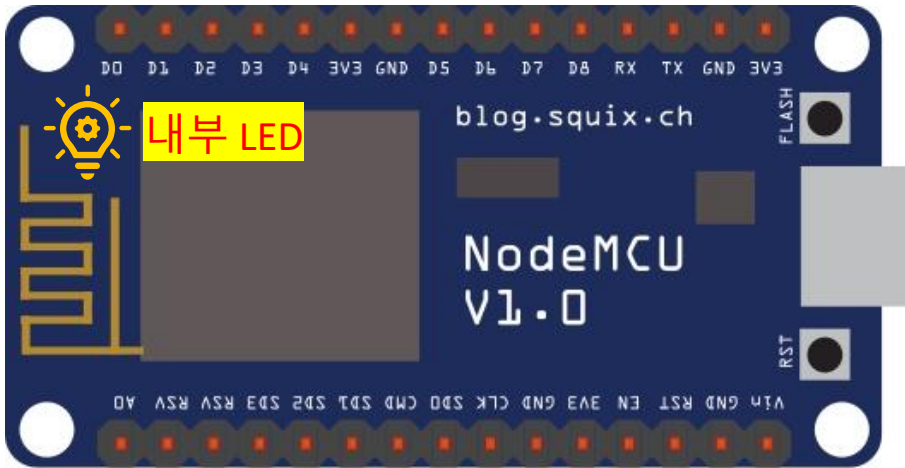
The screenshot shows the Arduino IDE interface. The title bar reads "Blink_Internal_LED | Arduino 1.8.9 (Windows Store 1.8.21.0)". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for saving, running, and other functions. The main text area displays the following code:

```
1 // Note that this sketch uses LED_BUILTIN to find the pin with the internal LED
2
3 void setup() {
4   pinMode(LED_BUILTIN, OUTPUT);    // Initialize the LED_BUILTIN pin as an output
5 }
6
7 // the loop function runs over and over again forever
8 void loop() {
9   digitalWrite(LED_BUILTIN, LOW);  // Turn the LED on (Note that LOW is the voltage level
10  // but actually the LED is on; this is because
11  // it is active low on the ESP-01)
12   delay(1000);                    // Wait for a second
13   digitalWrite(LED_BUILTIN, HIGH); // Turn the LED off by making the voltage HIGH
14   delay(2000);                    // Wait for two seconds (to demonstrate the active
15 }
```

Below the code editor, a status bar shows "Done Saving." and a dark area for the serial monitor. At the bottom, the footer indicates "15" on the left and "NodeMCU 1.0 (ESP-12E Module) on COM13" on the right.

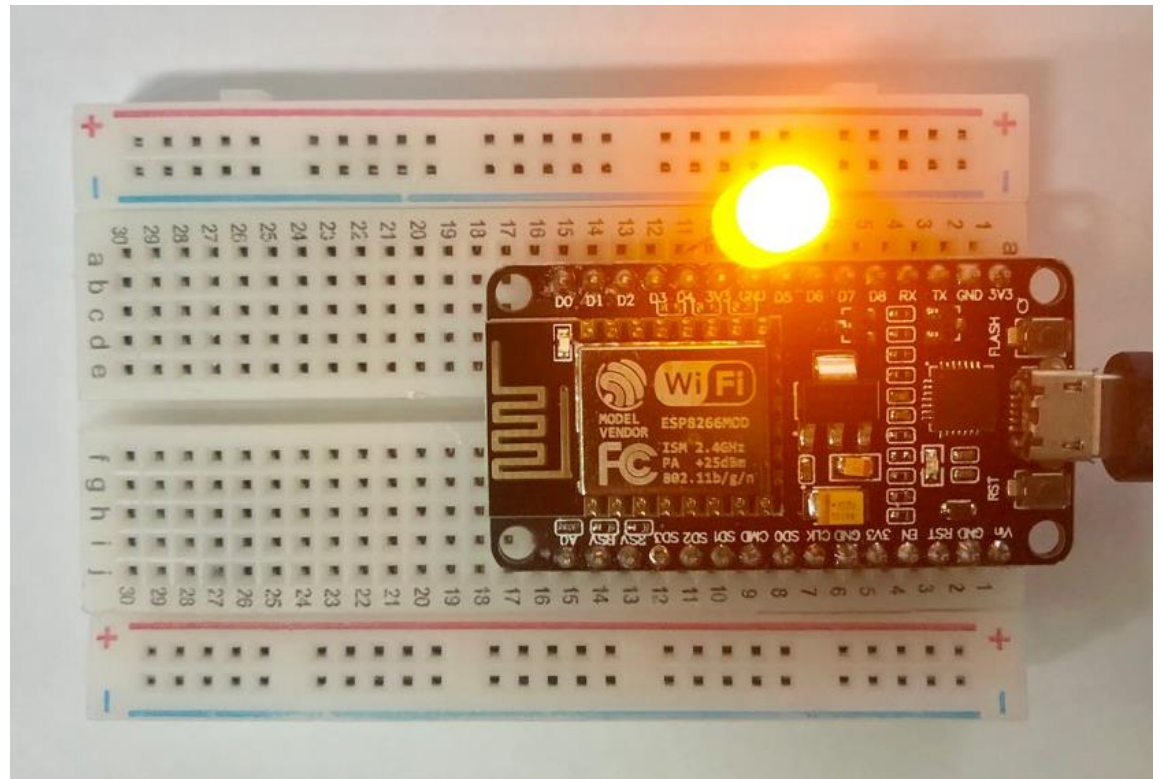
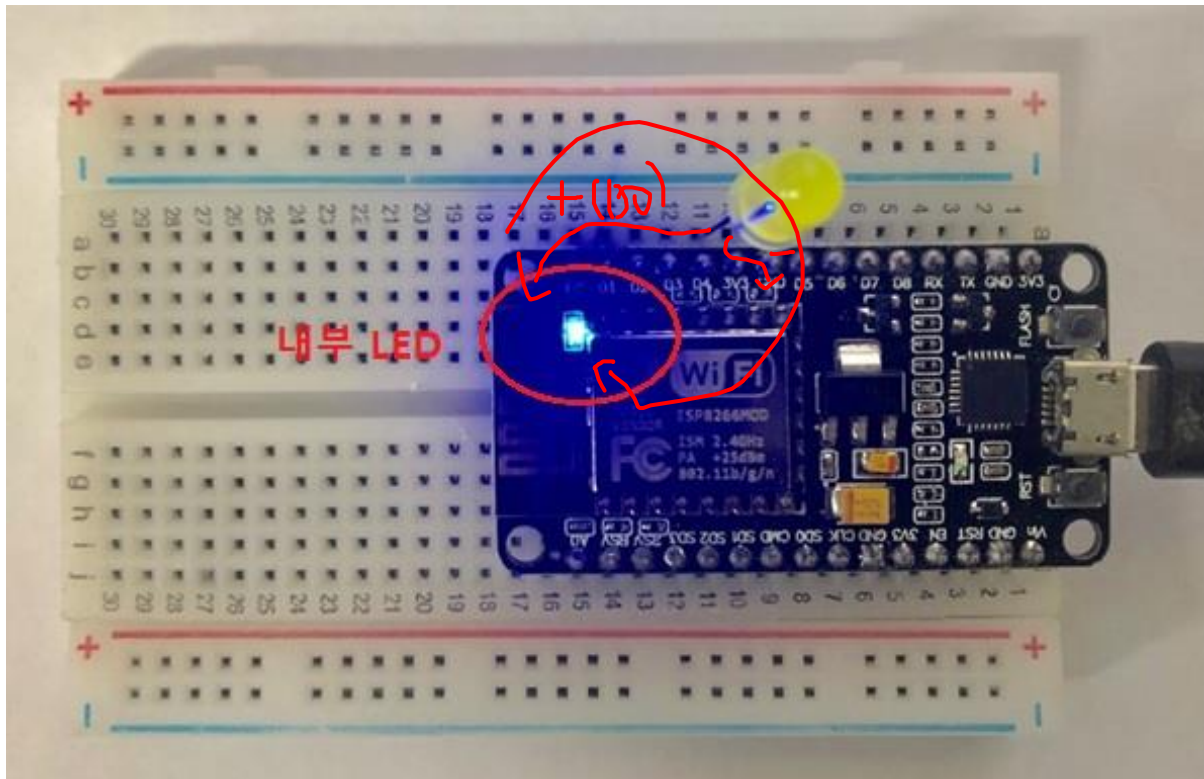
회로도 그리기 : Fritzing

- 내부 LED : LED_BUILTIN은 D0 핀



내부 LED 작동 확인 : D0(GPIO4) 핀

- 현재 저항(220 Ω)이 없기 때문에 오래 켜두면 과부하로 LED가 뜨거워짐(주의!!!)



NodeMCU LED 저항 계산

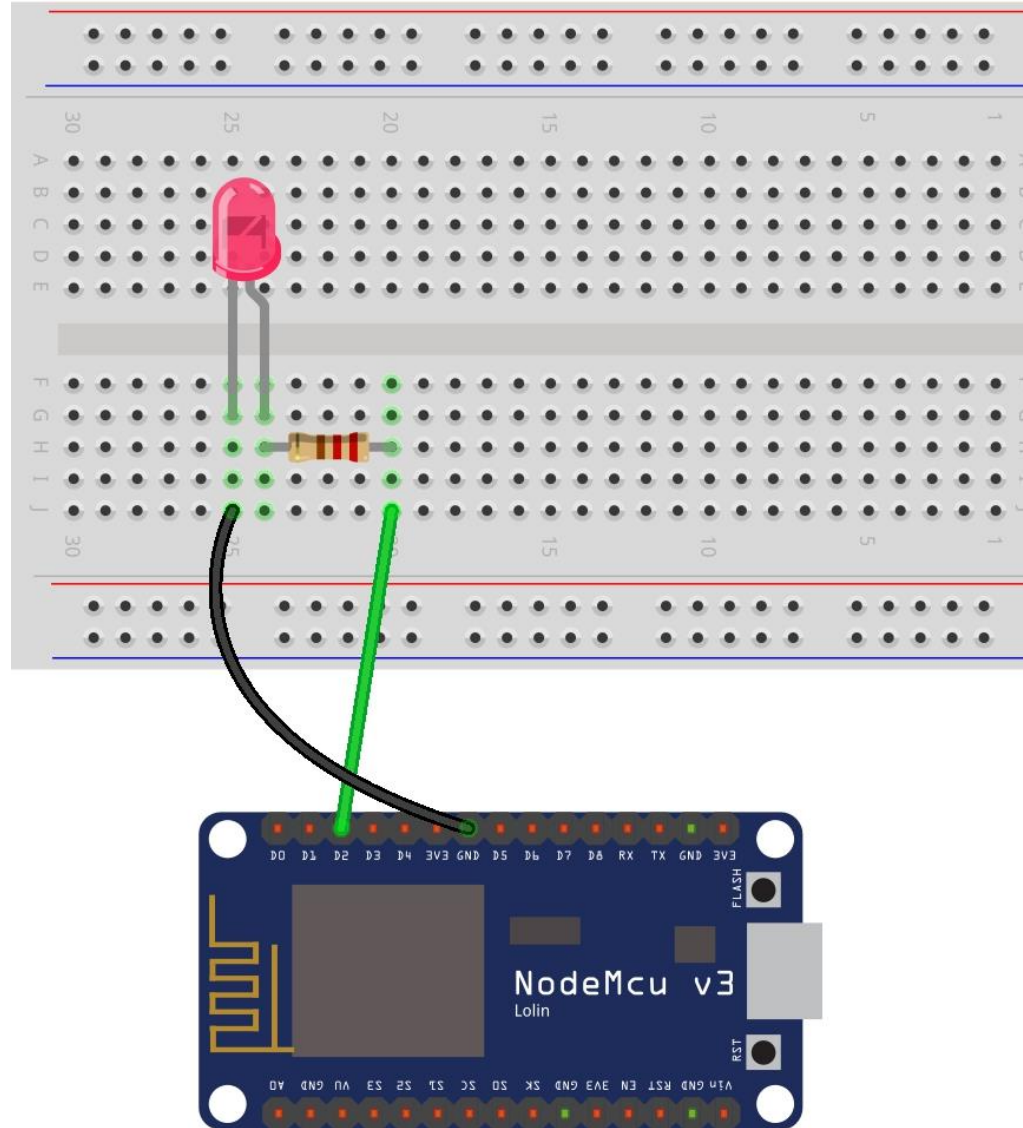
$$V = IR$$

$$R = V / I$$

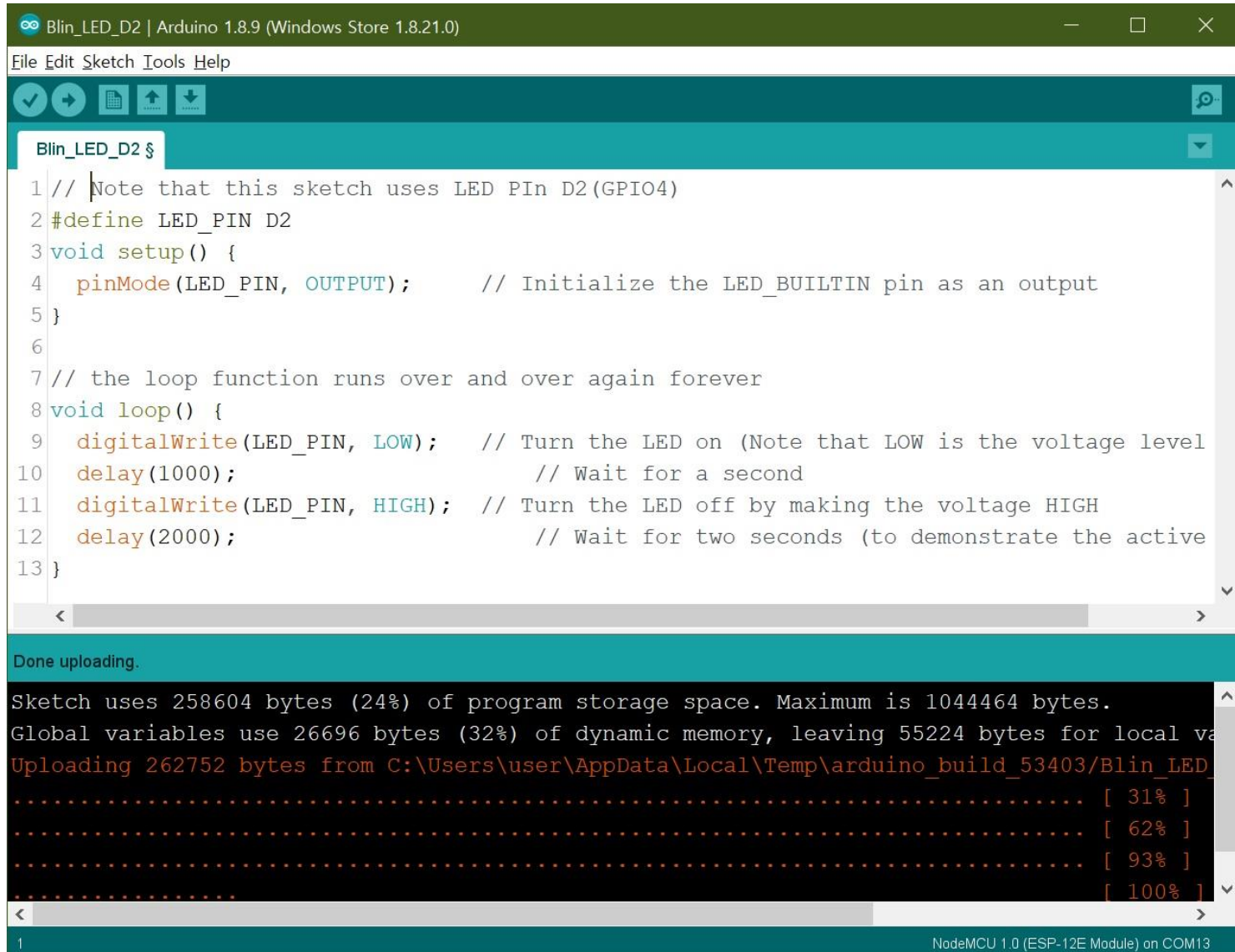
$$R = 3.3 \text{ V} / 0.02 \text{ A} = 165 \Omega$$

- 165 Ω 이상의 저항 사용,
- 구하기 쉬운 저항 220 Ω 많이 사용

1개 LED 제어 : D2(GPIO4) 핀



스케치 : D2(GPIO4) 핀



```
Blin_LED_D2 | Arduino 1.8.9 (Windows Store 1.8.21.0)
File Edit Sketch Tools Help

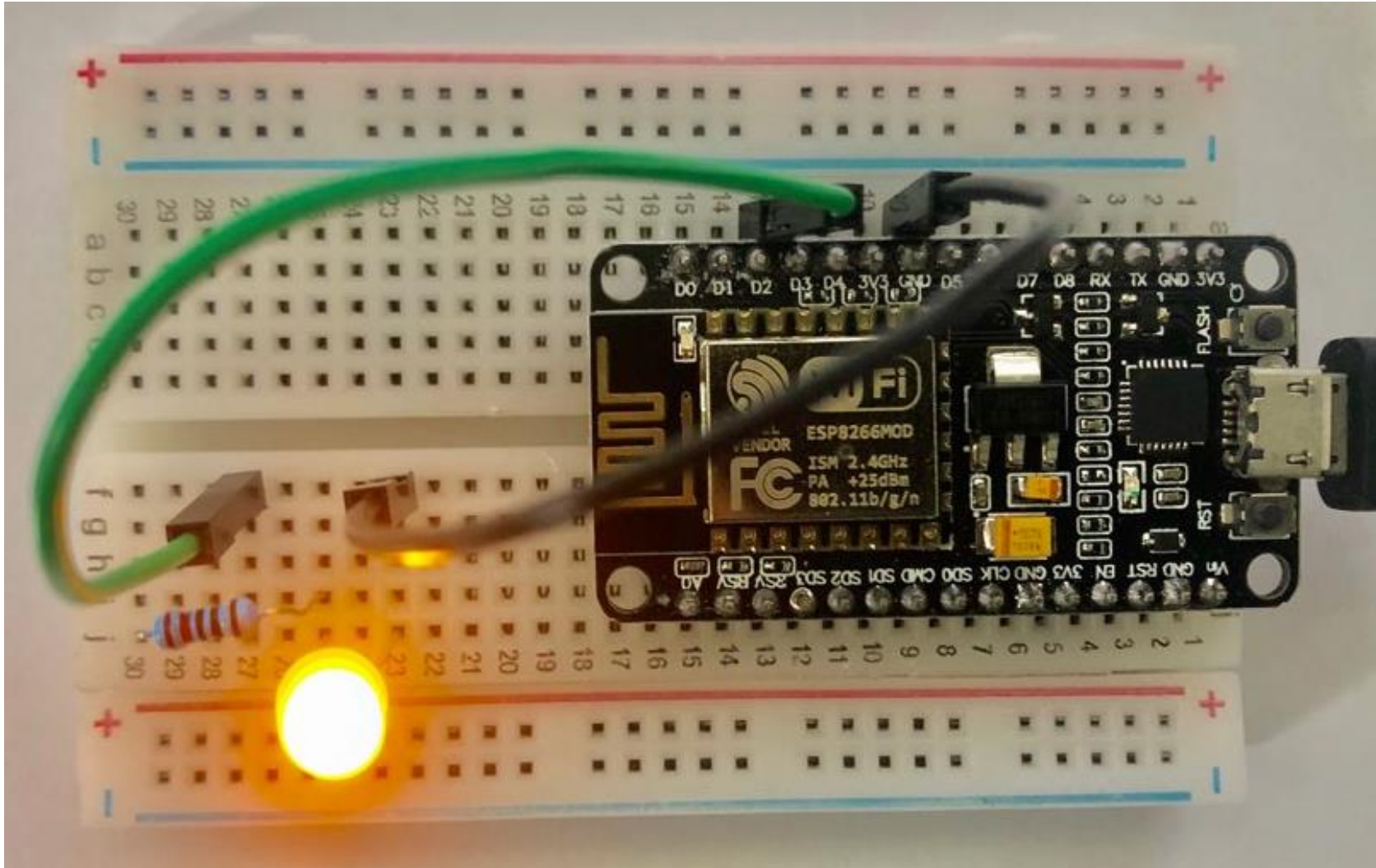
Blin_LED_D2 $
1 // Note that this sketch uses LED PIN D2 (GPIO4)
2 #define LED_PIN D2
3 void setup() {
4   pinMode(LED_PIN, OUTPUT);    // Initialize the LED_BUILTIN pin as an output
5 }
6
7 // the loop function runs over and over again forever
8 void loop() {
9   digitalWrite(LED_PIN, LOW);  // Turn the LED on (Note that LOW is the voltage level
10  delay(1000);                  // Wait for a second
11  digitalWrite(LED_PIN, HIGH); // Turn the LED off by making the voltage HIGH
12  delay(2000);                  // Wait for two seconds (to demonstrate the active
13 }
```

Done uploading.

Sketch uses 258604 bytes (24%) of program storage space. Maximum is 1044464 bytes.
Global variables use 26696 bytes (32%) of dynamic memory, leaving 55224 bytes for local variables.
Uploading 262752 bytes from C:\Users\user\AppData\Local\Temp\arduino_build_53403\Blin_LED_D2 to COM13
..... [31%]
..... [62%]
..... [93%]
..... [100%]

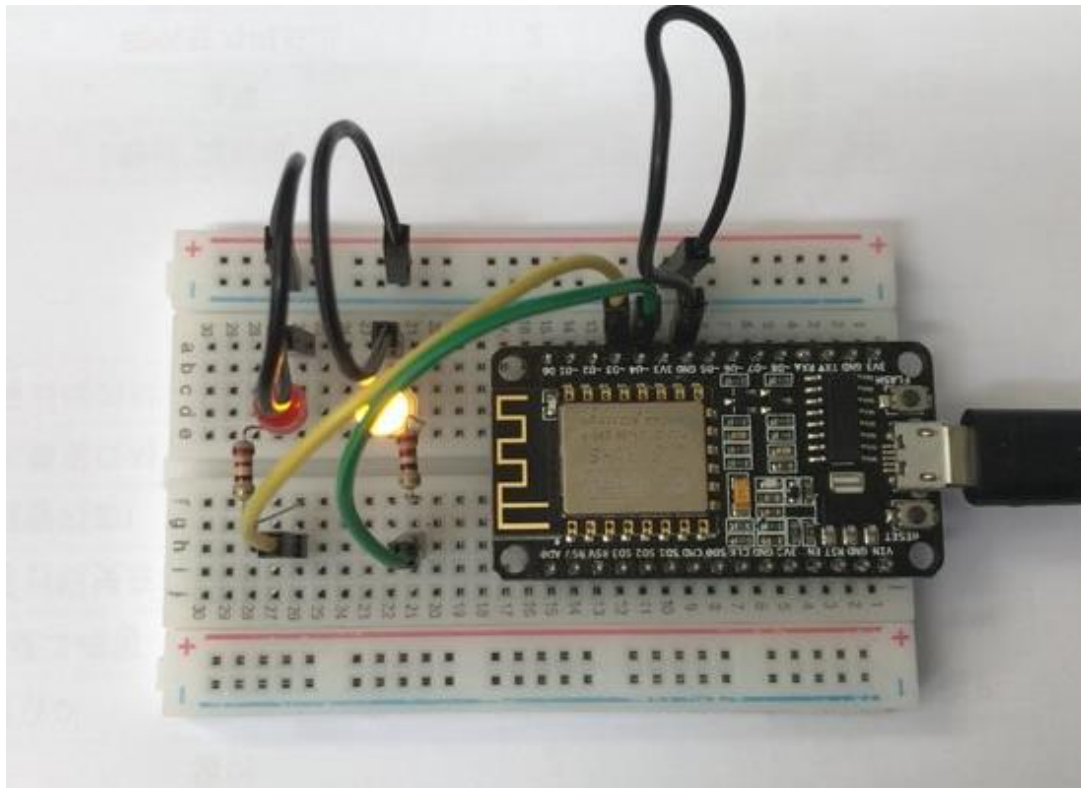
1 NodeMCU 1.0 (ESP-12E Module) on COM13

LED 작동 : D2 핀, 220 Ω 저항

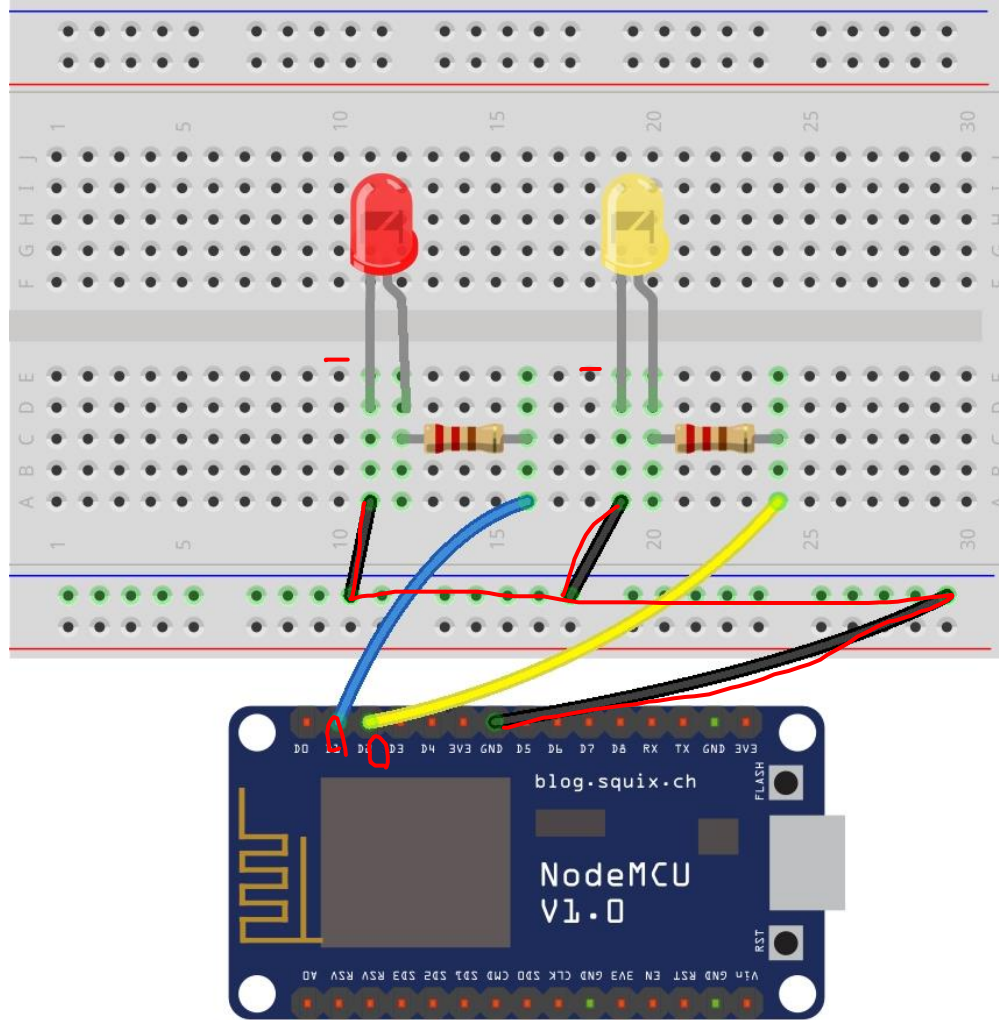


2개 LED ON/OFF

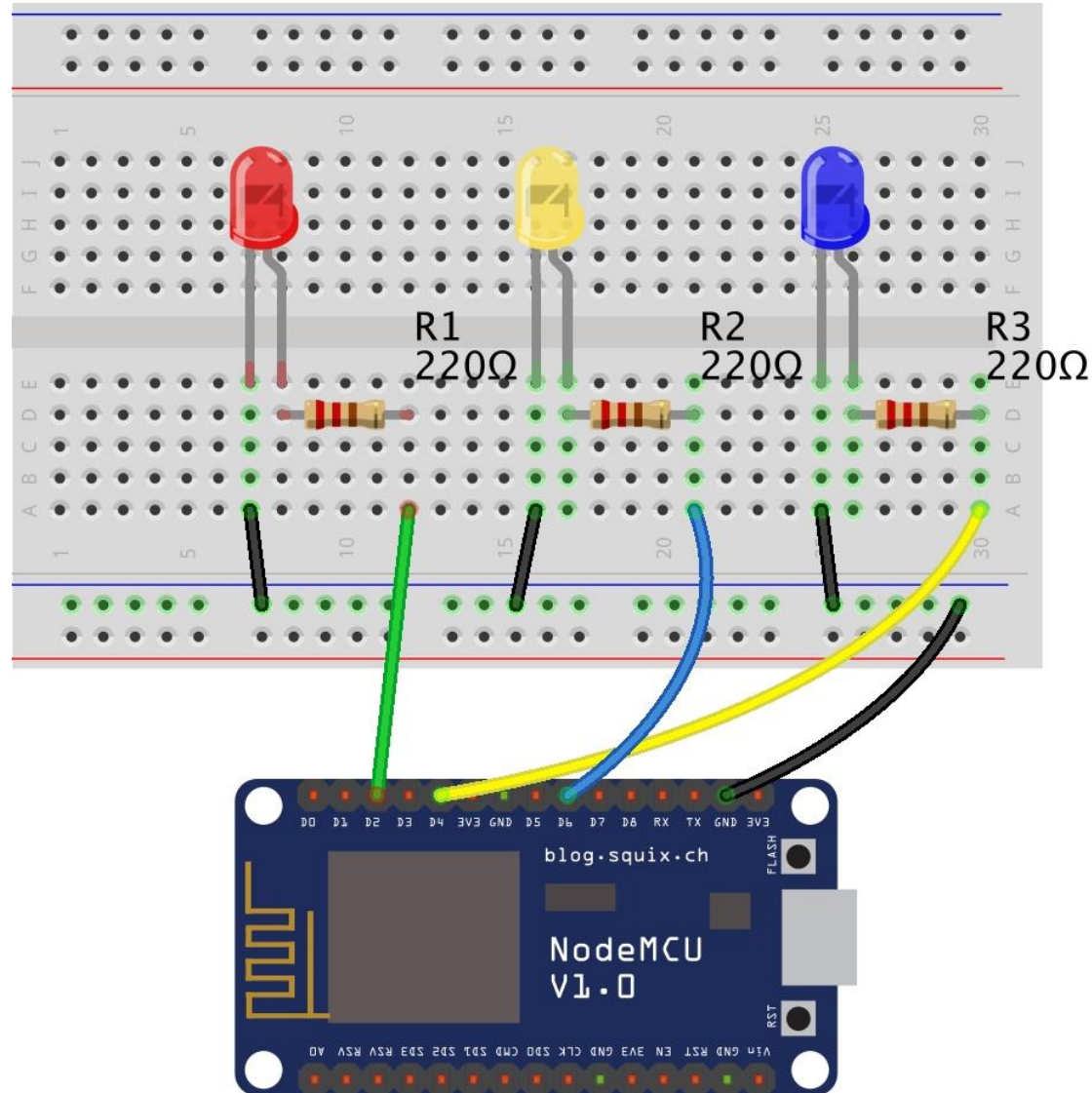
- 2개의 LED를 사용하여 0.5초 간격으로 blink 하기(빨강->노랑->빨강->노랑....)
- `digitalWrite()`, 220 Ω 저항



Fritzing 회로 작성



실습 과제 : 3개의 LED ON/OFF 하기



- Fritzing 회로 그리기
- H/W 회로 구성
- 스케치 코드 작성, 컴파일, 업로드, 결과 확인
- 함수(function)를 사용한 코드 작성하기

```
void blink_LED(int pinNo, int delayTime) {  
    // pinNO : 핀번호, delayTime : delay 시간  
    .....  
}
```

- LED가 랜덤하게 ON/OFF 하기

- 아두이노 random() 함수 사용

- 참고 : <https://www.arduino.cc/reference/ko/language/functions/random-numbers/random/>

*생각해보기 : 코딩 때 random() 함수로 생성된 0,1,2 값을 D1, D2, D3로 바꿔줘야 한다.