

10. 와이파이(WiFi)

학습내용

5. 와이파이를 이용해 4개의 LED를 켜다 껐다하는 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>

회 로 구 성	 <p> <컴퓨터> <USB Micro 5Pin 케이블> <ET보드> </p>
회 로 구 성	<div data-bbox="301 797 794 1211">  <p>① 컴퓨터 본체의 USB포트에 케이블을 연결합니다.</p> </div> <div data-bbox="301 1261 794 1675">  <p>② 다른 한쪽 케이블을 ET보드에 연결합니다.</p> </div>
소 스 코 드	<p>// 이 라이브러리를 통해 Arduino 보드를 인터넷에 연결할 수 있습니다.</p> <pre> #include <WiFi.h> int led2 = D2; // 빨강 LED int led3 = D3; // 파랑 LED int led4 = D4; // 초록 LED int led5 = D5; // 노랑 LED </pre>

```

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 = sever.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>");

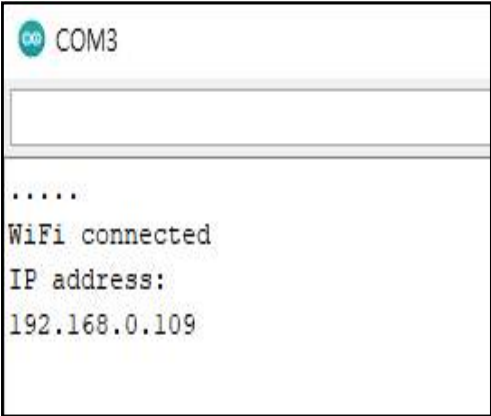
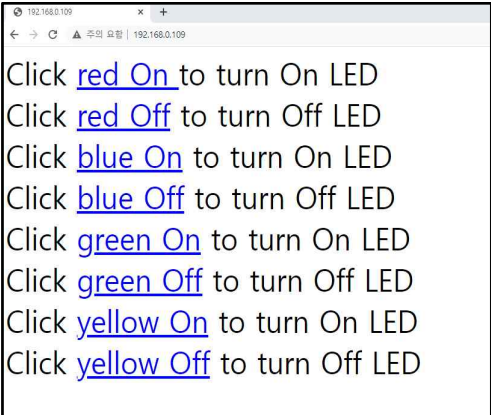
```

```

        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
        // 현재라인 뒤에 붙임
        currentLine += c;
    }
    // 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")) {

```

	<pre> digitalWrite(led5, LOW); // GET /L LED 끄 } } } // 연결 종료: client.stop(); Serial.println("Client Disconnected."); } } </pre>	
동작 과정	 <p>..... WiFi connected IP address: 192.168.0.109</p>  <p>Click red On to turn On LED Click red Off to turn Off LED Click blue On to turn On LED Click blue Off to turn Off LED Click green On to turn On LED Click green Off to turn Off LED Click yellow On to turn On LED Click yellow Off to turn Off LED</p>	<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>
참고 사항	<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 	