WIZnet Academy 2017

IoT Device! mbed로 쉽게 연결하자 !

정찬미

becky@wiznet.io

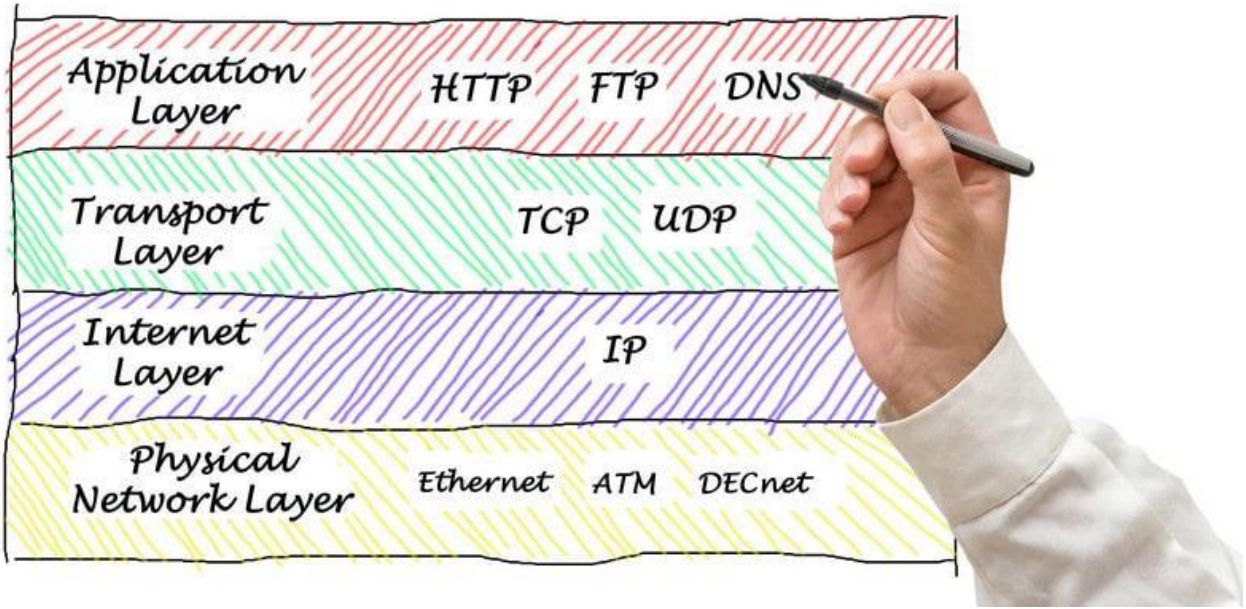
목차

- IoT Network 상식
  - TCP/UDP, DHCP, DNS, HTTP 알아보기
- mbed로 IoT Device 개발하기
  - Ethernet Interface
- IoT Device 실습
  - Web Client 개념 이해하기
  - Web Client 실습
  - Web Server 개념 이해하기

IoT Network 상식

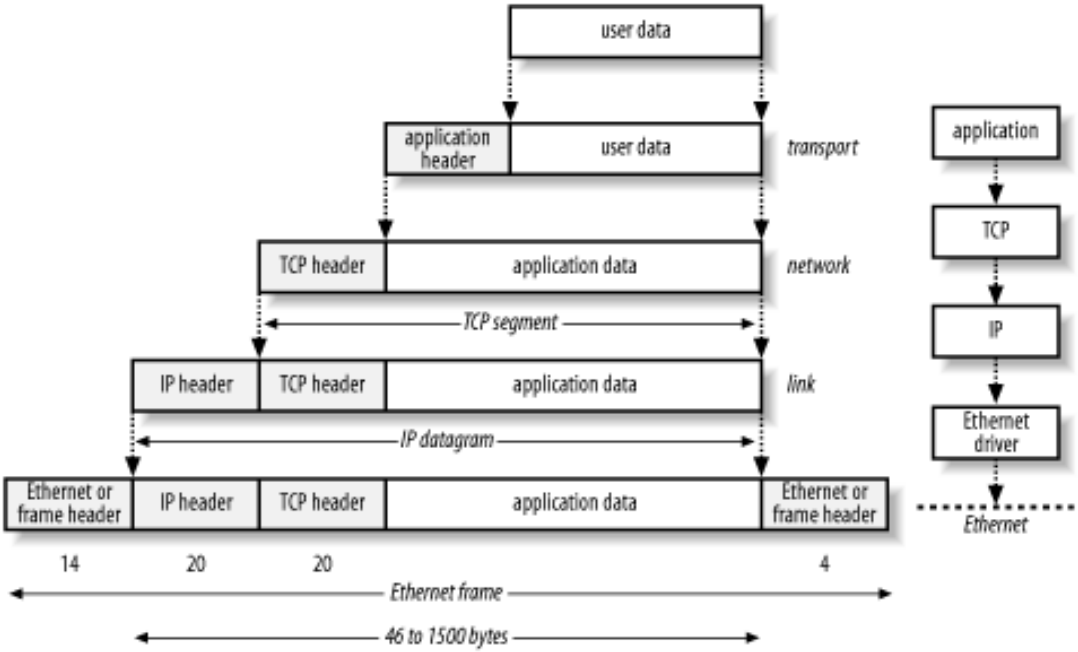
IoT Network 상식

» Networ layer



IoT Network 상식

» Networ layer



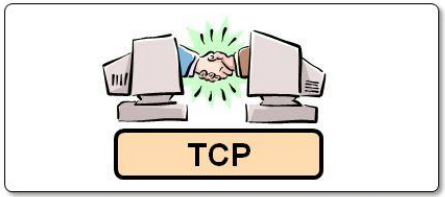
IoT Network 상식

TCP vs UDP

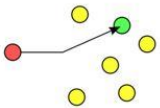
TCP (Transmission Control Protocol)	UDP (User Datagram Protocol)
연결형 프로토콜	비연결형 프로토콜
신뢰성 있는 데이터 전송 (데이터의 재전송 존재)	비신뢰성 데이터 전송 (데이터의 재전송 없음)
1 : 1통신(Unicast)	1 : 1통신 (Unicast) 1 : N 통신 (Broadcast) N : N 통신 (Multicast)

IoT Network 상식

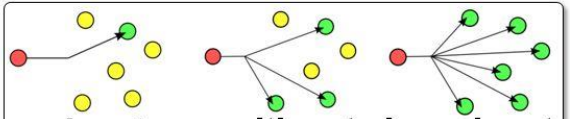
TCP vs UDP



- Slower but reliable transfers
- Typical applications:
  - Email
  - Web browsing



- Fast but non-guaranteed transfers ("best effort")
- Typical applications:
  - VoIP
  - Music streaming



IoT Network 상식

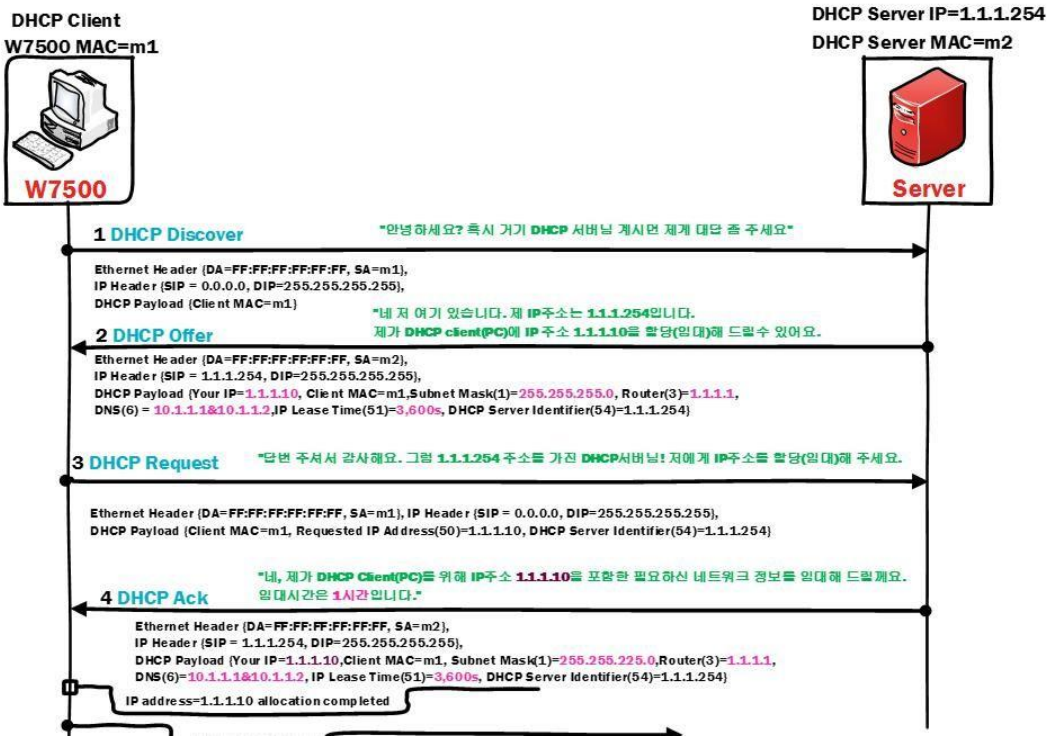
» DNS(Dopmain name system)

WHO CARES ABOUT NAVER SERVER IP ADDRESS ?????

DNS 동영상 자료

IoT Network 상식

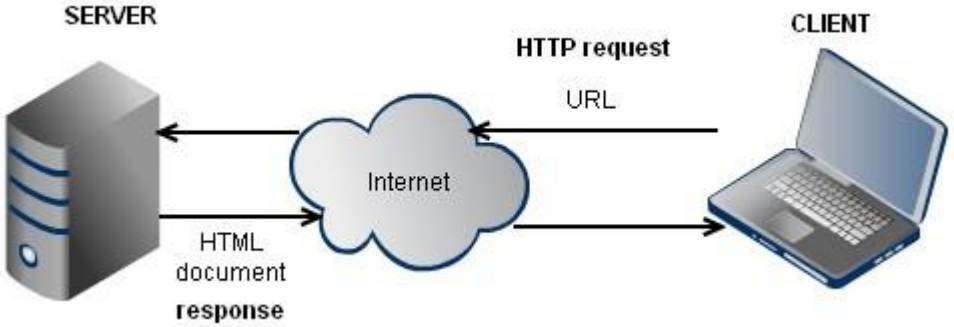
» DHCP(Dynamic Host Configuration Protocol)



DHCP 동영상 자료

IoT network 상식

HTTP (Hypertext Transfer Protocol)



- 인터넷에서 hypertext 문서를 교환하기 위하여 사용되는프로토콜
- TCP 기반 통신방식
- Connectionless
- Stateless
- Request-Response

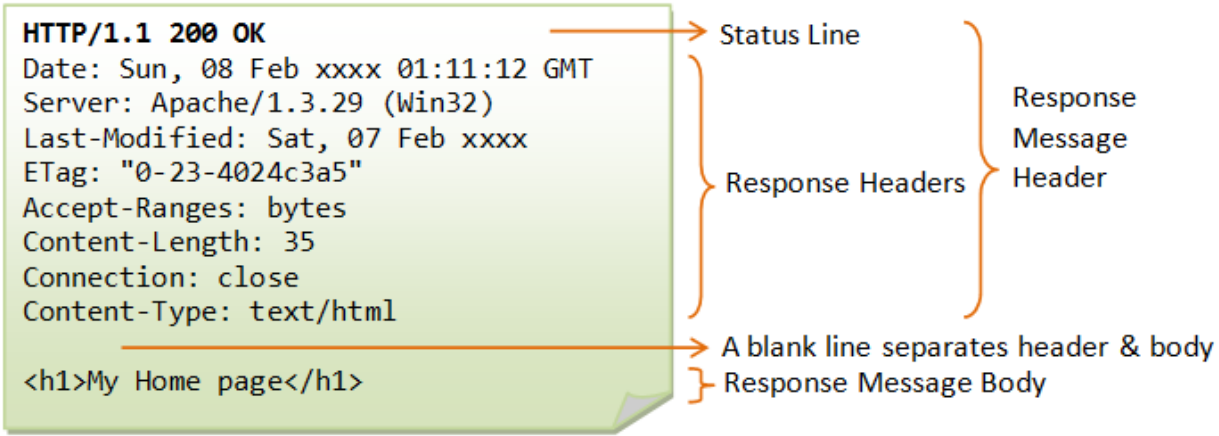
IoT Network 상식

HTTP Method

- GET
  - Web Server에 있는 특정 URI가 가지고 있는 Data를 얻기 위한 방법
- POST
  - Web Server에 있는 특정 URI가 가지고 있는 Data를 변경하기 위한 방법
- PUT
  - Web Server에 있는 특정 URI에 Data를 저장하기 위한 방법
- DELECT
  - Web Server에 있는 특정 URI에 있는 Data를 지우기 위한 방법

IoT Network 상식

HTTP Response Sturcture



mbed로 IoT Device 개발하기

mbed로 IoT Device 개발하기

Ethernet Interface

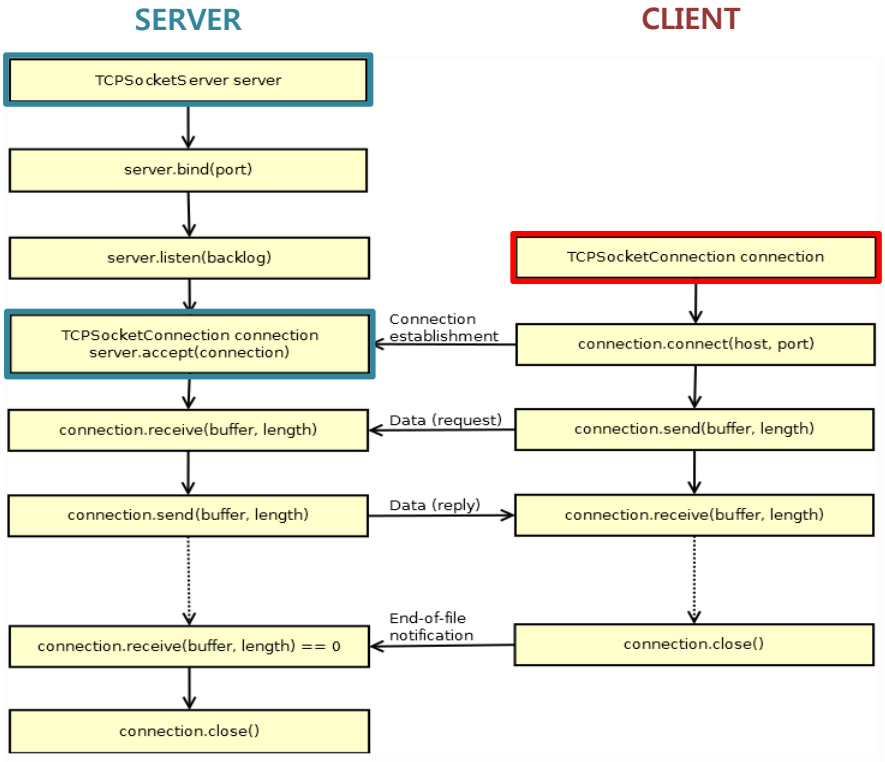
- mbed에서 제공하는 소프트웨어 스택을 사용하기 위해서 사용하는 라이브러리
  EthernetInterface

- WIZnet 칩을 사용하여 하드웨어 스택을 사용하기 위해서 사용하는 라이브러리
  WIZnetInterface

- 사용되는 Class 들
  - EthernetInterface — Network 세팅을 위한 클래스
  - TCPSocketServer — TCP Server 클래스
  - TCPSocketConnection — TCP 연결 성립과 데이터를 주고 받기 위한 클래스

mbed로 IoT Device 개발하기

Ethernet Interface



mbed로 IoT Device 개발하기

» Ethernet Interface

ETHERNET CLASS

```
EthernetInterface eth;
```

```
#ifdef DHCP
    eth.init(mac_addr);
#else
    eth.init(mac_addr, ip_addr, subnet_mask, gateway_addr);
#endif
```

```
if(eth.link() == true);
eth.connect();
```

SERVER CLASS

CLIENT CLASS

```
#ifdef SERVER
    TCPSocketServer server;
    server.bind(SERVER_PORT);
    server.listen();
#else
    TCPSocketConnection client;
    client.connect(SERVER_IP, SERVER_PORT);
#endif
```

IoT Device 실습

Web Server, Web Client

Web Server, Web Client

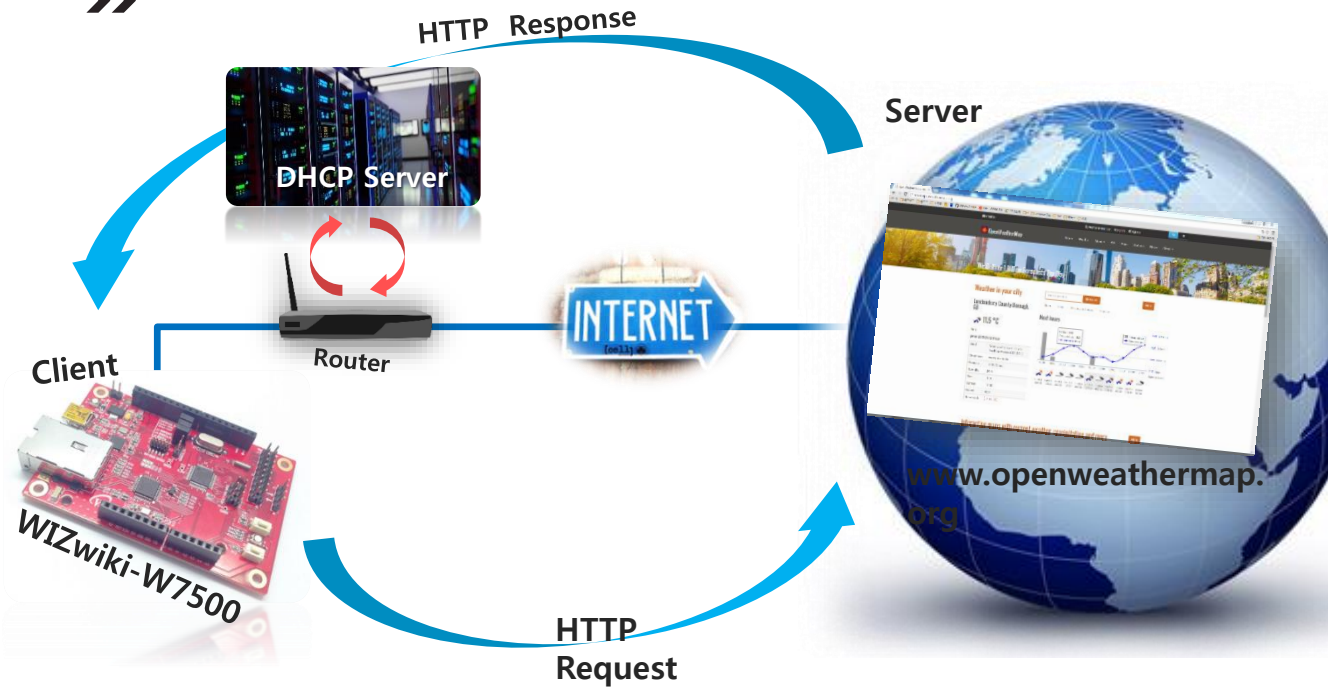
›› 개요

웹 서버

인터넷 서비스를 하기 위한 서버의 일종으로서, Web Client(웹 브라우저)에게 컨텐츠를 제공하는 서버로서, 정적인 HTML이나 jpeg, gif 같은 이미지를 HTTP 프로토콜을 통해 웹 브라우저에 전송한다.

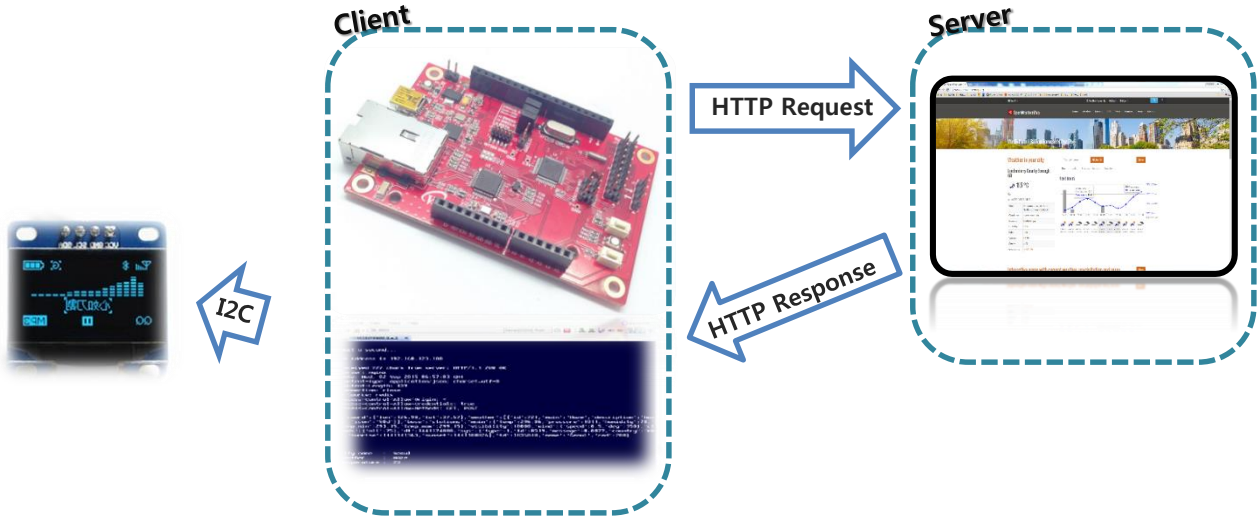
Web Client 실습

Web Client 실습



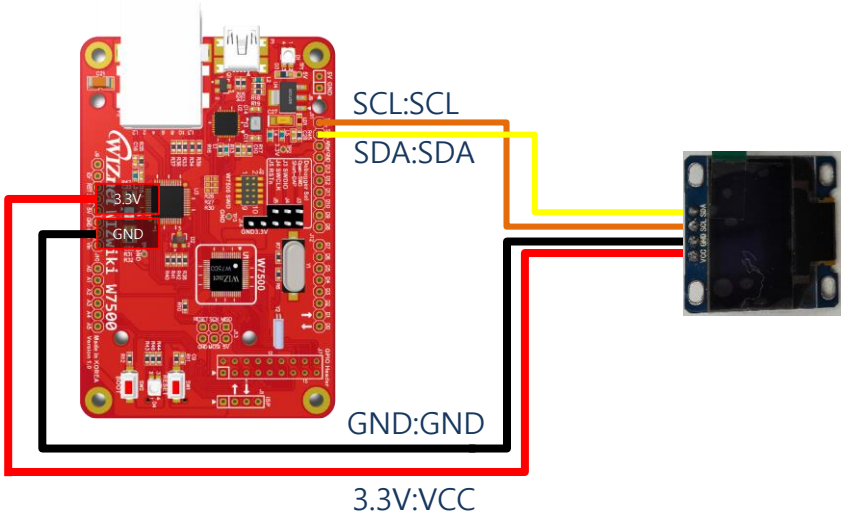
Web Client 실습

Weather forecast 개요



Web Client 실습

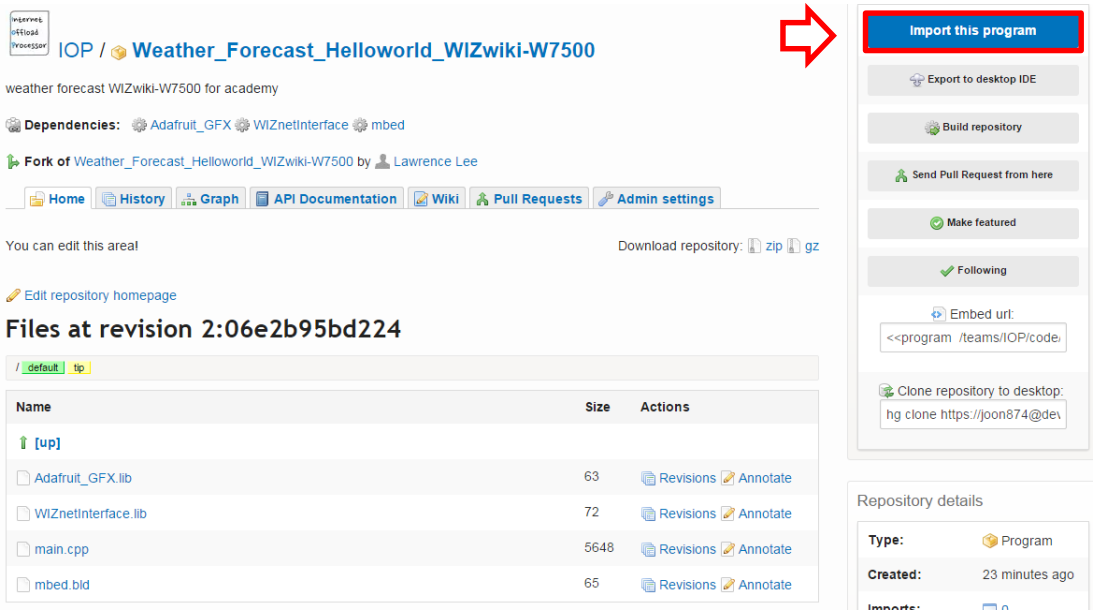
HW



Web Client 실습

Import Project

https://os.mbed.com/teams/IOP/code/HTTPWebClient-WIZwiki-7500/



Web Client 실습

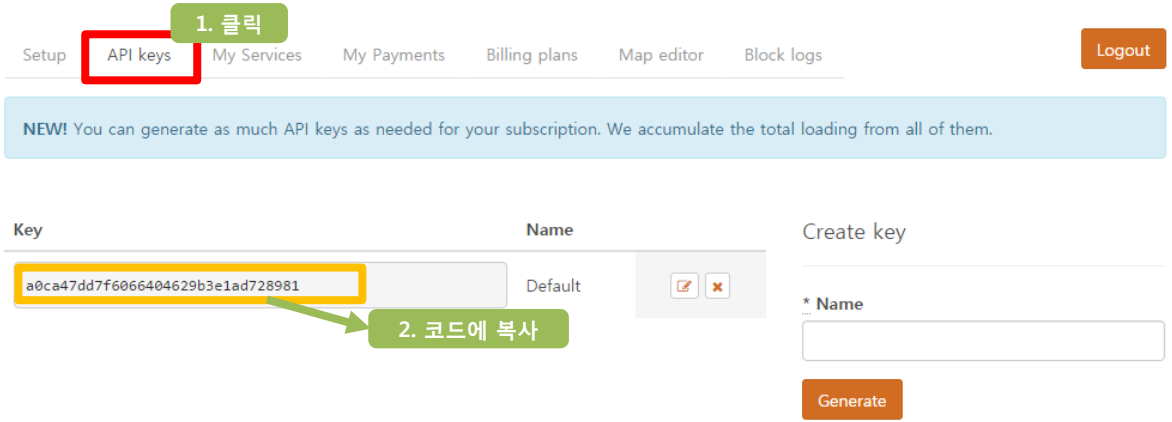
Openweather Map

http://openweathermap.org/

Web Client 실습

Openweather Map

http://openweathermap.org/



Web Client 실습

Server Weather Data

```
Received 785 chars from server: HTTP/1.1 200 OK
Server: nginx
Date: Thu, 03 Sep 2015 00:47:26 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 497
Connection: close
X-Source: redis
Access-Control-Allow-Origin: *
Access-Control-Allow-Credentials: true
Access-Control-Allow-Methods: GET, POST
```

HTTP Header


```
{"coord":{"lon":126.98,"lat":37.57},"weather":[{"id":701,"main":"Mist","desc
ription":"mist","icon":"50n"},{"id":721,"main":"Haze","description":"haze","
icon":"50n"}],"base":"stations","main":{"temp":294.83,"pressure":1012,"humid
ity":78,"temp_min":293.15,"temp_max":296.15},"visibility":10000,"wind":{"spe
ed":0.5,"deg":20},"clouds":{"all":75},"dt":1441237800,"sys":{"type":1,"id":8
519,"message":0.0093,"country":"KR","sunrise":1441227800,"sunset":1441274361
},"id":1835848,"name":"Seoul","cod":200}
```


HTTP Body

Web Client 실습

Weather Data Parsing

```
165     char *date;
166     char *weather;
167     char *city;
168     char *temper;
169
170     char cur_date[17] = {0};
171     char weather_con[15] = {0};
172     char city_name[10] = {0};
173     char temper_data[3] = {0};
```

Buffer에서 원하는 문자열의 시작주소를 가져오기 위한 포인터변수

Buffer에서 원하는 문자열을 가져오기 위한 배열 선언

```
181     date = strstr(buffer, "Date");
182     for(int x=0;x<17;x++){
183         cur_date[x] = date[x+6];
184     }
```

srtsrt()을 이용해 Buffer안에 "Data"라는 문자열의 첫번째 주소값 저장

cur_data[x] = Data[x+6]을 통해 저장된 주소에서 다음 6번째 주소부터 차례로 저장

```
189     weather = strstr(buffer, "main");
190     for(int i=0; i<15;i++){
191         weather_con[i] = weather[i+7];
192         if(weather_con[i] == 34){
193             weather_con[i] = 0;
194             break;
195         }
196     }
```

Web Client 실습

Temp Data Parsing

```
218  /*
219   *    kelvin to celius converter
220   */
221   num100 = temper_data[0] - 48 - 2;
222   num10 = temper_data[1] - 48 - 7;
223   num1 = temper_data[2] - 48 - 3;
224
225   temp = num100*100 + num10*10 + num1;
```

Char형으로 저장된 변수를 십진수로 변환
또한 -273함으로 절대온도를 섭씨로 변환

```
234  /*
235   *    OLED Display
236   */
237   gOled.begin();
238   gOled.clearDisplay();
239
240   gOled.printf("%s\n", cur_date);
241   gOled.printf("City    : %s\n", city_name);
242   gOled.printf("Weather : %s\n", weather_con);
243   gOled.printf("Temper  : %d\n", temp);
244   gOled.display();
245   gOled.setTextCursor(0,0);
```

OLED에 날씨 정보 Display

Web Client 실습 응용

Web Client 실습 응용

Country parsing을 추가해보자

```
char *date;
char *weather;
char *city;
char *temper;
char *country;
```

country 포인터변수 선언

```
char cur_date[17] = {0};
char weather_con[15] = {0};
char city_name[10] = {0};
char temper_data[3] = {0};
char country_name[5]={0};
int temp;
int num100, num10, num1;
```

country_name 배열 선언

```
// parsing country name
country = strstr(buffer, "country");
for(int j=0; j<5;j++){
    country_name[j] = country[j+10];
    if(country_name[j] == 34){
        country_name[j] = 0;
        break;
    }
}
```

country 정보 data parsing

Web Client 실습 응용

Country parsing 추가

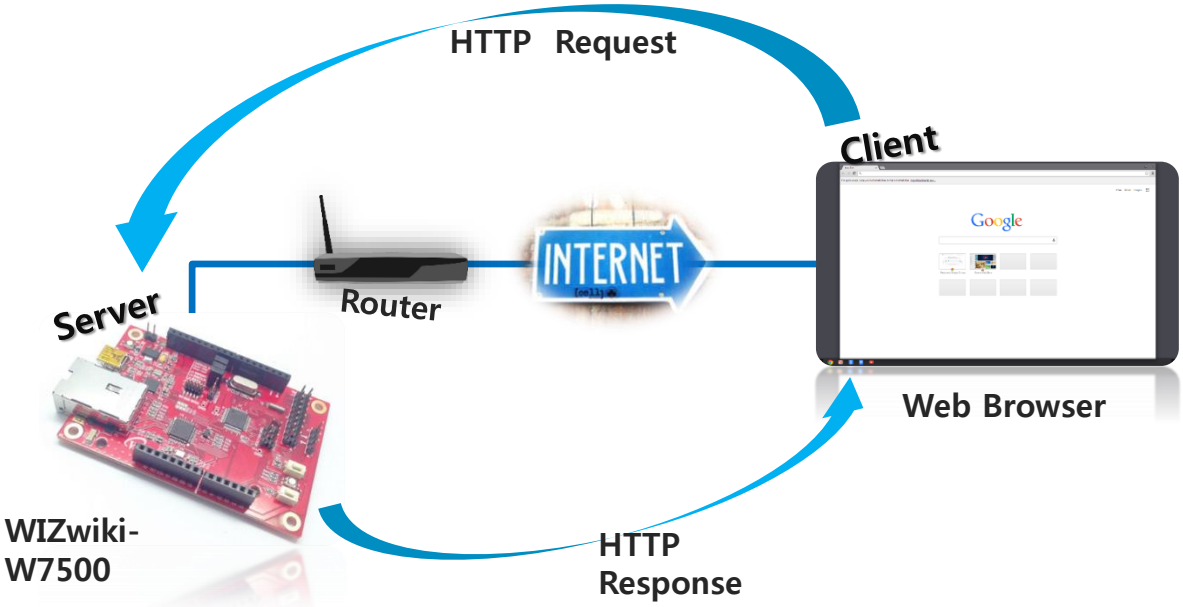
```
// Debug message
printf("city name    :  %s\r\n",  city_name);
printf("country name    :  %s\r\n",  country_name);
printf("weather      :  %s\r\n",  weather_con);
printf("temperature  :  %d\r\n\r\n",  temp);

// OLED Display
gOled.begin();
gOled.clearDisplay();
gOled.printf("%s\n\n", cur_date);
gOled.printf("City     : %s\n", city_name);
gOled.printf("Country     : %s\n", country_name);
gOled.printf("Weather  : %s\n", weather_con);
gOled.printf("Temper   : %d\n", temp);
gOled.display();
gOled.setTextCursor(0,0);
```

country debug message 출력!

Web Server

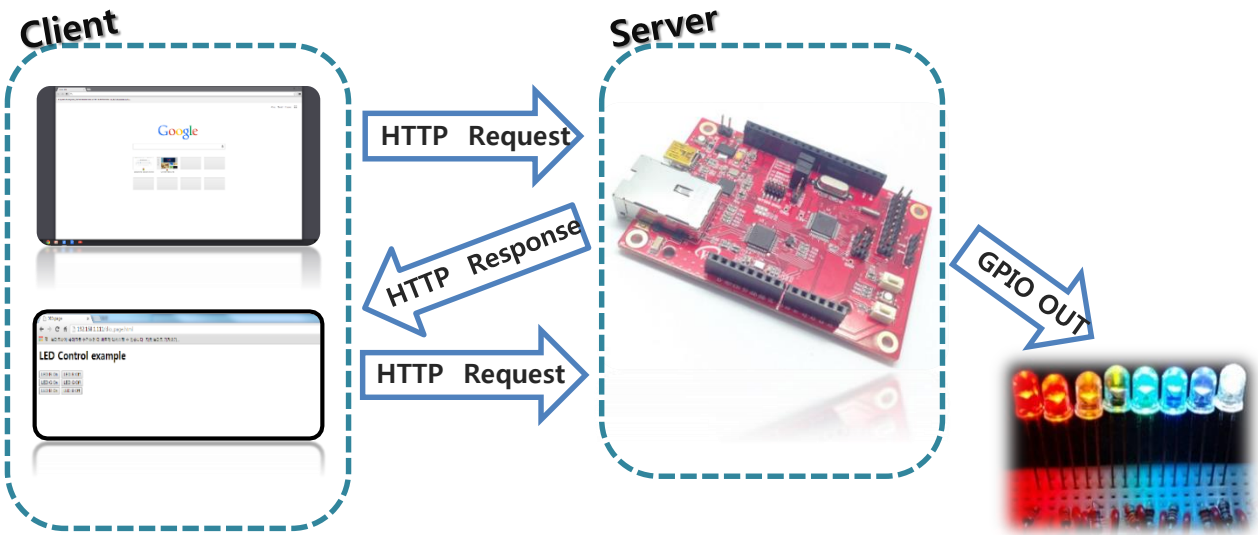
Web Server 실습

Embedded Web Server



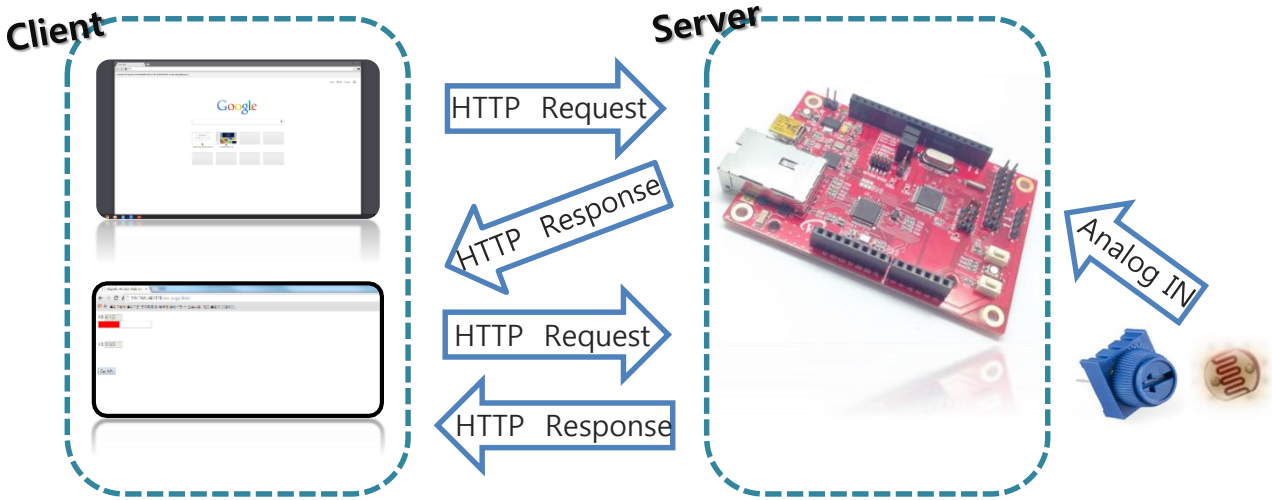
Web Server

Web page로 Web Server LED 컨트롤



Web Server

Web page로 Web Server에 Analog Sensor 값 요청



감사합니다

WIZnet Academy 2017

IoT Device! mbed로 쉽게 연결하자!

Thank you

Team SoC
Becky

becky@wiznet.io