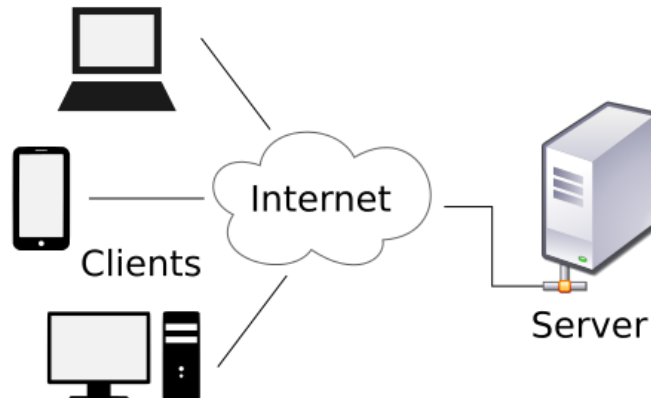


1. 서버와 클라이언트

(1) 서버와 클라이언트란?



서버는 일반적으로 클라이언트에게 네트워크를 통해 서비스를 제공하는 시스템을 말하는데, 여기서 클라이언트는 서비스를 사용하는 사용자 혹은 사용자의 단말기를 가리킨다.

자세히는, 서버는 웹페이지, 사이트, 또는 앱 등을 저장하는 컴퓨터이다. 클라이언트의 장비가 웹페이지에 접근하기를 원할 때, 서버로부터 클라이언트의 장치로 웹페이지의 사본이 다운로드 된다. 클라이언트는 웹 사용자의 인터넷이 연결된 장치들과 이런 장치들에서 웹에 접근할 수 있도록 하는 소프트웨어(크롬, 파이어폭스 등)이다.

서버와 클라이언트가 통신을 하는데(클라이언트가 서버에 원하는 것을 요청하면 서버가 그에 대한 응답을 하는 데) 필요한 것들이 있다.

- 인터넷 연결 : 웹에서 데이터를 보내고 받을 수 있게 해준다.
- TCP/IP : Transmission Control Protocol (전송 제어 규약)과 Internet Protocol (인터넷 규약)은 데이터가 어떻게 웹을 건너 여행해야 하는지 정의하는 통신 규약이다.
- DNS : Domain Name System Servers (도메인 이름 시스템 서버)는 웹사이트를 위한 주소록과 같다. 브라우저에 웹 주소를 입력할 때, 브라우저는 그 웹사이트를 검색하기 전에 DNS 를 살펴본다.브라우저는 HTTP 메시지를 올바른 장소로 전송하기 위해 그 웹사이트가 있는 서버가 어떤 것인지 찾아야 한다.
- HTTP : Hypertext Transfer Protocol (하이퍼텍스트 전송 규약)은 클라이언트와 서버가 서로 통신할 수 있게 하기 위한 언어를 정의하는 어플리케이션 규약이다.

브라우저에 웹 주소를 입력할 때, 아래와 같은 과정을 거쳐 서버와 클라이언트간의 통신이 일어난다.

1. 브라우저는 DNS 서버로 가서 웹사이트가 있는 서버의 진짜 주소를 찾는다.
2. 브라우저는 HTTP 요청 메시지를 서버로 전송한다. 이 메시지, 그리고 클라이언트와 서버 사이에 보내진 다른 데이터들은 TCP/IP 를 사용한 인터넷 연결을 건너 보내진다.
3. 제공된 서버는 클라이언트의 요청을 승인하고, 서버는 웹사이트를 볼 수 있다는 의미를 가진 '200 OK'라는 메시지를 클라이언트에게 보낸다. 그리고 웹사이트의 파일들을 브라우저에게 보낸다.
4. 브라우저는 이 파일들을 받아 웹사이트를 만들어 사용자에게 보여준다.

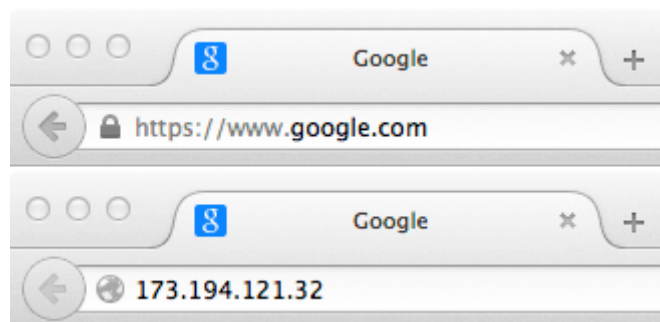
(2) DNS 란?

실제 웹 주소는 우리가 사용하는 URL 처럼 이름을 가지고 있지 않고, 숫자들로 이루어져있다.

(예: 173.194.121.320)

이것은 IP 주소라고 하고, 웹의 하나뿐인 특정 위치를 나타낸다. 하지만 이 숫자들을 기억하기 쉽지 않고, 이를 해결하기 위해 DNS 라는 것이 생겼다. DNS 는 사용자가 브라우저에 입력하는 웹 주소(www.google.com 같은)를 웹사이트의 실제 IP 주소에 맞춰주는 특별한 서버이다.

아래 그림처럼, 실제로 특정 웹 사이트에 접속할 때, url 이 아닌 웹사이트의 실제 주소를 입력해도 접속이 가능하다.



2. HTTP 통신

(1) HTTP 에 대하여

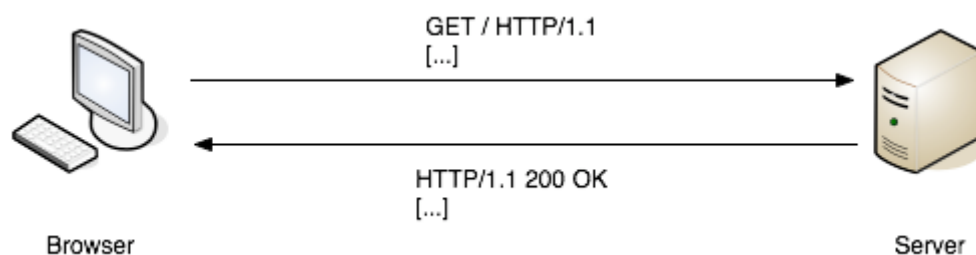
HTTP 는 인터넷상에서 데이터를 주고 받기 위한 서버-클라이언트 모델을 따르는 프로토콜이다. 어플리케이션 레벨의 프로토콜로 TCP/IP 위에서 작동한다. HTTP 는 어떤 종류의 데이터든지 전송할 수 있도록 설계되어 있는데, 인터넷 상에서 흔히 볼 수 있는 HTML 로 작성된 문서외에도 이미지, 동영상, 오디오 등이 있다.

(2) 작동방식

HTTP 는 서버-클라이언트 모델을 따른다. 클라이언트에서 서버에 요청(request)를 보내면 서버는 그 요청을 처리해서 응답(response)한다. 웹서버는 보통 80 번 포트로 서비스한다. HTTP 는 Connectless 방식으로 작동한다. 서버에 연결하고, 요청해서 응답을 받으면 연결을 끊어버린다. 기본적으로는 자원 하나에 대해서 하나의 연결을 만드는데 이런 작동방식은 아래와 같은 장점과 단점을 가진다.

- 장점 : 불특정 다수를 대상으로 하는 서비스에 적합한 방식이다. 수십만명이 웹 서비스를 사용하더라도 접속유지는 최소한으로 할 수 있기 때문에, 더 많은 유저의 요청을 처리할 수 있다.
- 단점 : 연결을 끊어버리기 때문에, 클라이언트의 이전 상태를 알 수가 없다. 이러한 HTTP 의 특징을 stateless 라고 하는데, connectless 로 부터 파생되는 특징이라고 할 수 있다. 클라이언트의 이전 상태 정보를 알 수 없게 되면, 웹 서비스를 하는데 당장에 문제가 생긴다. 예를 들어 클라이언트가 과거에 로그인을 성공하더라도 로그 정보를 유지할 수 없다. HTTP 는 이를 해결하기 위해 cookie 를 이용하고 있다.

(3) HTTP request



클라이언트가 서버에 어떤 것을 요청(request)하면 서버는 그 요청에 응답(response)한다. 이 때, 서버와 클라이언트는 HTTP 라는 프로토콜을 사용해 서로에게 요청과 응답을 전달한다.

그리고 당연히 클라이언트의 요청에는 다양한 종류가 있으며, 이를 메서드(method)라 부르기도 한다. 다음은 요청에 사용할 수 있는 메서드들이다.

- GET : 정보를 요청하기 위해 사용한다. URL(URI)형식으로 웹서버측의 리소스(데이터)를 요청한다.
- HEAD : 메시지의 헤더 정보만 요청한다. GET 과 비슷하나, 실제 문서를 요청하는 것이 아니라, 문서의 정보를 요청한다. 따라서 응답 메시지에 본문(Body)이 없이, 헤더만을 보내게 된다. 해당 자원이 존재하는지 혹은 서버에 문제가 없는지를 확인하기 위해 사용한다.
- POST : 정보를 보내기 위해 사용한다. 클라이언트에서 서버로 어떤 정보를 제출하고자 할 때, 요청 데이터를 HTTP 의 바디에 담아서 전송한다. 만약 새롭게 작성된 리소스라면 서버측은 HTTP 헤더 항목 중 'Location:'에 URI 주소 정보를 포함시켜 응답한다.
- PUT : 정보를 갱신하기 위해 사용한다. POST 처럼 정보를 서버로 제출하는 것으로 형식은 동일하나, 갱신 위주이므로 응답메시지의 HTTP 헤더 항목 중 'Location'을 보내지 않아도 된다.
- DELETE : 정보를 삭제하기 위해 사용한다.
- OPTIONS : 웹서버가 제공하는 메서드의 종류를 요청한다. 응답메시지의 HTTP 헤더에서 'Allow : GET, POST, HEAD' 처럼 보내진다.
- TRACE : 요청 리소스가 수신되는 경로를 보여준다.

한편, 보안상의 이유로 GET, POST 2 개 혹은 OPTIONS 포함 3 개만을 허용하는 웹서버가 대부분이다.

3. 웹클라이언트 예제

다음은 JSN270 와이파이 모듈을 활용해 아두이노를 웹클라이언트로 만든 후, 웹 서버와 통신하는 과정에 대한 설명이다.

웹서버와 웹클라이언트의 통신 과정을 다시 한번 살펴보면, 구글에 접속을 하려고 할 때, 우리는 PC 또는 스마트폰 등의 기기에서 internet explorer 또는 chrome 등의 웹 브라우저를 실행한 후 주소창에 "www.google.com"을 입력한다. 그럼 구글의 메인 화면이 나오고, 원하는 동작을 수행할 수 있게 된다. 이 때 internet explorer 나 chrome 은 클라이언트가 되고, 구글은 서버가 된다.

이 과정을 조금 더 자세히 들여다보면 다음과 같다.

1. 웹 브라우저를 실행한 후 구글의 주소(URL)을 입력하면 클라이언트인 웹 브라우저가 "www.google.com" 이라는 이름의 서버인 구글의 웹서버로 연결을 요청한다.
2. 구글의 웹서버는 클라이언트가 접속하기를 기다리고 있다가 연결 요청이 오면 연결을 수락한 후 데이터를 주고 받기 위한 길을 만든다.
3. 연결 요청이 수락되면 클라이언트는 서버에게 필요한 데이터를 요청한다.
4. 웹 서버는 요청 받은 데이터를 클라이언트로 전송한다.
5. 클라이언트인 웹 브라우저는 데이터 수신이 완료되면 서버와의 연결을 해제한다.
6. 그리고 수신된 데이터를 해석하여 웹 브라우저에 화면을 출력한다.

다음은 위와 같은 과정을 통해 아두이노를 클라이언트로 만든 후 웹 서버와 통신하는 것에 대한 예제이다.

(1) Google 에 'JSN270' 검색 결과 요청



```
JSN270_Web_Client | 아두이노 1.8.1
파일 편집 스케치 툴 도움말
JSN270_Web_Client$
#include <Debug.h>
#include <JSN270.h>
#include <Arduino.h>
#include <SoftwareSerial.h>

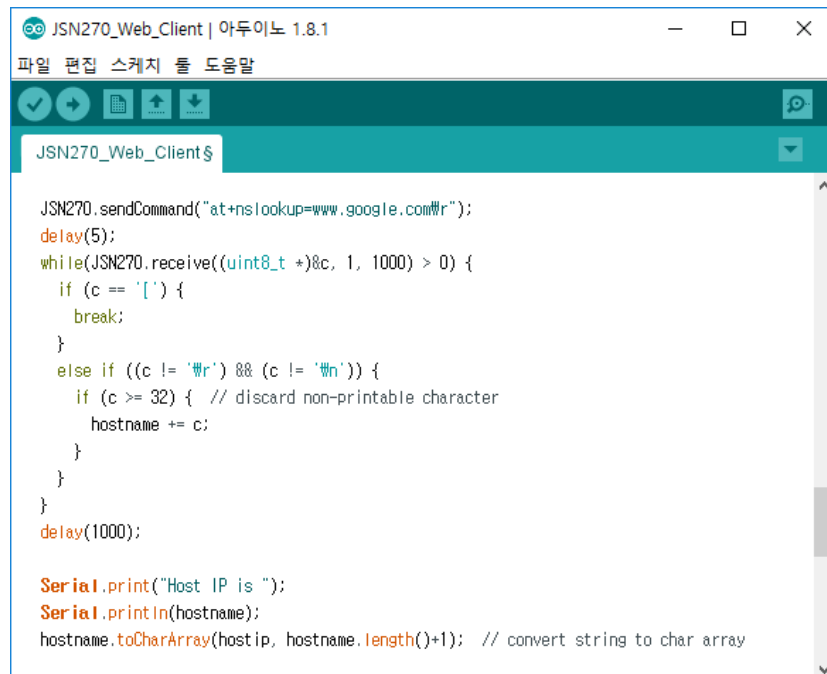
#define SSID "JSN270_28" // your wifi network SSID
#define KEY "12345678" // your wifi network password
#define AUTH "WPA2" // your wifi network security (NONE, WEP, WPA,

#define USE_DHCP_IP 1

#if !USE_DHCP_IP
#define MY_IP "192.168.1.133"
#define SUBNET "255.255.255.0"
#define GATEWAY "192.168.1.254"
#endif

#define HOST_IP "74.125.232.128" // www.google.com
#define REMOTE_PORT 80
#define PROTOCOL "TCP"
```

→ SSID 에는 JSN270 와이파이 모듈을 연결할 와이파이의 이름, KEY 에는 비밀번호를 입력한다. USE_DHCP_IP 는 1 로 설정되어 있는데, 이는 IP 주소를 할당받을 때, 유동 IP 를 받기 위해 사용된다. 만약 고정 IP 를 할당받으려면 'MY_IP'~'GATEWAY'에 할당받을 IP 주소를 입력한다. REMOTE_PORT 는 접속하게 될 서버의 포트이다. 기본적으로 웹서버는 80 을 사용한다.



```
JSN270_Web_Client | 아두이노 1.8.1
파일 편집 스케치 툴 도움말

JSN270_Web_Client$

JSN270.sendCommand("at+nslookup=www.google.com\r");
delay(5);
while(JSN270.receive((uint8_t *)&c, 1, 1000) > 0) {
    if (c == '[') {
        break;
    }
    else if ((c != '\r') && (c != '\n')) {
        if (c >= 32) { // discard non-printable character
            hostname += c;
        }
    }
}
delay(1000);

Serial.print("Host IP is ");
Serial.println(hostname);
hostname.toCharArray(hostip, hostname.length()+1); // convert string to char array
```

→ 'at+nslookup' 명령어를 사용해 DNS 서버에서 domain name 의 IP 주소를 검색한다. (IP 를 고정으로 설정할 경우 at+nset 명령의 옵션으로 DNS 서버를 미리 설정해야 한다. IP 를 DHCP 로 받아올 경우 자동으로 DNS 서버 주소를 받아온다.) 이 과정에서 서버의 IP 주소는 hostname 에 저장되고, 그 후 출력을 통해 IP 주소를 확인 할 수 있다.

```

JSN270_Web_Client | 아두이노 1.8.1
파일 편집 스케치 툴 도움말

JSN270_Web_Client$

if (!JSN270.client(hostip, REMOTE_PORT, PROTOCOL)) {
  Serial.print("Failed connect to ");
  Serial.println(hostip);
  Serial.println("Restart System");
} else {
  Serial.print("Socket connect to ");
  Serial.println(hostip);
  //delay(2000);

  // Enter data mode
  JSN270.sendCommand("at+exit\r");
  delay(100);

  Serial.println("connected to server");
  // Make a HTTP request:
  JSN270.println("GET /search?q=JSN270 HTTP/1.1");
  JSN270.println("Host: www.google.com");
  JSN270.println("Connection: close");
  JSN270.println();
}
}

```

→ 웹서버 IP, 포트, 프로토콜이 모두 조건에 충족되면 소켓이 연결된 후, 서버에 원하는 정보를 요청을 하게 된다. 여기서는 'GET'을 통해 데이터를 요청하는데, 'search?q=JSN270'이라는 명령이 추가적으로 붙었다. 이것은 www.google.com 을 통해 JSN270 을 검색 한 후 그 결과를 보내달라는 뜻이다. Host 에는 접속할 사이트의 url 혹은 IP 주소를 입력한다.

```

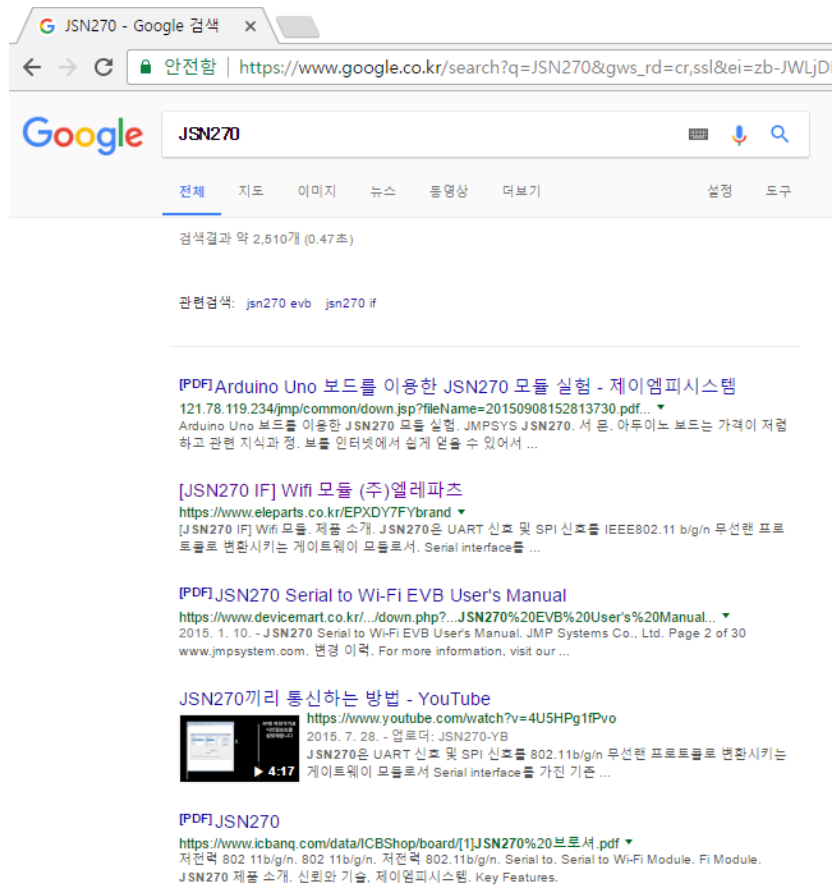
COM3 (Arduino/Genuino Uno)

Netmask=255.255.255.0
Gateway=192.168.43.1
DNS=192.168.43.1
[OK]
CMD: at+nslookup=www.google.com
Host IP is 64.233.188.106
CMD: at+nauto=0,1,64.233.188.106,80
CMD: at+socket
Socket connect to 64.233.188.106
CMD: at+exit
connected to server
[OK]
HTTP/1.1 302 Found
Location: http://www.google.co.kr/search?q=JSN270&gws_rd=cr&ei=zb-JVLJDDMP90gTspbrQBA
Cache-Control: private
Content-Type: text/html; charset=utf-8
P3P: CP="This is not a P3P policy! See https://www.google.com/support/accounts/answer/151657?hl=en for more info."
Date: Thu, 26 Jan 2017 09:22:21 GMT
Server: gws
X-XSS-Protection: 1; mode=block
X-Frame-Options: SAMEORIGIN
Content-Length: 280
Connection: close
Set-Cookie: NID=95=md6R6tsw9H4Fjnl9ozRrEaz58giu94BAbjz0KAR4f1RPcc3_NyLatgR9KvWHD6J6w_s-FoldjrAK5S-ZK1sS3YtrFxiY60pT4boxTdU3ZDWR4C

<HTML><HEAD><meta http-equiv="content-type" content="text/html; charset=utf-8">
<TITLE>302 Moved</TITLE></HEAD><BODY>
<H1>302 Moved</H1>
The document has moved
<A HREF="http://www.google.co.kr/search?q=JSN270&gws_rd=cr&ei=zb-JVLJDDMP90gTspbrQBA">here</A>.
</BODY></HTML>
DISCONNECT 0

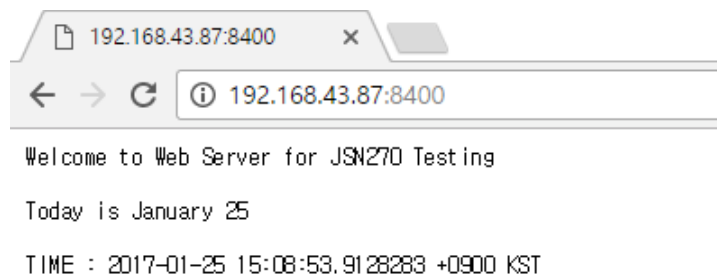
```

→ 실행결과는 위와 같다. 콘솔창에 JSN270 을 검색한 서버의 주소가 출력된 것을 확인 할 수 있다. 해당 주소로 접속하게 되면 아래 처럼 JSN270 검색 결과 화면이 뜬다.



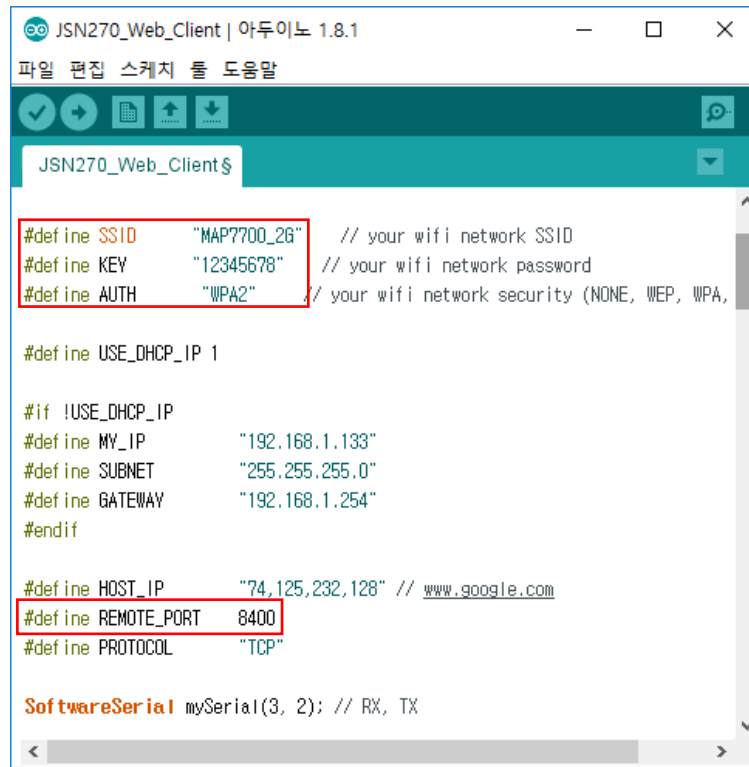
다음은 기본 웹클라이언트 예제를 활용해 다양한 서버와 통신하는 예제이다.

(2) JSN 모듈에서 웹서버 접속 후 현재 시각 문자열 수신



웹서버 내용 (PORT : 8400)

- 서버에 한 번만 접속해서 현재 시각 받아오기



```
JSN270_Web_Client | 아두이노 1.8.1
파일 편집 스케치 툴 도움말

JSN270_Web_Client$

#define SSID "MAP7700_26" // your wifi network SSID
#define KEY "12345678" // your wifi network password
#define AUTH "WPA2" // your wifi network security (NONE, WEP, WPA,

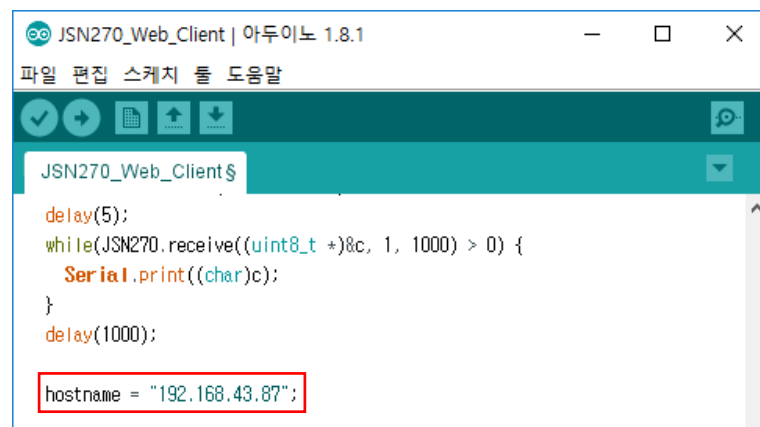
#define USE_DHCP_IP 1

#if !USE_DHCP_IP
#define MY_IP "192.168.1.133"
#define SUBNET "255.255.255.0"
#define GATEWAY "192.168.1.254"
#endif

#define HOST_IP "74.125.232.128" // www.google.com
#define REMOTE_PORT 8400
#define PROTOCOL "TCP"

SoftwareSerial mySerial(3, 2); // RX, TX
```

→ SSID 와 KEY 설정 후, REMOTE_PORT 를 8400 으로 변경한다.



```
JSN270_Web_Client | 아두이노 1.8.1
파일 편집 스케치 툴 도움말

JSN270_Web_Client$

delay(5);
while(JSN270.receive((uint8_t *)&c, 1, 1000) > 0) {
  Serial.print((char)c);
}
delay(1000);

hostname = "192.168.43.87";
```

- 접속할 주소를 나타내는 hostname 에 웹서버의 주소를 입력한다. 웹서버의 IP 주소를 알고 있기 때문에 위의 예제에서 'at+nslookup' 명령어를 사용해 DNS 서버에서 domain name 의 IP 주소를 검색하는 과정을 생략하고, 바로 IP 주소를 입력한다.

```

Serial.println("connected to server");
JSN270.println("GET / HTTP/1.1");
JSN270.println("Host: 192.168.43.87");
JSN270.println();
}
}

void loop() {

    while(JSN270.available()){
        Serial.write((char)JSN270.read());
    }

}

```

→ 해당 서버의 내용을 가져오라는 요청을 하기 위해 "GET/ HTTP/1.1"을 입력한다.
Host 에는 위의 hostname 과 같은 웹서버의 주소를 입력한다.

```

[OK]
CMD: at+nsstat
DHCP=1
IP=192.168.43.143
Netmask=255.255.255.0
Gateway=192.168.43.1
DNS=192.168.43.1
[OK]
Host IP is 192.168.43.87
CMD: at+nauto=0,1,192.168.43.87,8400
CMD: at+socket
Socket connect to 192.168.43.87
CMD: at+exit
connected to server
[OK]
HTTP/1.1 200 OK
Date: Wed, 25 Jan 2017 06:12:35 GMT
Content-Length: 108
Content-Type: text/plain; charset=utf-8
Welcome to Web Server for JSN270 Testing
Today is January 25
TIME : 2017-01-25 15:12:35.6944414 +0900 KST

```

→ 웹서버의 내용이 아두이노 콘솔창에 출력 된 것을 확인할 수 있다.

- n 초마다 서버에 접속해서 현재 시각 받아오기

```

JSN270_Web_Client | 아두이노 1.8.1
파일 편집 스케치 툴 도움말

JSN270_Web_Client$

    Serial.println("connected to server");
}
}

unsigned long prev_time=0;

void loop() {

    unsigned long current_time = millis();

    if(current_time - prev_time > 10000){
        JSN270.println("GET / HTTP/1.1");
        JSN270.println("Host: 192.168.43.87");
        JSN270.println();
        prev_time = current_time;
    }

    while(JSN270.available()){
        Serial.write((char)JSN270.read());
    }

}

```

→ 서버에 접속하는 명령어를 loop 문으로 옮긴 후, 10 초마다 서버에 접속하도록 코드를 추가한다.

```

COM3 (Arduino/Genuino Uno)

Today is January 25

TIME : 2017-01-25 15:13:15.7021079 +0900 KST
HTTP/1.1 200 OK
Date: Wed, 25 Jan 2017 06:13:25 GMT
Content-Length: 108
Content-Type: text/plain; charset=utf-8

Welcome to Web Server for JSN270 Testing

Today is January 25

TIME : 2017-01-25 15:13:25.6929279 +0900 KST
HTTP/1.1 200 OK
Date: Wed, 25 Jan 2017 06:13:36 GMT
Content-Length: 108
Content-Type: text/plain; charset=utf-8

Welcome to Web Server for JSN270 Testing

Today is January 25

TIME : 2017-01-25 15:13:36.4897008 +0900 KST
HTTP/1.1 200 OK
Date: Wed, 25 Jan 2017 06:13:45 GMT
Content-Length: 108
Content-Type: text/plain; charset=utf-8

Welcome to Web Server for JSN270 Testing

Today is January 25

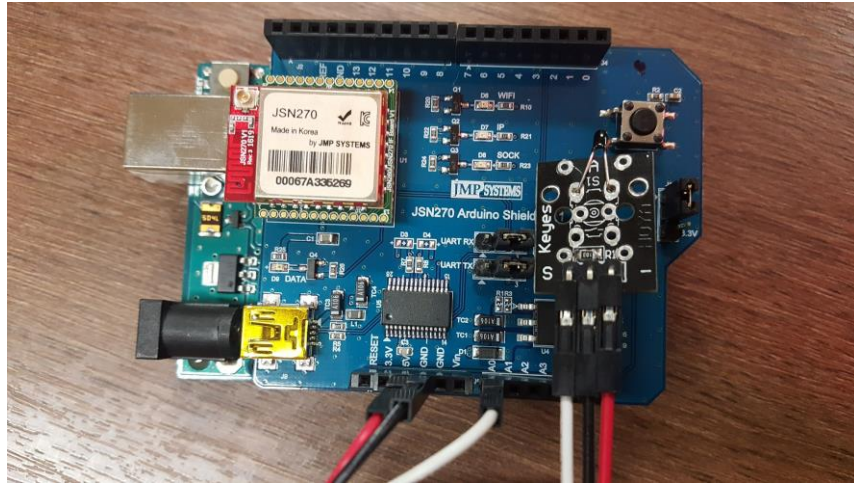
TIME : 2017-01-25 15:13:45.6893463 +0900 KST

```

→ 10 초마다 서버내용이 출력되는 것을 확인할 수 있다.

(3) 아두이노에서 온도 센서로 온도 측정 후 서버로 데이터 전송

- 온도센서 연결 방법



→ 위 사진처럼 온도센서의 핀을 차례대로 A0 핀, GND(그라운드), 5V 에 연결한다.
온도센서는 KY-013 모델을 사용했다.

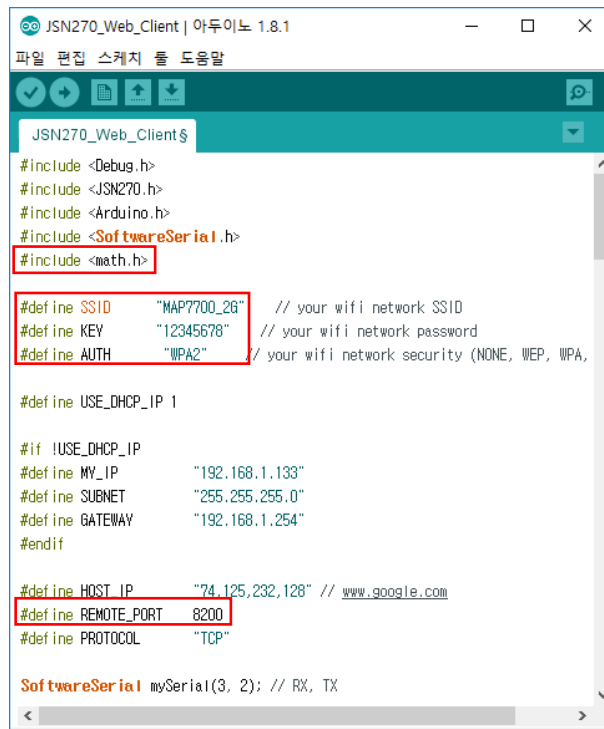
```
192.168.43.87:8200
Welcome~~ JSN270 Web Server!

-----
Total Query: 0

Temp: 온도 값 미입력...
Humi: 습도 값 미입력...

Below is what I received.
```

웹서버 내용은 위와 같다. (PORT : 8200)



```
JSN270_Web_Client$
#include <Debug.h>
#include <JSN270.h>
#include <Arduino.h>
#include <SoftwareSerial.h>
#include <math.h>

#define SSID "MAP7700_26" // your wifi network SSID
#define KEY "12345678" // your wifi network password
#define AUTH "WPA2" // your wifi network security (NONE, WEP, WPA,

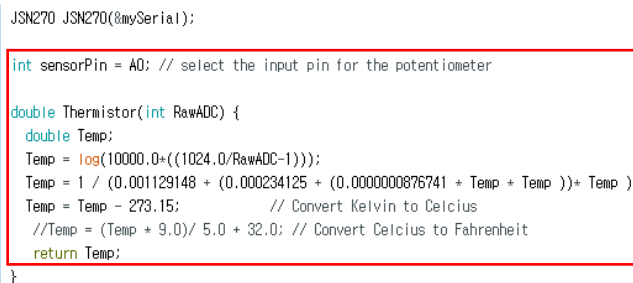
#define USE_DHCP_IP 1

#if !USE_DHCP_IP
#define MY_IP "192.168.1.133"
#define SUBNET "255.255.255.0"
#define GATEWAY "192.168.1.254"
#endif

#define HOST_IP "74.125.232.128" // www.google.com
#define REMOTE_PORT 8200
#define PROTOCOL "TCP"

SoftwareSerial mySerial(3, 2); // RX, TX
```

→ <math.h> 헤더파일 추가 후 REMOTE_PORT 를 8200 으로 수정한다.

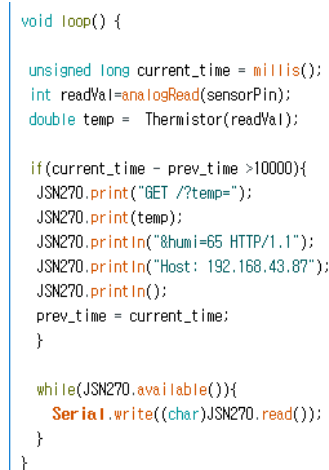


```
JSN270 JSN270(mySerial);

int sensorPin = A0; // select the input pin for the potentiometer

double Thermistor(int RawADC) {
  double Temp;
  Temp = log(10000.0+((1024.0/RawADC-1)));
  Temp = 1 / (0.001129148 + (0.000234125 + (0.0000000876741 * Temp * Temp ))* Temp );
  Temp = Temp - 273.15; // Convert Kelvin to Celcius
  //Temp = (Temp * 9.0)/ 5.0 + 32.0; // Convert Celcius to Fahrenheit
  return Temp;
}
```

→ 센서의 핀을 연결한 A0 핀을 설정한다. 온도 값을 계산하는 새로운 함수를 추가한다.



```
void loop() {

  unsigned long current_time = millis();
  int readVal=analogRead(sensorPin);
  double temp = Thermistor(readVal);

  if(current_time - prev_time >10000){
    JSN270.print("GET /?temp=");
    JSN270.print(temp);
    JSN270.println("&humi=65 HTTP/1.1");
    JSN270.println("Host: 192.168.43.87");
    JSN270.println();
    prev_time = current_time;
  }

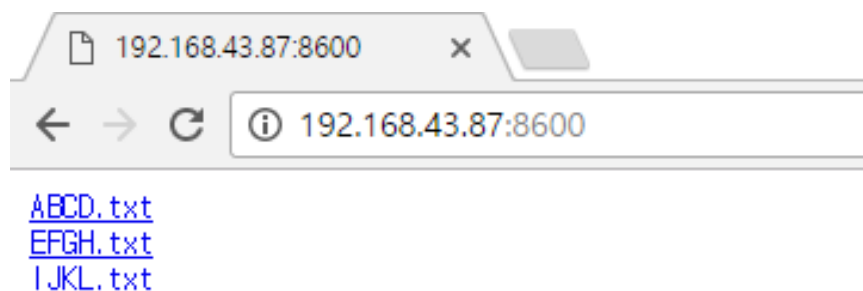
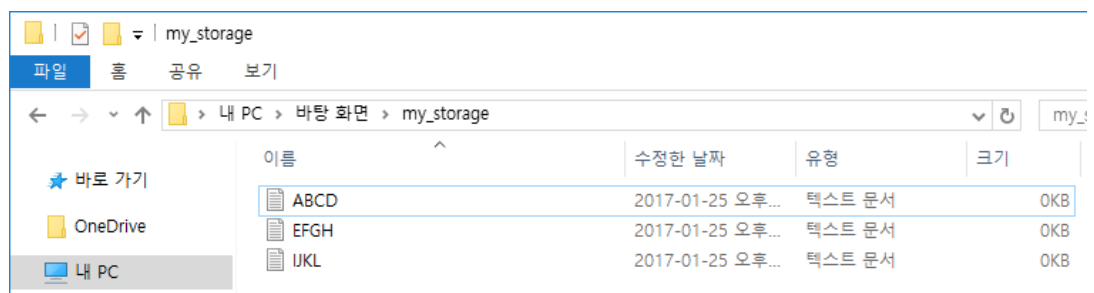
  while(JSN270.available()){
    Serial.write((char)JSN270.read());
  }
}
```

→ GET 을 사용해 측정된 온도값을 서버에 전달한다. 습도는 측정하지 못하기 때문에 임의의 값으로 설정했다.

→ 실행결과와 위와 같다. 측정된 온도가 서버에 전송되었고, 서버는 이 정보를 바탕으로 온도와 습도 상태를 판단한 후 클라이언트에게 전달한다.

(4) 폴더 구조 수신

웹서버가 존재하는 위치에 있는 my_storage 라는 폴더의 파일 목록을 수신하는 예제이다.
my_storage 폴더의 파일 목록은 다음과 같다.



→ 웹서버에 접속하면 위처럼 파일 목록을 보여주는 것을 확인할 수 있다.



```
#define USE_DHCP_IP 1

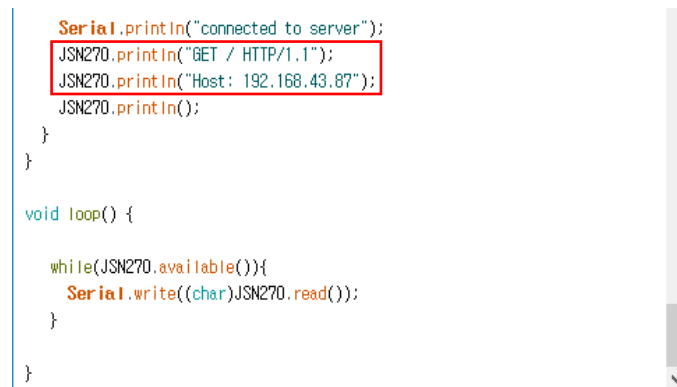
#if !USE_DHCP_IP
#define MY_IP      "192.168.1.133"
#define SUBNET     "255.255.255.0"
#define GATEWAY    "192.168.1.254"
#endif

#define HOST_IP     "74.125.232.128" // www.google.com
#define REMOTE_PORT 8600
#define PROTOCOL    "TCP"

SoftwareSerial mySerial(3, 2); // RX, TX

JSN270 JSN270(&mySerial);
```

→ REMOTE_PORT 를 8600 으로 바꾼다.



```
Serial.println("connected to server");
JSN270.println("GET / HTTP/1.1");
JSN270.println("Host: 192.168.43.87");
JSN270.println();
}
}

void loop() {

  while(JSN270.available()){
    Serial.write((char)JSN270.read());
  }
}
```

→ GET 을 사용해 서버에 요청을 한다.



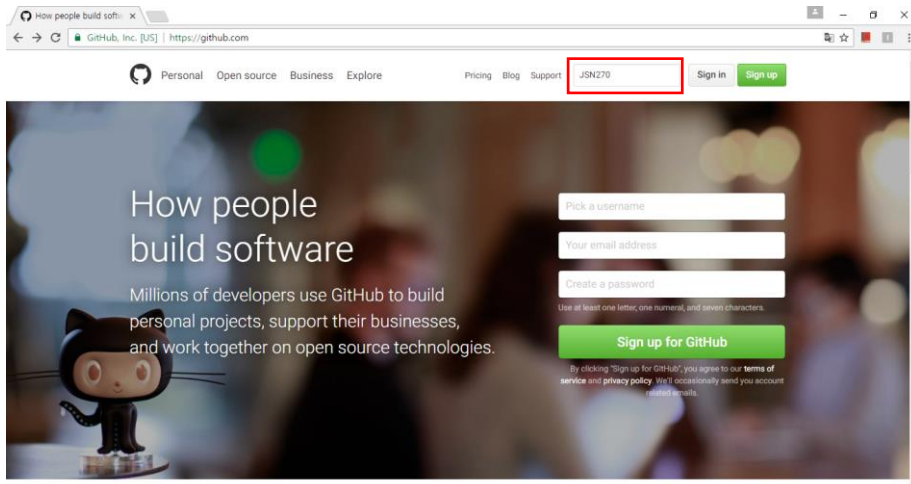
```
COM3 (Arduino/Genuino Uno)
[OK]
Host IP is 192.168.43.87
CMD: at+nauto=0,1,192.168.43.87,8600
CMD: at+socket
Socket connect to 192.168.43.87
CMD: at+exit
connected to server
[OK]
HTTP/1.1 200 OK
Content-Type: text/html; charset=utf-8
Last-Modified: Wed, 25 Jan 2017 08:19:58 GMT
Date: Wed, 25 Jan 2017 08:22:37 GMT
Content-Length: 109

<pre>
<a href="ABCD.txt">ABCD.txt</a>
<a href="EFGH.txt">EFGH.txt</a>
<a href="IJKL.txt">IJKL.txt</a>
</pre>
```

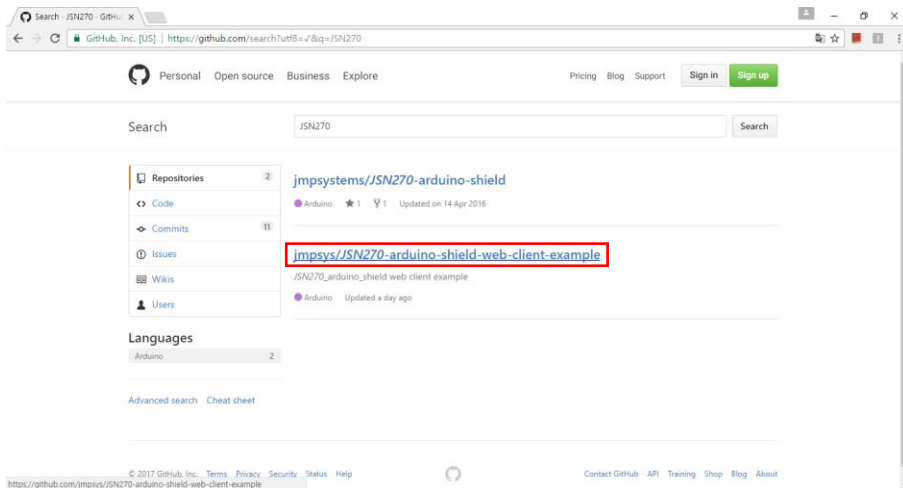
→ 실행결과와 위와 같다. 서버에서 출력된 것 처럼 my_storage 폴더의 파일들의 목록이 출력된 것을 확인할 수 있다.

4. 웹클라이언트 예제 다운로드 방법

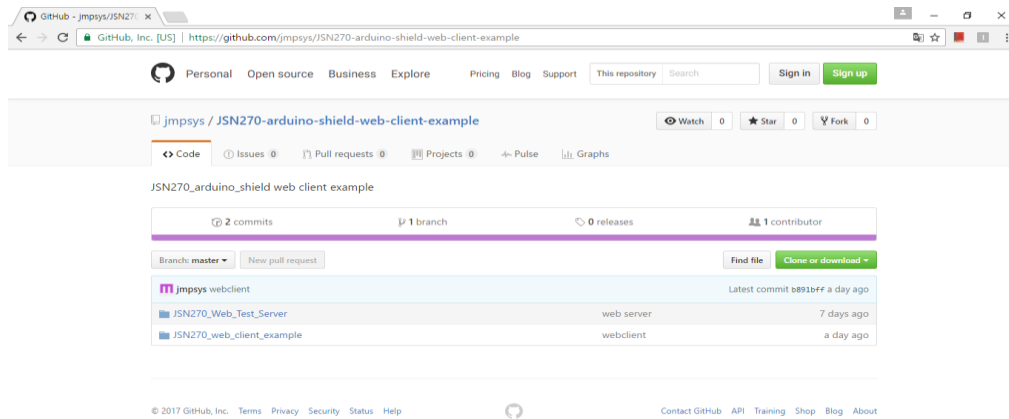
1. 주소창에 github.com 을 입력한 후 다음과 같은 창에서 JSN270 을 검색한다.



2. jmpsys/JSN270-arduino-shield-web-client-example 을 누른다.

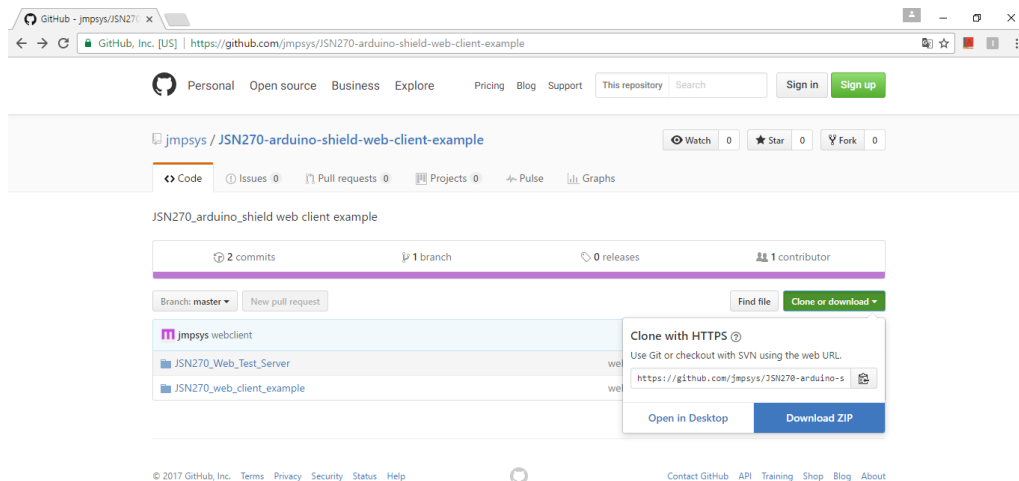


3. JSN270_Web_Test_Server 폴더에는 테스트용 웹서버 프로그램, JSN270_web_client_example 에는 웹클라이언트 예제에 대한 아두이노 파일이 들어있다.

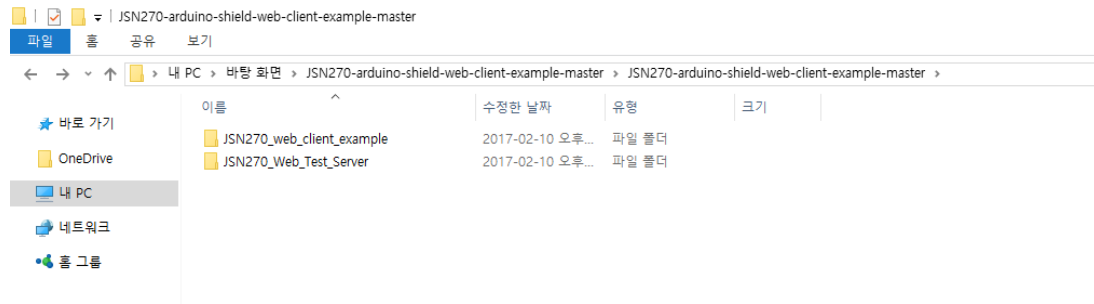


(1~3 을 생략하고 바로 <https://github.com/jmpsys/JSN270-arduino-shield-web-client-example> 에 접속해도 된다.)

4. 다운로드를 위해 Clone or download 버튼을 누른 후 Download Zip 을 선택한다.



5. 다운로드 후 압축을 풀면 아래와 같이 두개의 폴더가 포함된 것을 확인 할 수 있다.



6. Web_Test_Server 에 들어가 파일을 실행하면 다음과 같은 창이 뜬다.

모든 웹클라이언트 예제는 이 파일을 실행한 상태로 진행한다.

```

선택 C:\Users\JMP\Desktop\JSN270-arduino-shield-web-client-example-master\JSN270-arduino-shield-web-client-example-master\JSN270_We...
[ 웹 시험용 소프트웨어 for JSN270 series ] -- (주)제이엠포시스템

이 서버는 2 개의 CPU 코어를 가지고 있습니다.
2017-02-10 17:58:08.7465088 +0900 KST --- 최고 성능으로 동작합니다
본 예제를 시험하는 동안에는 윈도우 방화벽 통과를 허용해 주세요

JSN 와이파이 모듈 시험용 웹서버 입니다 (문의: www.jmpsyst.com)

[ex 1] JSN 모듈에서 웹서버에 8400포트로 접속하여 아래 형태의 현재시각 문자열을 수신해보세요

=====
Welcome to Web Server for JSN270 Testing
February 10
2017-02-10 17:58:08.8871978 +0900 KST
=====

[ex 2] JSN 모듈에서 웹서버에 8200포트로 접속하면서 다음과 같은 쿼리를 전송해보세요
http://192.168.xxx.xxx:8200/?temp=36&humi=65
<온도가 36도일 경우와 그렇지 않을 경우에 대한 반응 비교>

[ex 3] 파일서버는 8600포트이며 ./my_storage 폴더에 서비스할 파일을 저장하고 폴더구조를 수신해보세요
자세한 내용 구글 검색 키워드 : "JSN270 IoT 예제"

```