



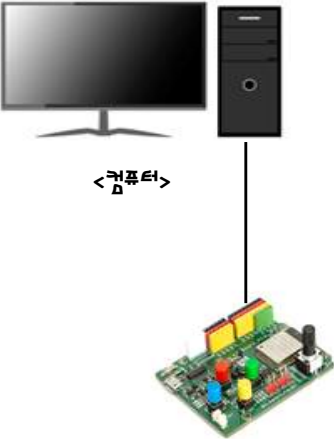


ET보드 level2 콘텐츠

211130 최수범

학습내용	WiFi에 연결하여 받은 IP주소를 출력 해보기
소스	01._WiFi_IPAddress_print.ino
개념	 <p>와이파이는 일상생활에서 흔하게 볼 수 있습니다.</p>
준비물	   <p><컴퓨터> <USB Micro 5Pin 케이블> <ET보드></p>
회로구성	 <p><컴퓨터> ← <USB Micro 5Pin 케이블> <ET보드></p>

<div data-bbox="140 607 220 687" data-label="Section-Header"> <h2>회로 구성</h2> </div>	<div data-bbox="304 215 799 631" data-label="Image"> </div> <div data-bbox="871 383 1362 463" data-label="List-Group"> <ol style="list-style-type: none"> ① 컴퓨터 본체의 USB포트에 케이블을 연결합니다. </div> <div data-bbox="304 667 799 1084" data-label="Image"> </div> <div data-bbox="871 855 1450 893" data-label="List-Group"> <ol style="list-style-type: none"> ② 다른 한쪽 케이블을 ET보드에 연결합니다. </div>
<div data-bbox="140 1503 220 1583" data-label="Section-Header"> <h2>소스 코드</h2> </div>	<div data-bbox="247 1187 1211 1227" data-label="Text"> <p>// 이 라이브러리를 통해 Arduino 보드를 인터넷에 연결할 수 있습니다.</p> </div> <div data-bbox="247 1234 521 1270" data-label="Text"> <pre>#include <WiFi.h></pre> </div> <div data-bbox="247 1321 1043 1361" data-label="Text"> <pre>const char *ssid = "와이파이 아이디를 입력하세요";</pre> </div> <div data-bbox="247 1364 592 1404" data-label="Text"> <p>// 와이파이 아이디 입력</p> </div> <div data-bbox="247 1411 1075 1449" data-label="Text"> <pre>const char *password = "와이파이 비밀번호를 입력하세요";</pre> </div> <div data-bbox="247 1453 619 1494" data-label="Text"> <p>// 와이파이 비밀번호 입력</p> </div> <div data-bbox="247 1545 443 1583" data-label="Text"> <pre>void setup()</pre> </div> <div data-bbox="247 1592 277 1630" data-label="Text"> <pre>{</pre> </div> <div data-bbox="277 1632 612 1673" data-label="Text"> <pre> Serial.begin(115200);</pre> </div> <div data-bbox="277 1677 443 1718" data-label="Text"> <pre> delay(10);</pre> </div> <div data-bbox="849 1632 1197 1673" data-label="Text"> <p>// 시리얼 통신속도 설정</p> </div> <div data-bbox="849 1677 1136 1718" data-label="Text"> <p>// 0.01초 기다리기</p> </div> <div data-bbox="277 1767 550 1807" data-label="Text"> <pre> Serial.println();</pre> </div> <div data-bbox="277 1812 550 1852" data-label="Text"> <pre> Serial.println();</pre> </div> <div data-bbox="277 1856 764 1897" data-label="Text"> <pre> Serial.print("Connecting to ");</pre> </div> <div data-bbox="277 1901 612 1942" data-label="Text"> <pre> Serial.println(ssid);</pre> </div>

```

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

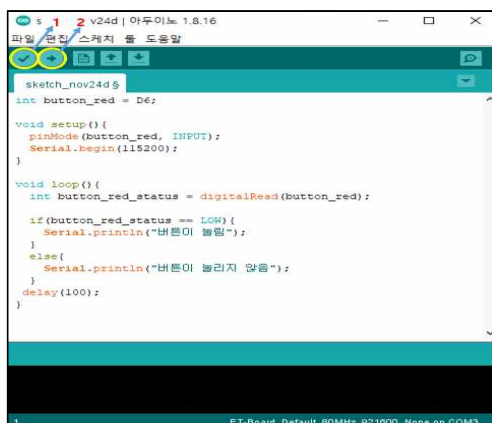
//연결이 될 때까지 계속 대기
while(WiFi.status() != WL_CONNECTED){
    delay(500);
    Serial.print(".");
}

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

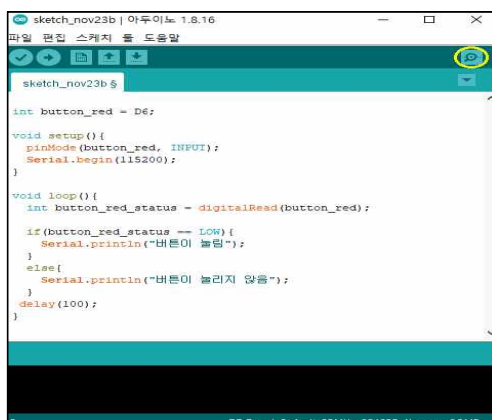
void loop()
{
    delay(10);
}

```

동작과정






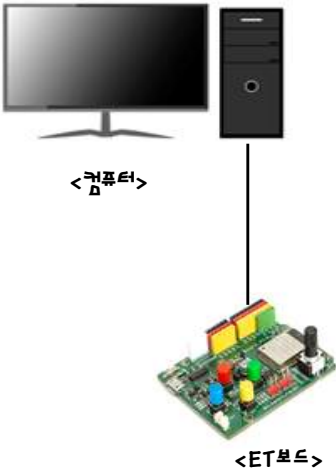


①아두이노 코드입력 후 상단 1[확인] 아이콘을 클릭하고 완료되면 2[업로드] 아이콘도 클릭 해 펌웨어를 ET보드에 업로드합니다.



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

<p>동 작 과 정</p>		<p>③ 시리얼 모니터에 IP가 출력됩니다.</p>
<p>참 고 사 항</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.ino
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 5Pin 케이블></p> <p><ET보드></p>

<div data-bbox="143 609 220 687" data-label="Section-Header"> <h2>회로 구성</h2> </div>	<div data-bbox="300 215 794 631" data-label="Image"> </div> <div data-bbox="868 383 1358 461" data-label="List-Group"> <ol style="list-style-type: none"> ① 컴퓨터 본체의 USB포트에 케이블을 연결합니다. </div> <div data-bbox="300 667 794 1084" data-label="Image"> </div> <div data-bbox="868 855 1447 891" data-label="List-Group"> <ol style="list-style-type: none"> ② 다른 한쪽 케이블을 ET보드에 연결합니다. </div>
<div data-bbox="143 1503 220 1581" data-label="Section-Header"> <h2>소스 코드</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 521 1225" data-label="Text"> <pre>#include <WiFi.h></pre> </div> <div data-bbox="248 1236 612 1270" data-label="Text"> <pre>#include <WiFiClient.h></pre> </div> <div data-bbox="248 1281 596 1314" data-label="Text"> <pre>#include <WebServer.h></pre> </div> <div data-bbox="248 1366 984 1404" data-label="Text"> <pre>const char *ssid = "와이파이 아이디를 입력하세요";</pre> </div> <div data-bbox="248 1411 588 1447" data-label="Text"> <p>// 와이파이 아이디 입력</p> </div> <div data-bbox="248 1456 1074 1494" data-label="Text"> <pre>const char *password = "와이파이 비밀번호를 입력하세요";</pre> </div> <div data-bbox="248 1500 617 1536" data-label="Text"> <p>// 와이파이 비밀번호 입력</p> </div> <div data-bbox="248 1590 580 1628" data-label="Text"> <pre>WebServer server(80);</pre> </div> <div data-bbox="836 1590 1254 1628" data-label="Text"> <p>// 서버에서 사용할 포트 설정</p> </div> <div data-bbox="248 1635 544 1671" data-label="Text"> <pre>const int led = D2;</pre> </div> <div data-bbox="836 1635 1152 1671" data-label="Text"> <p>// 빨간 LED의 핀 번호</p> </div> <div data-bbox="248 1724 536 1760" data-label="Text"> <pre>void handleRoot(){</pre> </div> <div data-bbox="836 1724 1410 1760" data-label="Text"> <p>// root(/)로 접속했을 때 처리하는 함수</p> </div> <div data-bbox="279 1769 655 1807" data-label="Text"> <pre>digitalWrite(led, HIGH);</pre> </div> <div data-bbox="836 1769 1050 1805" data-label="Text"> <p>// LED를 켜기</p> </div> <div data-bbox="279 1814 734 1850" data-label="Text"> <pre>Serial.println("root call!");</pre> </div> <div data-bbox="836 1814 1407 1850" data-label="Text"> <p>// 페이지로 접속했다고 시리얼로 알려줌</p> </div> <div data-bbox="279 1859 1144 1895" data-label="Text"> <pre>server.send(200 , "text/plain", "hello from ET-board!");</pre> </div> <div data-bbox="279 1904 639 1939" data-label="Text"> <pre>digitalWrite(led, LOW);</pre> </div> <div data-bbox="836 1904 1046 1939" data-label="Text"> <p>// LED를 끄기</p> </div> <div data-bbox="248 1948 276 1982" data-label="Text"> <pre>}</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());

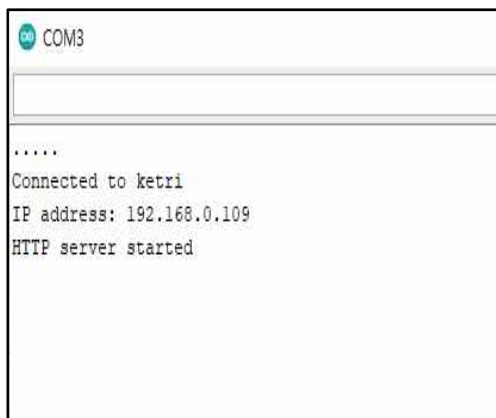
    // root(/)로 접속했을 때 처리하는 함수랑 연결
    server.on("/", handleRoot);

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

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





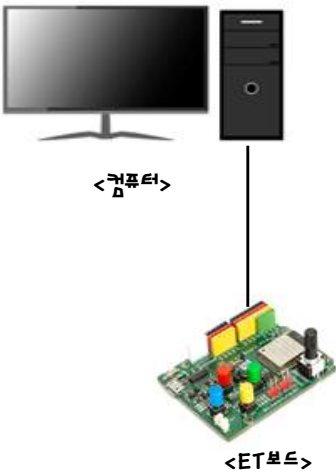
```

동작
과정



① 시리얼 모니터에 IP가 출력됩니다.

	<div data-bbox="304 219 799 633" data-label="Image"> </div> <div data-bbox="879 322 1414 443" data-label="Text"> <p>② 주소란에 할당받은 IP주소를 입력하면 이미지와 같이 “hello from ET-board!” 문구가 뜹니다.</p> </div>
<div data-bbox="148 884 217 958" data-label="Section-Header"> <p>참 고 사 항</p> </div>	<div data-bbox="256 714 1318 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
소 스	03._WiFi_web_server_variable_resistance.ino
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> ← <USB Micro 5Pin 케이블> <ET보드></p>

<div data-bbox="143 609 220 687" data-label="Section-Header"> <h2>회로 구성</h2> </div>	<div data-bbox="300 215 794 631" data-label="Image"> </div> <div data-bbox="873 383 1362 461" data-label="Text"> <p>① 컴퓨터 본체의 USB포트에 케이블을 연결합니다.</p> </div> <div data-bbox="300 667 794 1084" data-label="Image"> </div> <div data-bbox="873 855 1450 891" data-label="Text"> <p>② 다른 한쪽 케이블을 ET보드에 연결합니다.</p> </div>
<div data-bbox="143 1503 220 1581" data-label="Section-Header"> <h2>소스 코드</h2> </div>	<pre> // 이 라이브러리를 통해 Arduino 보드를 인터넷에 연결할 수 있습니다. #include <WiFi.h> #include <WiFiClient.h> #include <WebServer.h> const char * ssid = "와이파이 아이디를 입력하세요"; // 와이파이 아이디 입력 const char * password = "와이파이 비밀번호를 입력하세요"; // 와이파이 비밀번호 입력 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 </pre>

```

}

// 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);
}

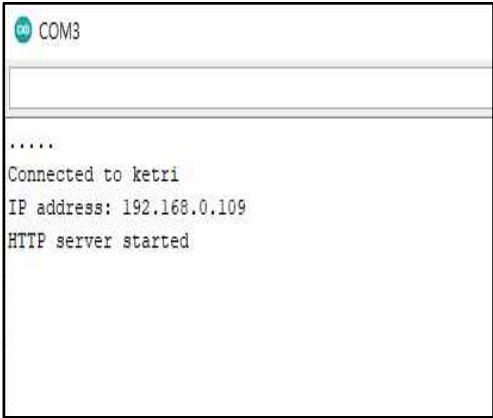
void setup(void){
    pinMode(led, OUTPUT);           // LED를 출력상태로 설정
    digitalWrite(led, LOW);         // LED를 끈 상태로 설정
    Serial.begin(115200);            // 시리얼 통신속도 설정
    WiFi.mode(WIFI_STA);            // 연결모드로 설정
    WiFi.begin(ssid, password);      // WiFi에 접속을 시도
    Serial.println("");

    // 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");
}

```






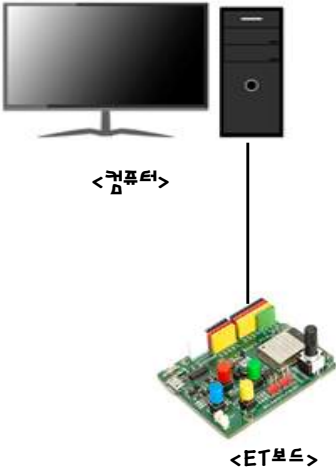
	<pre>void loop(void){ server.handleClient(); // 클라이언트의 접속을 받음 delay(2); }</pre>
동작과정	<div data-bbox="304 427 799 844">  </div> <div data-bbox="884 598 1417 678"> <p>① 할당 받은 IP주소를 인터넷창에 입력하여 접속합니다.</p> </div> <div data-bbox="304 880 799 1296">  </div> <div data-bbox="884 987 1485 1149"> <p>② 주소란에 할당받은 IP주소를 입력하여 이미지와 같이 “hello from ET-board!” 문구가 뜨면 IP주소 뒤에 /read_a0 입력하고 접속합니다.</p> </div> <div data-bbox="304 1332 799 1749">  </div> <div data-bbox="884 1503 1337 1583"> <p>③ 현재 ET보드의 가변저항값을 확인할 수 있습니다.</p> </div>
참고사항	<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	html 페이지에서 버튼을 눌러 LED를 켜고 꺼 보기
소스	04._WiFi_led_control.ino
개념	  <p>와이파이는 일상생활에서 흔하게 볼 수 있습니다.</p>
준비물	   <p><컴퓨터> <USB Micro 5Pin 케이블> <ET보드></p>
회로구성	 <p><컴퓨터> <ET보드></p> <p>← <USB Micro 5Pin 케이블></p>

<div data-bbox="143 609 220 687" data-label="Section-Header"> <h2>회로 구성</h2> </div>	<div data-bbox="300 215 794 631" data-label="Image"> </div> <div data-bbox="868 383 1358 461" data-label="List-Group"> <ol style="list-style-type: none"> ① 컴퓨터 본체의 USB포트에 케이블을 연결합니다. </div> <div data-bbox="300 667 794 1084" data-label="Image"> </div> <div data-bbox="868 855 1447 891" data-label="List-Group"> <ol style="list-style-type: none"> ② 다른 한쪽 케이블을 ET보드에 연결합니다. </div>
<div data-bbox="143 1503 220 1581" data-label="Section-Header"> <h2>소스 코드</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 1314" data-label="Text"> <pre>#include <WiFi.h> #include <WiFiClient.h> #include <WebServer.h></pre> </div> <div data-bbox="248 1366 1086 1536" data-label="Text"> <pre>const char * ssid = "와이파이 아이디를 입력하세요"; // 와이파이 아이디 입력 const char * password = "와이파이 비밀번호를 입력하세요"; // 와이파이 비밀번호</pre> </div> <div data-bbox="248 1590 1268 1628" data-label="Text"> <pre>WebServer server(80); // 서버에서 사용할 포트 설정</pre> </div> <div data-bbox="248 1680 1495 1895" 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>

```

// root(/)로 접속했을 때 처리하는 함수
void handleRoot(){
    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());
}

```


	<pre>// root(/)로 접속했을 때 처리하는 함수랑 연결 server.on("/", handleRoot); // 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>
동 작 과 정	<div>  <p>① 할당 받은 IP주소를 인터넷창에 입력하여 접속합니다.</p> </div> <div>  <p>② 접속한 페이지에서 red On을 클릭하면 빨간색 LED가 켜지고 red Off를 클릭하면 빨간색 LED가 꺼집니다.</p> </div>
참 고 사 항	<p>① 회로</p> <p>○ 와이파이에 대한 참고사항 링크</p> <p>https://ko.wikipedia.org/wiki/%EC%99%80%EC%9D%B4%ED%8C%8C%EC%9D%B4</p>






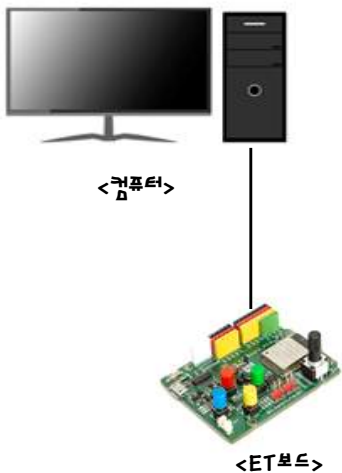
○ 웹서버에 대한 참고사항 링크

https://ko.wikipedia.org/wiki/%EC%9B%B9_%EC%84%9C%EB%B2%84

② 소스코드

○ 와이파이 사용법에 대한 설명 문서

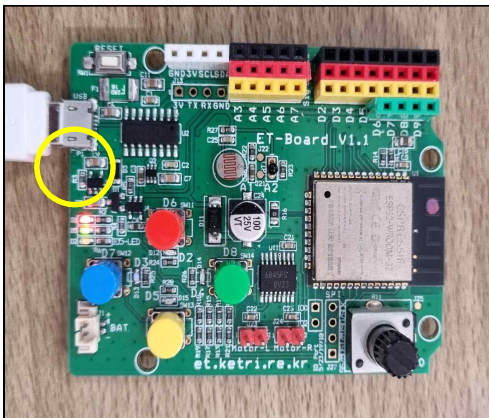
<https://docs.espressif.com/projects/arduino-esp32/en/latest/api/wifi.html>

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

회로 구성



① 컴퓨터 본체의 USB포트에 케이블을 연결합니다.



② 다른 한쪽 케이블을 ET보드에 연결합니다.

소스 코드

// 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 = "와이파이 아이디를 입력하세요";
```

// 와이파이 아이디 입력

```
const char * password = "와이파이 비밀번호를 입력하세요";
```

// 와이파이 비밀번호 입력

```
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 = "";

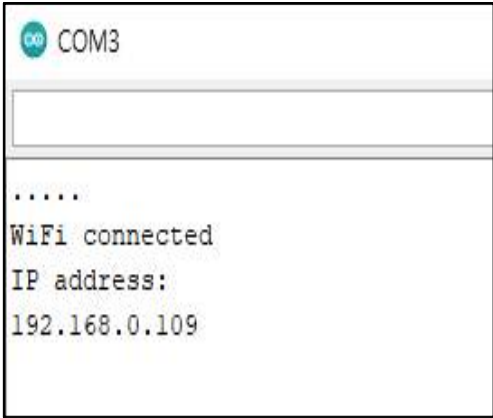
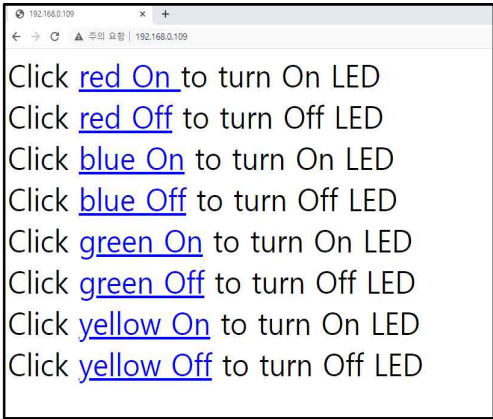
```

```

    }
  } 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>동 작 과 정</p>	<div data-bbox="304 293 799 710">  </div> <div data-bbox="304 745 799 1162">  </div> <div data-bbox="871 465 1453 994"> <p>① 할당 받은 IP주소를 인터넷창에 입력하여 접속합니다.</p> <p>② 접속한 페이지에서 red On을 클릭하면 빨간색 LED가 켜지고 red Off를 클릭하면 빨간색 LED가 꺼집니다.(3개의 LED도 확인)</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 <p>② 소스코드</p> <ul style="list-style-type: none"> ○ 와이파이 사용법에 대한 설명 문서 https://docs.espressif.com/projects/arduino-esp32/en/latest/api/wifi.html