

ESP8266

IoT 서비스를 위한 MCU 기초

ESP8266

ESP8266(ESP8266EX)은 Wi-Fi 무선 통신 기능을 탑재한 저전력 마이크로컨트롤러로, 중국 상하이에 소재한 에스프레시프 시스템즈(Espressif Systems)사에서 2014년에 출시하였다. 같은 회사에서 개발한 후속 제품인 ESP32와 함께, 저렴한 가격에 IoT 시나리오를 구현할 수 있는 마이크로컨트롤러로 널리 사용되고 있다.

Specipication

하드웨어	스펙
CPU	Tensilica Xtensa® L106(Diamond Standard 106Micro) @80/160MHz (32-bit RISC)
동작전압	2.7V~3.6V
Wi-Fi(MODE)	802.11 b/g/n(HT20) 프로토콜
I/O(pinout)	17
메모리(RAM)	160KB(iRAM64KB,dRAM96KB)
메모리(SRAM)	1KB
외장플래쉬	512KB(최소사양),OTA(on the air)최소사양(1MB)~16MB까지 지원

현재 나오는 모든 제품들은 대부분 ESP8266EX 베이스

특징

Wi-Fi, TCP/IP 스택을 한꺼번에 처리 가능한 칩을 저렴한 가격에 사용가능 - 세계적인기
제조사에서 공식적으로 Non-OS SDK와 FreeRTOS SDK를 제공
커뮤니티에서 개발한 아두이노 코어 호환 SDK를 사용하여 아두이노 호환 보드처럼 사용
NodeMCU를 이용해 Lua 언어 개발가능
와이파이를 지원하지 않는 보드와 조합하여 시리얼 Wi-Fi 어댑터 용도로도 사용
Arduino UNO Wi-Fi 보드는 "우노 + ESP8266" 조합으로 구현

모듈 및 개발 보드

ESP-01

ESP-05

NodeMCU

We Mos D1

차례

아두이노 IDE 설치

관련 드라이브(ch340) 설치

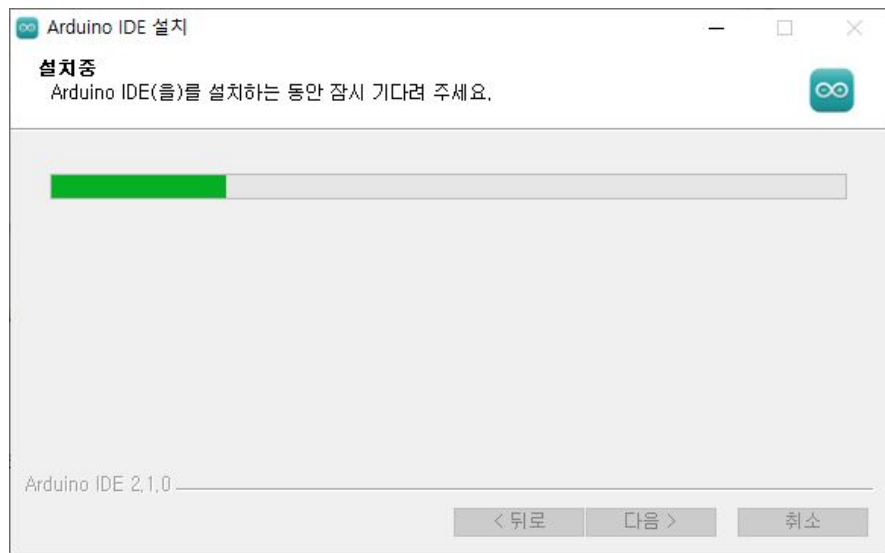
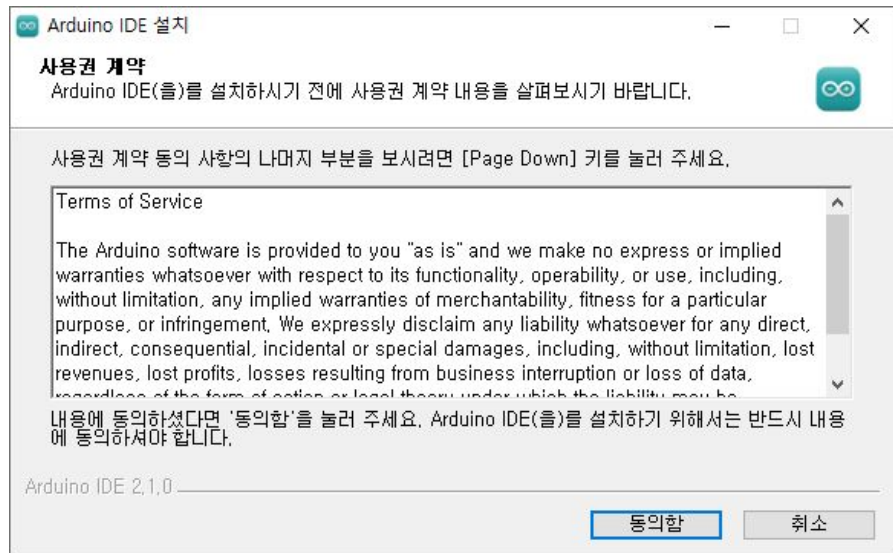
라이브러리 & 보드 업데이트

연결 후 LED Blink 코드 실행

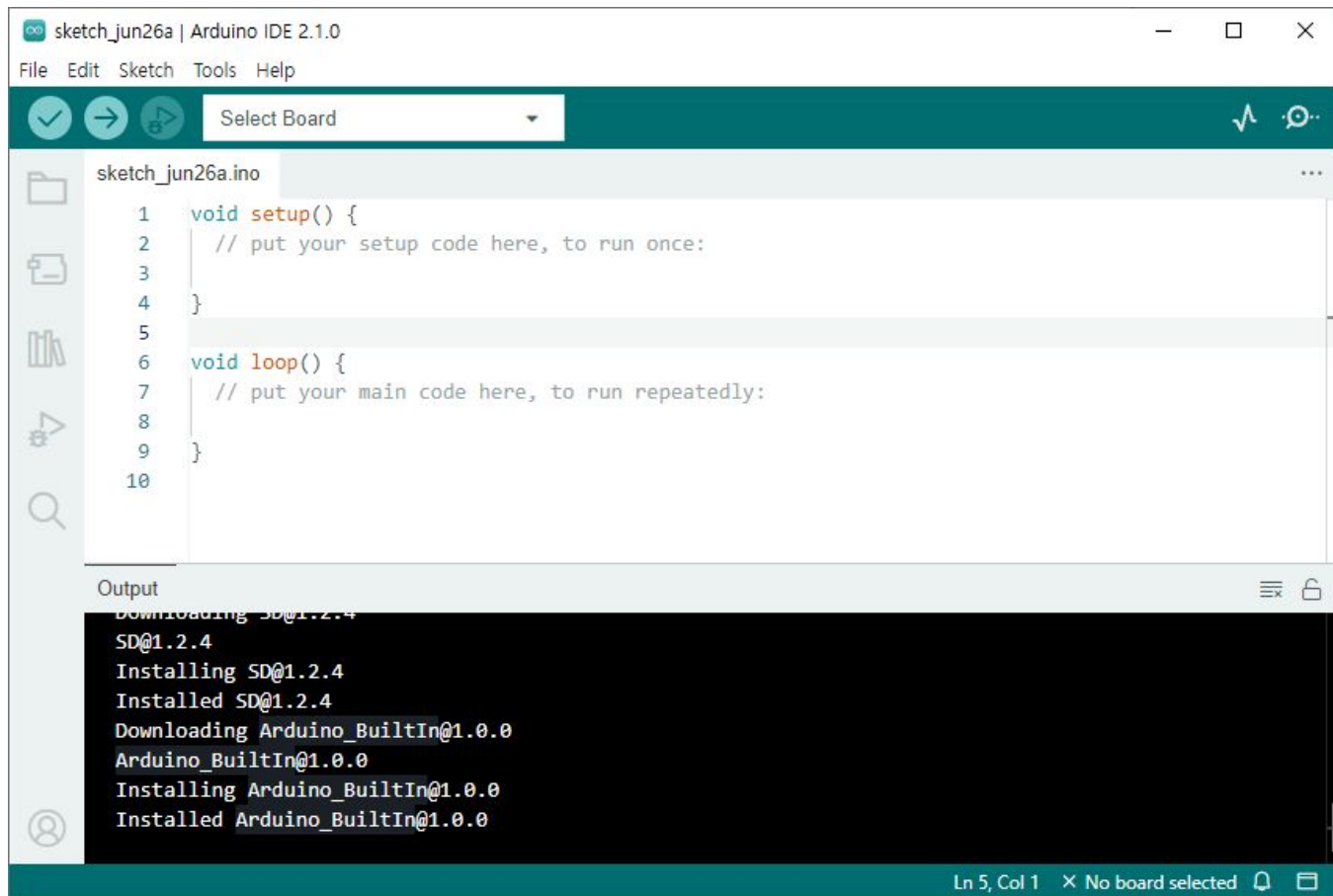
Arduino IDE 설치

아두이노 검색 후 IDE 설치

2.x 또는 1.8x 설치 - 2.x 권장(속도 빠름)



실행



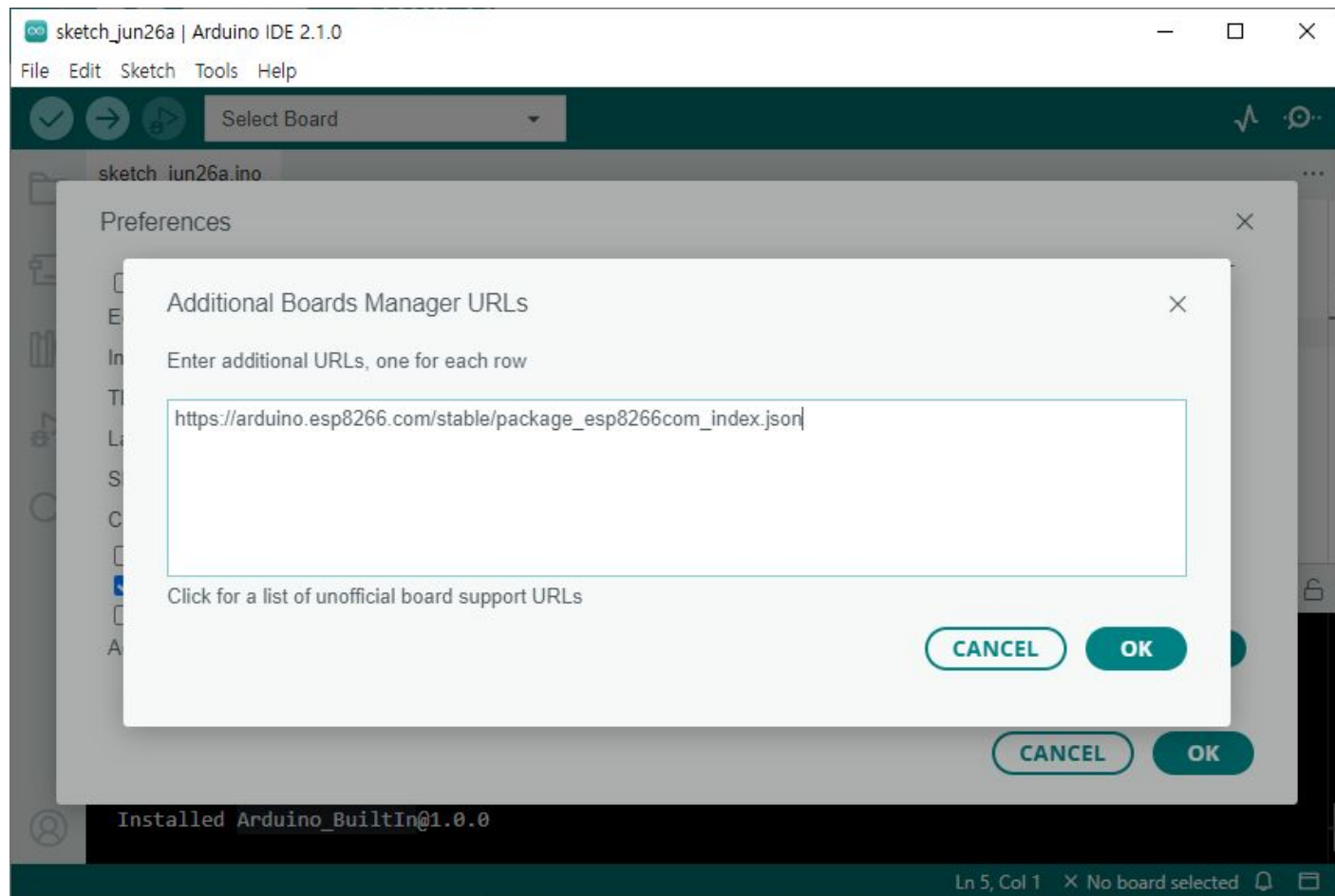
보드 추가

File – Preference – (맨아래) Additional board manager URLs:

Click for a list of unofficial board support URLs

- ESP8266 (혹은 NodeMCU) 검색
 - ESP8266 Community 의 json 복사
 - https://arduino.esp8266.com/stable/package_esp8266com_index.json
- 복사된 것은 위 박스에 추가 & OK 버튼 클릭

보드 추가



- https://arduino.esp8266.com/stable/package_esp8266com_index.json

보드 추가

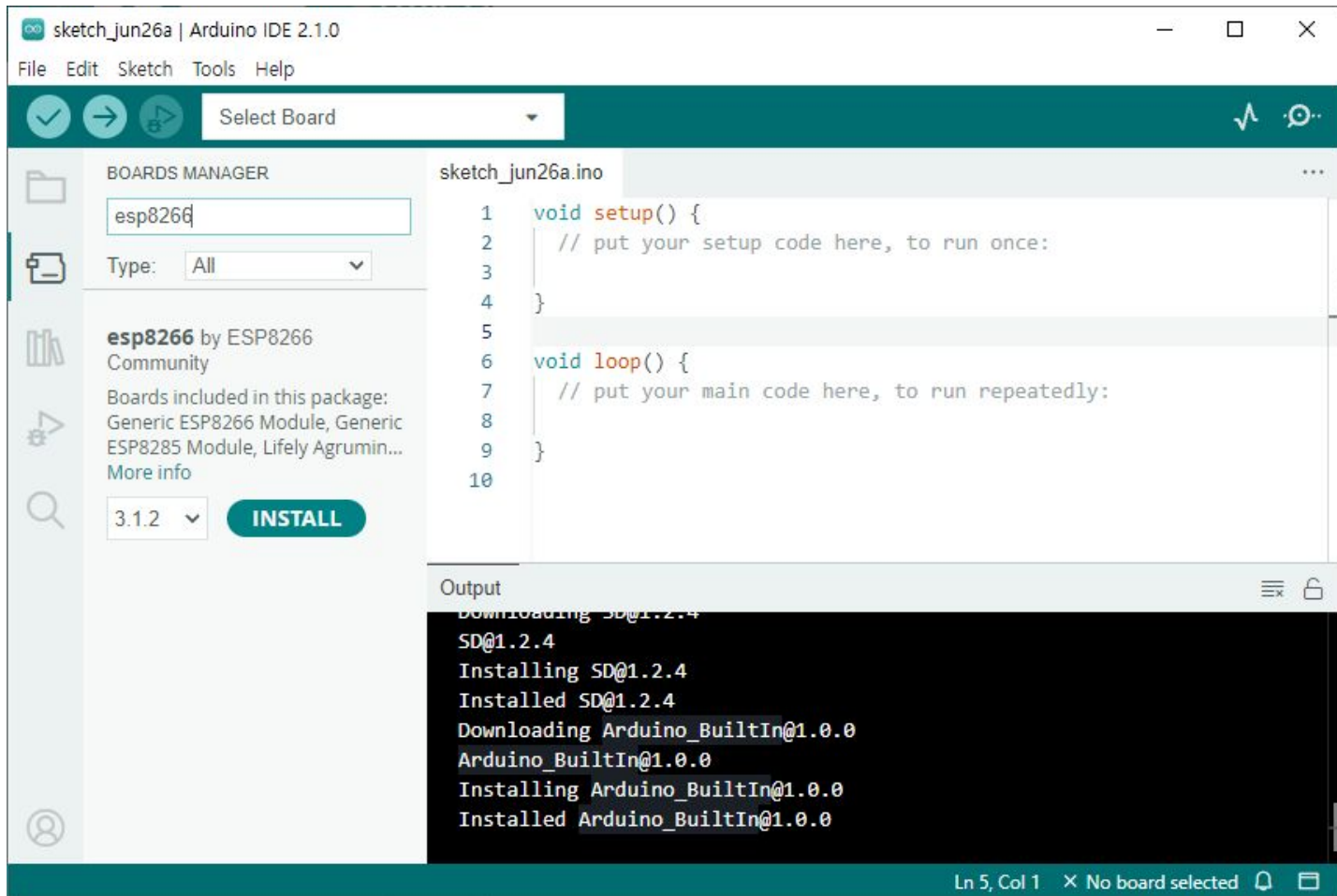
Tools ->

Board ->

Board Manager

esp8266 검색

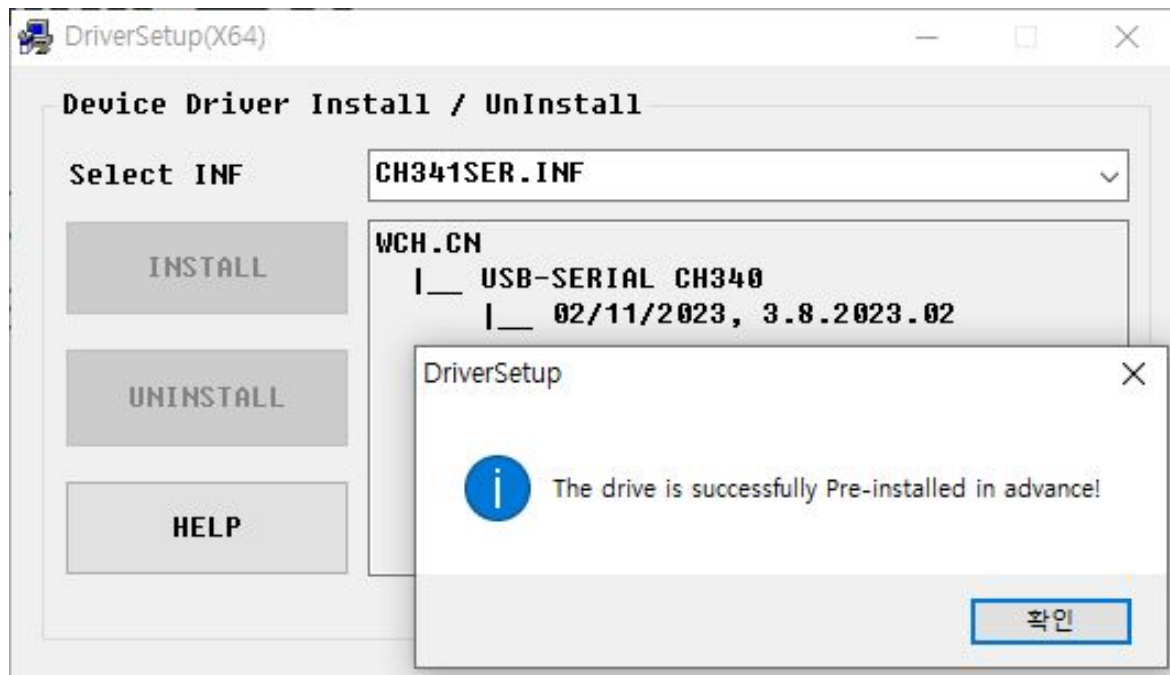
>> INSTALL



CH340 드라이버 추가

CH340 검색 : https://www.wch.cn/download/CH341SER_EXE.html

Download & Install



ESP8266 , PC 연결

USB 케이블로 연결

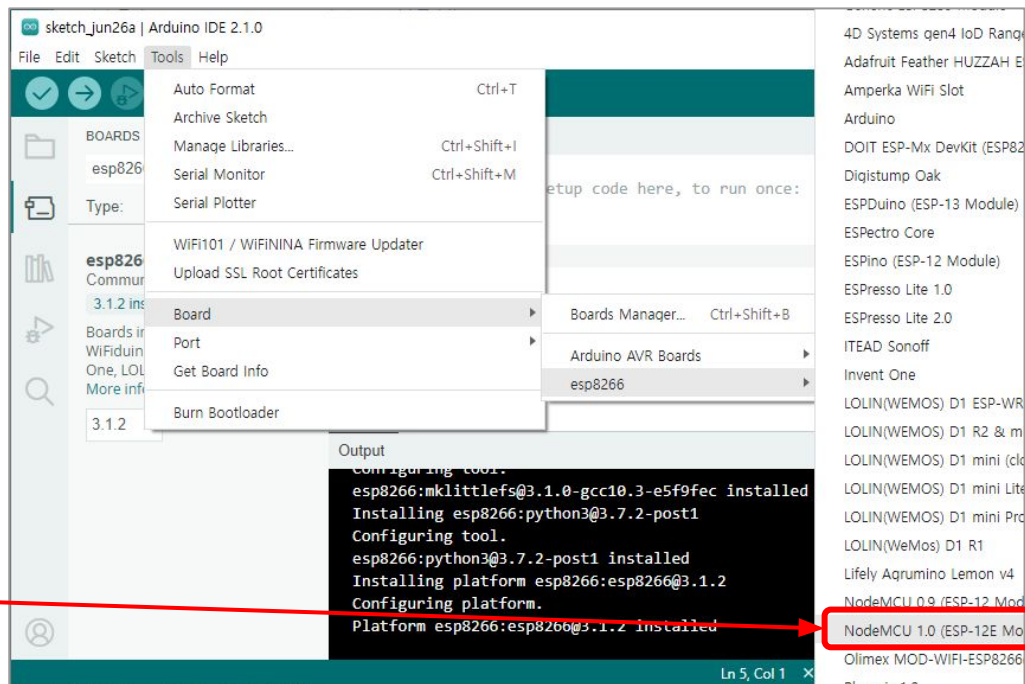
Arduino IDE 실행

Tool

>> Board

>> ESP8266

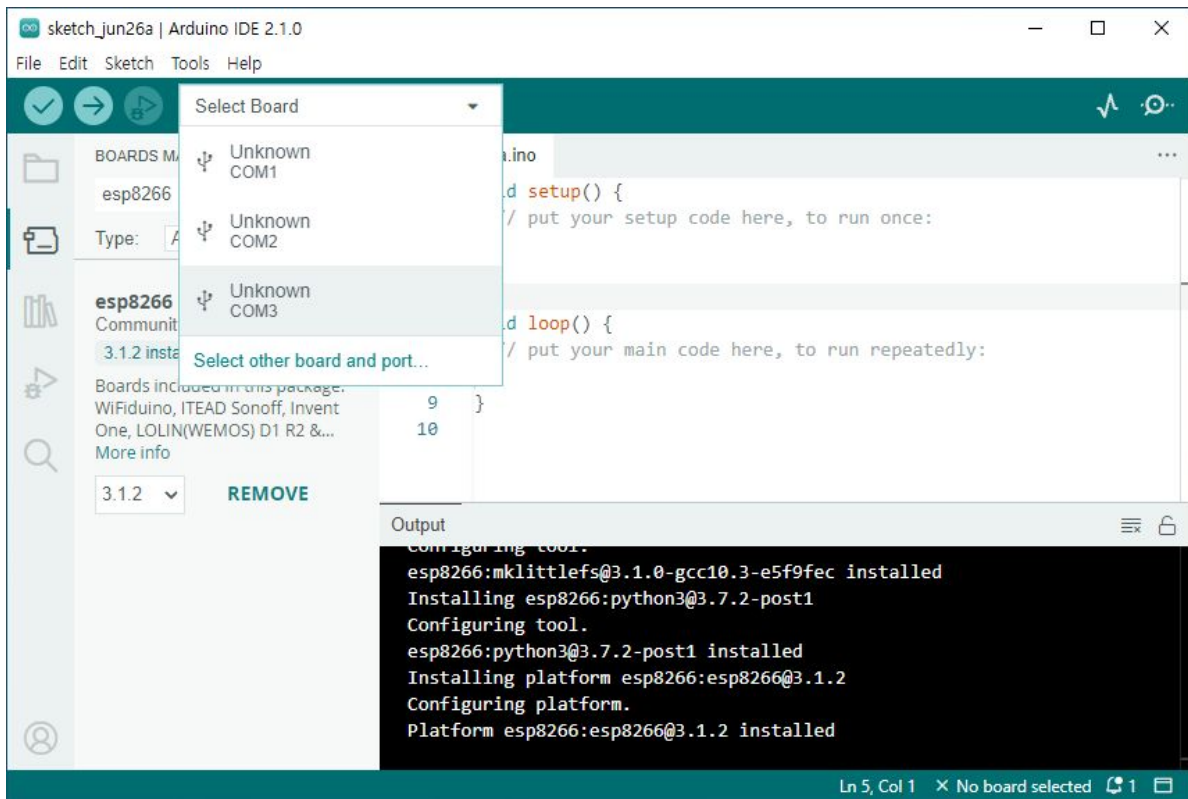
>> NodeMCU 1.0 ...



연결

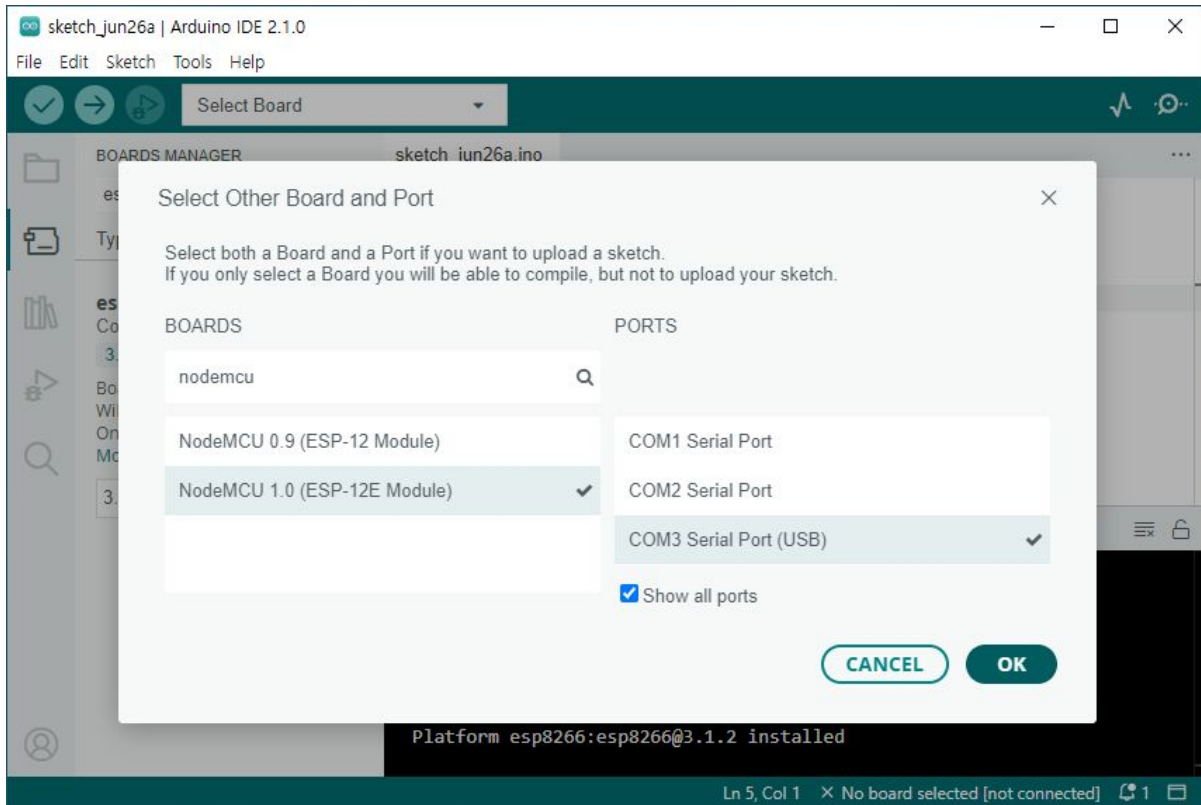
Select Board

>> COMx 로 연결



연결

보드 & 포트 선택



예제 코드 실행

File

>> Examples

>> ESP8266

>> Blink

업로드 아이콘 클릭

깜박임 시작

The screenshot shows the Arduino IDE interface with the Blink example code loaded. The code is as follows:

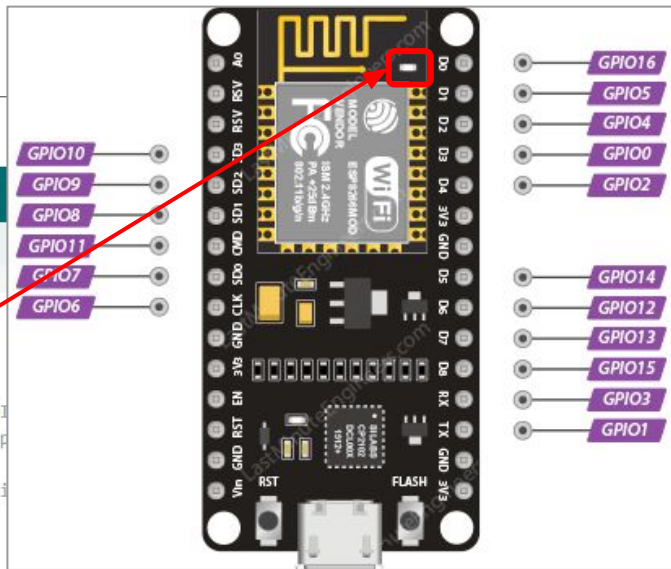
```
1  /*
2   ESP8266 Blink by Simon Peter
3   Blink the blue LED on the ESP-01 module
4   This example code is in the public domain
5
6   The blue LED on the ESP-01 module is connected to GPIO
7   (which is also the TXD pin; so we cannot use Serial.p
8
9   Note that this sketch uses LED_BUILTIN to find the pi
10  */
11
12  void setup() {
13    pinMode(LED_BUILTIN, OUTPUT); // Initialize the LED_BUILTIN pin as an output
14  }
15
```

The output window shows the upload progress:

```
Writing at 0x00020000... (75 %)
Writing at 0x00024000... (83 %)
Writing at 0x00028000... (91 %)
Writing at 0x0002c000... (100 %)
Wrote 265616 bytes (195727 compressed) at 0x00000000 in 17.2 seconds (effective 123.4 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
```

At the bottom of the IDE, the status bar indicates: "Ln 1, Col 1 NodeMCU 1.0 (ESP-12E Module) on COM3".



레퍼런스 링크

1.

<https://randomnerdtutorials.com/esp8266-nodemcu-http-get-post-arduino/#http-get-1>

2. esp8266 예제

ESP8266HTTPClient >> BasicHttpClient

3.

<https://blog.naver.com/roboholic84/221887137012>

관련 링크

KEYWORDS : arduino heart rate sensor

<https://lastminuteengineers.com/pulse-sensor-arduino-tutorial/>

<https://microcontrollerslab.com/pulse-sensor-esp8266-nodemcu-tutorial/>

Heart Rate Pulse Sensor



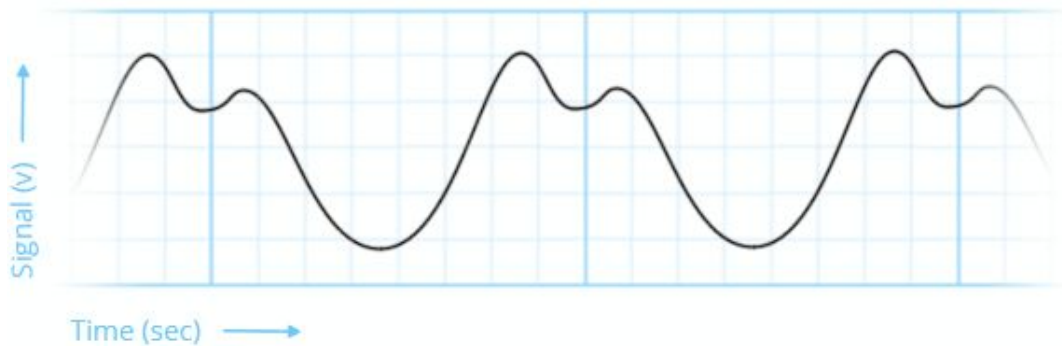
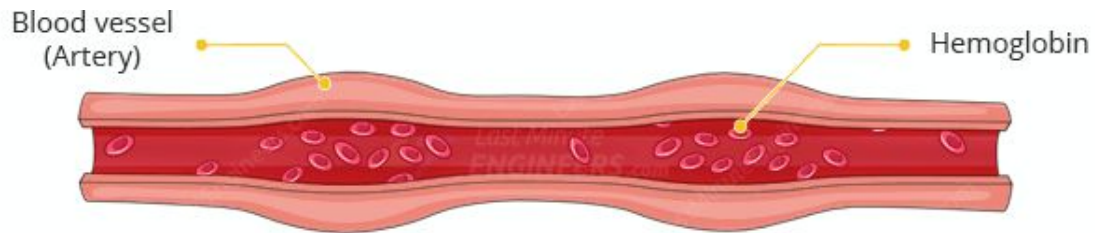
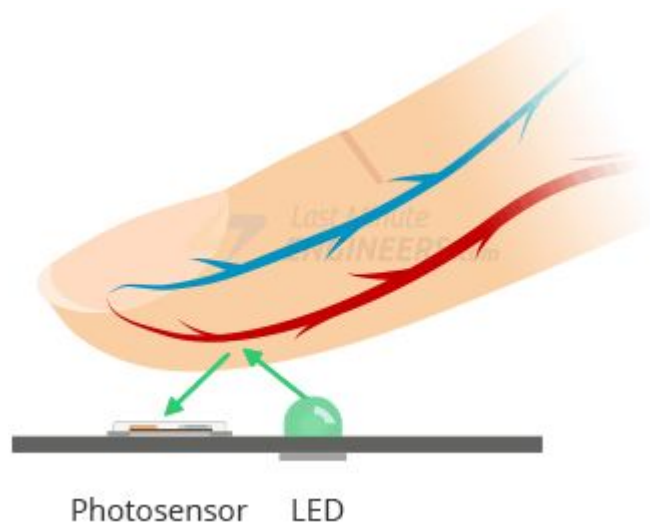
APDS-9008 Light
Photo Sensor

Reverse mount LED

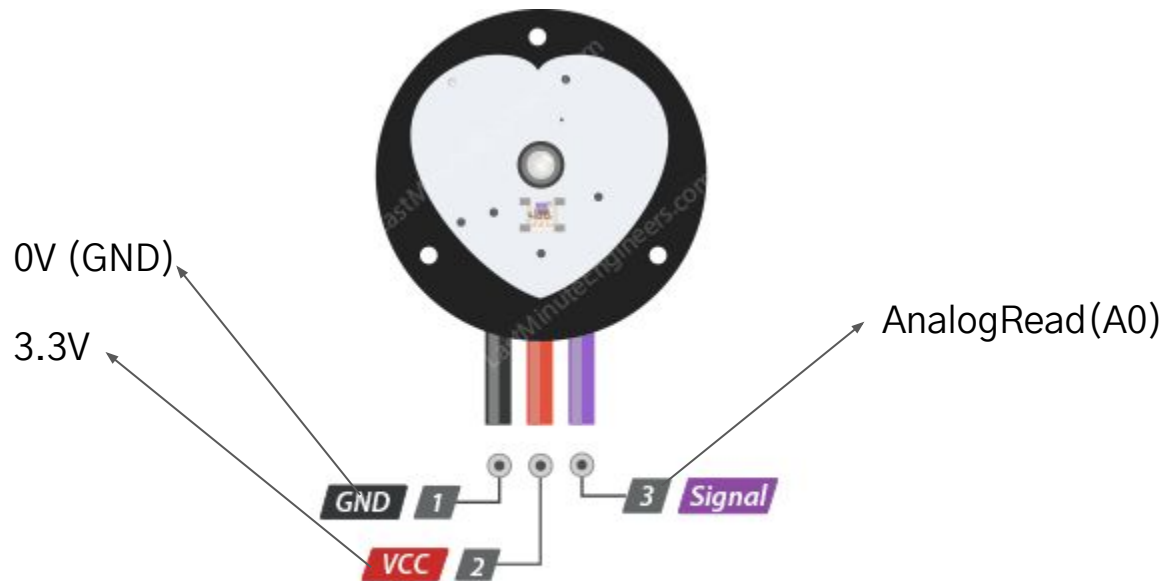
Reverse protection
diode

MCP6001 Op-Amp

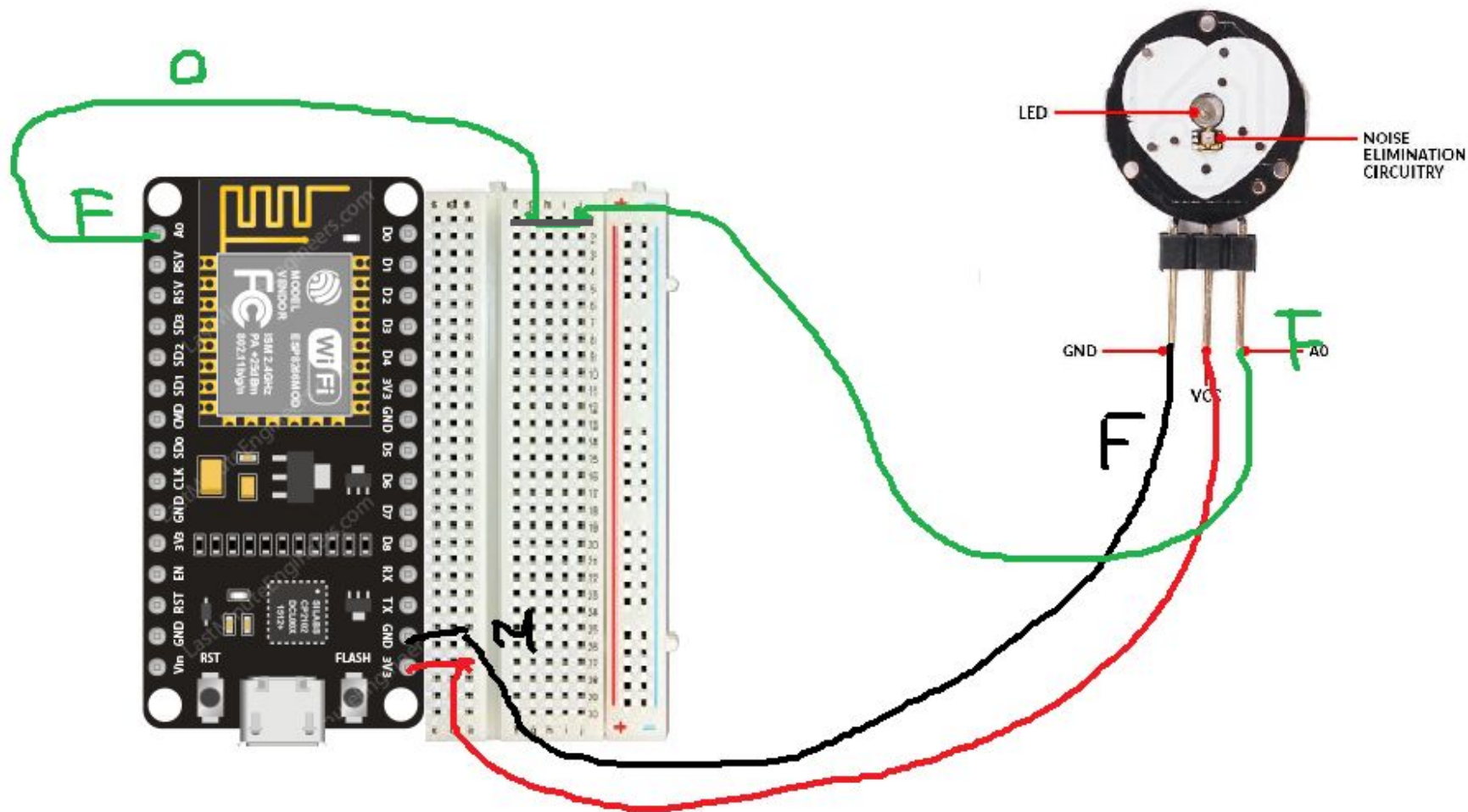
센서 작동 방식



핀아웃



Pulse Sensor Pinout



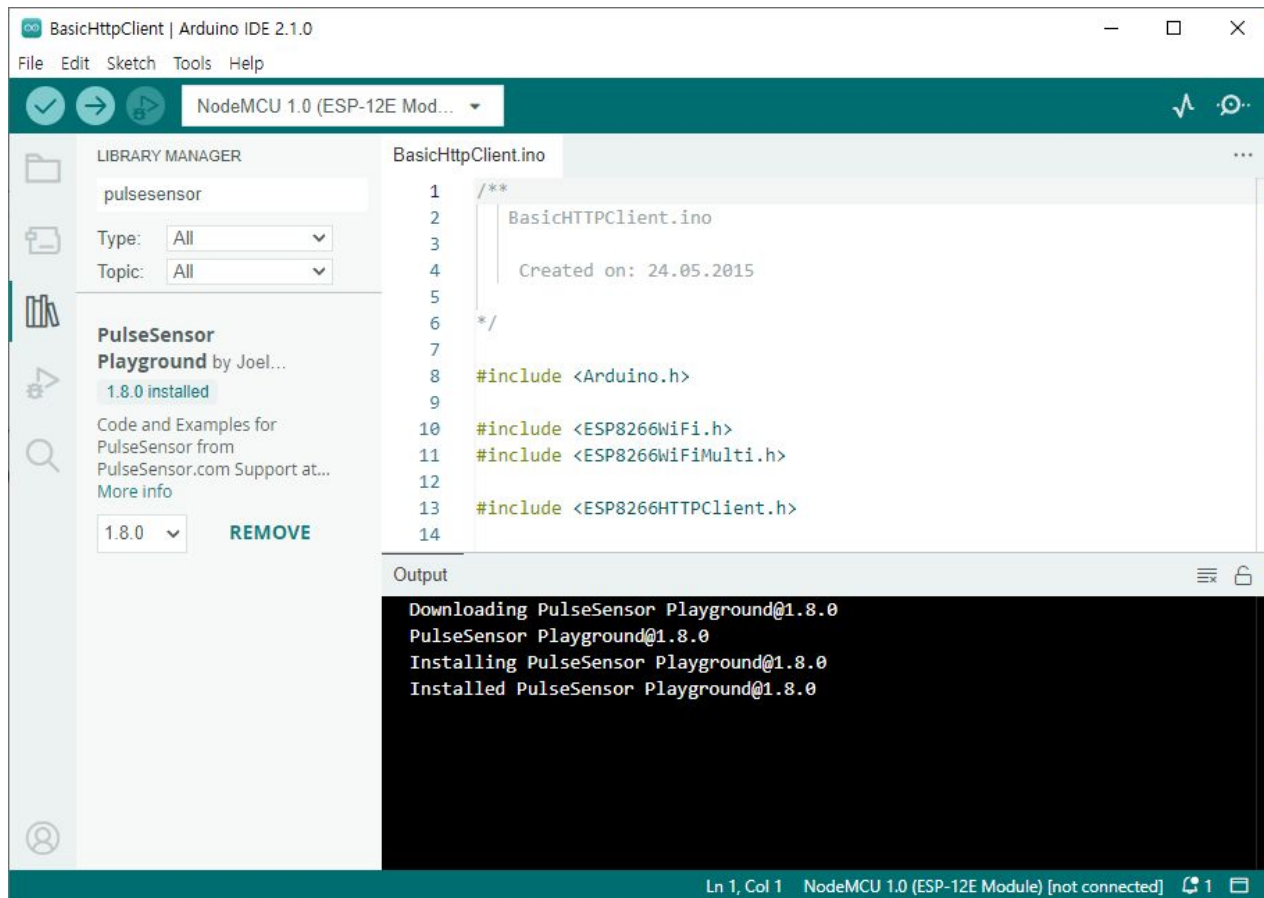
라이브러리 추가

Sketch

>> Include Library

>> Manage Libraries

pulsesensor 추가



기본 예제

파일

>> examples

>> PulseSensor Playground

>> GettingStartedProject

-> A0 값을 읽어서 시리얼로 보내는 코드



```
GettingStartedProject.ino
22
23 // Variables
24 int PulseSensorPurplePin = 0; // Pulse Sensor PURPLE WIRE connected to ANALOG PIN 0
25 int LED = LED_BUILTIN; // The on-board Arduino LED
26 int Signal; // holds the incoming raw data. Signal value can range from 0-1024
27 int Threshold = 580; // Determine which Signal to "count as a beat", and which to ignore.
28
29 // The Setup Function:
30 void setup() {
31   pinMode(LED, OUTPUT); // pin that will blink to your heartbeat!
32   Serial.begin(9600); // Set's up Serial Communication at certain speed.
33 }
34
35 // The Main Loop Function
36 void loop() {
37   Signal = analogRead(PulseSensorPurplePin); // Read the PulseSensor's value.
38   Serial.println(Signal); // Send the Signal value to Serial Plotter.
39   if(Signal > Threshold){ // If the signal is above "550", then "turn-on" Arduino's on-board LED.
40     digitalWrite(LED, HIGH);
41   } else {
42     digitalWrite(LED, LOW); // Else, the signal must be below "550", so "turn-off" this LED.
43   }
44   delay(10);
45 }
46
```

Ln 28, Col 1 NodeMCU 1.0 (ESP-12E Module) on COM3

기본 예제

파일

>> examples

>> ESP8266HTTPClient

>> BasicHttpClient

-> HTTP 프로토콜을 이용한

클라이언트 코드

```
BasicHttpClient.ino
1  /**
2   * BasicHTTPClient.ino
3   *
4   * Created on: 24.05.2015
5   *
6   */
7
8  #include <Arduino.h>
9
10 #include <ESP8266WiFi.h>
11 #include <ESP8266WiFiMulti.h>
12
13 #include <ESP8266HTTPClient.h>
14
15 #include <WiFiClient.h>
16
17 ESP8266WiFiMulti WiFiMulti;
18
19 void setup() {
20
21   Serial.begin(115200);
22   // Serial.setDebugOutput(true);
```

Ln 37, Col 1 NodeMCU 1.0 (ESP-12E Module) on COM3

코드 수정

```
WiFiMulti.addAP("4Lab", "");    // WIFI 공유기 이름과 비밀번호
```

```
if (http.begin(client, "http://http.com"))
```

```
    // http 프로토콜사용하는 사이트의 주소
```

이 부분("http://http.com")을 다음과 같은 방식으로 수정

["http://192.168.0.10:8181/iot/write?mWriter=d901&mContent="+String\(Signal\)](http://192.168.0.10:8181/iot/write?mWriter=d901&mContent=)

GET 을 이용하여 write 에 mWriter 와 mContent 에 값을 담아 전달

mWriter 는 기기번호, mContent 는 센서값

수정된 코드

```
#include <Arduino.h>

#include <ESP8266WiFi.h>
#include <ESP8266WiFiMulti.h>

#include <ESP8266HTTPClient.h>

#include <WiFiClient.h>

ESP8266WiFiMulti WiFiMulti;

int PulseSensorPurplePin = A0;
int LED = LED_BUILTIN;
int Signal;
int Threshold = 580;

void setup() {
  pinMode(LED,OUTPUT);
  Serial.begin(115200);
  // Serial.setDebugOutput(true);

  Serial.println();
  Serial.println();
  Serial.println();

  for (uint8_t t = 4; t > 0; t--) {
    Serial.printf("[SETUP] WAIT %d...\n", t);
    Serial.flush();
    delay(1000);
  }

  WiFi.mode(WIFI_STA);
  WiFiMulti.addAP("4Lab", "");
}
```

```
void loop() {
  Signal = analogRead(PulseSensorPurplePin);
  if(Signal > Threshold){
    digitalWrite(LED,HIGH); // OFF
  } else {
    digitalWrite(LED,LOW); // ON
  }

  // wait for WiFi connection
  if ((WiFiMulti.run() == WL_CONNECTED)) {

    WiFiClient client;

    HTTPClient http;

    Serial.print("[HTTP] begin...\n");
    if (http.begin(client, "http://192.168.0.10:8181/iot/write?mWriter=d901&mContent="+String(Signal))) {

      Serial.print("[HTTP] GET...\n");
      // start connection and send HTTP header
      int httpCode = http.GET();

      // httpCode will be negative on error
      if (httpCode > 0) {
        // HTTP header has been send and Server response header has been handled
        Serial.printf("[HTTP] GET... code: %d\n", httpCode);

        // file found at server
        if (httpCode == HTTP_CODE_OK || httpCode == HTTP_CODE_MOVED_PERMANENTLY) {
          String payload = http.getString();
          Serial.println(payload);
        }
      } else {
        Serial.printf("[HTTP] GET... failed, error: %s\n", http.errorToString(httpCode).c_str());
      }

      http.end();
    } else {
      Serial.println("[HTTP] Unable to connect");
    }
  }

  delay(30000);
}
```

스프링 게시판

기존에 만들어 둔 게시판을 사용

- mId, mWriter, mContent 항목을 가진 작은 게시판을 사용
- 글을 쓸때 ... write?mWriter=deviceNumber&mContent=SensorValue 를 이용

작동 방식

- ESP8266 (IoT디바이스)
 - GET 방식으로 서버에 센서값 전송
 - 여러 디바이스가 접속할 수 있도록 기기별 ID 부여
- 스프링 서버
 - ESP8266 에서 오는 값과 디바이스 ID를 DB에 저장

이후 할 작업

1. 맥박수 계산

단순한 센서 값을 읽는 것에서 마치기보다는 센서값을 해석하여 실제 맥박수를 계산하고, 맥박수를 전달하게 한다.

2. 부가 정보 추가

시간 정보와 건강에 대한 다른 정보(온도, 습도, 미세먼지 등)를 추가하여 이후 데이터 사용에 연관성을 부여할 수 있게 한다.

3. 클라우드 서비스에 서버 올리기