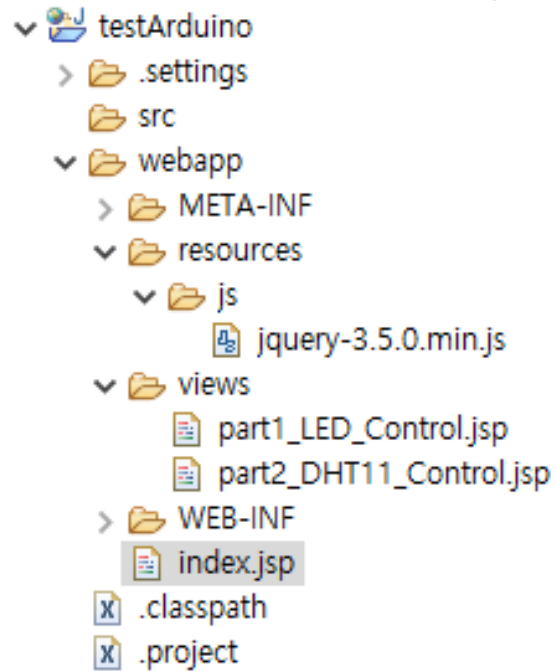


이더넷 스위드 & 웹 서버 연결

▶ JSP 웹 서버 준비하기

➤ file tree



➤ index.jsp

아두이노 이더넷 센서 제어

1. [part.1 LED 제어](#)
2. [part.2 온습도 제어](#)

▶ LED 원격 제어하기

➤ part1_LED_Control.jsp

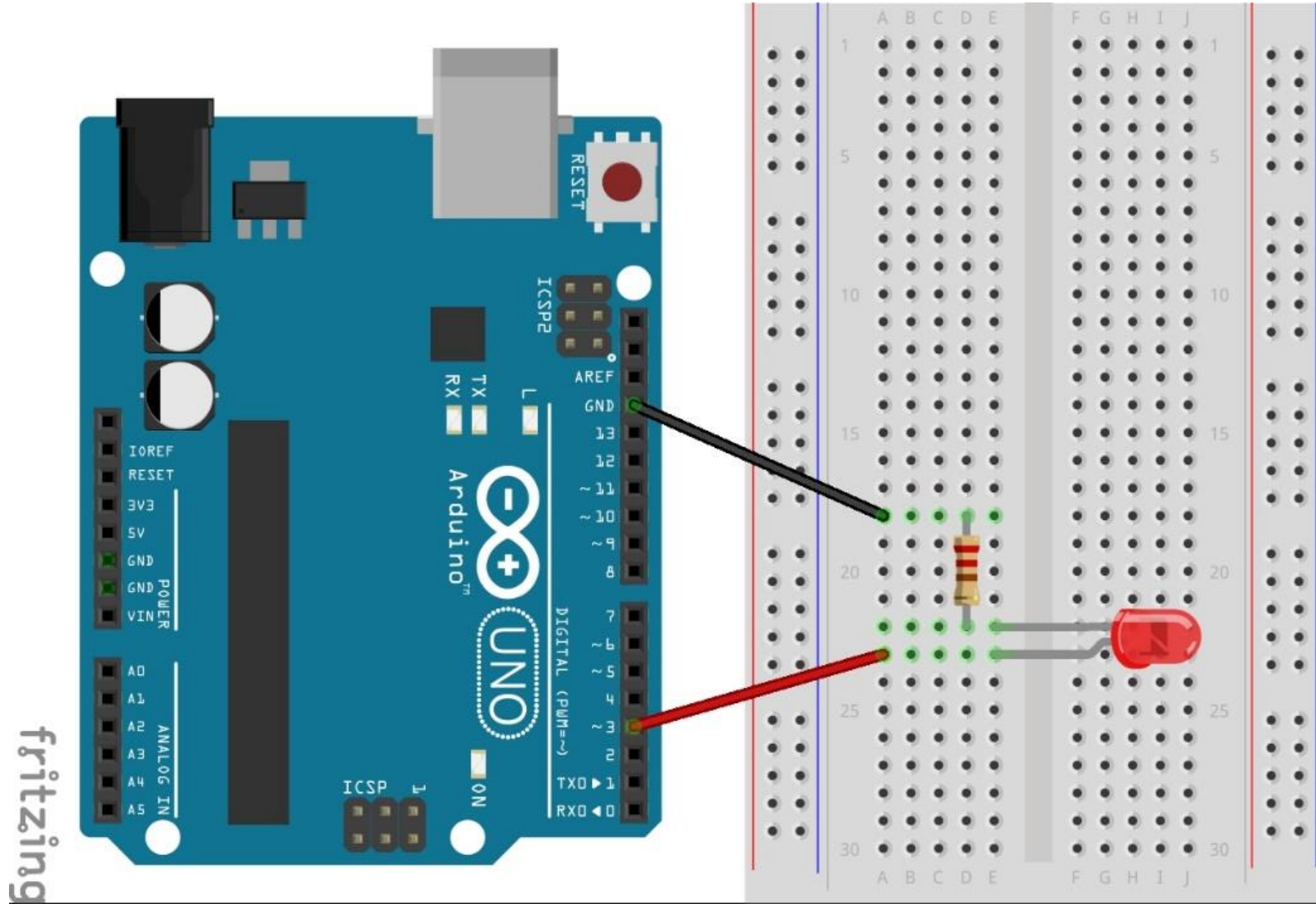
아두이노 LED 연결하기

RED ☒ On / ☐ Off

전송하기

```
<form method="get" action="http://아두이노_IP/led.cgi">
  <!-- cgi (common gateway interface) :
        사용자의 요구를 아두이노에게 돌려주기 위한 방법
  -->
  <p>
    RED
    <input type="radio" name="LEDstatus" value="1" /> On /
    <input type="radio" name="LEDstatus" value="0" /> Off
    <br>
  </p>
  <input type="submit" value="전송하기">
</form>
```

▶ LED 원격 제어하기



▶ LED 원격 제어하기

```
1 #include <SPI.h>
2 #include <Dhcp.h>
3 #include <Dns.h>
4 #include <Ethernet.h>
5 #include <EthernetClient.h>
6 #include <EthernetServer.h>
7
8 //mac 주소 입력
9 byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED};
10 // 로컬 네트워크에서 잡힌 IP Address
11 IPAddress ip(192,168,20,5);
12
13 EthernetServer server(80); //80은 포트번호
14
15 #define RED_PORT 3          //RED rgb 핀
16
17 boolean statusLED = false;  // LED 상태
18 String buffer = "";         // 수신 데이터 버퍼
```

```
20 void setup() {
21     Serial.begin(9600);
22
23     //이더넷 디바이스 초기화
24     Ethernet.begin(mac, ip);
25
26     Serial.println("WebServerControlLED");
27     Serial.print("server is at ");
28     Serial.println(Ethernet.localIP());
29     //서버 연결 시작
30     server.begin();
31
32     delay(2000);          // 이더넷 쉘드의 초기화 대기
33
34     pinMode(RED_PORT, OUTPUT);    //RED 핀 출력 설정
35     digitalWrite(RED_PORT, LOW); //RED 핀 초기 설정으로 꺼놓기
36 }
37
```

▶ LED 원격 제어하기

```
38 void loop() {
39   //수신 클라이언트가 있는 지 확인
40   EthernetClient client = server.available();
41
42   // 웹 브라우저가 접속한 경우
43   if (client) {
44     int response_type = -1;
45     Serial.println("*** New client connected...");
46
47     while (client.connected()) {
48       // 줄 단위의 데이터 수신
49       // HTML 요청의 끝을 알아내기 위해
50       // 빈 줄이 수신되었는지의 여부를 반환함
51       boolean empty_line = read_one_line(client);
52       Serial.println(buffer);
53
54       if(buffer.indexOf("GET /led.cgi?LEDstatus=1") >= 0)
55         show_on_off_page(client, true);
56       else if(buffer.indexOf("GET /led.cgi?LEDstatus=0") >= 0)
57         show_on_off_page(client, false);
58       if(empty_line) break; // HTML 요청이 끝나면 결과 전송
59     }
60
61     client.stop(); // 웹 브라우저 연결 종료
62     Serial.println("*** Client disconnected...");
63   }
64 }
```

```
66 boolean read_one_line(EthernetClient client)
67 {
68   buffer = ""; // 버퍼 비움
69   while(client.available()){
70     char c = client.read(); // 바이트 데이터 읽기
71     if(c == '\r'){ // 개행문자를 수신한 경우
72       client.read(); // '\n' 문자 버림
73       break;
74     }
75     // 개행문자 이외의 문자는 버퍼에 저장
76     buffer = buffer + c;
77   }
78
79   return (buffer.length() == 0); // 빈 줄 여부 반환
80 }
```

▶ LED 원격 제어하기

```
82 void show_on_off_page(EthernetClient client, boolean status)
83 {
84     statusLED = status;    // LED 상태 저장
85     digitalWrite(RED_PORT, status);    // LED 상태 바꿈
86     String str = "LED 상태 : ";
87     Serial.println(str + (status ? "HIGH" : "LOW"));
88
89     // HTML 헤더
90     client.println("HTTP/1.1 200 OK");
91     client.println("Content-Type: text/html");
92     client.println();
93
94     // HTML 내용
95     client.println("<html>");
96     client.println("<body><script>");
97     client.println("window.location.href = \"http://192.168.120.3:8088/test/\";");
98     client.print("alert('Currently... LED is '); // 현재 LED 상태 표시
99     client.println(statusLED ? "ON.'" : "OFF.'");");
100
101     client.println("</script></body>");
102     client.println("</html>");
103 }
```

▶ DHT11 모듈 연결하기

➤ part2_DHT11_Control.jsp

아두이노 온습도 & ajax 연결하기

온습도 확인하기

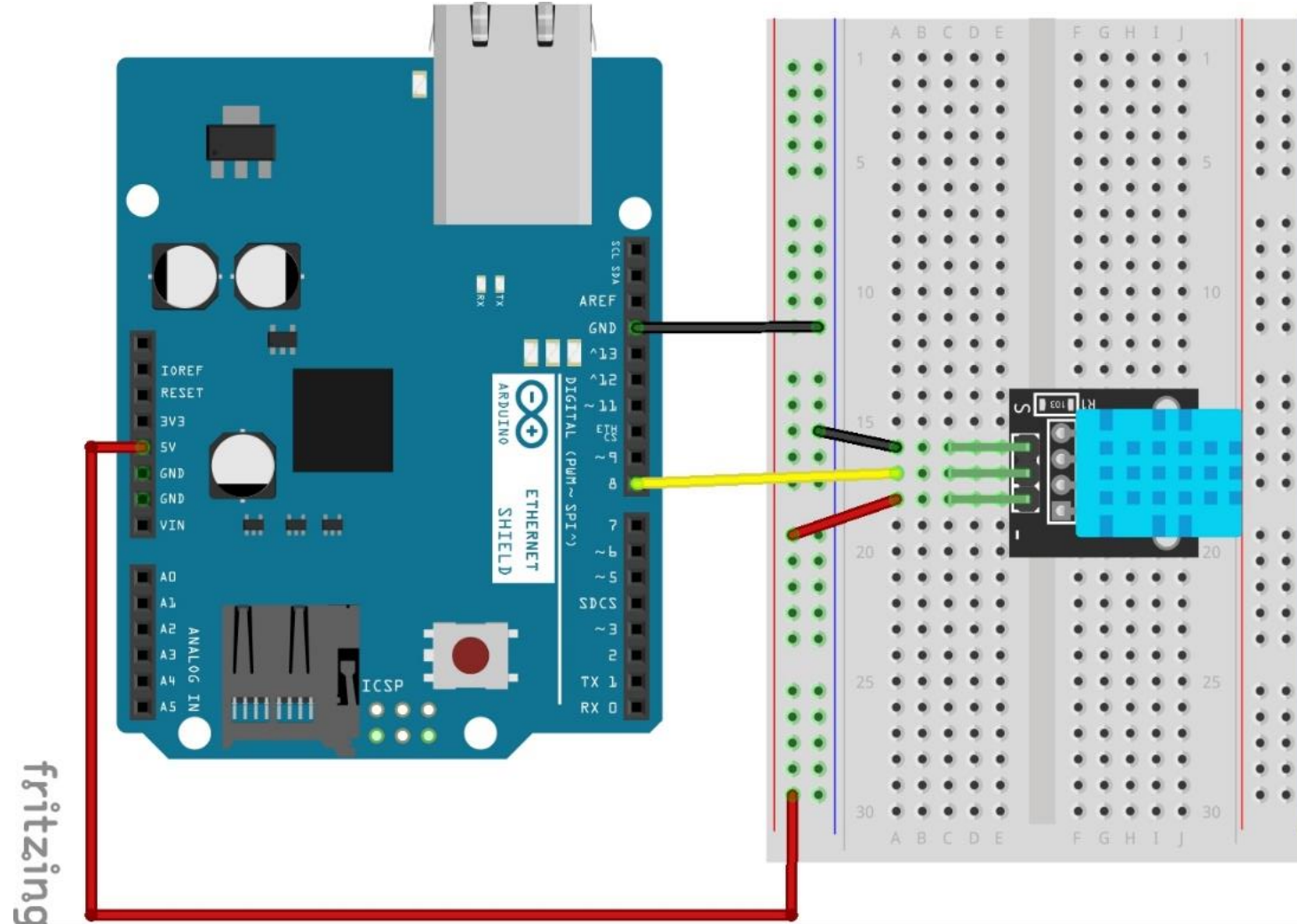
시간	습도	온도
2020-05-09 pm 12:00	50%	20°C
2020-05-09 pm 12:01	40%	21°C
2020-05-09 pm 12:02	35%	23°C
2020-05-09 pm 12:03	56%	27°C
2020-05-09 pm 12:04	60%	22°C

```
<script>
$('#checkDHT').on('click', function() {
    $.ajax({
        method : "GET",
        url : "http://아두이노_IP/DHT11.cgi",
        dataType : "json",
        // jquery 에서 CORS 허용하도록 jsonp방식으로 우회한다.
        jsonpCallback : "callback",
        // 받아 올 객체의 이름
        // (반드시 전달하는 객체의 이름과 같아야 한다.)
        success : function(data) {
            console.log(data);

            // 서버에 작성할 내용
            // . . .
        }, error : function(error, code, msg) {
            console.log(msg);
        }
    });
});
</script>
```

웹 서버의 주소가
아두이노의 IP 주소와
다르기 때문에
CORS 이슈를 해결해야
함

▶ DHT11 모듈 연결하기



▶ DHT11 모듈 연결하기

```
1 #include <SPI.h>
2 #include <Dhcp.h>
3 #include <Dns.h>
4 #include <Ethernet.h>
5 #include <EthernetClient.h>
6 #include <EthernetServer.h>
7 #include <ArduinoJson.h>
8 #include "DHT.h"
9 #define DHTPIN 8
10 #define DHTTYPE DHT11
11 DHT dht(DHTPIN, DHTTYPE);
12
13 //mac 주소 입력
14 byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED};
15
16 // 로컬 네트워크에서 잡힌 IP Address
17 IPAddress ip(192,168,0,11);
18
19 EthernetServer server(80); //80은 포트번호
20 String buffer = ""; // 수신 데이터 버퍼
```

```
22 void setup() {
23     Serial.begin(9600);
24
25     //이더넷 디바이스 초기화
26     Ethernet.begin(mac, ip);
27
28     Serial.println("WebServer Send DHT11 Sensor DATA" );
29     Serial.print("server is at ");
30     Serial.println(Ethernet.localIP());
31     //서버 연결 시작
32     server.begin();
33     dht.begin();
34
35     delay(2000); // 이더넷 쉴드의 초기화 대기
36 }
```

▶ DHT11 모듈 연결하기

```
38 void loop () {
39     delay (2000);
40     int h = dht.readHumidity ();
41     int t = dht.readTemperature ();
42     Serial.print ("Humidity: " );
43     Serial.print (h);
44     Serial.print (" %\t");
45     Serial.print ("Temperature: " );
46     Serial.print (t);
47     Serial.println (" C");
48
49     //수신 클라이언트가 있는지 확인
50     EthernetClient client = server.available ();
51
52     // 클라이언트가 없다면 복귀
53     if (!client) return;
54     else { // 웹 브라우저가 접속한 경우
55
56         while (client.available ()) client.read ();
57
58         // JsonData를 저장할 객체 선언
59         DynamicJsonDocument doc (500);
```

```
61     // 아날로그 객체 필드 선언
62     doc["Humidity"] = dht.readHumidity ();
63     doc["Temperature"] = dht.readTemperature ();
64
65     client.println (F("HTTP/1.0 200 OK" ));
66     // json으로 contentType 설정
67     client.println (F("Content-Type: application/json; charset=utf-8" ));
68     client.println (F("Connection: close" ));
69     client.print (F("Content-Length: " ));
70     client.println (measureJsonPretty (doc));
71     client.println ();
72
73     // json을 담을 임시 변수 설정
74     String str = "";
75     // doc에 담긴 json 데이터를 문자열로 변환
76     // 일반 DynamicJsonDocument 객체로는 문자열 혼입이 불가능하다.
77     serializeJson (doc, str);
78
79     // 서버에 전달할 객체의 이름인 callback을 붙여 서버에 전송
80     Serial.println ("callback(" +str+")");
81     client.println ("callback(" +str+")");
82
83     client.stop (); // 웹 브라우저 연결 종료
84     Serial.println ("** Client disconnected..." );
85 }
86 }
```