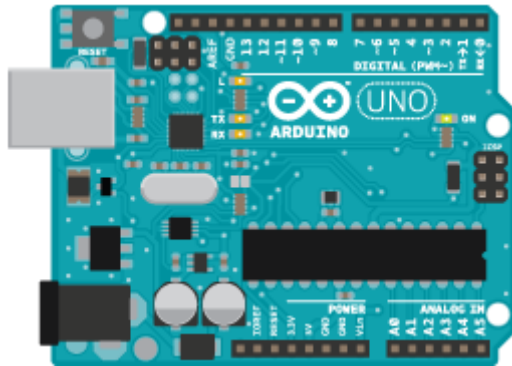


프로젝트1 현관등 만들기



미니 프로젝트(현관등 만들기)

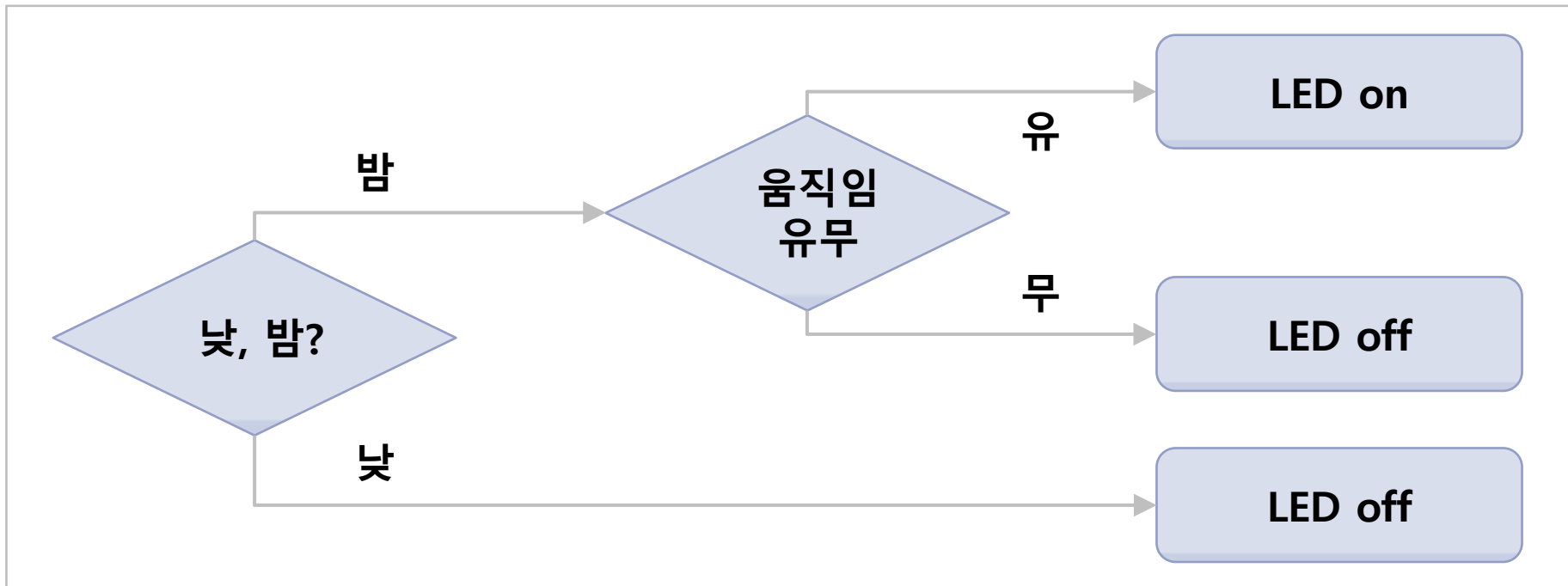
■ 아날로그 신호 입 · 출력

낮

- 움직임의 유무와 관계 없이 → LED off

밤

- 움직임 있음 → LED on
- 움직임 없음 → LED off

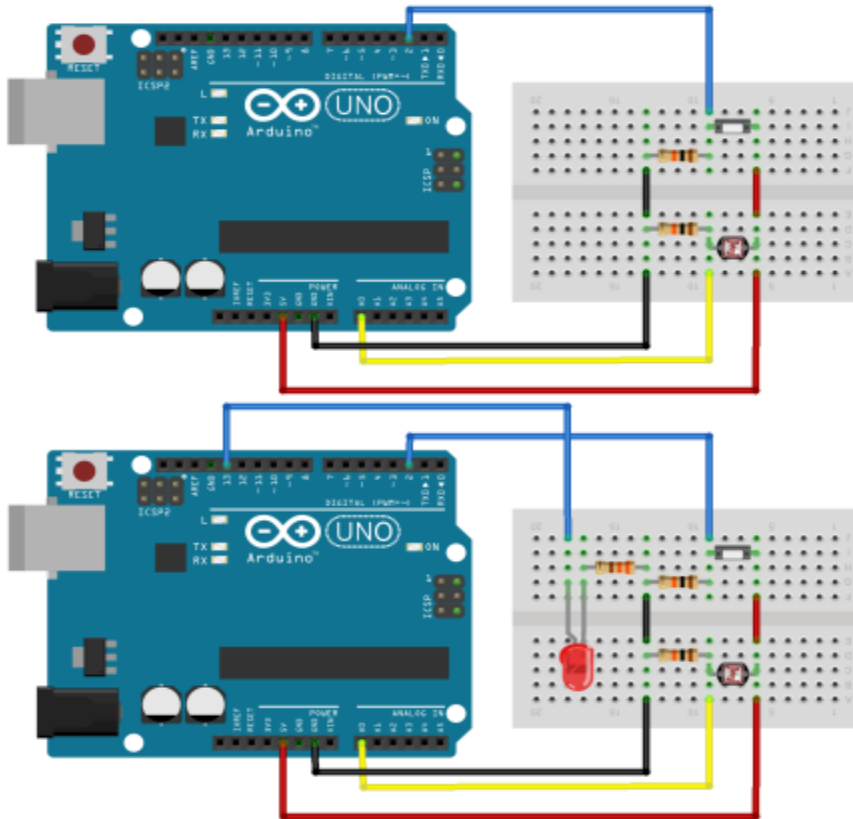


미니 프로젝트(현관등 만들기)

■ 아날로그 신호 입 · 출력

▶ 회로 구성

- Cds → A0핀
- 푸시 스위치 → 2번 핀



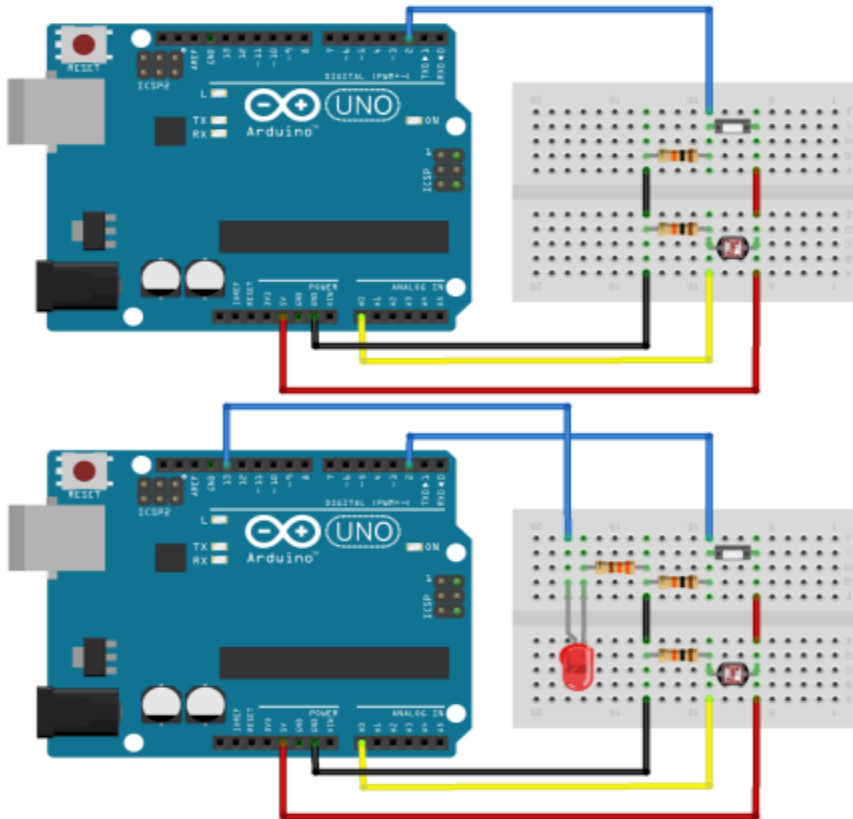
```
void setup() {  
  Serial.begin(9600);  
  pinMode(13, OUTPUT);  
  pinMode(2, INPUT);  
}  
  
void loop() {  
  int sensorValue = analogRead(A0);  
  if( sensorValue < 300 ) {  
    if( digitalRead(2) == HIGH ) {  
      digitalWrite(13, HIGH);  
    } else {  
      digitalWrite(13, LOW);  
    }  
  } else {  
    digitalWrite(13, LOW);  
  }  
  Serial.println(sensorValue);  
}
```

미니 프로젝트(현관등 만들기)

■ 아날로그 신호 입 · 출력

▶ 회로 구성

- Cds → A0핀
- 푸시 스위치 → 2번 핀



```
void setup() {  
    Serial.begin(9600);  
    pinMode(13, OUTPUT);  
    pinMode(2, INPUT);  
}  
  
void loop() {  
    int sensorValue = analogRead(A0);  
    if( sensorValue < 300 ) {  
        int motion = digitalRead(2);  
        Serial.println(motion);  
        if( motion == HIGH ) {  
            digitalWrite(13, HIGH);  
        } else {  
            digitalWrite(13, LOW);  
        }  
    } else {  
        digitalWrite(13, LOW);  
    }  
    Serial.println(sensorValue);  
}
```

미니 프로젝트(현관등 만들기)

조건 1	조건 2	LED
밤	움직임 있음	on
밤	움직임 없음	off
낮	움직임 있음	off
낮	움직임 없음	off

if

밤
and
움직임 있음

}

else

미니 프로젝트(현관등 만들기)

■ 자동 점멸 현관등 동작과 유사함

```
void setup() {  
  Serial.begin(9600);  
  pinMode(13, OUTPUT);  
  pinMode(2, INPUT);  
}  
  
void loop() {  
  int sensorValue = analogRead(A0);  
  if( sensorValue < 300 && digitalRead(2) ==HIGH) {  
    digitalWrite(13, HIGH);  
  } else {  
    digitalWrite(13, LOW);  
  }  
  
  Serial.println(sensorValue);  
}
```

■ 자동 점멸 현관등 동작과 유사함

```
void setup() {  
  Serial.begin(9600);  
  pinMode(13, OUTPUT);  
}  
  
void loop() {  
  int sensorValue = analogRead(A0);  
  if( sensorValue < 300 && digitalRead(2) ) {  
    digitalWrite(13, HIGH);  
    delay(3000);  
  } else {  
    digitalWrite(13, LOW);  
  }  
  
  Serial.println(sensorValue);  
}
```



node mcu

NodeMCU

ESP-12E development board

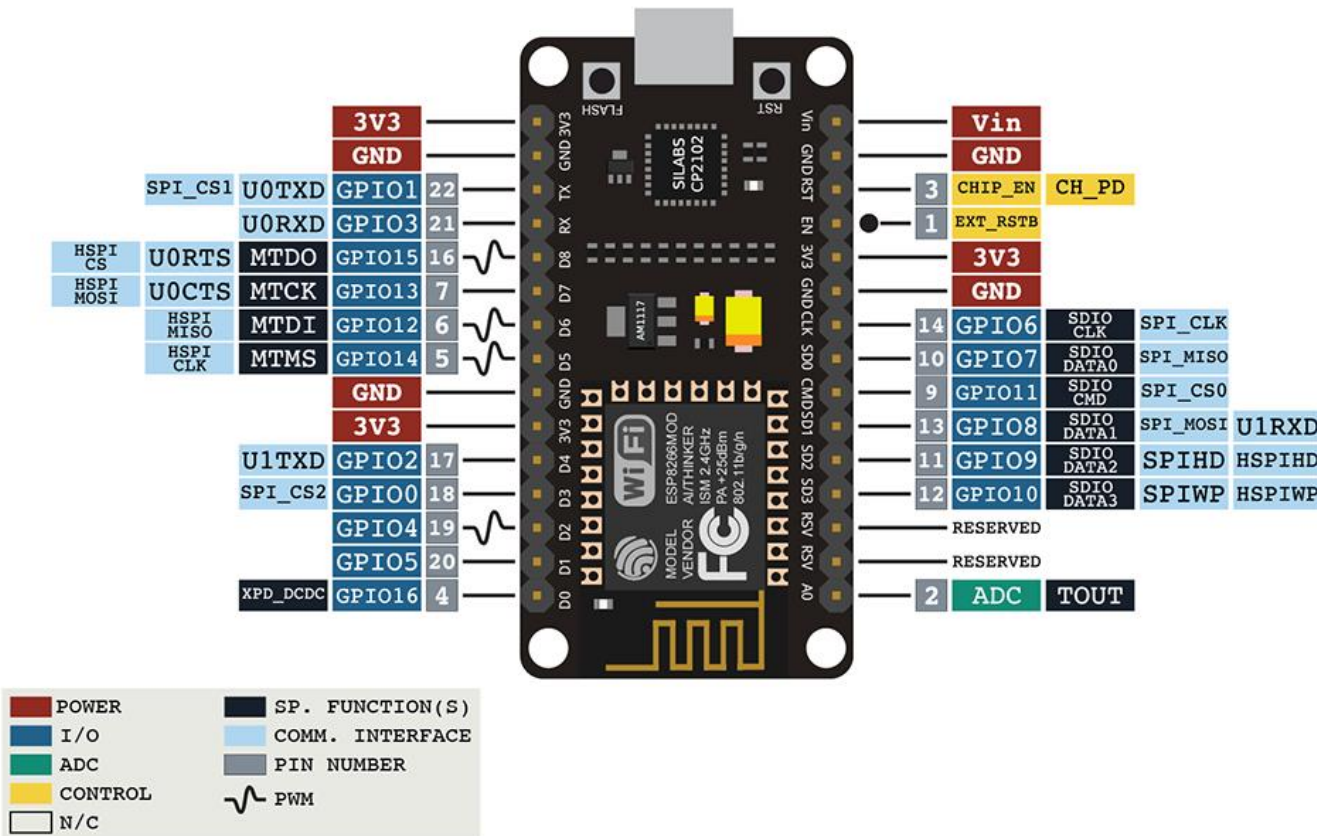
- ESP-12E + (SiLabs)CP2102
- 구동 전압 : 3.3V
- 구동 전류 : 80mA
- NCP1117, NCV1117
 - 1.0[A]
- 참고 사이트
 - <https://nodemcu.readthedocs.io/>
 - Nodemcu.com

ESP-12E DEVELOPMENT BOARD

PINOUT

NOTES:

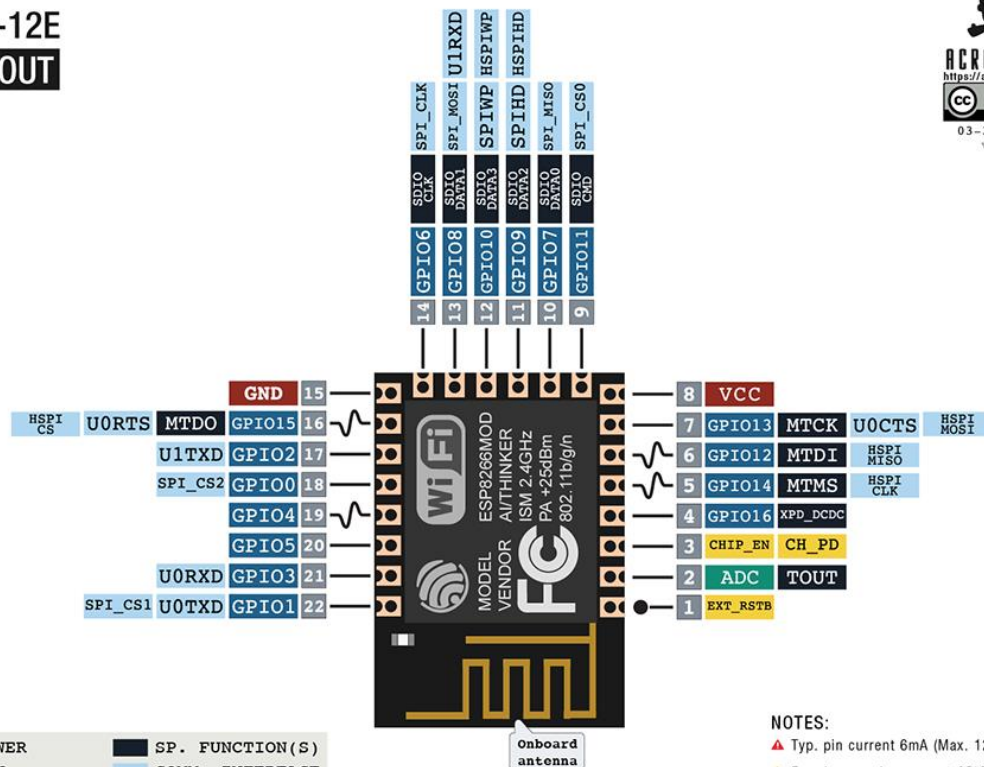
- ▲ Typ. pin current 6mA (Max. 12mA)
- ▲ For sleep mode, connect GPIO16 and EXT_RSTB. On wakeup, GPIO16 will output LOW for system reset.
- ▲ On boot/reset/wakeup, keep GPIO15 LOW and GPIO2 HIGH.



NodeMCU

► **ESP-12E**

- ESP8266-12의 업그레이드 버전.
- 32-bit RISC CPU:Tensilica Xtensa L106 running at 80MHz
- 64-KiB of instruction RAM, 96 KiB of data RAM
- 10-bit ADC
- WiFi 2.4GHz, WPA 지원, Station/softAP/SoftAp+station
- 대기 모드/슬립 모드 가능
- ESP-12 모듈과의 차이점
 - GPIO Pin 2개 추가, SPI pin 추가
 - EMI 차단 기능 향상



NOTES:

- ▲ Typ. pin current 6mA (Max. 12mA)
- ▲ For sleep mode, connect GPIO16 and EXT_RSTB. On wakeup, GPIO16 will output LOW for system reset.
- ▲ On boot/reset/wakeup, keep GPIO15 LOW and GPIO2 HIGH.



■ NodeMCU

- ▶ Board Name : Amica
 - ▶ Open Source IoT Platform
 - **Install CP2102 driver**
 - Use 9600 Baud Rate
 - Connect Wi-Fi and Enjoy
- VIN + 5V Recommended +10V Max



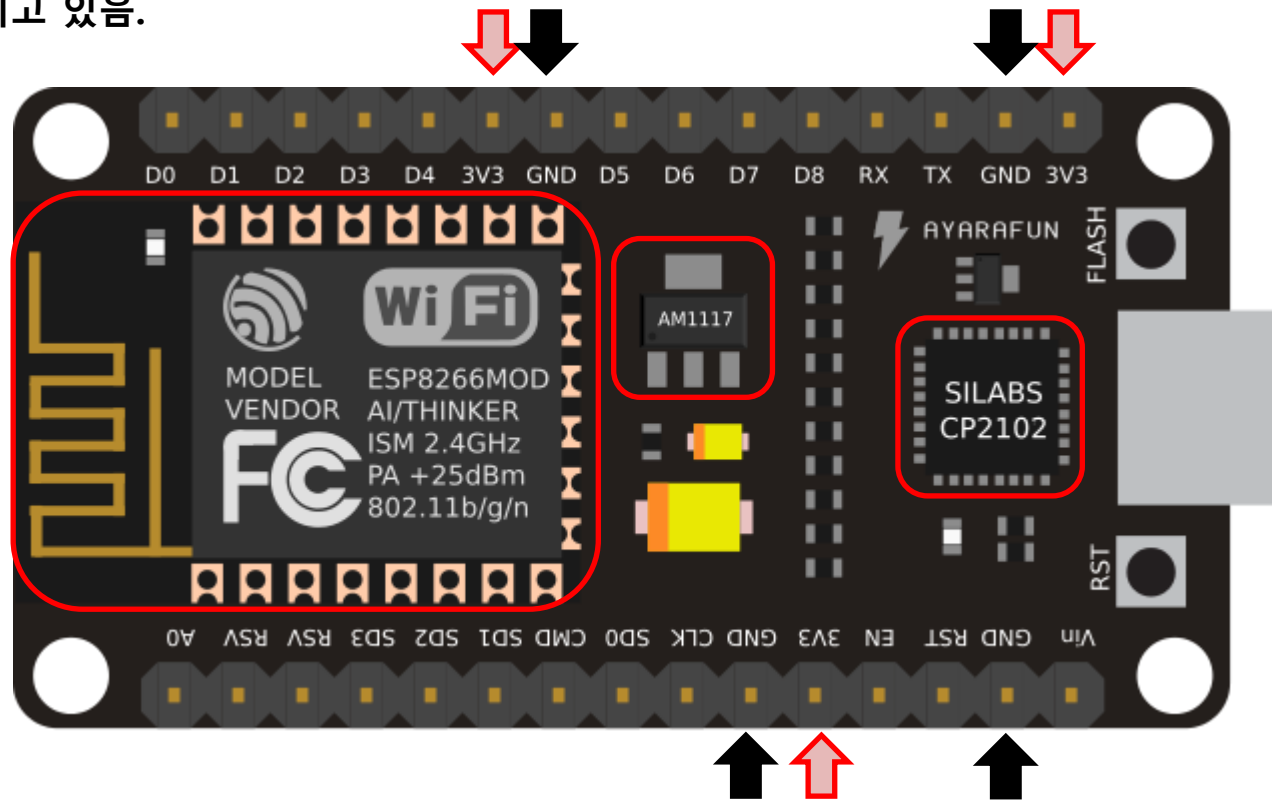
■ Install CP2102 driver

- ▶ 아두이노 배우기 카페
 - <https://cafe.naver.com/studyonarduino/5629>
 - CP210x_Windos_Drivers.zip 파일을 다운 로드 후 압축 풀어서
 - 운영체제에 맞는 파일 실행



■ NodeMCU

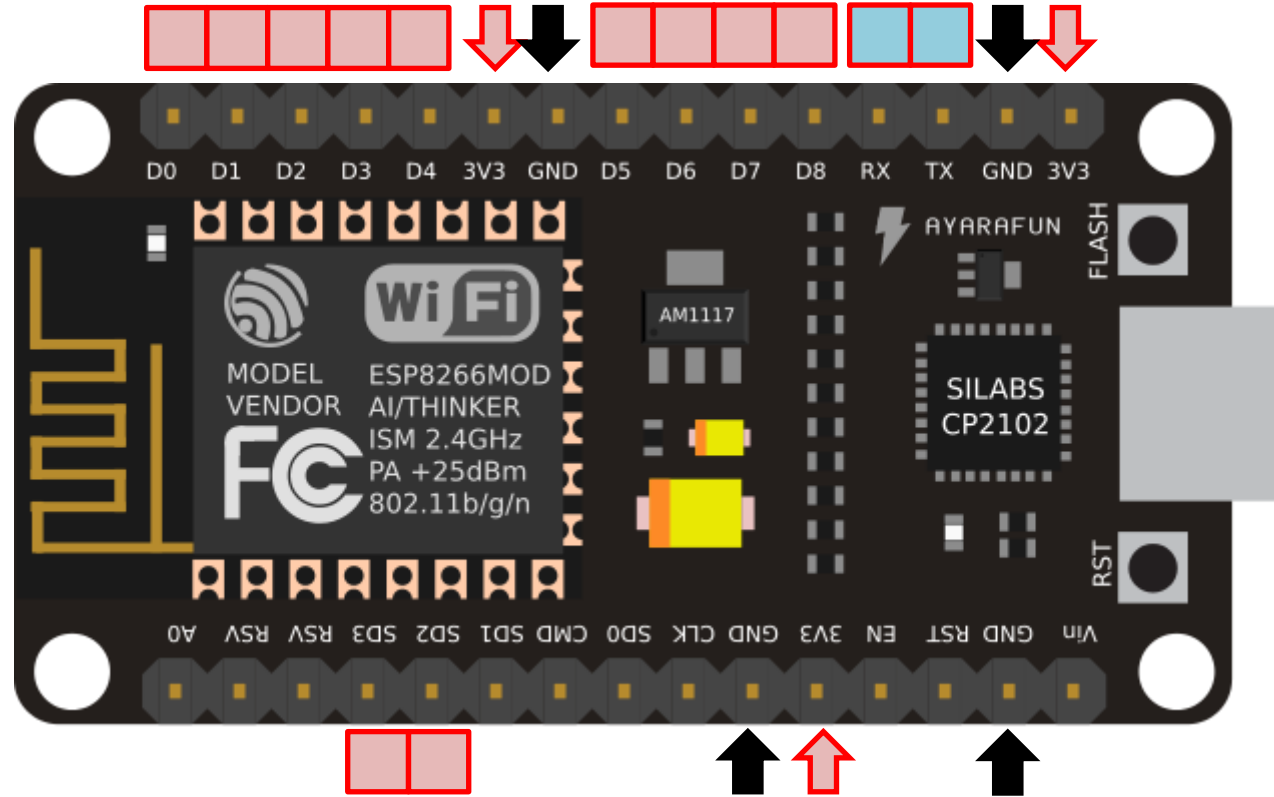
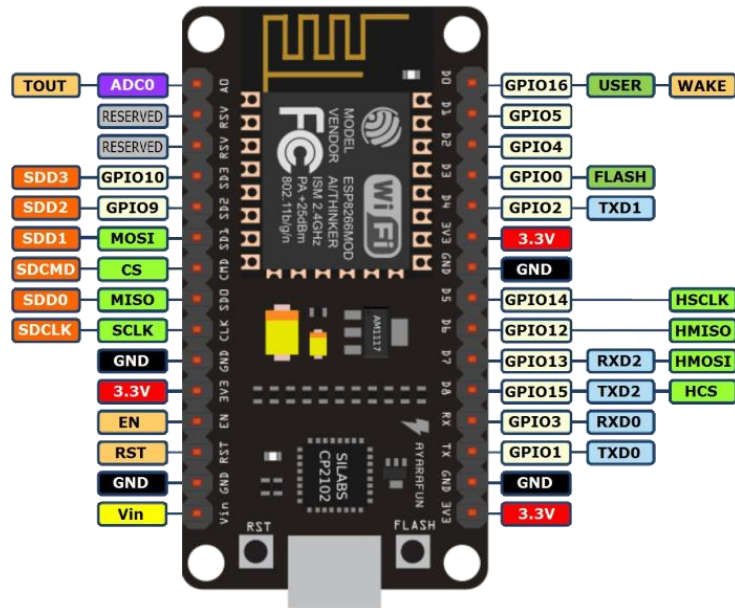
- ▶ ESP-12E
- ▶ SILABS CP2102
 - CP2102 이외에도 다양한 칩이 사용되고 있음.
- ▶ AM1117
- ▶ 전원 핀 확인



NodeMCU – layout/GPIO Pin

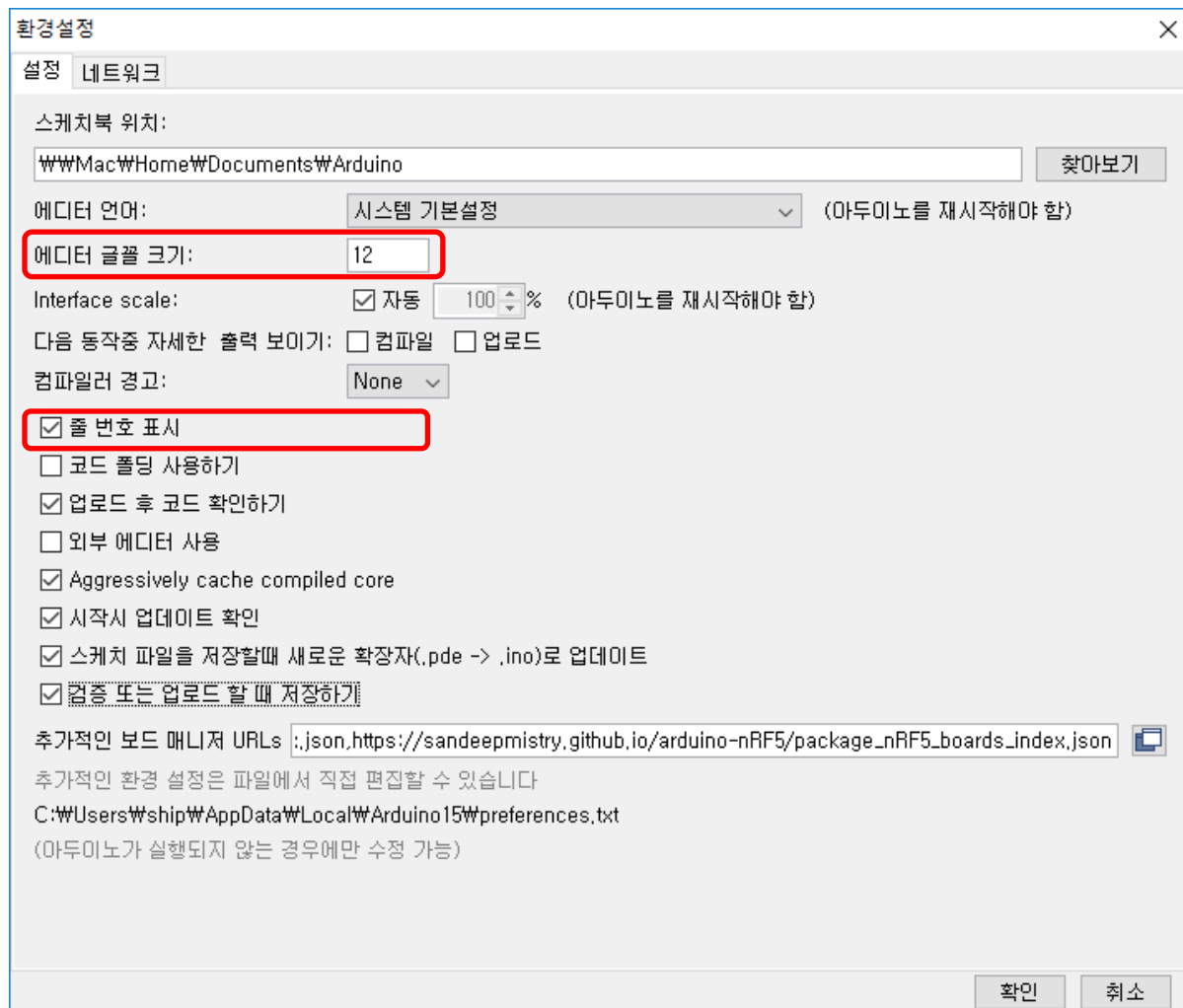
■ NodeMCU

- ▶ **ESP-12E**
- ▶ **SILABS CP2102**
- ▶ **AM1117**



■ 아두이노 통합 개발 환경 설정

- ▶ 에디터 글꼴 크기
- ▶ 줄 번호 표시

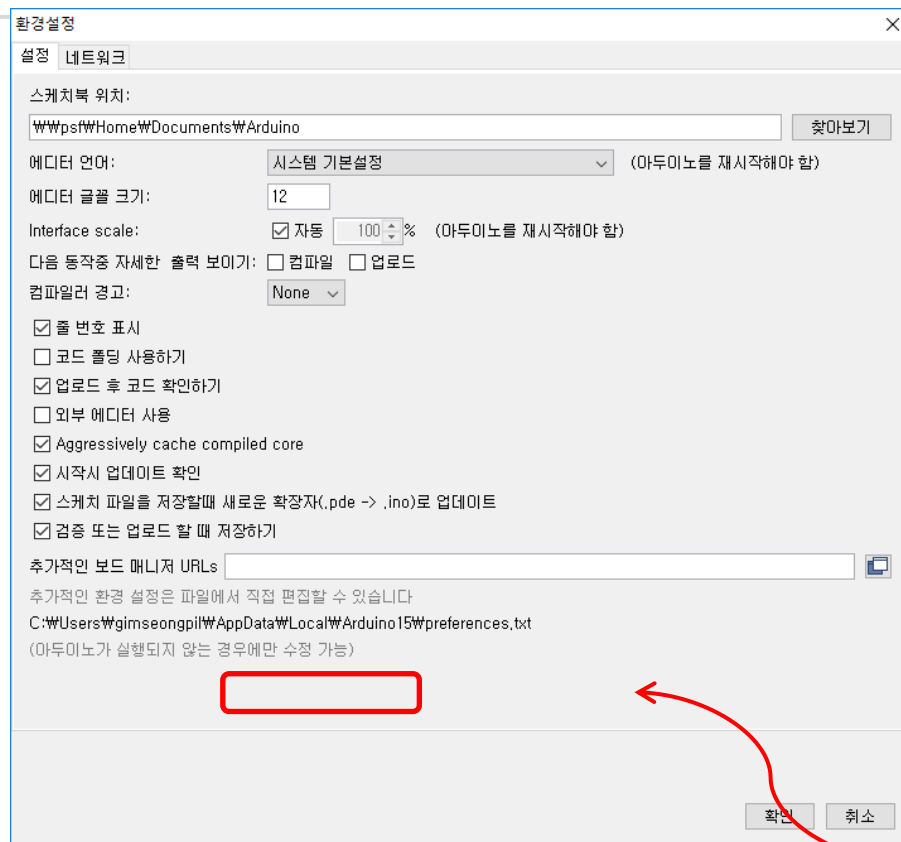


NodeMCU 개발 환경 구축

■ 개발 환경 구축

▶ NodeMCU 보드를 아두이노 IDE에 추가

- [단계1] Arduino IDE 설치
- [단계2] Arduino IDE 실행 → 파일/환경설정 선택
- [단계3] 추가적인 보드 매니저 URLs
 - » NodeMCU 보드에 대한 정보가 있는 URL



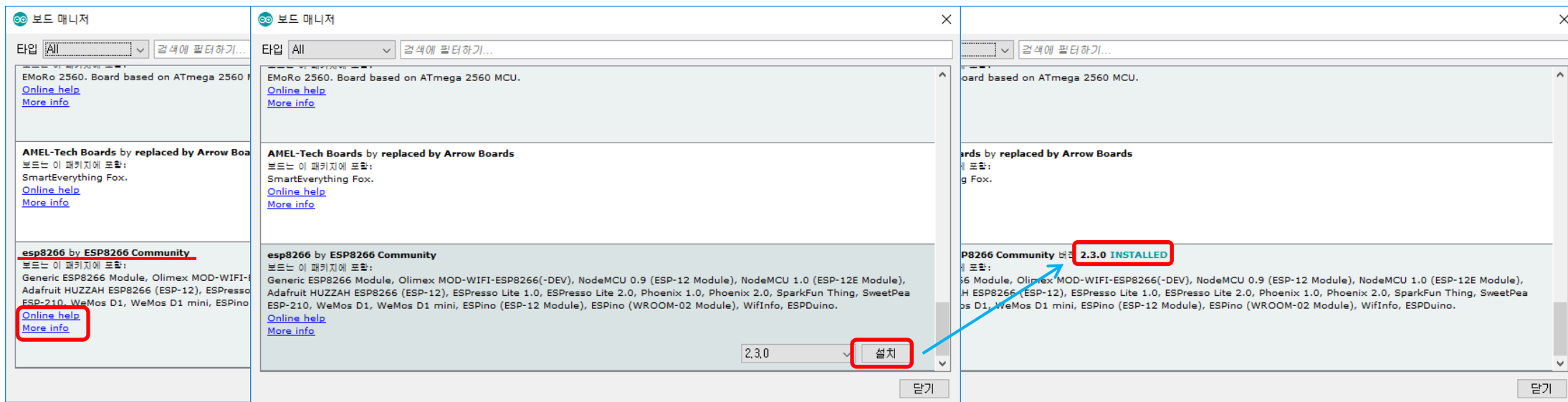
- 위 URL은 여기(<https://cafe.naver.com/studyonarduino/5629>)에서 복사하실 수 있습니다.

http://arduino.esp8266.com/stable/package_esp8266com_index.json

NodeMCU 개발 환경 구축

■ 개발 환경 구축

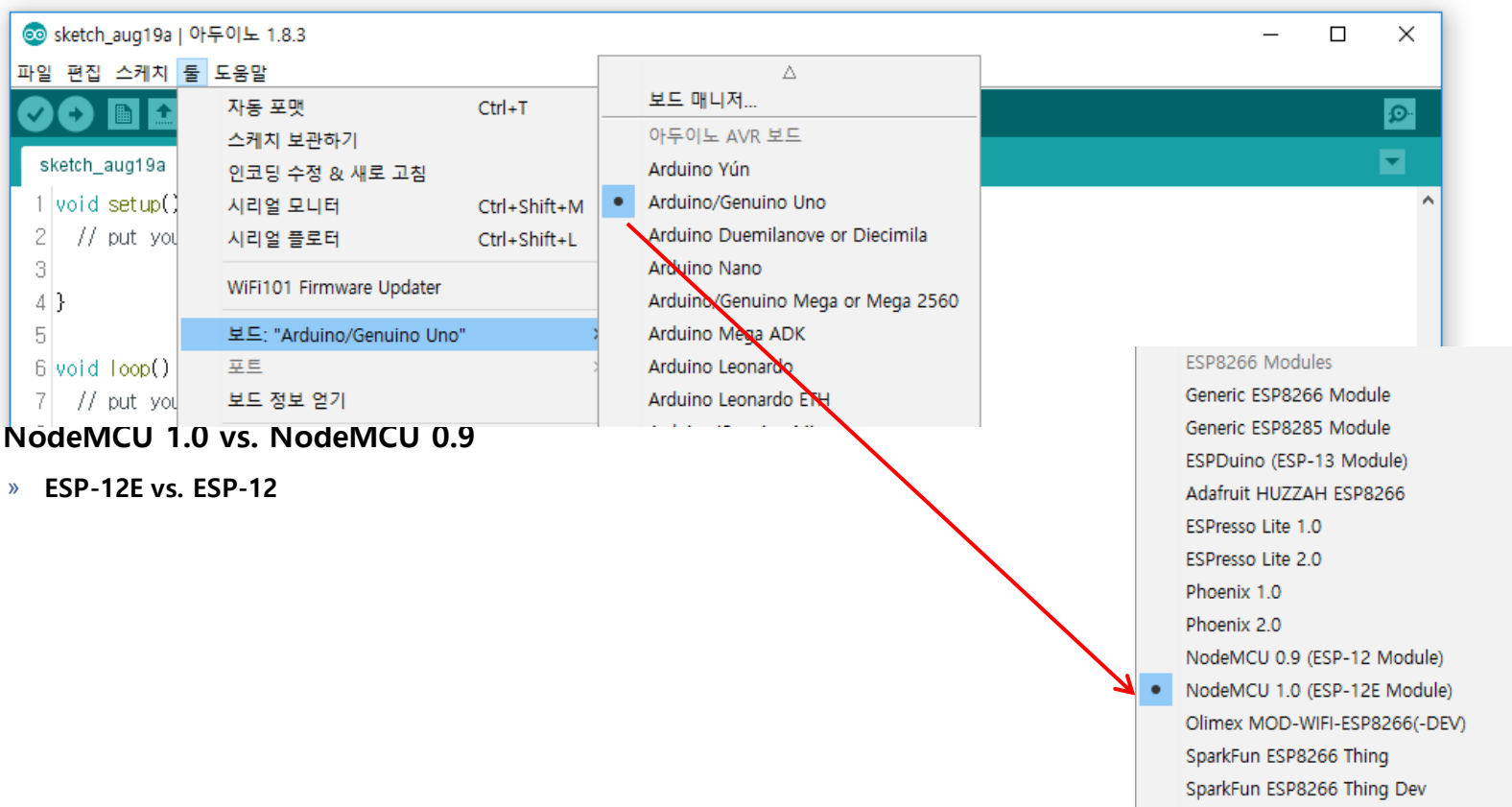
- [단계4] 메뉴 – 툴/보드/보드메니저... 선택 → esp8266 by ESP8266 Community에서 More info 선택 → 설치



NodeMCU 개발 환경 구축

■ 개발 환경 구축

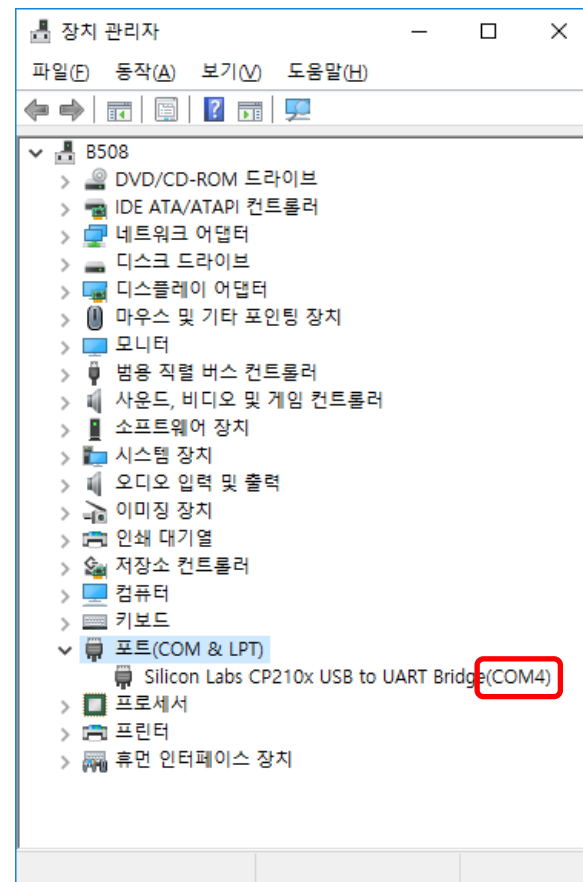
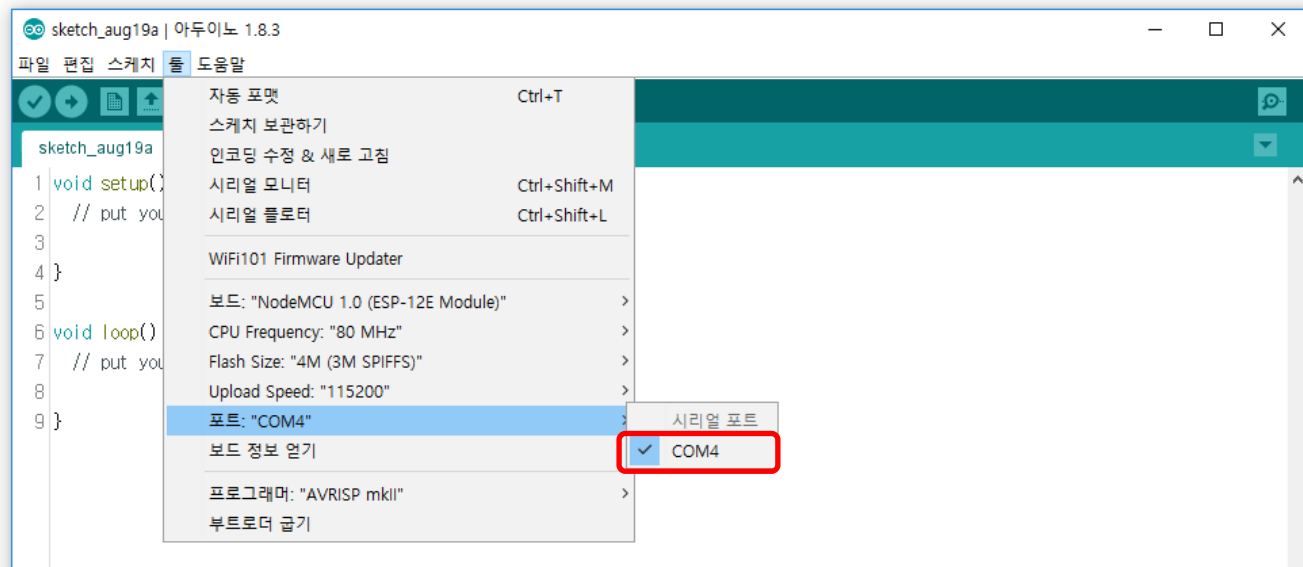
- [단계5] 메뉴 – 툴/보드 → NodeMCU 1.0(ESP-12E Module) 선택



NodeMCU 개발 환경 구축

■ 개발 환경 구축

- [단계6] 장치 관리자에서 보드가 연결된 포트 확인
 - Windows(Key) + Pause/Break(Key)를 동시에 누름
 - » 왼쪽 상단에 보이는 '장치관리자' 메뉴 클릭
 - 포트(COM & LPT)를 클릭하여 보드가 연결된 포트 번호 확인
- [단계7] 메뉴 - 툴/포트 클릭 → 보드가 연결된 포트 선택(클릭)



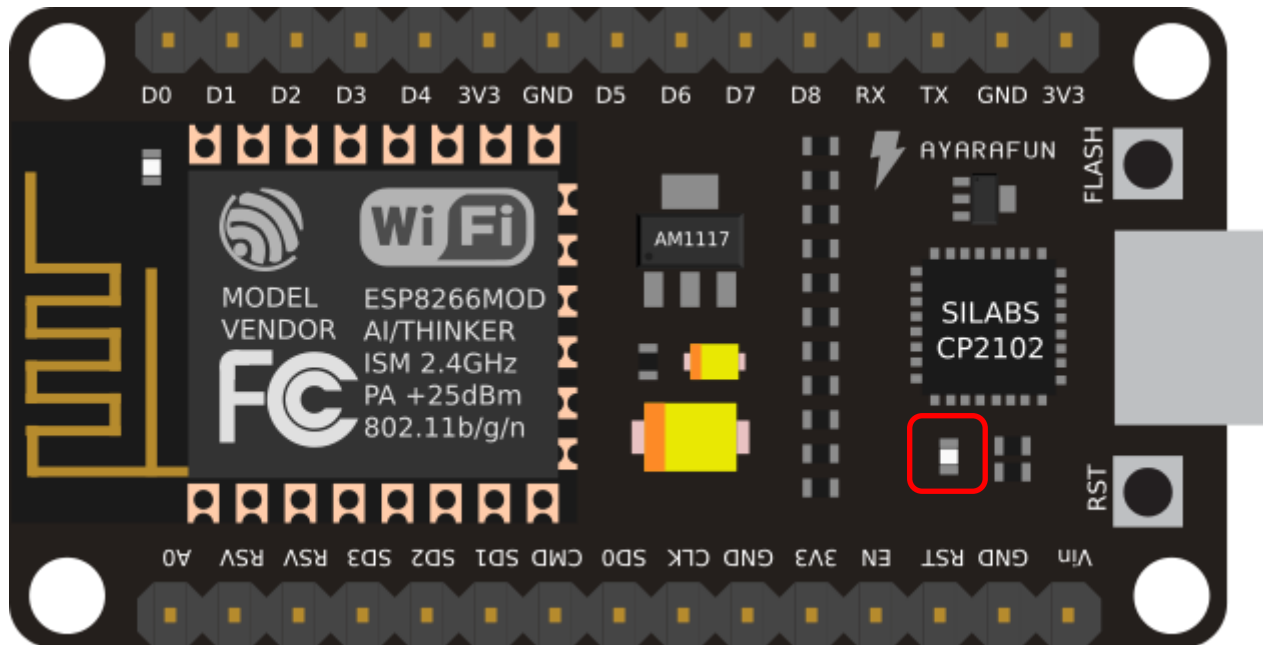
Digital OUTPUT

■ Digital Output

▶ 회로구성

- D0 핀에 LED 연결

▶ 메뉴-파일/예제/ESP8266/Blink



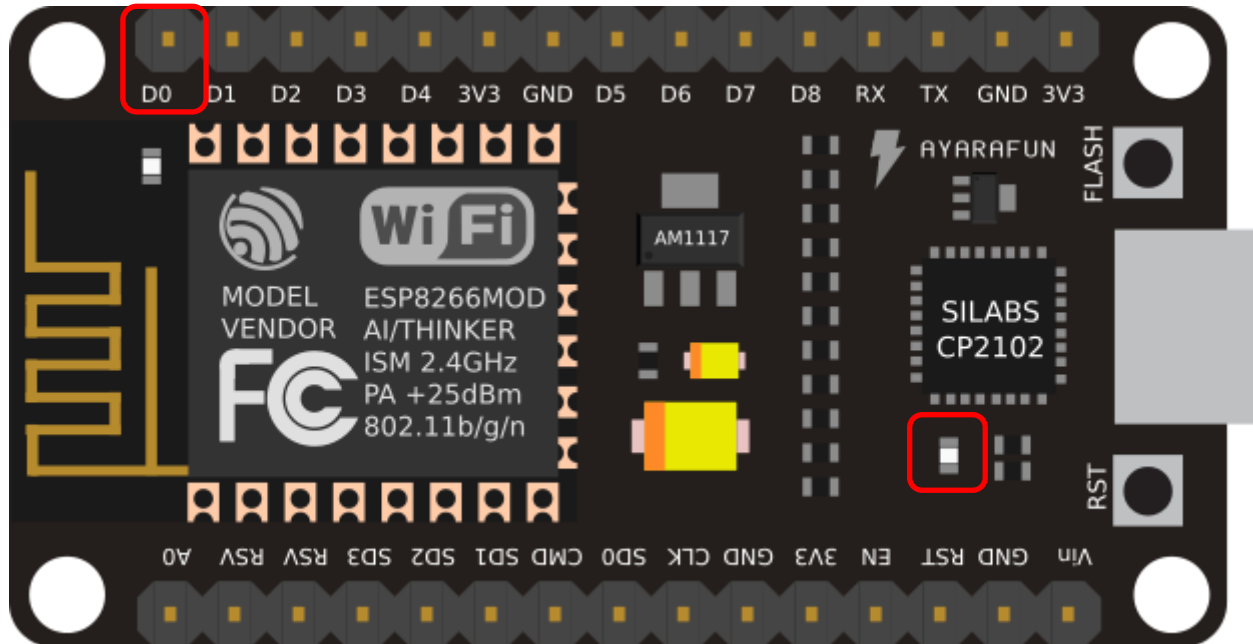
LED_BUILTIN

```
void setup() {  
    pinMode(LED_BUILTIN, OUTPUT);  
}  
  
void loop() {  
    digitalWrite(LED_BUILTIN, LOW);  
    delay(1000);  
    digitalWrite(LED_BUILTIN, HIGH);  
    delay(2000);  
}
```

Digital OUTPUT

■ Digital Output

- ▶ 회로구성
- ▶ 예제 파일 오픈
 - 메뉴-파일/예제/Basic/Blink → 아래와 같이 코드 수정



```
void setup() {  
    pinMode(D0, OUTPUT);  
}  
  
void loop() {  
    digitalWrite(D0, LOW);  
    delay(1000);  
    digitalWrite(D0, HIGH);  
    delay(1000);  
}
```

- **ESP-12E + (SiLabs)CP2102**

- 구동 전류 : 80mA

- NCP1117, NCV1117

$$-1.0[A]$$

- 참고 사이트

- <https://nodemcu.readthedocs.io/>

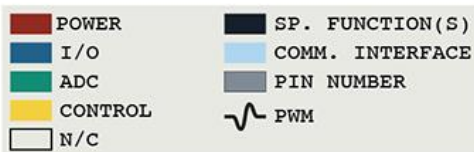
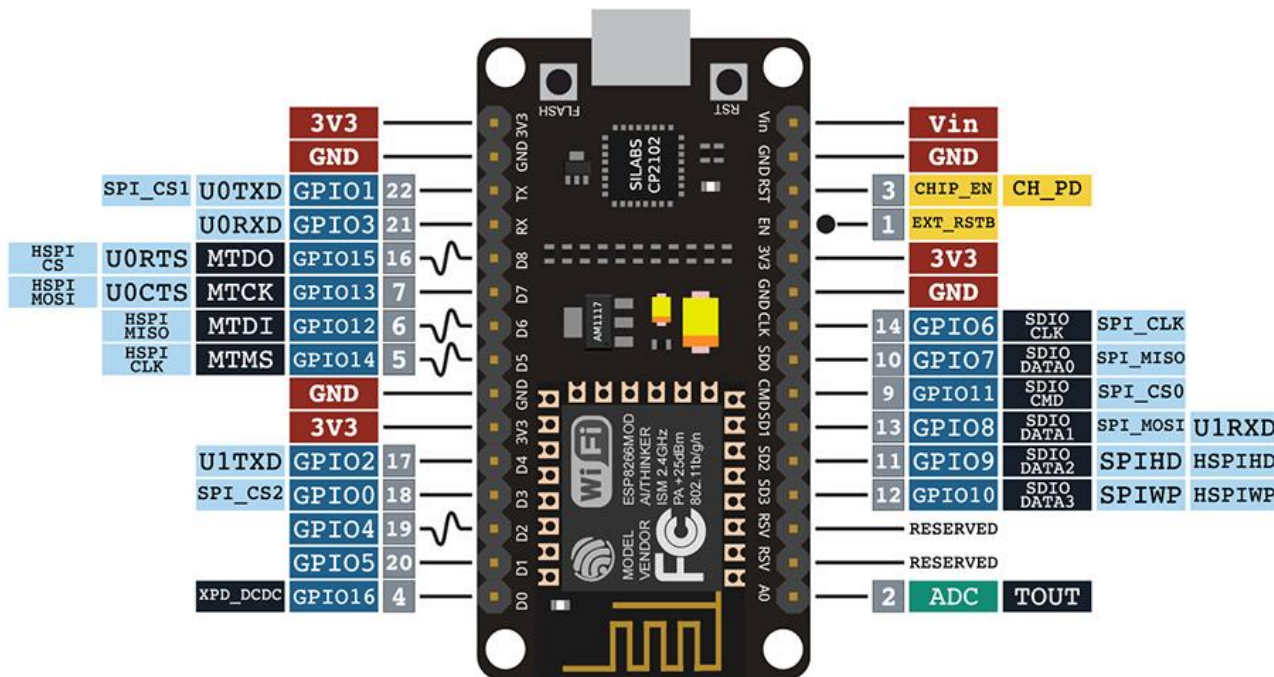
- **Nodemcu.com**

PINOUT

▲ Typ. pin current 6mA (Max. 12mA)

⚠ For sleep mode, connect GPIO16 and EXT_RSTB. On wakeup, GPIO16 will output LOW for system reset.

⚠ On boot/reset/wakeup, keep GPIO15 LOW and GPIO2 HIGH.

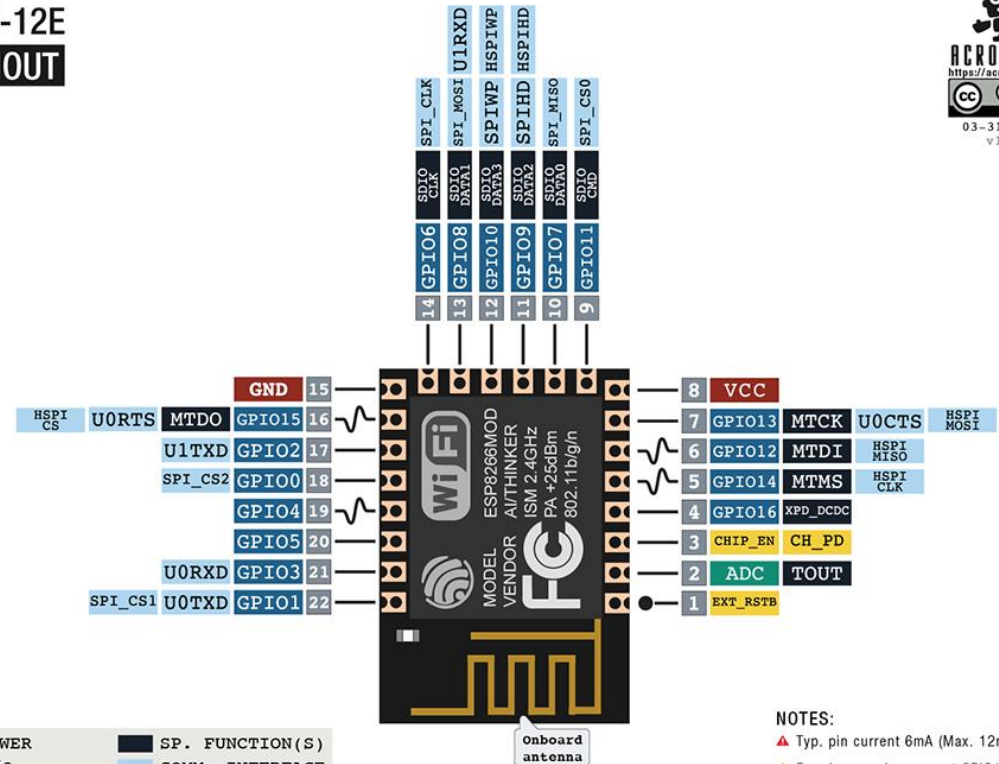


NodeMCU

▶ ESP-12E

- ESP8266-12의 업그레이드 버전.
- 32-bit RISC CPU:Tensilica Xtensa L106 running at 80MHz
- 64-KiB of instruction RAM, 96 KiB of data RAM
- 10-bit ADC
- WiFi 2.4GHz, WPA 지원, Station/softAP/SoftAp+station
- 대기 모드/슬립 모드 가능
- ESP-12 모듈과의 차이점
 - GPIO Pin 2개 추가, SPI pin 추가
 - EMI 차단 기능 향상

ESP-12E
PINOUT



NOTES:

- ▲ Typ. pin current 6mA (Max. 12mA)
- ▲ For sleep mode, connect GPIO16 and EXT_RSTB. On wakeup, GPIO16 will output LOW for system reset.
- ▲ On boot/reset/wakeup, keep GPIO15 LOW and GPIO2 HIGH.

■ NodeMCU

- ▶ Board Name : Amica
 - ▶ Open Source IoT Platform
 - **Install CP2102 driver**
 - Use 9600 Baud Rate
 - Connect Wi-Fi and Enjoy
- VIN + 5V Recommended +10V Max



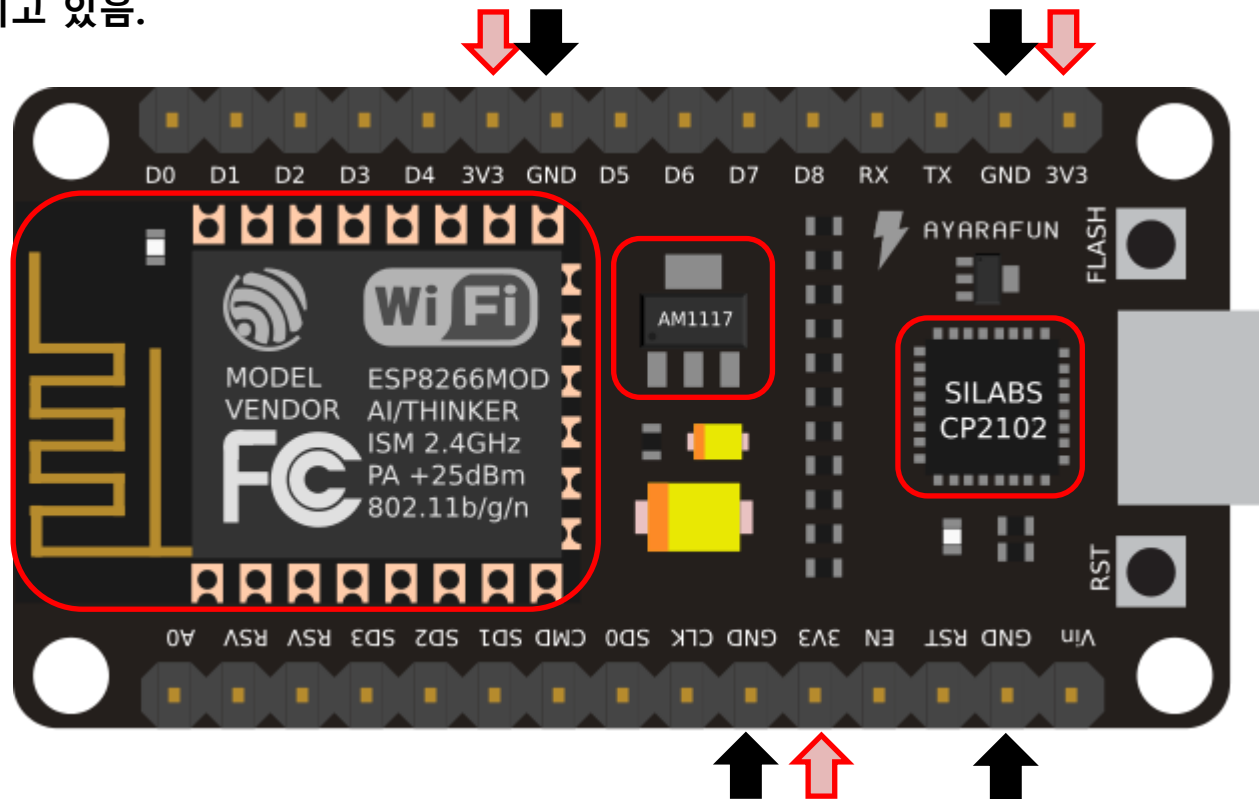
■ Install CP2102 driver

- ▶ 아두이노 배우기 카페
 - <https://cafe.naver.com/studyonarduino/5629>
 - CP210x_Windos_Drivers.zip 파일을 다운 로드 후 압축 풀어서
 - 운영체제에 맞는 파일 실행



■ NodeMCU

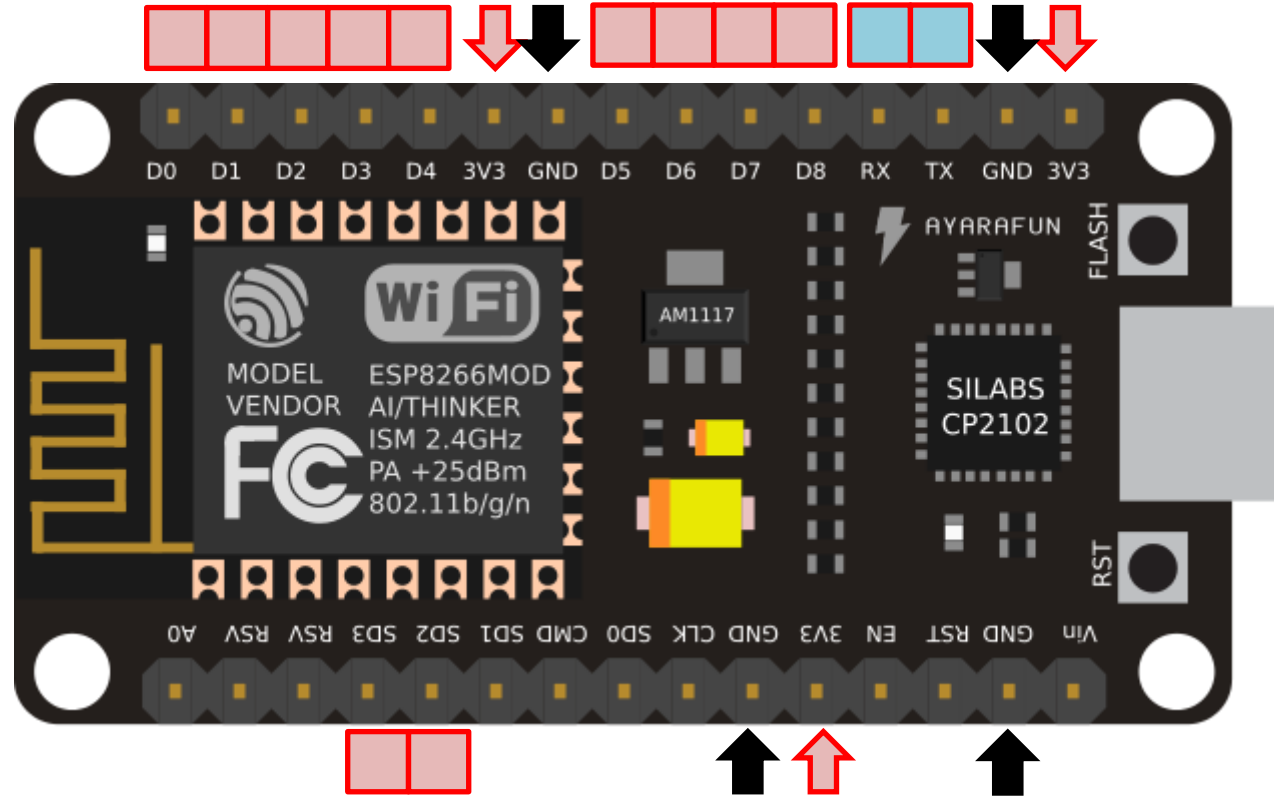
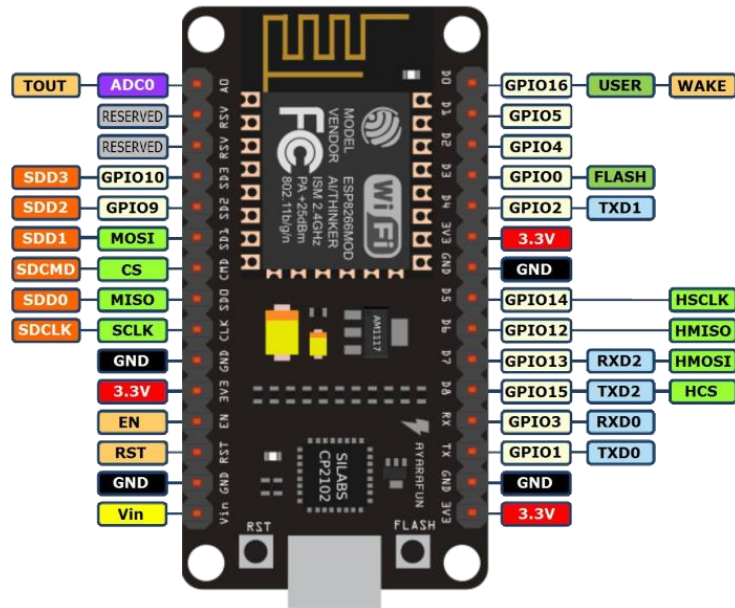
- ▶ ESP-12E
- ▶ SILABS CP2102
 - CP2102 이외에도 다양한 칩이 사용되고 있음.
- ▶ AM1117
- ▶ 전원 핀 확인



NodeMCU – layout/GPIO Pin

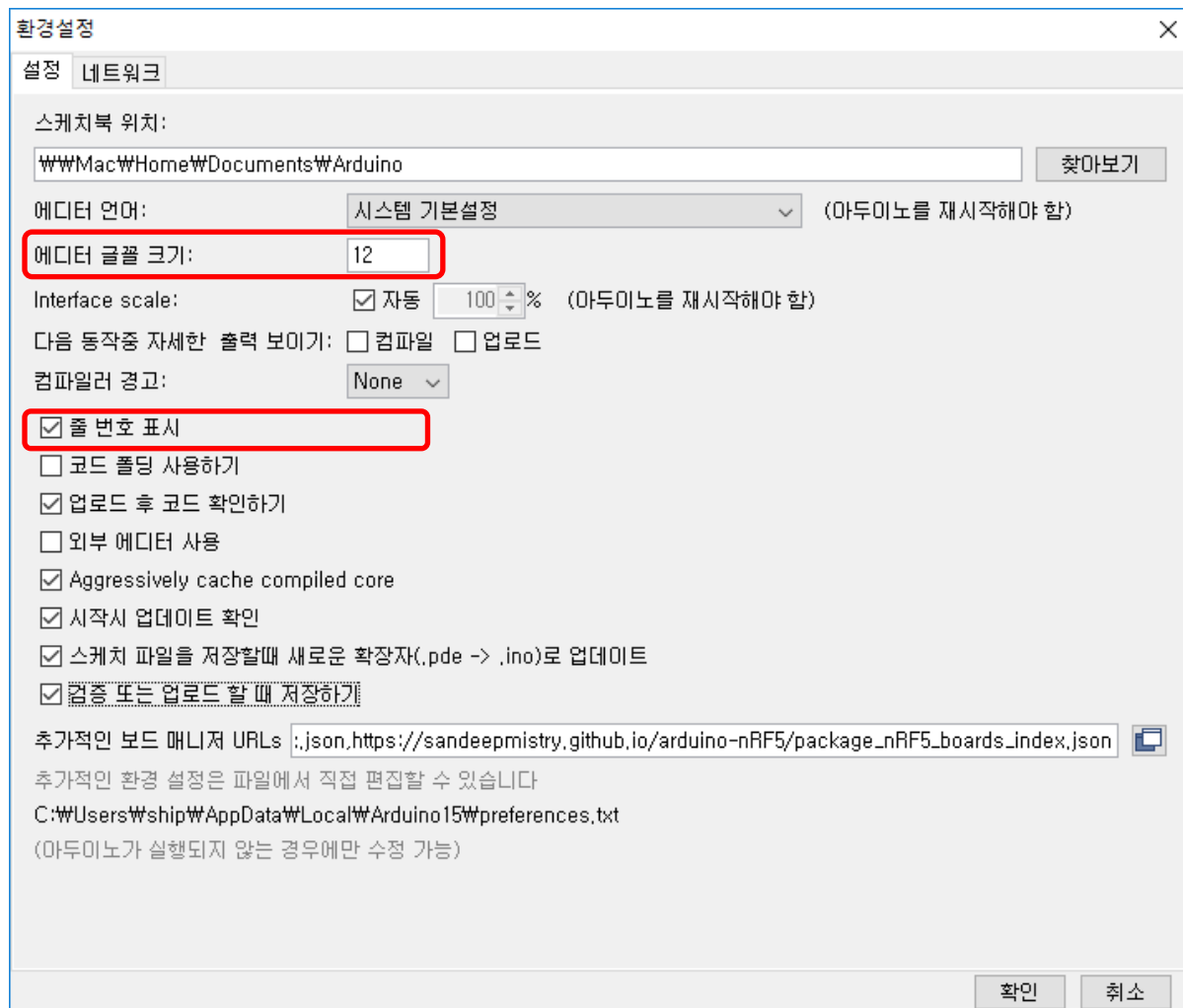
■ NodeMCU

- ▶ **ESP-12E**
- ▶ **SILABS CP2102**
- ▶ **AM1117**



■ 아두이노 통합 개발 환경 설정

- ▶ 에디터 글꼴 크기
- ▶ 줄 번호 표시



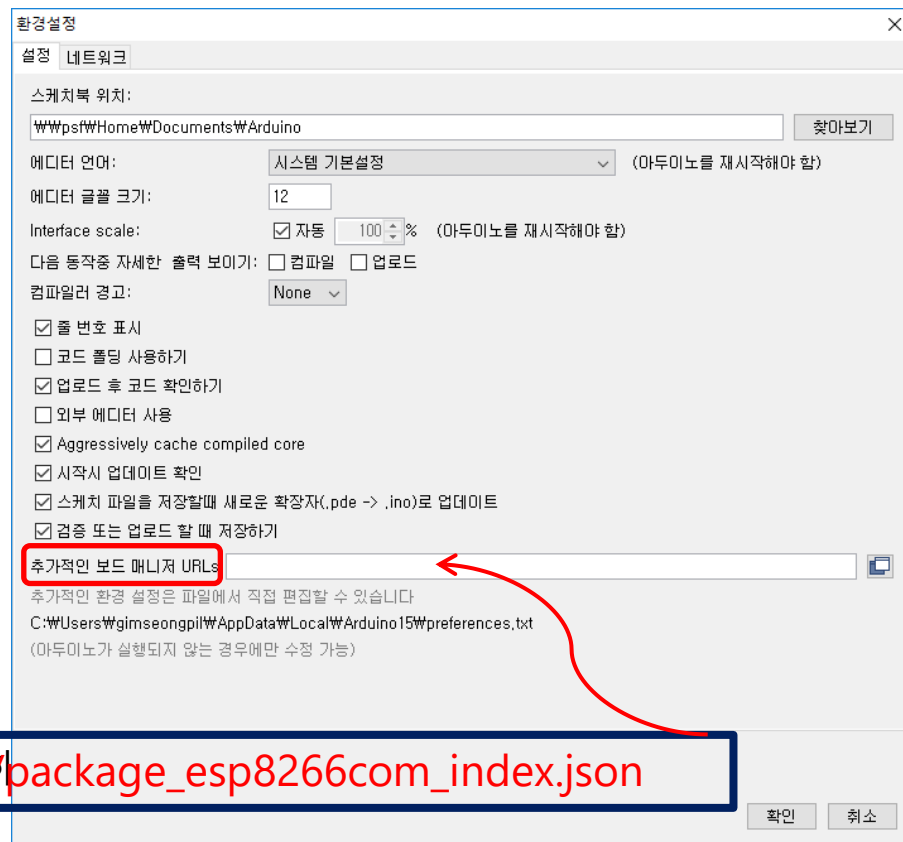
NodeMCU 개발 환경 구축

■ 개발 환경 구축

▶ NodeMCU 보드를 아두이노 IDE에 추가

- [단계1] Arduino IDE 설치
- [단계2] Arduino IDE 실행 → 파일/환경설정 선택
- [단계3] 추가적인 보드 매니저 URLs
 - » NodeMCU 보드에 대한 정보가 있는 URL

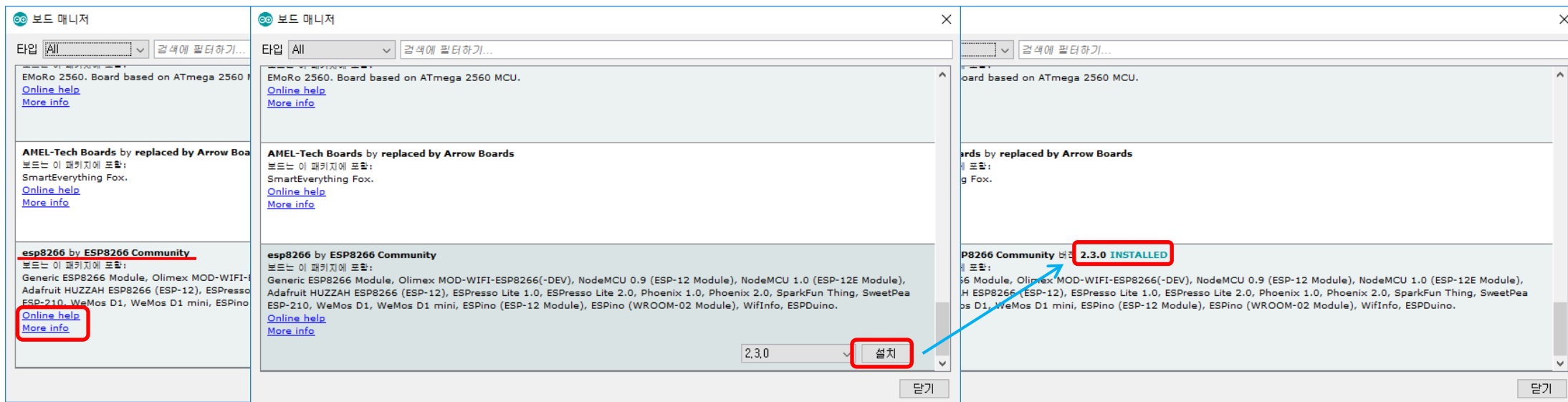
- 위 URL은 여기(<https://cafe.naver.com/studyonarduino/5629>)에서 복사하



NodeMCU 개발 환경 구축

■ 개발 환경 구축

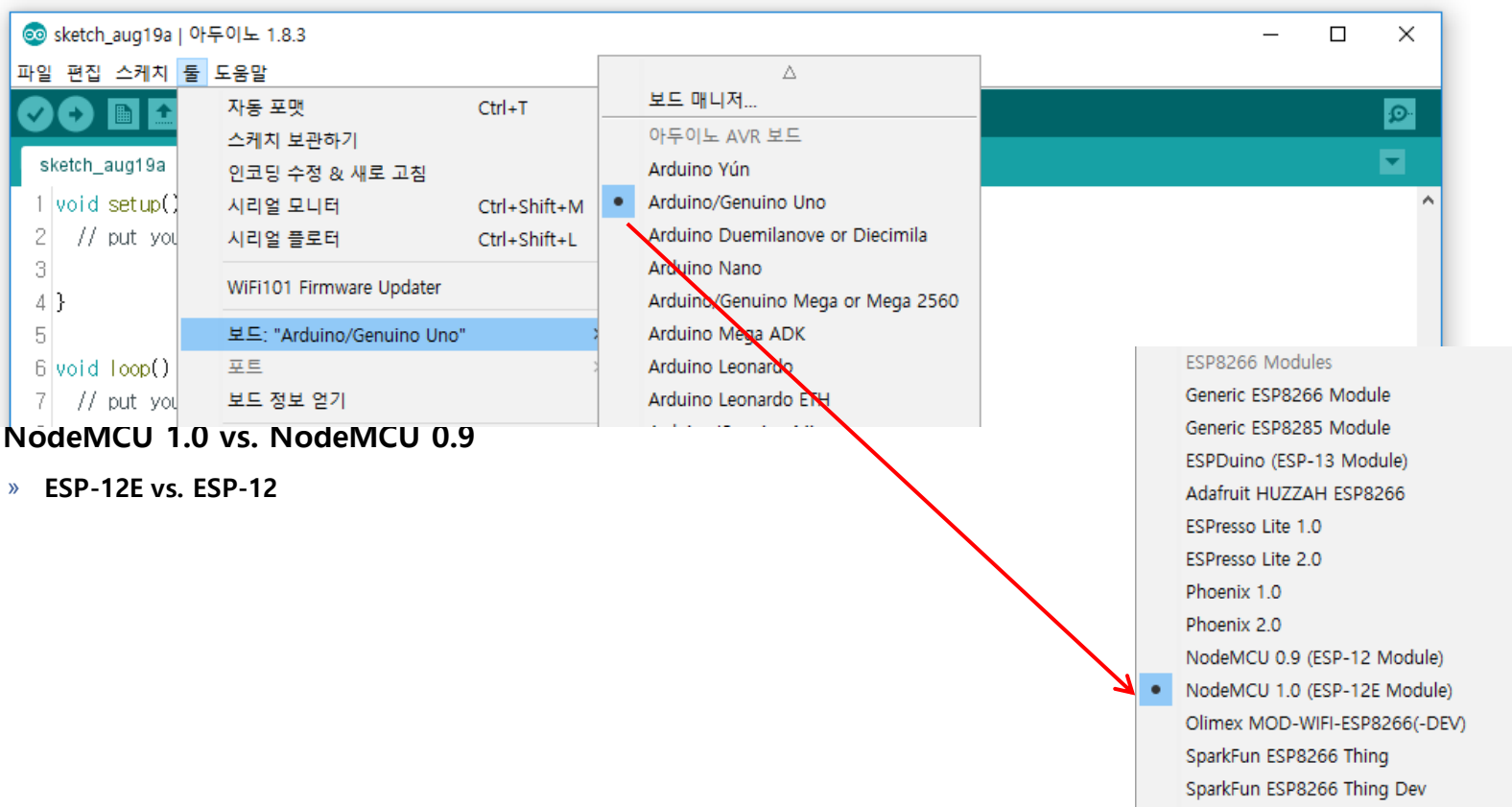
- [단계4] 메뉴 – 툴/보드/보드메니저... 선택 → esp8266 by ESP8266 Community에서 More info 선택 → 설치



NodeMCU 개발 환경 구축

■ 개발 환경 구축

- [단계5] 메뉴 – 툴/보드 → NodeMCU 1.0(ESP-12E Module) 선택



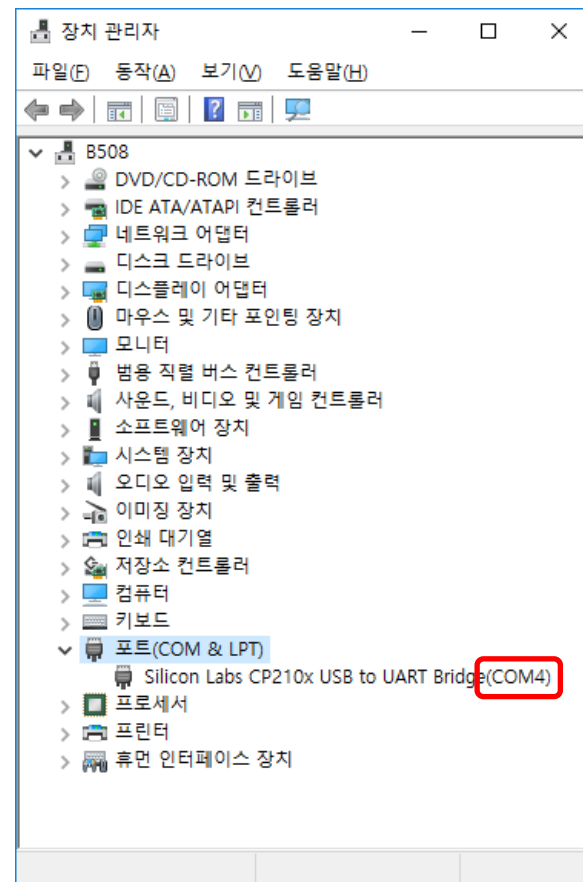
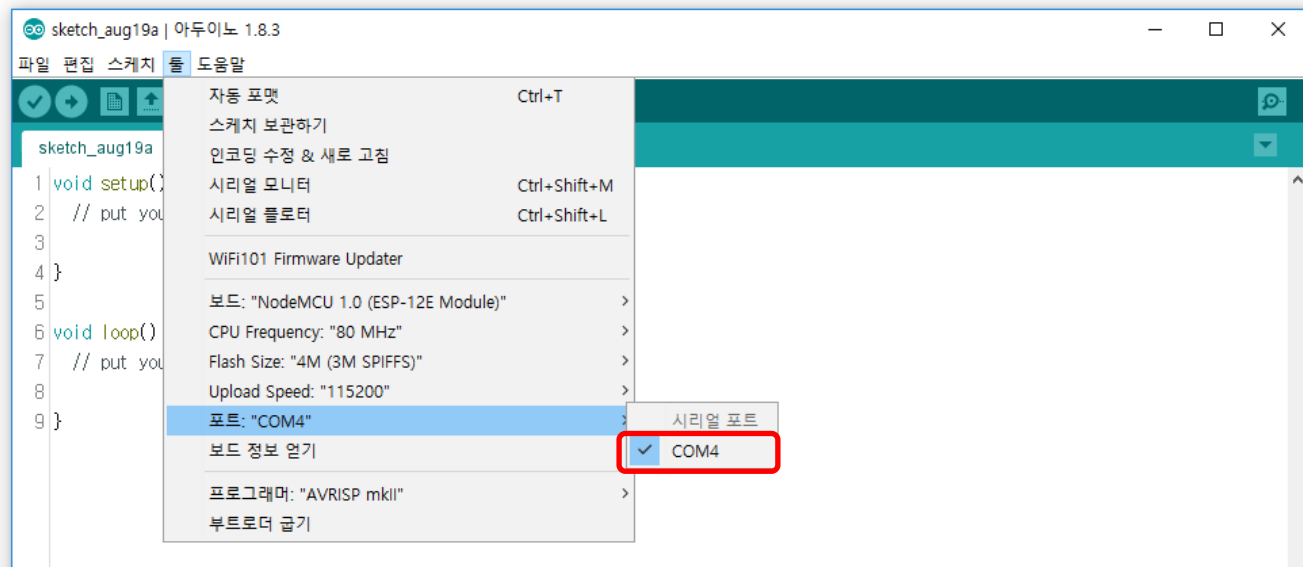
– NodeMCU 1.0 vs. NodeMCU 0.9

» ESP-12E vs. ESP-12

NodeMCU 개발 환경 구축

■ 개발 환경 구축

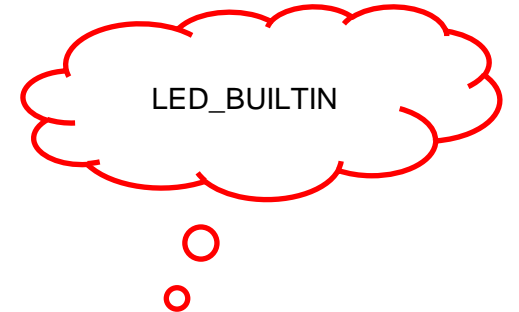
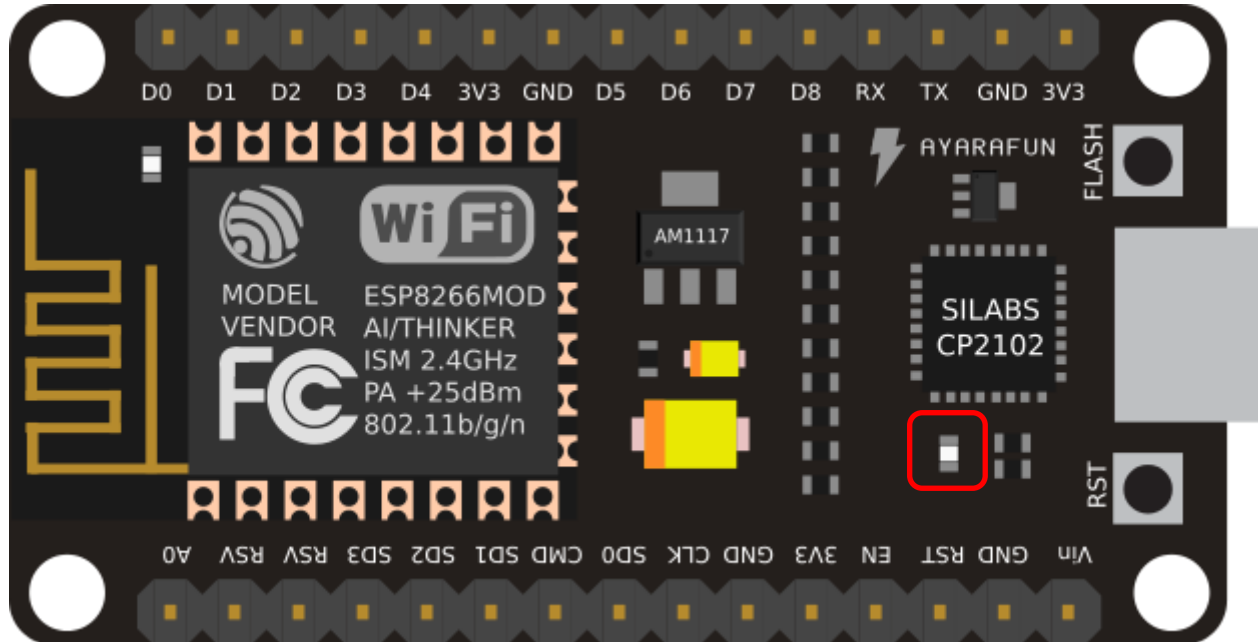
- [단계6] 장치 관리자에서 보드가 연결된 포트 확인
 - Windows(Key) + Pause/Break(Key)를 동시에 누름
 - » 왼쪽 상단에 보이는 '장치관리자' 메뉴 클릭
 - 포트(COM & LPT)를 클릭하여 보드가 연결된 포트 번호 확인
- [단계7] 메뉴 - 툴/포트 클릭 → 보드가 연결된 보트 선택(클릭)



Digital OUTPUT

■ Digital Output

- ▶ 회로구성
 - D0 핀에 LED 연결
- ▶ 메뉴-파일/예제/ESP8266/Blink

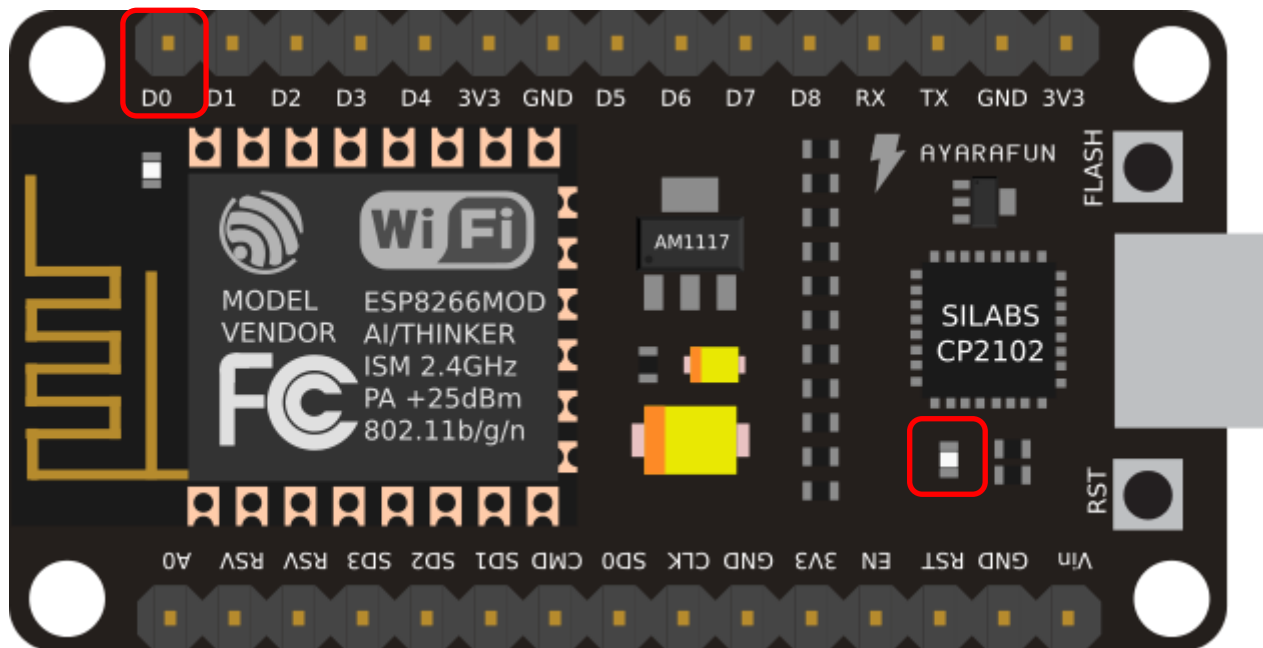


```
void setup() {  
    pinMode(LED_BUILTIN, OUTPUT);  
}  
  
void loop() {  
    digitalWrite(LED_BUILTIN, LOW);  
    delay(1000);  
    digitalWrite(LED_BUILTIN, HIGH);  
    delay(2000);  
}
```

Digital OUTPUT

■ Digital Output

- ▶ 회로구성
- ▶ 예제 파일 오픈
 - 메뉴-파일/예제/Basic/Blink → 아래와 같이 코드 수정



```
void setup() {  
    pinMode(D0, OUTPUT);  
}  
  
void loop() {  
    digitalWrite(D0, LOW);  
    delay(1000);  
    digitalWrite(D0, HIGH);  
    delay(1000);  
}
```

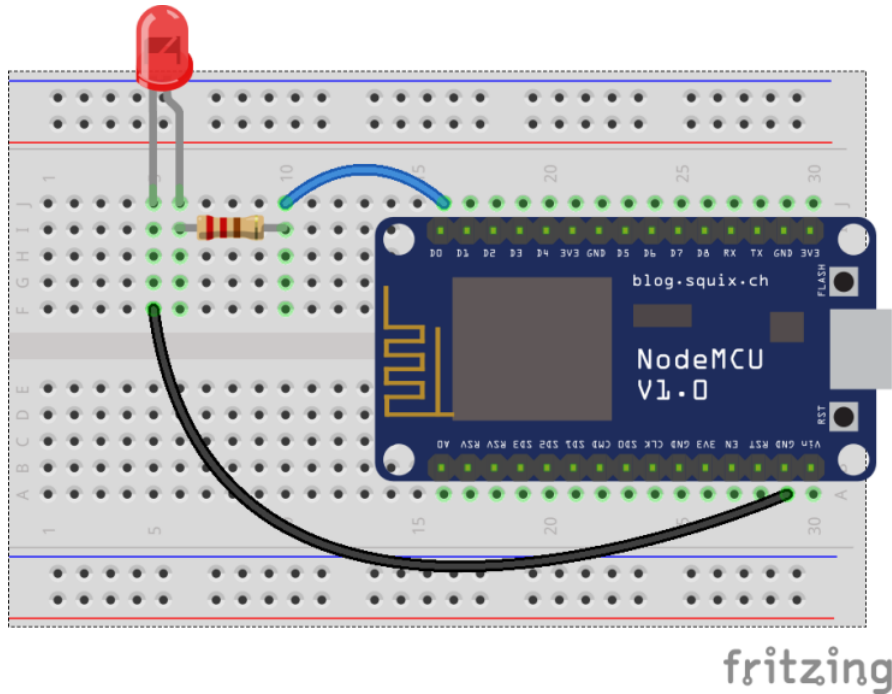

Digital OUTPUT

■ Digital Output

▶ 회로구성

- D0 핀에 LED 연결

▶ 메뉴-파일/예제/Basic/Blink



```
void setup() {  
  pinMode(D0, OUTPUT);  
}
```

```
void loop() {  
  digitalWrite(D0, LOW);  
  delay(1000);  
  digitalWrite(D0, HIGH);  
  delay(1000);  
}
```

Digital OUTPUT

Digital Output

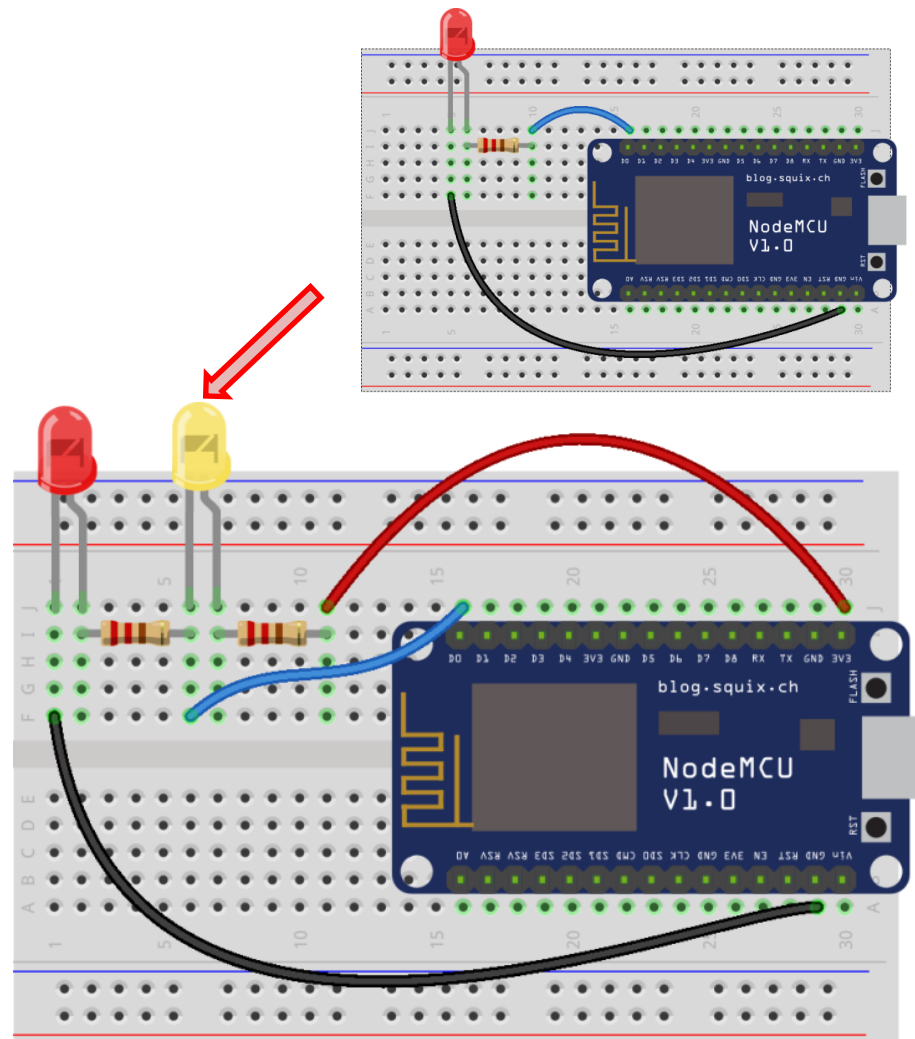
회로구성

- D0 핀에 LED 연결 추가 연결

메뉴-파일/예제/ESP8266/Blink

```
void setup() {  
  pinMode(D0, OUTPUT);  
}
```

```
void loop() {  
  digitalWrite(D0, LOW);  
  delay(1000);  
  digitalWrite(D0, HIGH);  
  delay(2000);  
}
```



fritzing

Digital OUTPUT

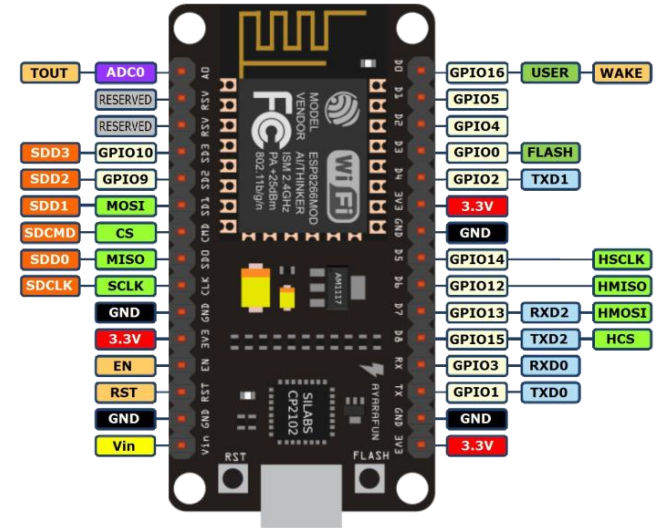
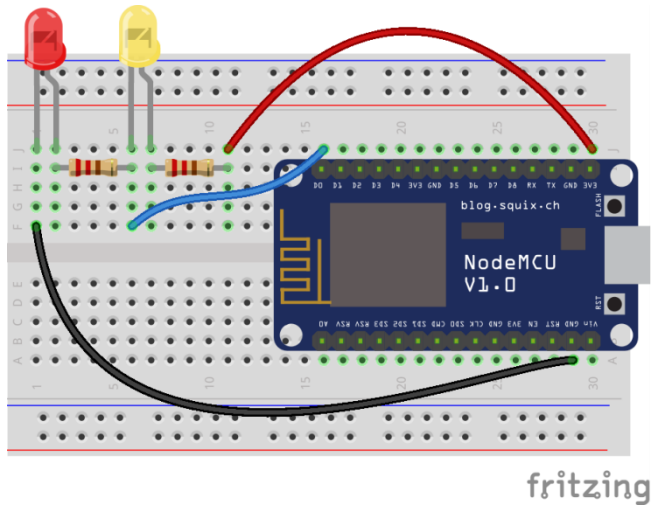
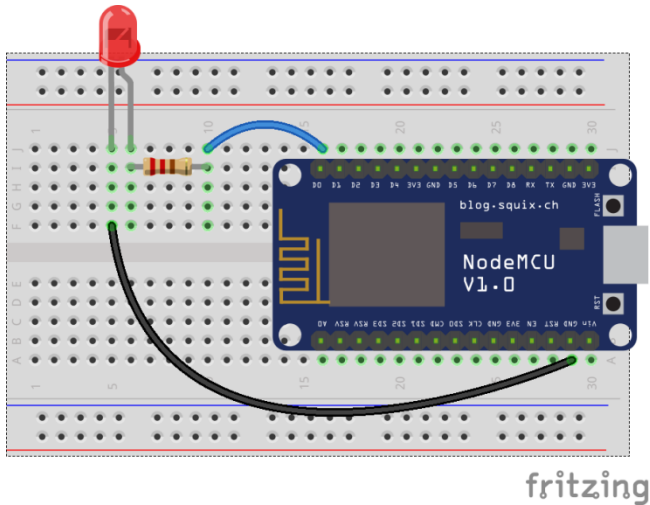
Digital Output

회로구성

- D0 핀에 LED 연결

메뉴-파일/예제/ESP8266/Blink

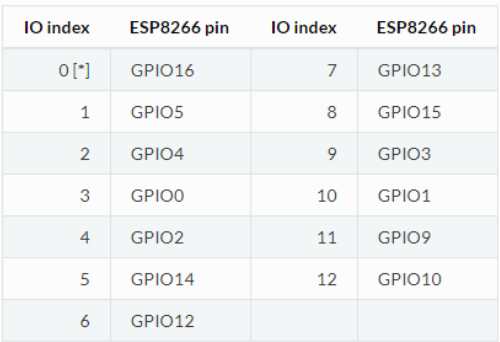
- 코드 수정 후 재업로드



```
void setup() {  
    pinMode(16, OUTPUT);  
}
```

```
void loop() {  
    digitalWrite(16, LOW);  
    delay(1000);  
    digitalWrite(16, HIGH);  
    delay(1000);  
}
```

■ Wiring between ESP-12E and NodeMCU



The pinMode Command

- ▶ In our setup(), we use the pinMode command to tell the ESP8266EX what we want it to do.

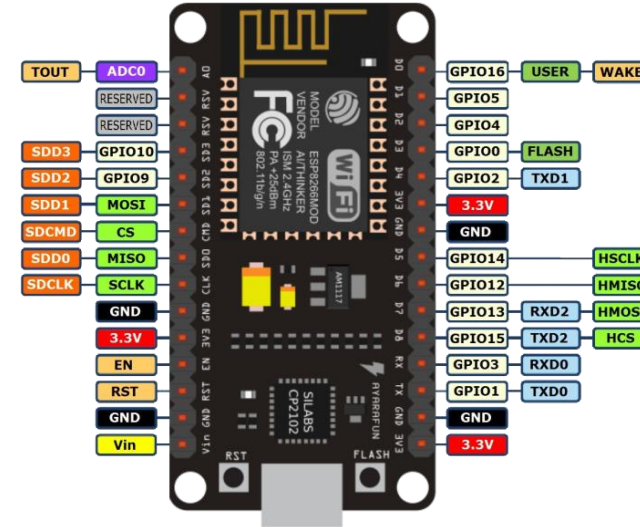
The GPIO Pin Number

`pinMode(pinNum, nMode);`

The Mode

- INPUT
- OUTPUT
- INPUT_PULLUP
- INPUT_PULLDOWN

- ▶ The key thing to remember is that you do not specify the development board pin number, but the GPIO number of the ESP8266EX Chip.
- ▶ Another determination that has to be made is whether or not to use the **INPUT_PULLUP** or **INPUT_PULLDOWN**.



IO index	ESP8266 pin	IO index	ESP8266 pin
0[*]	GPIO16	7	GPIO13
1	GPIO5	8	GPIO15
2	GPIO4	9	GPIO3
3	GPIO0	10	GPIO1
4	GPIO2	11	GPIO9
5	GPIO14	12	GPIO10
6	GPIO12		

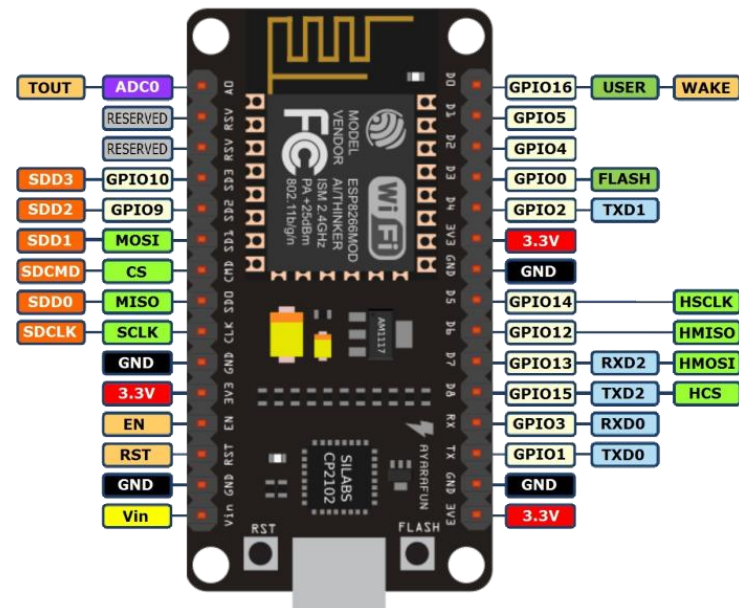
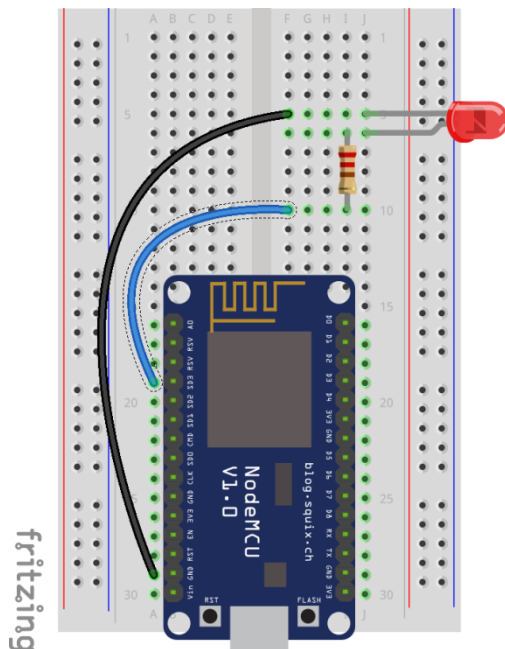
Digital OUTPUT

Digital Output

회로구성

- GPIO10 핀에 LED 연결 후 LED On/Off 반복
- GPIO9 핀에 LED 연결 후 LED On/Off 반복
→ 여기서 우리는 “모든 GPIO 핀의 상태를 HIGH 또는 LOW로 만들 수 있다!”

```
void setup() {  
  pinMode(10, OUTPUT);  
}  
void loop() {  
  digitalWrite(10, LOW);  
  delay(1000);  
  digitalWrite(10, HIGH);  
  delay(1000);  
}
```



■ nodeMCU의 UART

- ▶ UARTs(0 and 1) available but UART1 is not capable of receiving data and is therefore transmit only.
 - 256-byte TX and RX buffers
- ▶ Serial uses UART0
 - RXD0(GPIO3)
 - TXD0(GPIO1)
 - CP12xx를 통해 USB 케이블로 연결
 - Serial may be remapped to GPIO15(TXD2) and GPIO13(RXD2) by calling `Serial.swap()` after `Serial.begin()`.
 - Call swap again maps UART0 back to GPIO1 and GPIO3
 - 참고-<https://github.com/esp8266/Arduino/blob/master/doc/reference.rst#serial>
- ▶ UART1 - Only Tx
 - TXD1

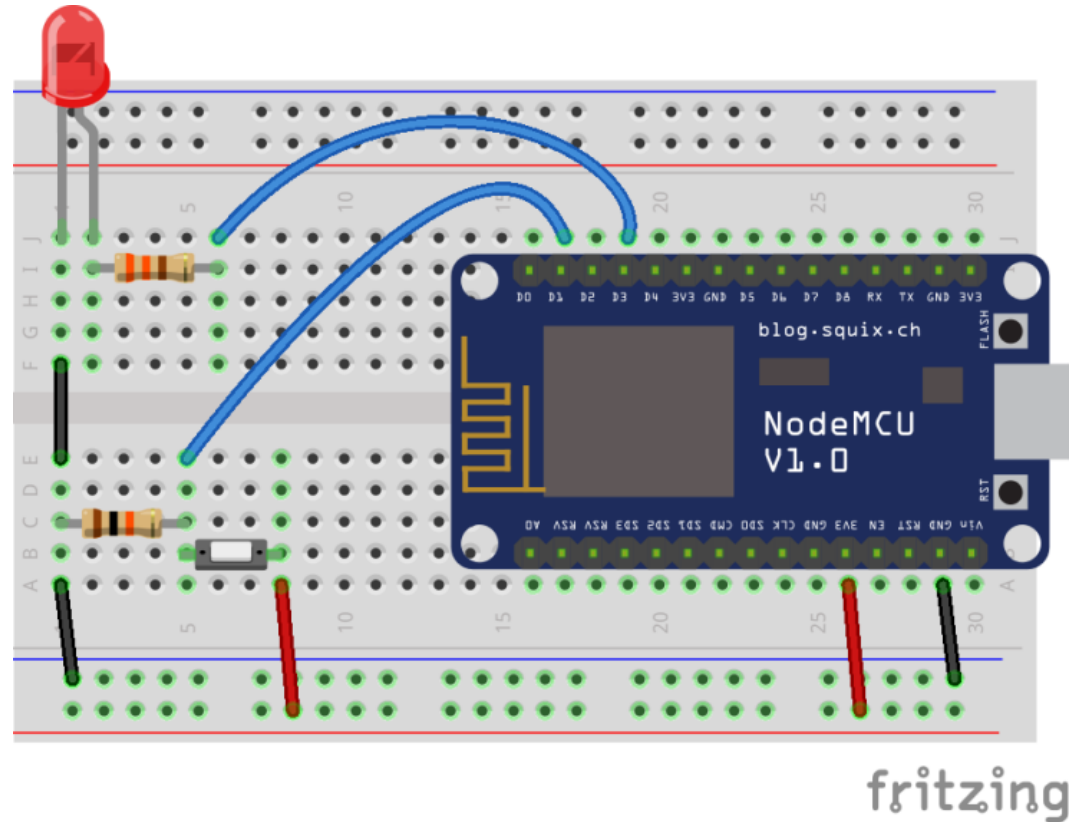
Digital Input / Serial Output

■ Digital Input

▶ 회로 구성

- D3 핀에 LED 연결
- D1 핀에 스위치 연결

```
void setup() {  
  Serial.begin(9600);  
  pinMode(D3, OUTPUT);  
  pinMode(D1, INPUT);  
}  
  
void loop() {  
  int swState = digitalRead(D1);  
  digitalWrite(D3, swState);  
  Serial.println( swState );  
}
```

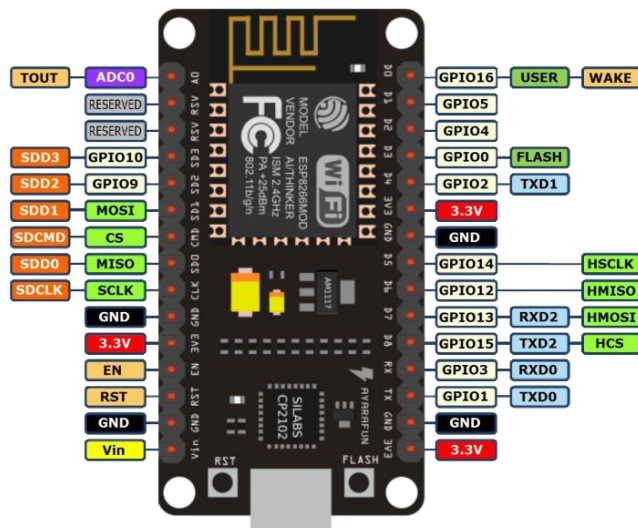


Digital Input / Serial Output

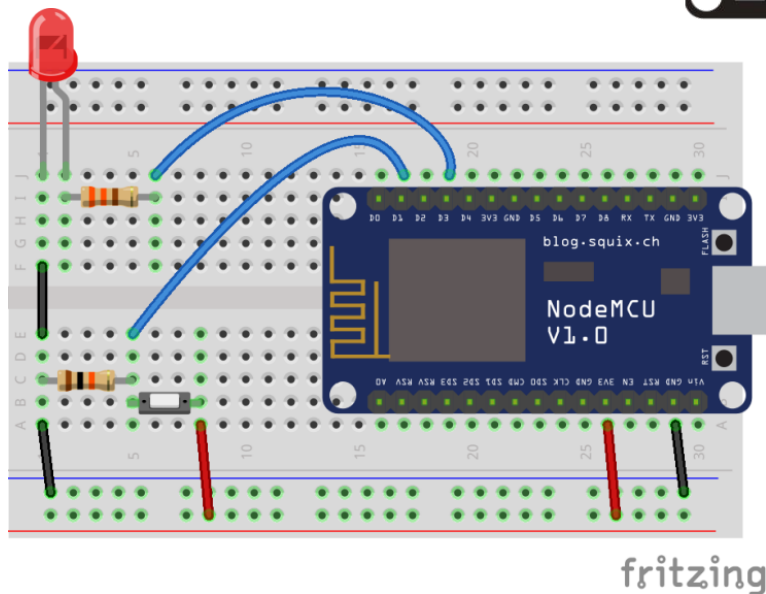
Digital Input

회로 구성

- D3 핀에 LED 연결
- D2 핀에 스위치 연결



IO index	ESP8266 pin	IO index	ESP8266 pin
0[*]	GPIO16	7	GPIO13
1	GPIO5	8	GPIO15
2	GPIO4	9	GPIO3
3	GPIO0	10	GPIO1
4	GPIO2	11	GPIO9
5	GPIO14	12	GPIO10
6	GPIO12		



```
void setup() {  
  Serial.begin(9600);  
  pinMode(0, OUTPUT);           //D3 – GPIO0  
  pinMode(5, INPUT);           //D1 – GPIO5  
}
```

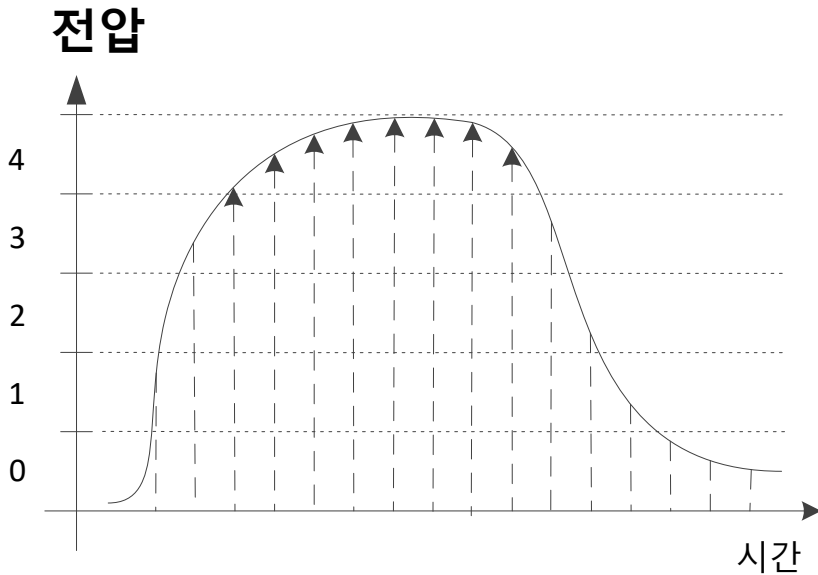
```
void loop() {  
  int swState = digitalRead(5);  
  digitalWrite(0, swState);  
  Serial.println( swState );  
}
```



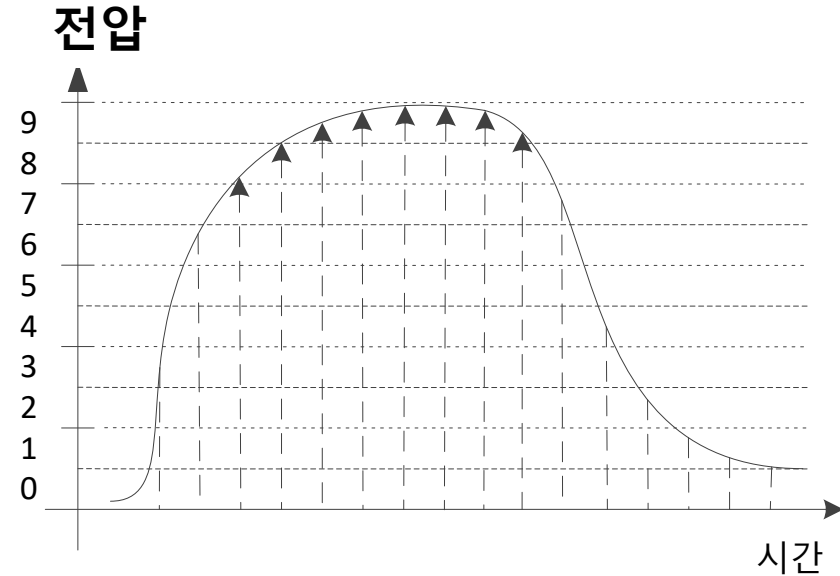
■ ADC(Analog-to-Digital Converter)

- ▶ 전압 크기를 전체 크기 등급 중 몇 번째 등급에 해당하는지 등급(칸) 번호 알려줌
- ▶ RaspberryPi는 ADC를 가지고 있지 않음.
- ▶ ADC of nodemcu : 10bit, 1024 분해능

전압 크기 등급을 5단계로 나눔



전압 크기 등급을 10단계로 나눔

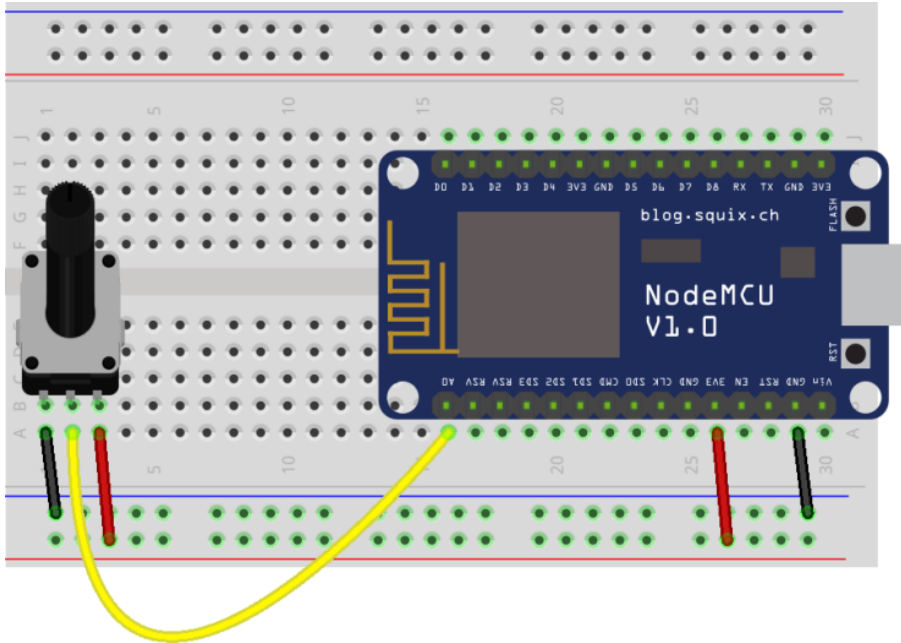


Analog Input

■ Analog Input

▶ 회로 구성

- A0 핀에 가변저항(포텐서미터) 연결



fritzing

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

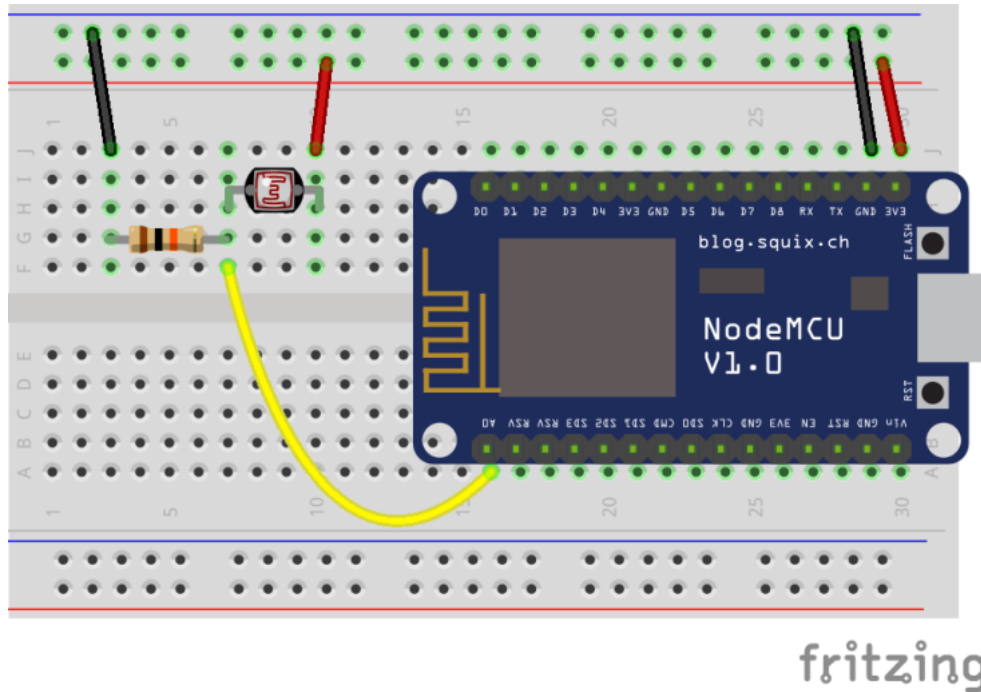
```
void loop() {  
    int signalLevel = analogRead(A0);  
    Serial.println( signalLevel );  
}
```

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

```
void loop() {  
    int signalLevel = analogRead(A0);  
    float voltage = signalLevel * 3.3/1024.0;  
    Serial.println( voltage, 4 );  
}
```

■ Analog Signal

■ Wiring



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

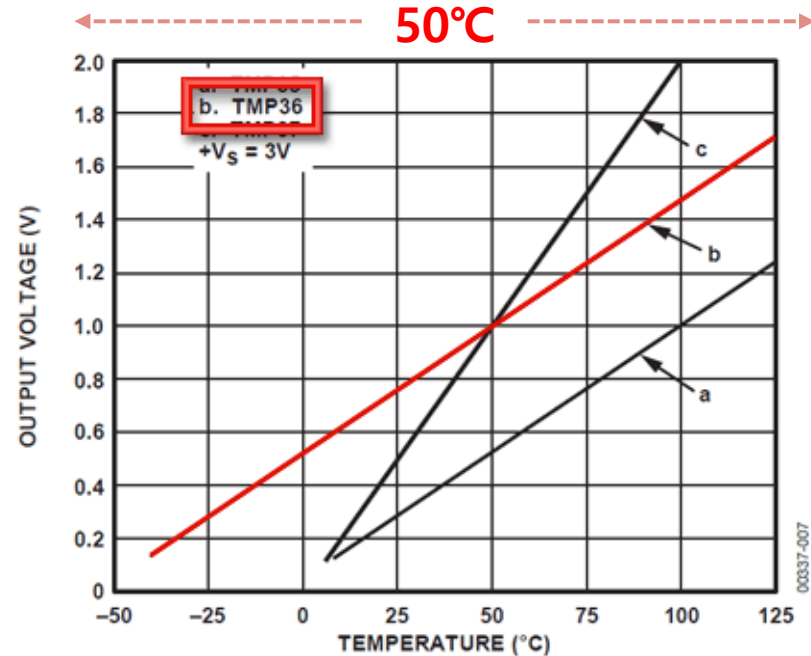
```
void loop() {  
  int signalLevel = analogRead(A0);  
  Serial.println(signalLevel);  
  delay(100);  
}
```

■ 온도 센서(TMP36)

▶ 50°C → 온도센서(TMP36)의 출력 전압 : 1[V]

1°C 감소 시
출력 전압 10[mv] 감소

1°C 상승 시
출력 전압 10[mv] 증가



[Temperature vs. Output Voltage]

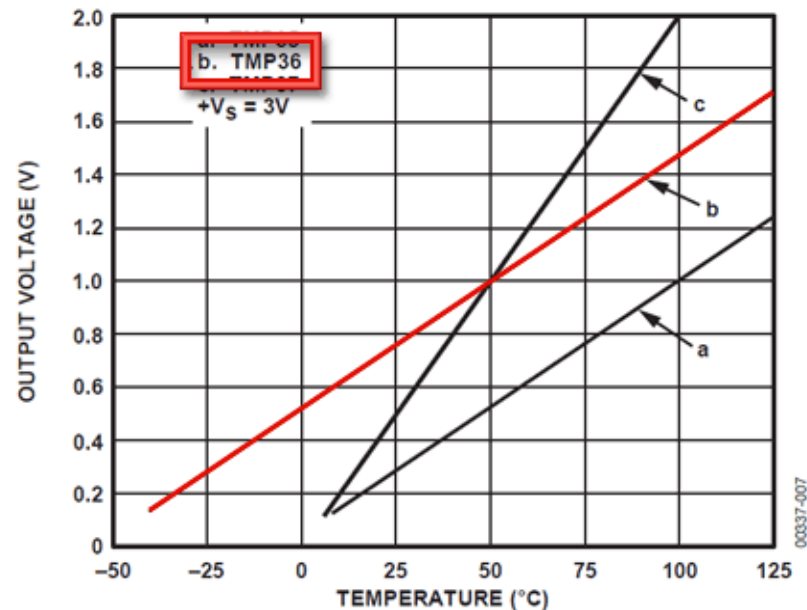
■ 온도 센서(TMP36)

▶ 전압을 온도로 환산하는 식

$$v = \frac{0.01}{1} (t - 50) + 1 = \frac{1}{100} t + \frac{50}{100}$$

$$\frac{1}{100} t = v - \frac{50}{100}$$

$$t = 100v - 50$$



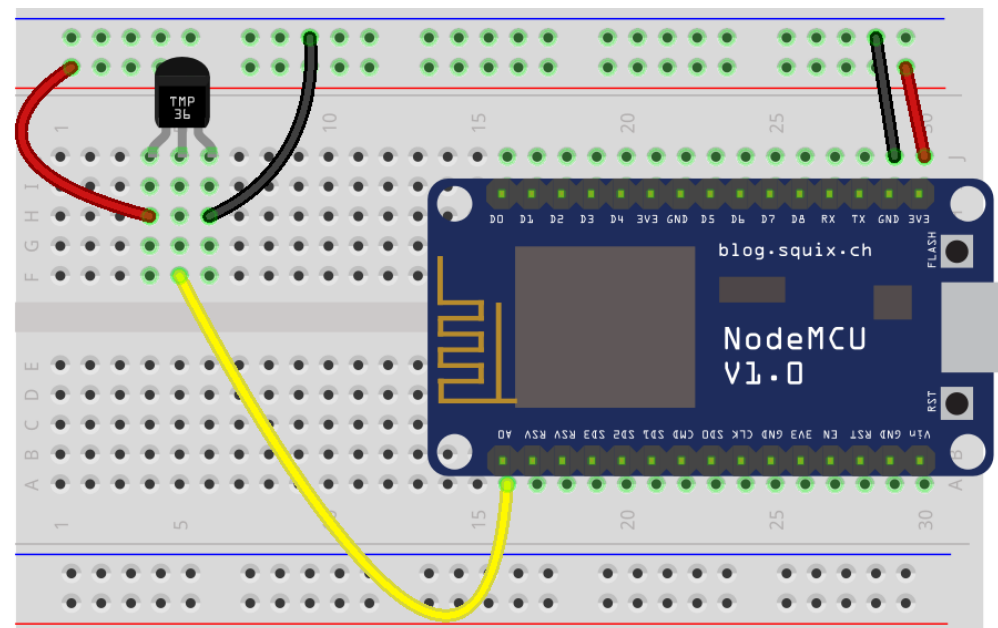
Analog Input

■ 온도센서 TMP36

▶ 회로 구성

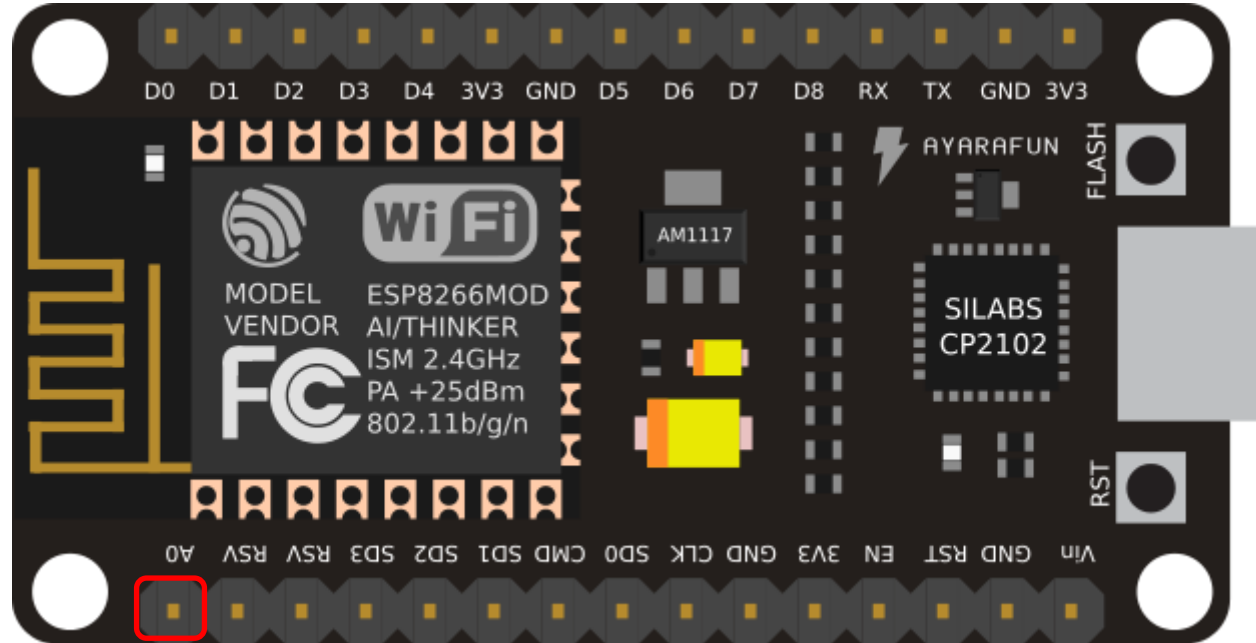
- A0 핀에 TMP36 연결

```
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  int signalLevel = analogRead(A0);  
  float voltage = signalLevel * 3.3/1024.0;  
  float temp = 100.0 * voltage - 50;  
  Serial.println(temp, 1);  
  delay(500);  
}
```



fritzing

Analog Output(PWM)



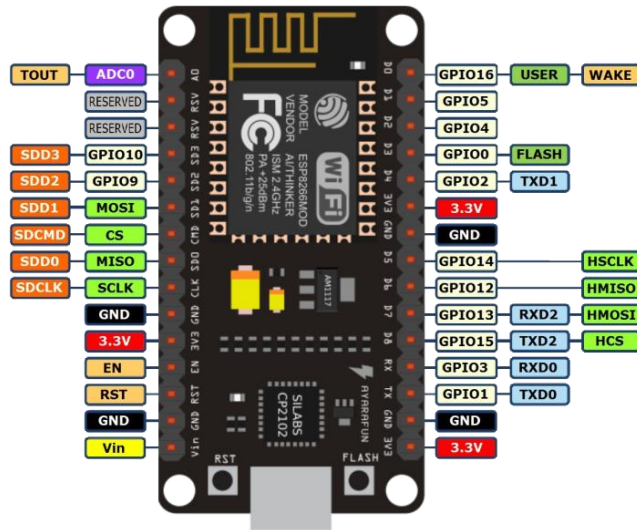
■ PWM(Pulse Width Modulation)

► Command - `analogWrite(PinNumber, TimeOn);`

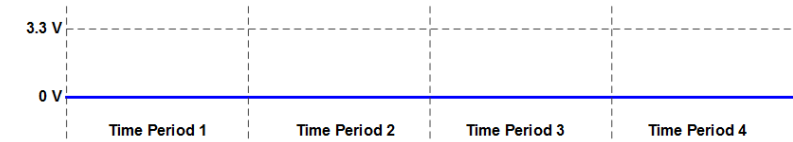
- PinNumber is **one of the digital output pins** on the NodeMCU.
 - » Pins 0 to 16
- TimeOn is **the amount of time the output is high** within a time period.
 - » TimeOn : 0 to PWMRange, which is equal to 1023 by default

► PWM Freq. – 1kHz default

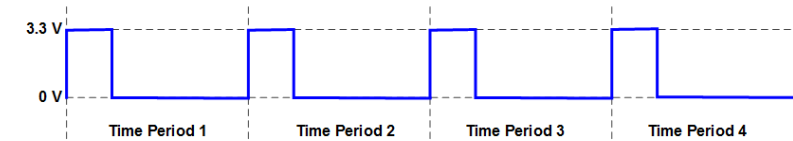
- `analogWriteFreq(new_frequency)` to change the frequency



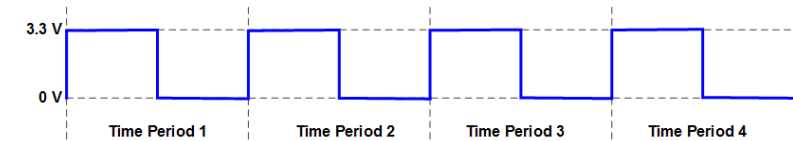
`analogWrite(LedPin, 0);`



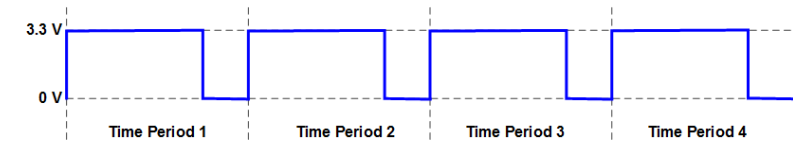
`analogWrite(LedPin, 255);`



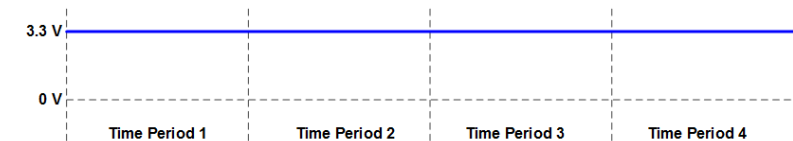
`analogWrite(LedPin, 512);`



`analogWrite(LedPin, 767);`



`analogWrite(LedPin, 1023);`

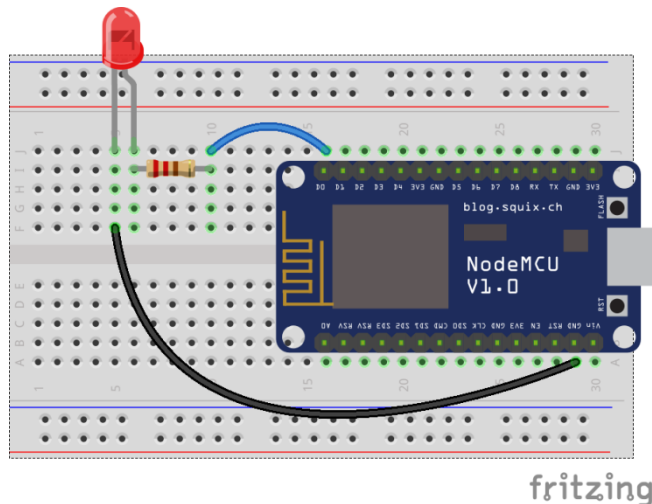


■ PWM(Pulse Width Modulation)

▶ 회로 구성

- D0 핀에 LED 연결

```
int ledPin = D0;  
void setup() {  
  pinMode(ledPin, OUTPUT);  
}  
  
void loop() {  
  analogWrite(ledPin, 512);  
  delay(2000);  
  analogWrite(ledPin, 1023);  
  delay(2000);  
}
```



```
int ledPin = D0;  
void setup() {  
  pinMode(ledPin, OUTPUT);  
  analogWriteFreq(10000);  
}  
  
void loop() {  
  analogWrite(ledPin, 512);  
  delay(2000);  
  analogWrite(ledPin, 1023);  
  delay(2000);  
}
```

■ PWM(Pulse Width Modulation)

▶ 회로 구성

- 옆 그림을 보고 회로 결선 - D0 핀에 트랜지스터의 베이스 연결

```
int motorPin = D0;  
  
void setup() {  
  pinMode(motorPin, OUTPUT);  
}  
  
void loop() {  
  analogWrite(motorPin, 512);  
  delay(2000);  
  analogWrite(motorPin, 1023);  
  delay(2000);  
}
```

