

Chap06. 원격 제어

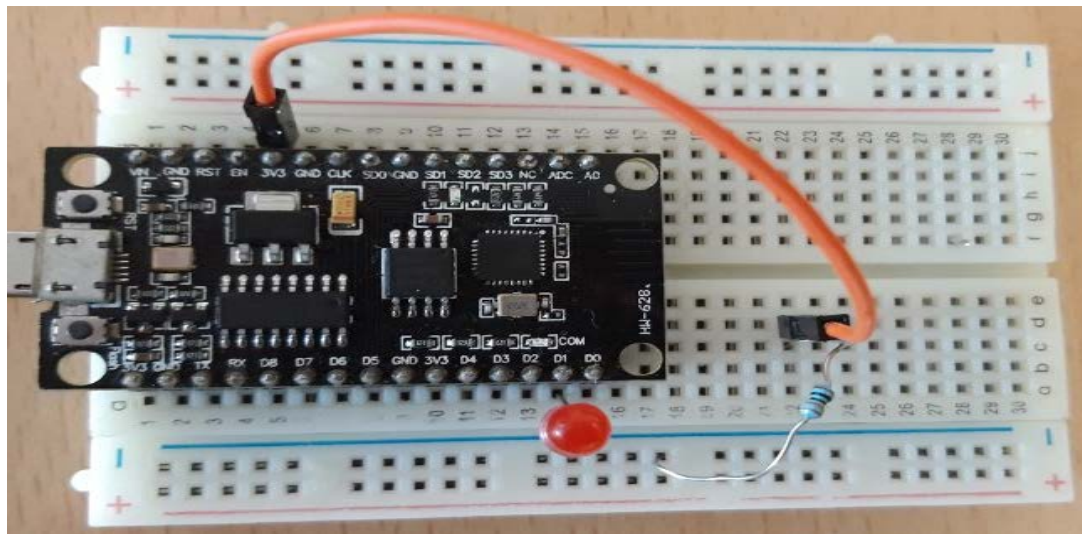
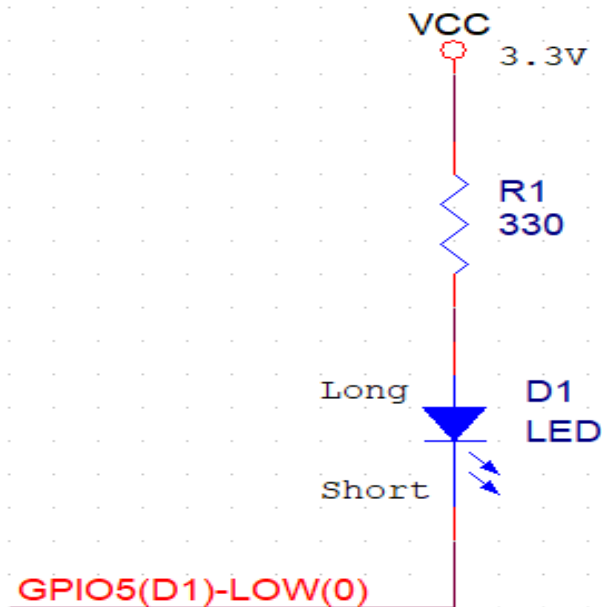
Contents

6.1 웹서버 기반 LED 제어

6.2 웹서버 기반 온습도 센서

6.1 웹서버 기반 LED 제어

◆ 회로도 및 연결도



6.1 웹서버 기반 LED 제어

◆ 소스 코드

```
2 //LED Control -> led_control_remote(스마트폰 -> NodeMCU)
3
4 // Including the ESP8266 WiFi library
5 #include <ESP8266WiFi.h>
6
7 // Replace with your network details
8 const char* ssid = "sjpark";
9 const char* password = "12345678";
10
11 WiFiServer server(80); // 웹서버용 포트 번호
```

6.1 웹서버 기반 LED 제어

◆ 소스 코드

```
13 void setup()
14 {
15     Serial.begin(9600);
16     delay(100);
17
18     pinMode(D1, OUTPUT);
19     digitalWrite(D1, 1);
20
21     // 와이파이 네트워크에 접속
22     Serial.println();
23     Serial.print("Connecting to ");
24     Serial.println(ssid);
25
26     WiFi.begin(ssid, password);
```

```
28     while(WiFi.status() != WL_CONNECTED) {
29         delay(500);
30         Serial.print(".");
31     }
32     Serial.println();
33     Serial.println("WiFi connected!!");
34
35     // 웹서버 개시
36     server.begin();
37     Serial.println("Server started");
38
39     // 접속할 IP 주소 표시
40     Serial.print("Use this URL to connect: ");
41     Serial.print("http://");
42     Serial.print(WiFi.localIP());
43     Serial.println("/");
44     Serial.println("");
45 }
```

6.1 웹서버 기반 LED 제어

◆ 소스 코드

```
47 void loop()
48 {
49 // 클라이언트가 접속되었는지 확인
50 WiFiClient client = server.available();
51 if(!client) {
52     return;
53 }
54
55 // 클라이언트가 데이터를 보낼 때까지 기다림
56 Serial.println("new client");
57 while(!client.available()) {
58     delay(1);
59 }
```

```
61 // 리퀘스트의 첫번째 줄을 읽어들이
62 String request = client.readStringUntil('\r');
63 Serial.println(request);
64 client.flush();
65
66 // 리퀘스트에 따른 LED 제어
67 int value = LOW;
68 if(request.indexOf("/LED=ON") != -1) {
69     digitalWrite(D1, LOW);
70     value = HIGH;
71 }
72 if(request.indexOf("/LED=OFF") != -1) {
73     digitalWrite(D1, HIGH);
74     value = LOW;
75 }
```

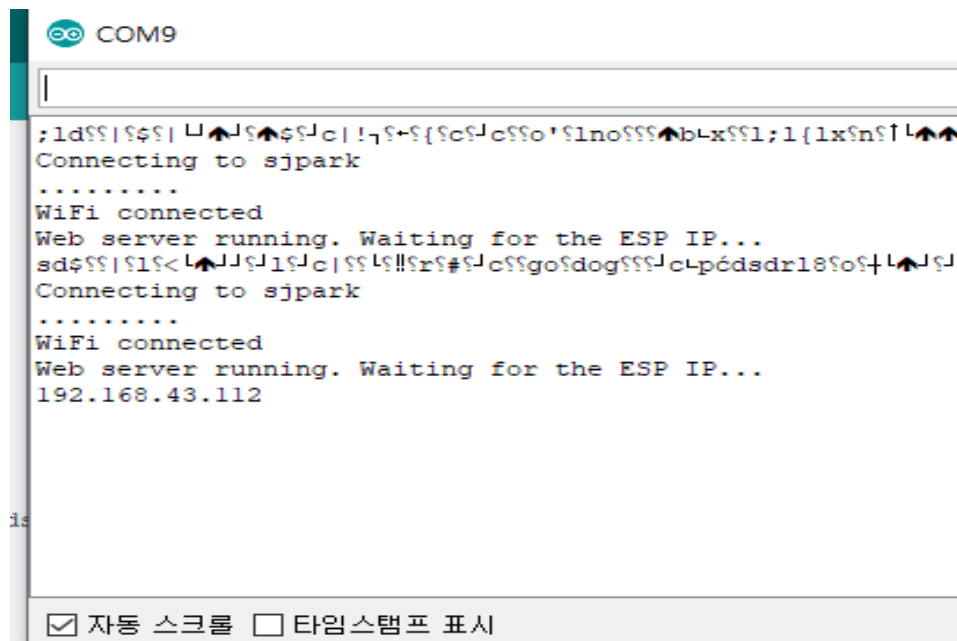
6.1 웹서버 기반 LED 제어

◆ 소스 코드

```
77 // 클라이언트 리퀘스트에 대한 응답 메시지
78 client.println("HTTP/1.1 200 OK");
79 client.println("Content-Type: text/html");
80 client.println("");
81 client.println("<!DOCTYPE HTML>");
82 client.println("<html>");
83 client.println("LED is turned ");
84 if(value)
85     client.print("On");
86 else
87     client.print("Off");
88 client.println("<br><br>");
89 client.println("<a href=\"/LED=ON\"><button>Turn On</button></a>");
90 client.println("<a href=\"/LED=OFF\"><button>Turn Off</button></a>");
91 client.println("</html>");
92
93 delay(1);
94 Serial.println("Client disconnected!!");
95 Serial.println();
96 }
```

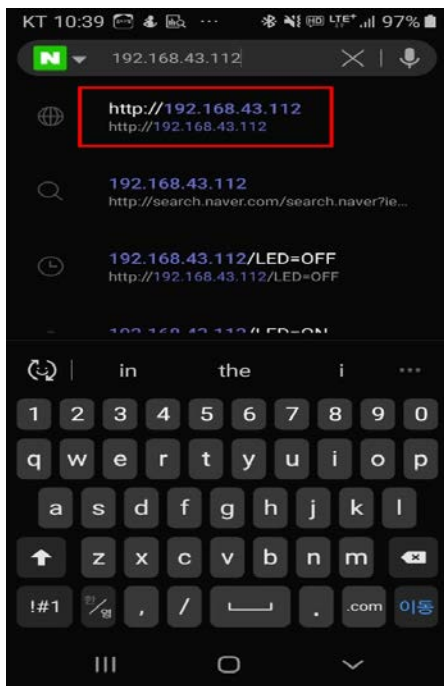
6.1 웹서버 기반 LED 제어

◆ 시리얼 모니터(웹서버 주소 확인)



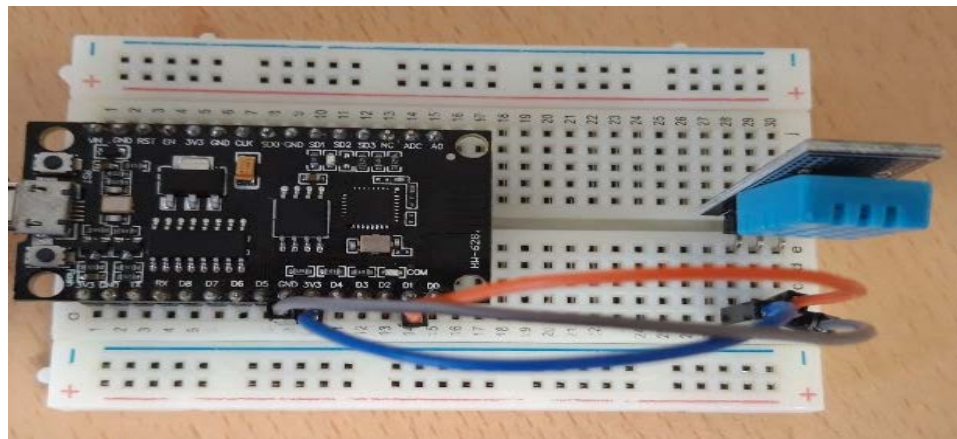
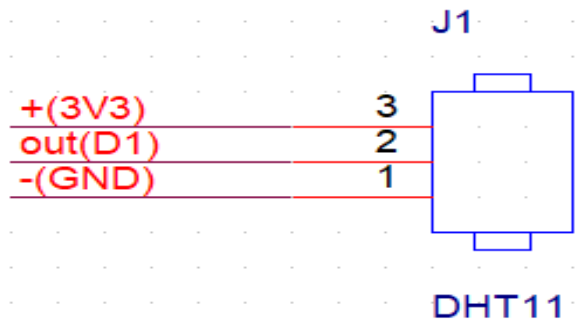
6.1 웹서버 기반 LED 제어

◆ 스마트폰(웹서버 주소 입력 및 결과 확인)



6.2 웹서버 기반 온습도 센서

◆ 회로도 및 연결도



6.2 웹서버 기반 온습도 센서

◆ 소스 코드

```
2 //온습도센서 (DHT11) Control -> dht_control_remote(NodeMCU->스마트폰)
3
4 // Including the ESP8266 WiFi library
5 #include <ESP8266WiFi.h>
6 #include "DHT.h"
7
8 // Uncomment one of the lines below for whatever DHT sensor type you're using!
9 #define DHTTYPE DHT11 // DHT 11
10
11 // Replace with your network details
12 const char* ssid = "sjpark";
13 const char* password = "12345678";
14
15
16 WiFiServer server(80); // Web Server on port 80
17
18 const int DHTPin = D1; // DHT Sensor
19 DHT dht(DHTPin, DHTTYPE); // Initialize DHT sensor.
20
21 // Temporary variables
22 static char celsiusTemp[7];
23 static char fahrenheitTemp[7];
24 static char humidityTemp[7];
```

6.2 웹서버 기반 온습도 센서

◆ 소스 코드

```
26 // only runs once on boot
27 void setup()
28 {
29     // Initializing serial port for debugging purposes
30     Serial.begin(115200);
31     delay(10);
32
33     dht.begin();
34
35     // Connecting to WiFi network
36     Serial.println();
37     Serial.print("Connecting to ");
38     Serial.println(ssid);
39
40     WiFi.begin(ssid, password);
41
42     while (WiFi.status() != WL_CONNECTED) {
43         delay(500);
44         Serial.print(".");
45     }
46     Serial.println("");
47     Serial.println("WiFi connected");
48
49     // Starting the web server
50     server.begin();
51     Serial.println("Web server running. Waiting for the ESP IP...");
52     delay(10000);
53
54     // Printing the ESP IP address
55     Serial.println(WiFi.localIP());
56 }
```

6.2 웹서버 기반 온습도 센서

◆ 소스 코드

```
58 // runs over and over again
59 void loop()
60 {
61   // Listennig for new clients
62   WiFiClient client = server.available();
63
64   if (client) {
65     Serial.println("New client");
66     // boolean to locate when the http request ends
67     boolean blank_line = true;
68     while (client.connected()) {
69       if (client.available()) {
70         char c = client.read();
71
72         if (c == '\n' && blank_line) {
73           // Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
74           float h = dht.readHumidity();
75           // Read temperature as Celsius (the default)
76           float t = dht.readTemperature();
77           // Read temperature as Fahrenheit (isFahrenheit = true)
78           float f = dht.readTemperature(true);
79           // Check if any reads failed and exit early (to try again).
80           if (isnan(h) || isnan(t) || isnan(f)) {
81             Serial.println("Failed to read from DHT sensor!");
82             strcpy(celsiusTemp, "Failed");
83             strcpy(fahrenheitTemp, "Failed");
84             strcpy(humidityTemp, "Failed");
85           }

```

6.2 웹서버 기반 온습도 센서

◆ 소스 코드

```
86         else{
87             // Computes temperature values in Celsius + Fahrenheit and Humidity
88             float hic = dht.computeHeatIndex(t, h, false);
89             dtostrf(hic, 6, 2, celsiusTemp);
90
91             dtostrf(h, 6, 2, humidityTemp);
92             // You can delete the following Serial.print's, it's just for debugging purposes
93
94         }
95         client.println("HTTP/1.1 200 OK");
96         client.println("Content-Type: text/html");
97         client.println("Connection: close");
98         client.println();
99         // your actual web page that displays temperature and humidity
100         client.println("<!DOCTYPE HTML>");
101         client.println("<html>");
102         client.println("<head></head><body><h1>ESP8266 - Temperature and Humidity</h1><h3>Temperature in Celsius: ");
103         client.println(celsiusTemp);
104         client.println("<C</h3><h3>Humidity: ");
105         client.println(humidityTemp);
106         client.println("</h3><h3>");
107         client.println("</body></html>");
108         break;
109     }
```

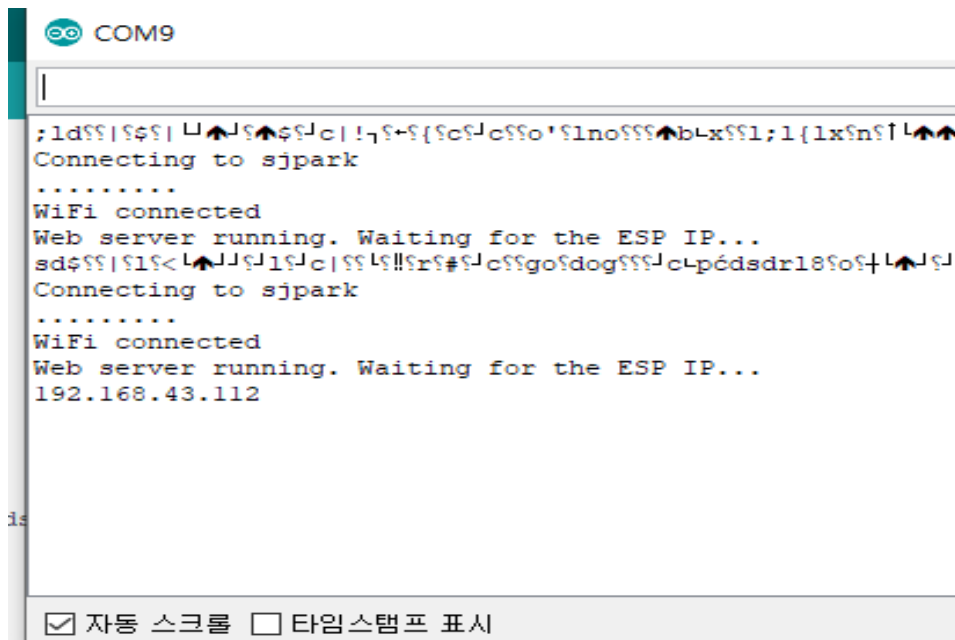
6.2 웹서버 기반 온습도 센서

◆ 소스 코드

```
110         if (c == '\n') {
111             // when starts reading a new line
112             blank_line = true;
113         }
114         else if (c != '\r') {
115             // when finds a character on the current line
116             blank_line = false;
117         }
118     }
119 }
120 // closing the client connection
121 delay(1);
122 client.stop();
123 Serial.println("Client disconnected.");
124 }
125 }
```

6.2 웹서버 기반 온습도 센서

◆ 시리얼 모니터(웹서버 주소 확인)



6.2 웹서버 기반 온습도 센서

◆ 스마트폰(웹서버 주소 입력 및 결과 확인)

