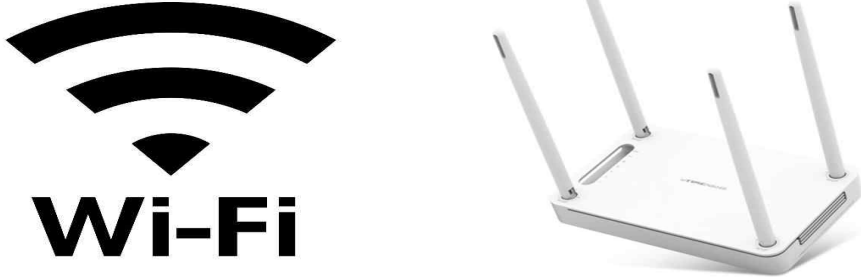



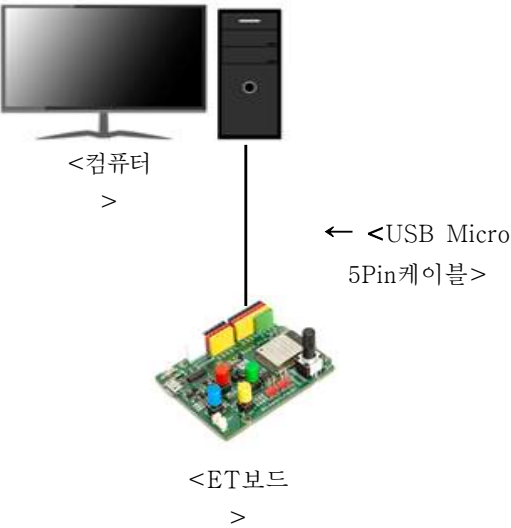


ET보드 level2 콘텐츠

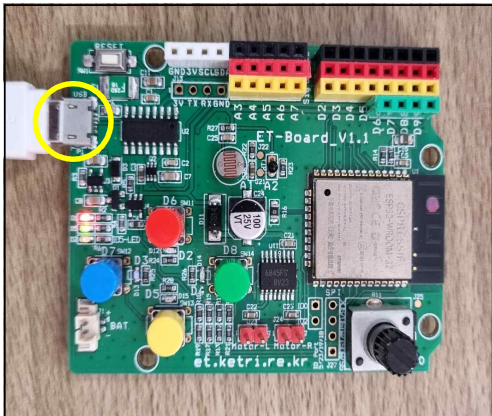
211130 최수범

학습내용	WiFi에 연결하여 받은 IP주소를 출력 해보기
소 스	01._WiFi_IPAddress_print
개 념	 <p>와이파이는 일상생활에서 흔하게 볼 수 있습니다.</p>
준비물	   <p><컴퓨터> <USB Micro 5Pin 케이블> <ET보드></p>
회 로 구 성	 <p><컴퓨터> ></p> <p>← <USB Micro 5Pin케이블></p> <p><ET보드> ></p>

외
로
구
성



① 컴퓨터 본체의 USB포트에 케이블을 연결 합니다.



② 다른 한쪽 케이블을 ET보드에 연결 합니다.

소
스
코
드

// 이 라이브러리를 통해 Arduino 보드를 인터넷에 연결할 수 있습니다.

```
#include <WiFi.h>
```

```
const char *ssid      = "ssid";
```

// 와이파이 아이디

```
const char *password = "password";
```

// 와이파이 비밀번호

```
void setup()
```

```
{
```

```
  Serial.begin(115200);
```

// 시리얼 통신속도 설정

```
  delay(10);
```

// 0.01초 기다리기

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

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

```
  Serial.print("Connecting to ");
```

```
  Serial.println(ssid);
```

//ssid와 password를 이용해서 와이파이에 접속을 시도

```
  WiFi.begin(ssid, password);
```

```

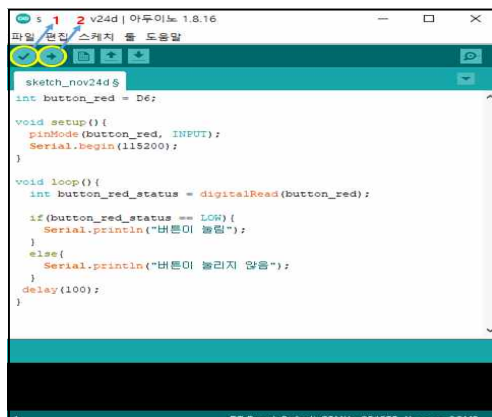
//연결이 될 때까지 계속 대기
while(WiFi.status() != WL_CONNECTED){
    delay(500);
    Serial.print(".");
}

Serial.println("");
Serial.println("WiFi connected");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());    //연결이 됐다면 할당받은 아이피를 출력함
}

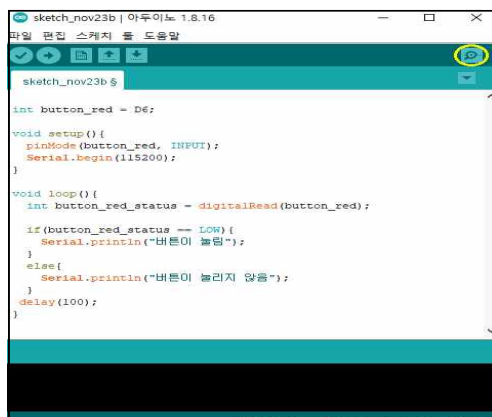
void loop()
{
    delay(10);
}

```

동작과정






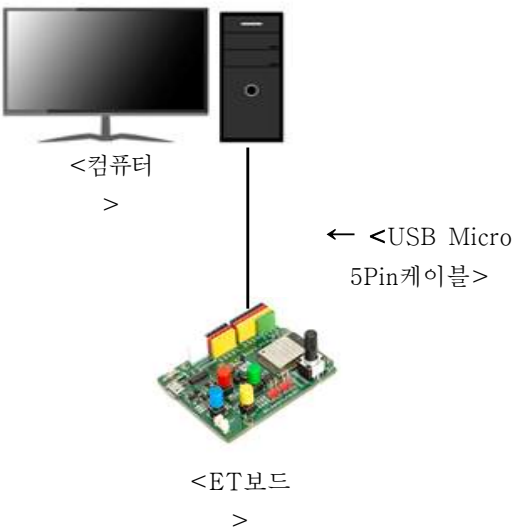


①아두이노 코드입력 후 상단 1[확인] 아이콘을 클릭하고 완료되면 2[업로드] 아이콘도 클릭 해 펌웨어를 ET보드에 업로드 합니다.



② 업로드까지 마치면 우측상단 돋보기 모양의 “시리얼모니터”를 클릭 해 줍니다.

<p>동작 과정</p>		<p>③ 시리얼 모니터에 IP가 출력 됩니다.</p>
<p>참고 사항</p>	<p>① 회로 ○ 와이파이에 대한 참고사항 링크 https://ko.wikipedia.org/wiki/%EC%99%80%EC%9D%B4%ED%8C%8C%EC%9D%B4</p> <p>② 소스코드 ○ 와이파이에 사용법에 대한 설명 문서 https://docs.espressif.com/projects/arduino-esp32/en/latest/api/wifi.html</p>	

학습내용	간단한 문자열을 보여주는 서버 만들어 보기
소스	02._WiFi_simple_web_server
개념	  <p>와이파이는 일상생활에서 흔하게 볼 수 있습니다.</p>
준비물	   <p><컴퓨터> <USB Micro 5Pin 케이블> <ET보드></p>
회로구성	 <p><컴퓨터> ></p> <p>← <USB Micro 5Pin케이블></p> <p><ET보드> ></p>

<div data-bbox="143 609 220 685" data-label="Section-Header"> <h2>외로구성</h2> </div>	<div data-bbox="301 217 794 629" data-label="Image"> </div> <div data-bbox="868 383 1356 461" data-label="List-Group"> <ol style="list-style-type: none"> ① 컴퓨터 본체의 USB포트에 케이블을 연결 합니다. </div> <div data-bbox="301 667 794 1079" data-label="Image"> </div> <div data-bbox="868 855 1431 891" data-label="List-Group"> <ol style="list-style-type: none"> ② 다른 한쪽 케이블을 ET보에 연결 합니다. </div>
<div data-bbox="143 1503 220 1579" data-label="Section-Header"> <h2>소스코드</h2> </div>	<div data-bbox="248 1144 1209 1180" data-label="Text"> <p>// 이 라이브러리를 통해 Arduino 보드를 인터넷에 연결할 수 있습니다.</p> </div> <div data-bbox="248 1191 612 1312" data-label="Text"> <pre>#include <WiFi.h> #include <WiFiClient.h> #include <WebServer.h></pre> </div> <div data-bbox="248 1364 1139 1447" data-label="Text"> <pre>const char *ssid = "ssid"; // 와이파이 아이디 const char *password = "password"; // 와이파이 비밀번호</pre> </div> <div data-bbox="248 1500 1254 1536" data-label="Text"> <pre>WebServer server(80); // 서버에서 사용할 포트 설정</pre> </div> <div data-bbox="248 1590 1150 1626" data-label="Text"> <pre>const int led = D2; // 빨간 LED의 핀 번호</pre> </div> <div data-bbox="248 1680 1409 1935" data-label="Text"> <pre>void handleRoot(){ // root(/)로 접속했을 때 처리하는 함수 digitalWrite(led, HIGH); // LED를 켜기 Serial.println("root call!"); // 페이지로 접속했다고 시리얼로 알려줌 server.send(200, "text/plain", "hello from ET-board!"); digitalWrite(led, LOW); // LED를 끄기 }</pre> </div>

```

void setup(void){
    pinMode(led, OUTPUT);           // LED를 출력상태로 설정
    digitalWrite(led, LOW);         // LED를 끈 상태로 설정
    Serial.begin(115200);           // 시리얼 통신속도 설정
    WiFi.mode(WIFI_STA);           // 연결모드로 설정
    WiFi.begin(ssid, password);     // WiFi에 접속을 시도
    Serial.println("");

    // 연결이 될 때까지 계속 대기
    while(WiFi.status() != WL_CONNECTED){
        delay(500);
        Serial.print(".");
    }
    Serial.println("");
    Serial.print("Connected to ");
    Serial.println(ssid);
    Serial.print("IP address: ");
    // 연결이 됐다면 할당받은 아이피를 출력함
    Serial.println(WiFi.localIP());

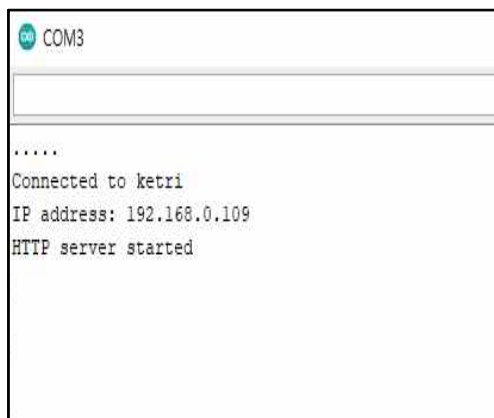
    // root(/)로 접속했을 때 처리하는 함수랑 연결
    server.on("/", handleRoot);

    server.begin();                 // 서버 시작
    Serial.println("HTTP server started");
}

void loop(void) {
    server.handleClient();           // 클라이언트의 접속을 받음
    delay(2);
}





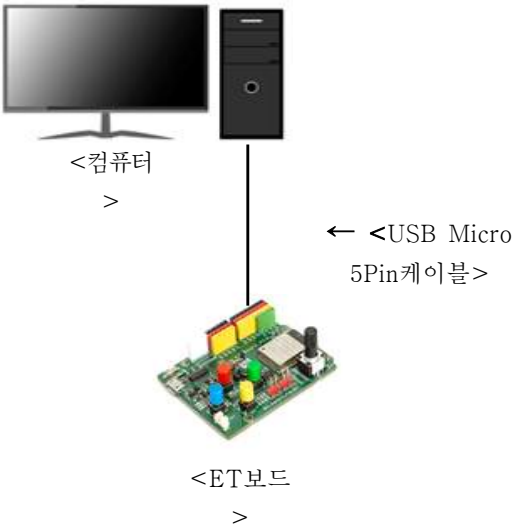
```

동작
과정



① 시리얼 모니터에 IP가 출력됩니다.

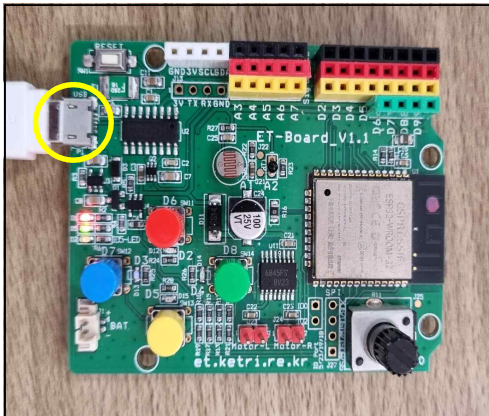
	<div data-bbox="304 219 799 633" data-label="Image"> </div> <div data-bbox="868 322 1422 445" data-label="Text"> <p>② 주소란에 할당받은 IP주소를 입력하면 이미지와 같이 “hello from ET-board!” 문구가 뜹니다.</p> </div>
<div data-bbox="145 882 220 958" data-label="Section-Header"> <p>참 고 사 항</p> </div>	<div data-bbox="250 712 1321 1131" data-label="List-Group"> <ul style="list-style-type: none"> ① 회로 <ul style="list-style-type: none"> ○ 와이파이예 대한 참고사항 링크 https://ko.wikipedia.org/wiki/%EC%99%80%EC%9D%B4%ED%8C%8C%EC%9D%B4 ○ 웹서버예 대한 참고사항 링크 https://ko.wikipedia.org/wiki/%EC%9B%B9_%EC%84%9C%EB%B2%84 ② 소스코드 <ul style="list-style-type: none"> ○ 와이파이예 사용법에 대한 설명 문서 https://docs.espressif.com/projects/arduino-esp32/en/latest/api/wifi.html </div>

학습내용	가변저항 값을 보여주는 웹서버 만들어 보기
소스	03._WiFi_web_server_variable_resistance
개념	 <p>와이파이는 일상생활에서 흔하게 볼 수 있습니다.</p>
준비물	   <p><컴퓨터> <USB Micro 5Pin 케이블> <ET보드></p>
회로구성	 <p><컴퓨터> ></p> <p>← <USB Micro 5Pin케이블></p> <p><ET보드> ></p>

외
로
구
성



① 컴퓨터 본체의 USB포트에 케이블을 연결 합니다.



② 다른 한쪽 케이블을 ET보에 연결 합니다.

소
스
코
드

```
// 이 라이브러리를 통해 Arduino 보드를 인터넷에 연결할 수 있습니다.
#include <WiFi.h>
#include <WiFiClient.h>
#include <WebServer.h>

const char * ssid = "ssid";           // 와이파이 아이디
const char * password = "password";   // 와이파이 비밀번호

WebServer server(80);                  // 서버에서 사용할 포트 설정

const int led = D2;                    // 빨강 LED의 핀 번호

// root(/)로 접속했을 때 처리하는 함수
void handleRoot(){
    digitalWrite(led, HIGH);           // LED를 켜기
    Serial.println("root call!");      // 페이지로 접속했다고 시리얼로 알려줌
    server.send(200, "text/plain", "hello from ET-board!");
    digitalWrite(led, LOW);            // LED를 끄기
}
```

```

// read_a0(/read_a0)로 접속했을 때 처리하는 함수
void handleA0(){
    int sensorValue = analogRead(A0);           // 가변 저항의 값을 읽어옴
    String sendData = "variable_resistance : "; // 단순 문자열 저장
    sendData = sendData + String(sensorValue);  // 출력할 값을 만들어냄
    // 페이지로 접속했다고 시리얼로 알려줌
    Serial.println("A0 call!");
    server.send(200, "text/plain", sendData);
}

void setup(void){
    pinMode(led, OUTPUT);           // LED를 출력상태로 설정
    digitalWrite(led, LOW);         // LED를 끈 상태로 설정
    Serial.begin(115200 );          // 시리얼 통신속도 설정
    WiFi.mode(WIFI_STA);           // 연결모드로 설정
    WiFi.begin(ssid, password);     // WiFi에 접속을 시도
    Serial.println("");

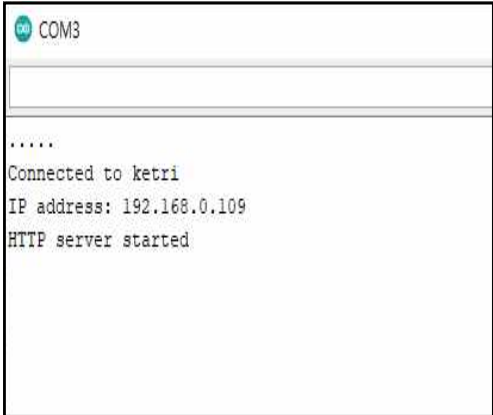


    // 연결이 될 때까지 계속 대기
    while(WiFi.status() != WL_CONNECTED){
        delay(500);
        Serial.print(".");
    }
    Serial.println("");
    Serial.print("Connected to ");
    Serial.println(ssid);
    Serial.print("IP address: ");
    // 연결이 됐다면 할당받은 아이피를 출력함
    Serial.println(WiFi.localIP());

    // root(/)로 접속했을 때 처리하는 함수랑 연결
    server.on("/", handleRoot);
    server.on("/read_a0", handleA0);

    // 서버 시작
    server.begin();
    Serial.println("HTTP server started");
}

void loop(void){

```

	<pre>server.handleClient(); // 클라이언트의 접속을 받음 delay(2); }</pre>
동작과정	<div data-bbox="304 383 799 797">  </div> <div data-bbox="884 555 1417 629"> <p>① 할당 받은 IP주소를 인터넷창에 입력하여 접속합니다.</p> </div> <div data-bbox="304 835 799 1249">  </div> <div data-bbox="884 1025 1477 1059"> <p>② IP주소 뒤에 /read_a0 입력하고 접속 합니다.</p> </div> <div data-bbox="304 1288 799 1702">  </div> <div data-bbox="884 1458 1334 1532"> <p>③ 현재 ET보드의 가변저항값을 확인 할 수 있습니다.</p> </div>
참고사항	<p>① 회로</p> <ul style="list-style-type: none"> ○ 와이파이예 대한 참고사항 링크 https://ko.wikipedia.org/wiki/%EC%99%80%EC%9D%B4%ED%8C%8C%EC%9D%B4 ○ 웹서버에 대한 참고사항 링크 https://ko.wikipedia.org/wiki/%EC%9B%B9_%EC%84%9C%EB%B2%84

② 소스코드

- 와이파이 사용법에 대한 설명 문서

<https://docs.espressif.com/projects/arduino-esp32/en/latest/api/wifi.html>

학습내용	html 페이지에서 버튼을 눌러 LED를 켜고 꺼 보기
소 스	04._WiFi_led_control
개 념	<div data-bbox="370 519 708 779" data-label="Image"> </div> <div data-bbox="858 488 1235 797" data-label="Image"> </div> <p>와이파이는 일상생활에서 흔하게 볼 수 있습니다.</p>
준비물	<div data-bbox="312 1028 601 1167" data-label="Image"> </div> <div data-bbox="671 994 879 1198" data-label="Image"> </div> <div data-bbox="979 1016 1198 1187" data-label="Image"> </div> <p><컴퓨터> <USB Micro 5Pin 케이블> <ET보드></p>
회 로 구 성	<div data-bbox="312 1382 601 1523" data-label="Image"> </div> <div data-bbox="375 1525 469 1590" data-label="Text"> <p><컴퓨터> ></p> </div> <div data-bbox="632 1594 831 1668" data-label="Text"> <p>← <USB Micro 5Pin케이블></p> </div> <div data-bbox="467 1675 651 1818" data-label="Image"> </div> <div data-bbox="496 1839 608 1904" data-label="Text"> <p><ET보드> ></p> </div>

<div data-bbox="143 607 220 685" data-label="Section-Header"> <h2>외로구성</h2> </div>	<div data-bbox="301 217 796 631" data-label="Image"> </div> <div data-bbox="868 383 1356 461" data-label="List-Group"> <ol style="list-style-type: none"> ① 컴퓨터 본체의 USB포트에 케이블을 연결 합니다. </div> <div data-bbox="301 667 796 1081" data-label="Image"> </div> <div data-bbox="868 855 1431 891" data-label="List-Group"> <ol style="list-style-type: none"> ② 다른 한쪽 케이블을 ET보에 연결 합니다. </div>
<div data-bbox="143 1503 220 1579" data-label="Section-Header"> <h2>소스코드</h2> </div>	<div data-bbox="248 1144 1209 1180" data-label="Text"> <p>// 이 라이브러리를 통해 Arduino 보드를 인터넷에 연결할 수 있습니다.</p> </div> <div data-bbox="248 1191 612 1312" data-label="Text"> <pre>#include <WiFi.h> #include <WiFiClient.h> #include <WebServer.h></pre> </div> <div data-bbox="248 1364 1153 1447" data-label="Text"> <pre>const char * ssid = "ssid"; // 와이파이 아이디 const char * password = "password"; // 와이파이 비밀번호</pre> </div> <div data-bbox="248 1500 1268 1536" data-label="Text"> <pre>WebServer server(80); // 서버에서 사용할 포트 설정</pre> </div> <div data-bbox="248 1588 1495 1803" data-label="Text"> <pre>const int led = D2; // 빨강 LED의 핀번호 char htmlPage[] = "Click red On to turn On LED
\" \"Click red Off to turn Off LED
\";</pre> </div> <div data-bbox="248 1897 810 1982" data-label="Text"> <pre>// root(/)로 접속했을 때 처리하는 함수 void handleRoot(){</pre> </div>

```

Serial.println("root call!"); // 페이지로 접속했다고 시리얼로 알려줌
server.send(200, "text/html", htmlPage);
}

// red_led_on(/red_led_on)로 접속했을 때 처리하는 함수
void handleD2On(){
    Serial.println("D2 On call!"); // 페이지로 접속했다고 시리얼로 알려줌
    digitalWrite(led, HIGH);
    server.send(200, "text/html", htmlPage);
}

// red_led_off(/red_led_off)로 접속했을 때 처리하는 함수
void handleD2Off(){
    Serial.println("D2 Off call!"); // 페이지로 접속했다고 시리얼로 알려줌
    digitalWrite(led, LOW);
    server.send(200, "text/html", htmlPage);
}

void setup(void){
    pinMode(led, OUTPUT);           // LED를 출력상태로 설정
    digitalWrite(led, LOW);         // LED를 끈 상태로 설정
    Serial.begin(115200);           // 시리얼 통신속도 설정
    WiFi.mode(WIFI_STA);           // 연결모드로 설정
    WiFi.begin(ssid, password);    // WiFi에 접속을 시도
    Serial.println("");

    // 연결이 될 때까지 계속 대기
    while(WiFi.status() != WL_CONNECTED){
        delay(500);
        Serial.print(".");
    }
    Serial.println("");
    Serial.print("Connected to ");
    Serial.println(ssid);
    Serial.print("IP address: ");

    // 연결이 됐다면 할당받은 아이피를 출력함
    Serial.println(WiFi.localIP());

    // root(/)로 접속했을 때 처리하는 함수랑 연결
    server.on("/", handleRoot);






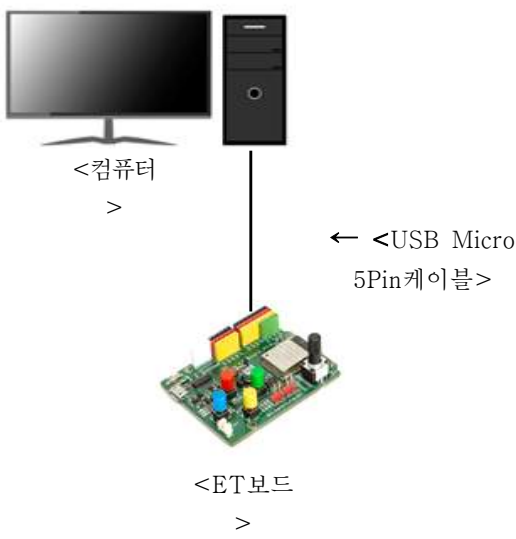
```


	<pre> // red_led_on(/red_led_on)로 접속했을 때 처리하는 함수랑 연결 server.on("/red_led_on", handleD2On); // red_led_off(/red_led_off)로 접속했을 때 처리하는 함수랑 연결 server.on("/red_led_off", handleD2Off); server.begin(); // 서버 시작 Serial.println("HTTP server started"); } void loop(void){ server.handleClient(); // 클라이언트의 접속을 받음 delay(2); } </pre>
<p>동 작 과 정</p>	<div data-bbox="304 831 799 1245">  </div> <div data-bbox="868 999 1422 1075"> <p>① 할당 받은 IP주소를 인터넷창에 입력하여 접속 합니다.</p> </div> <div data-bbox="304 1279 799 1693">  </div> <div data-bbox="868 1429 1437 1550"> <p>② 접속한 페이지에서 red On을 클릭하면 빨간색 LED가 켜지고 red Off를 클릭하면 빨간색 LED가 꺼집니다.</p> </div>
<p>참 고 사 항</p>	<p>① 회로</p> <ul style="list-style-type: none"> ○ 와이파이에 대한 참고사항 링크 https://ko.wikipedia.org/wiki/%EC%99%80%EC%9D%B4%ED%8C%8C%EC%9D%B4 ○ 웹서버에 대한 참고사항 링크 https://ko.wikipedia.org/wiki/%EC%9B%B9_%EC%84%9C%EB%B2%84

② 소스코드

- 와이파이 사용법에 대한 설명 문서

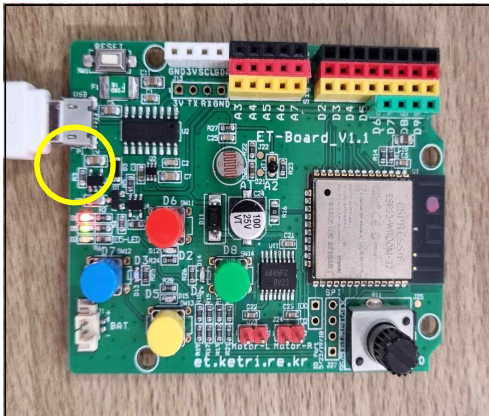
<https://docs.espressif.com/projects/arduino-esp32/en/latest/api/wifi.html>

학습내용	와이파이를 이용해 4개의 LED를 켜다 켜다하는 웹서버 만들어 보기
소스	05._WiFi_led_control_all
개념	  <p>와이파이는 일상생활에서 흔하게 볼 수 있습니다.</p>
준비물	   <p><컴퓨터> <USB Micro 5Pin 케이블> <ET보드></p>
회로구성	 <p><컴퓨터> ></p> <p>← <USB Micro 5Pin케이블></p> <p><ET보드> ></p>

외
로
구
성



① 컴퓨터 본체의 USB포트에 케이블을 연결 합니다.



② 다른 한쪽 케이블을 ET보에 연결 합니다.

소
스
코
드

// 이 라이브러리를 통해 Arduino 보드를 인터넷에 연결할 수 있습니다.

```
#include <WiFi.h>
```

```
int led2 = D2;
```

// 빨강 LED

```
int led3 = D3;
```

// 파랑 LED

```
int led4 = D4;
```

// 초록 LED

```
int led5 = D5;
```

// 노랑 LED

```
const char * ssid = "ssid";
```

// 와이파이 아이디

```
const char * password = "password";
```

// 와이파이 비밀번호

```
WiFiServer server(80);
```

// 지정된 포트에서 들어오는 연결을 수신하는 서버를 만듭니다.

```
void setup()
```

```
{
```

```
  Serial.begin(115200);
```

// 시리얼 통신속도 설정

```
  pinMode(led2, OUTPUT);
```

// 핀 모드 설정

```

pinMode(led3, OUTPUT);           // 핀 모드 설정
pinMode(led4, OUTPUT);           // 핀 모드 설정
pinMode(led5, OUTPUT);           // 핀 모드 설정
delay(10);                       // 0.01초 기다리기

// WiFi network에 접속

Serial.println();
Serial.println();
Serial.print("Connected to ");
Serial.println(ssid);

WiFi.begin(ssid, password);
//ssid와 password를 이용해서 와이파이에 접속을 시도

while(WiFi.status() != WL_CONNECTED){
    delay(500);
    Serial.print(".");
}
// 만약 WiFi 상태가 연결이 되지 않았다면 반복 딜레이는 500을 시리얼에서 출력
Serial.println("");
Serial.println("WiFi Connected."); // 출력
Serial.print("IP address: ");      // 출력
Serial.println(WiFi.localIP());
// 연결이 됐다면 할당받은 아이피를 출력함

server.begin();                   // 서버 시작
}

int value = 0;

void loop(){
    WiFiClient client = server.available(); // 새로운 클라이언트 생성

    if (client) {                  // 클라이언트가 만들어짐
        Serial.println("New Client."); // 시리얼 포트에 쓰고
        String currentLine = "";     // 클라이언트에서 받을 문자열 변수준비
        while (client.connected()) { // 연결될 때까지...
            if (client.available()) { // 클라이언트에서 문자를 받으면
                char c = client.read();

```

```

// 문자를 byte 단위로 읽고, 문자로 저장

Serial.write(c); // 시리얼 모니터에 표시
if (c == '\n') { // 문자가 개행문자이면..

    // 현재 라인에 아무것도 없으면, 행당 2개라인을 가짐
    // 이는 client HTTP 응답의 마지막이라는 것이므로 응답을 전송할 수 있다:
    if (currentLine.length() == 0) {
        // HTTP 헤더는 항상 응답코드 (e.g. HTTP/1.1 200 OK)로 시작하고
        // content-type이 나오고 빈칸이 나옴
        client.println("HTTP/1.1 200 OK");
        client.println("Content-type:text/html");
        client.println();

        // HTTP 응답은 다음과 같은 헤더를 가진다:
        client.print("<meta charset=\"UTF-8\">");
        client.print("<font size=16>Click <a href=\"/red_on\"> red On </a>
to turn On LED<br></font>");
        client.print("<font size=16>Click <a href=\"/red_off\"> red Off</a>
to turn Off LED<br></font>");
        client.print("<font size=16>Click <a href=\"/blue_on\"> blue On</a>
to turn On LED<br></font>");
        client.print("<font size=16>Click <a href=\"/blue_off\"> blue
Off</a> to turn Off LED<br></font>");
        client.print("<font size=16>Click <a href=\"/green_on\"> green
On</a> to turn On LED<br></font>");
        client.print("<font size=16>Click <a href=\"/green_off\"> green
Off</a> to turn Off LED<br></font>");
        client.print("<font size=16>Click <a href=\"/yellow_on\"> yellow
On</a> to turn On LED<br></font>");
        client.print("<font size=16>Click <a href=\"/yellow_off\"> yellow
Off</a> to turn Off LED<br></font>");

        // HTTP 응답은 빈칸으로 끝난다.
        client.println();
        // while loop를 종료
        break;
    } else { // 새로운 라인을 가지면, 현재라인 지움:
        currentLine = "";
    }
} else if (c != '\r') { // carriage return 문자라면

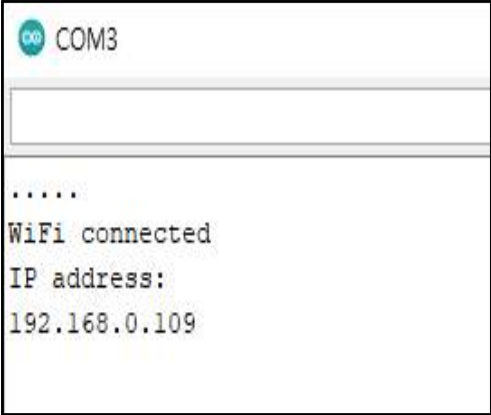
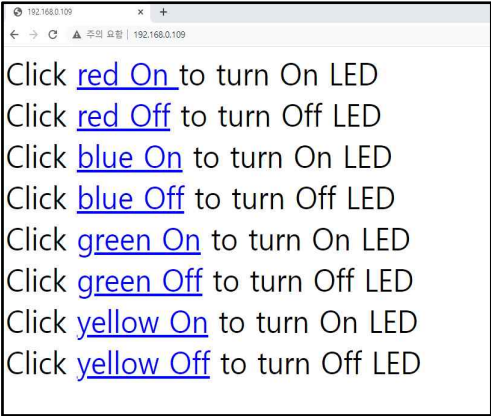
```

```

        currentLine += c;                                // 현재라인 뒤에 붙임
    }
    // 클라이언트 응답이 "GET /H" 또는 "GET /L" 였는지 확인:
    // 레드
    if (currentLine.endsWith("GET /red_on")) {
        digitalWrite(led2, HIGH);                        // GET /H LED 켜
    }
    if (currentLine.endsWith("GET /red_off")) {
        digitalWrite(led2, LOW);                          // GET /L LED 끄
    }
    // 블루
    if (currentLine.endsWith("GET /blue_on")) {
        digitalWrite(led3, HIGH);                        // GET /L LED 켜
    }
    if (currentLine.endsWith("GET /blue_off")) {
        digitalWrite(led3, LOW);                          // GET /L LED 끄
    }
    // 그린
    if (currentLine.endsWith("GET /green_on")) {
        digitalWrite(led4, HIGH);                        // GET /L LED 켜
    }
    if (currentLine.endsWith("GET /green_off")) {
        digitalWrite(led4, LOW);                          // GET /L LED 끄
    }
    // 옐로우
    if (currentLine.endsWith("GET /yellow_on")) {
        digitalWrite(led5, HIGH);                        // GET /L LED 켜
    }
    if (currentLine.endsWith("GET /yellow_off")) {
        digitalWrite(led5, LOW);                          // GET /L LED 끄
    }
}

// 연결 종료:
client.stop();
Serial.println("Client Disconnected.");
}
}

```

<p>동작 과정</p>	<div data-bbox="306 215 799 629">  </div> <div data-bbox="306 667 799 1081">  </div> <div data-bbox="871 383 1452 913"> <p>① 할당 받은 IP주소를 인터넷창에 입력하여 접속 합니다.</p> <p>② 접속한 페이지에서 red On을 클릭하면 빨간색 LED가 켜지고 red Off를 클릭하면 빨간색 LED가 꺼집니다.(3개의 LED도 확인)</p> </div>
<p>참고 사항</p>	<p>① 회로</p> <ul style="list-style-type: none"> ○ 와이파이에 대한 참고사항 링크 https://ko.wikipedia.org/wiki/%EC%99%80%EC%9D%B4%ED%8C%8C%EC%9D%B4 ○ 웹서버에 대한 참고사항 링크 https://ko.wikipedia.org/wiki/%EC%9B%B9_%EC%84%9C%EB%B2%84 <p>② 소스코드</p> <ul style="list-style-type: none"> ○ 와이파이에 사용법에 대한 설명 문서 https://docs.espressif.com/projects/arduino-esp32/en/latest/api/wifi.html