How to implement HTTP Sever in W5200

Version 0.9βeta

`

Table of Contents

1 Introduction

An embedded HTTP (Hyper Text Transfer Protocol) server, or web server (as it is commonly called), is an excellent addition to any network-enabled device. HTTP server capability allows an embedded device to be monitored and controlled remotely using any standard, off-the-shelf Internet browser. Owing to the ubiquitous deployment of Internet browsers, a web-enabled device can be accessed from almost any computer – desktop or mobile.

This WIZnet W5200 HTTP server application note and the included W5200E01-M3 board provide an HTTP Server module that can be integrated with almost any application on a STM32 Cortex M3 microcontroller product. IAR 5.41 is required to compile this HTTP server application note. W5200E01-M3 board is required to run this HTTP server application note.

All codes and files mentioned in this document are available for download from www.wiznet.co.kr/w5200/download.

1.1 ASSUMPTION

The author assumes that the reader is familiar with WIZnet W5200 driver. It is also assumed that the reader is familiar with C programming language and HTTP server concepts. Terminology from these technologies is used in this document, and only brief overviews of the concepts are provided. Advanced users are encouraged to read the associated specifications.

1.2 Features:

The HTTP server provided here does not implement all HTTP functionality; it is a minimal server targeted for embedded systems. The user can easily add new functionality as required.

The HTTP server presented here incorporates the following features:

• Provides portability across the 32-bit family of STM32 Cortex M3 microcontrollers

• Supports multiple HTTP connections (maximum 8 sockets)

• Supports the HTTP methods: HEAD, GET and POST

• Supports Common Gateway Interface (CGI) to invoke predefined functions from within the remote browser Function 1: configure the network parameters (IP, Subnet and Gateway) of the W5200E01-M3

Function 2: monitor and control the LED3 and LED4 of the W5200E01-M3

Monitor: IP address...LED status
Control: IP address...LED status
ioPlatform

Figure 1.

2 HTTP Protocol overview

HTTP is an abbreviation of Hyper Text Transfer Protocol. HTTP functions as a request-response protocol in the client-server computing model. In HTTP, a web browser, for example, acts as a *client*, while an application running on a computer hosting a web site functions as a *server*. The client submits an HTTP request message to the server. The server, which stores content, or provides *resources*, such as HTML files, or performs other functions on behalf of the client, returns a response message to the client. A response contains completion status information about the request and may contain any content requested by the client in its message body.

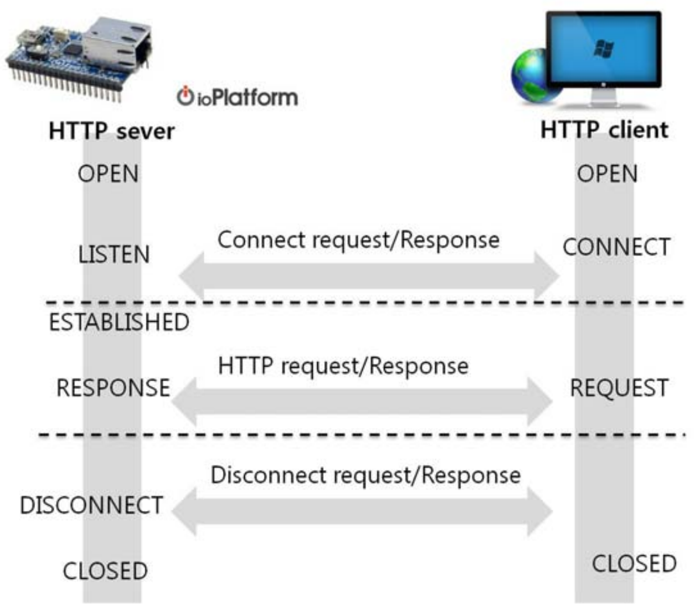


Figure 2. HTTP communication

The communication process of the HTTP server can be roughly divided into three steps.

1. Connection: The process of W5200 assigning the socket to HTTP server, opening the socket and listening.
2. Communication: The connection established. And the W5200 sending HTTP response after the HTTP request is received from the client.
3. Closing: The process of finishing connection after all HTTP request/response is done.

HTTP is an Application Layer protocol designed within the framework of the Internet Protocol Suite. The protocol definitions presume a reliable Transport Layer protocol for host-to-host data transfer. The TCP is the dominant protocol in use for this purpose.

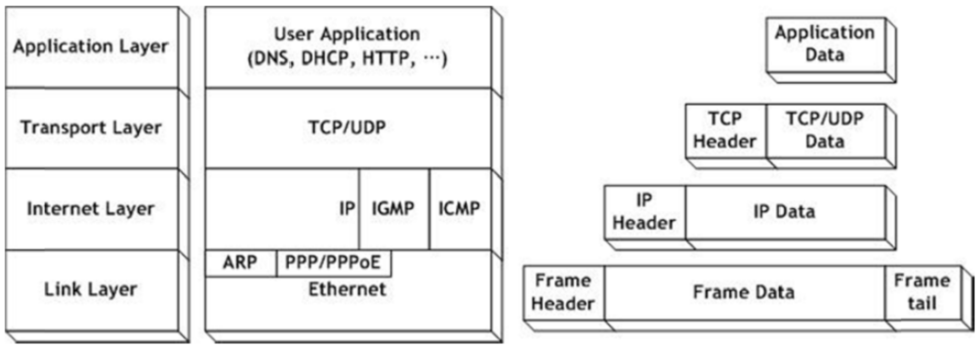


Figure 3 Encapsulation of data as it goes down the protocol stack

W5200 has already embedded Ethernet, IP and TCP layer. Thus, HTTP server can be easily implemented as long as using the W5200 API functions which come with W5200 driver. Fig. 4 below shows the block diagram of W5200.

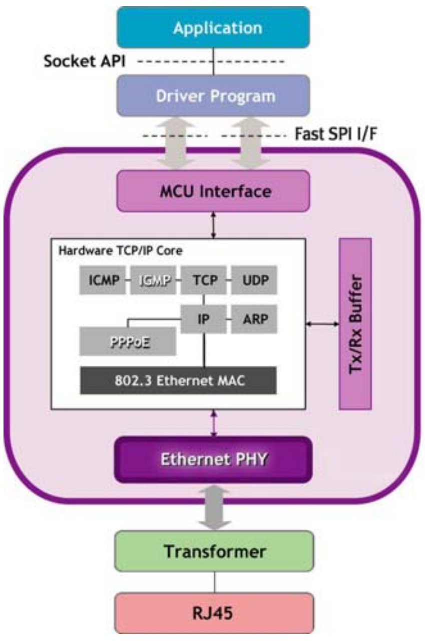


Figure 4. Block Diagram of W5200

2.~~2~~1 HTTP Session

An HTTP session is a sequence of network request-response transactions. An HTTP client initiates a request by establishing a TCP connection to a particular port on a server (typically port 80). An HTTP server listening on that port waits for a client's request message. Upon receiving the request, the server sends back a status line, such as "HTTP/1.1 200 OK", and a message of its own, the body of which is perhaps the requested resource, an error message, or some other information

2.2 Request message

The request message consists of the following:

- Request line, such as GET /images/logo.png HTTP/1.1, which requests a resource called /images/logo.png from server
- Headers, such as Accept-Language: en
- An empty line.
- An optional message body.

The request line and headers must all end with <CR><LF> (that is, a carriage return followed by a line feed). The empty line must consist of only <CR><LF> and no other whitespace. Although <CR><LF> is required <LF> alone is also accepted by most servers.

2.3 Response message

HTTP defines nine methods (sometimes referred to as "verbs") indicating the desired action to be performed on the identified resource. The HTTP server provided here does not implement all HTTP functionality; it is a minimal server targeted for embedded systems. Only HEAD, GET and POST method is available.

- HEAD: Asks for the response identical to the one that would correspond to a GET request, but without the response body.
- GET: Requests a representation of the specified resource.
- POST: Submits data to be processed (e.g., from an HTML form) to the identified resource. The data is included in the body of the request.

2.4 Example of session

~~2.4 Example of session~~

서식 있음: 글꼴: (한글) 맑은 고딕, (한글) 한국어

Below is a sample conversation between an HTTP client and an HTTP server running on W5200 http server 192.168.11.88, port 80.

2.4.1 Client request

```
GET / HTTP/1.1\r\n
Host: 192.168.11.33\r\n
\r\n
```

2.4.2 Server response

```
HTTP/1.1 200 OK\r\n
Content-Type: text/html\r\n
Content-Length: 348\r\n
<html>\r\n
<head>\r\n
<meta http-equiv="content-type" content="text/html; charset=ksc5601" />\r\n
<title>A.K. : WIZ-Embedded webserver - wiznet</title>\r\n
</head>\r\n
<frameset cols="197,*" border=0>\r\n
<frame name="left"   src="left.htm" marginheight=2 marginwidth=2>\r\n
<frame name="main"  src="main.htm" marginheight=5 marginwidth=5>\r\n
</frameset>\r\n
</html>\r\n
\r\n
```

A server response is followed by a blank line and text of the requested page. *Content-Type* specifies the Internet media type of the data conveyed by the http message, while *Content-Length* indicates its length in bytes.

3. HTTP Server Implementation

3.1 Network setting

Network parameters (IP, Subnet and Gateway) of HTTP server are saved in the Flash of STM32F103C8 which is the Host MCU of W5200E01-M3. They can use the webpage to change the Network setting. Of course, the default of Network parameters can be changed by modifying source code.

Note: below code is also the sample code to use our API SetNetInfo()

```
  /* Default and Configurable Network parameters-----------------*/

  void SetConfig(void)
  {
    uint8 i;
    unsigned char Valid[4];
    //Added by Gang 20119-27
    for (i=0 ; i < 4 ; i++)
    {
      Valid[i]= *((const unsigned char*)(FlashDestination + i));
    }


    if (Valid[0]== 0x00 && Valid[1]== 0x52 && Valid[2]== 0x00 && Valid[3]== 0x00)
    {

         (netinfonew).Mac[0] = 0x00;
         (netinfonew).Mac[1] = 0x08;
         (netinfonew).Mac[2] = 0xDC;
         (netinfonew).Mac[3] = 0x01;
         (netinfonew).Mac[4] = 0x02;
         (netinfonew).Mac[5] = 0x03;

         for (i=4 ; i < 8 ; i++)
         {
             (netinfonew).IP[i-4] = *((const unsigned char*)(FlashDestination + i));
         }
         for (i=8 ; i < 12 ; i++)
         {
             (netinfonew).Gateway[i-8] = *((const unsigned char*)(FlashDestination + i));
         }
         for (i=12 ; i < 16 ; i++)
         {
             (netinfonew).Subnet[i-12] = *((const unsigned char*)(FlashDestination + i));
         }

          SetNetInfo(&netinfonew);

          after_cgi_flag=1;
    }
    else
    {
      InitNetInfo();
    }

}
```

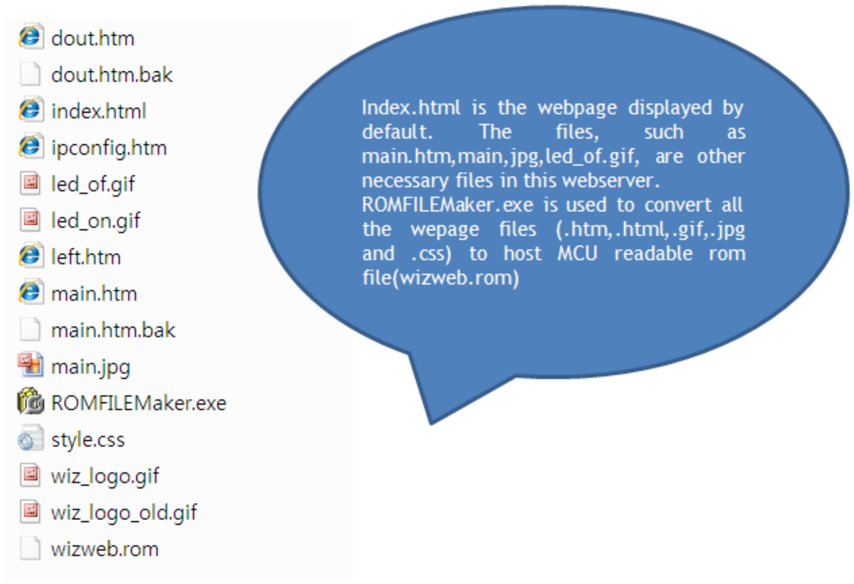
```
void InitNetInfo(void)
{
     (netinfo).Mac[0] = 0x00;
     (netinfo).Mac[1] = 0x08;
     (netinfo).Mac[2] = 0xDC;
     (netinfo).Mac[3] = 0x01;
     (netinfo).Mac[4] = 0x02;
     (netinfo).Mac[5] = 0x03;

     (netinfo).Gateway[0] = 192;
     (netinfo).Gateway[1] = 168;
     (netinfo).Gateway[2] = 11;
     (netinfo).Gateway[3] = 1;

     (netinfo).IP[0] = 192;
     (netinfo).IP[1] = 168;
     (netinfo).IP[2] = 11;
     (netinfo).IP[3] = 88;

     (netinfo).Subnet[0] = 255;
     (netinfo).Subnet[1] = 255;
     (netinfo).Subnet[2] = 255;
     (netinfo).Subnet[3] = 0;

     SetNetInfo(&netinfo);
}
```

3.2 Default homepage setting

The default webpage 'index.html' will be displayed when HTTP client is connected to this HTTP server. If user wants to change the default page to a different one, change the below code from the main.c file.

```
unsigned char *homepage_default = "index.html";
```

In order to understand what the "index.html" refers to, please see below figure:



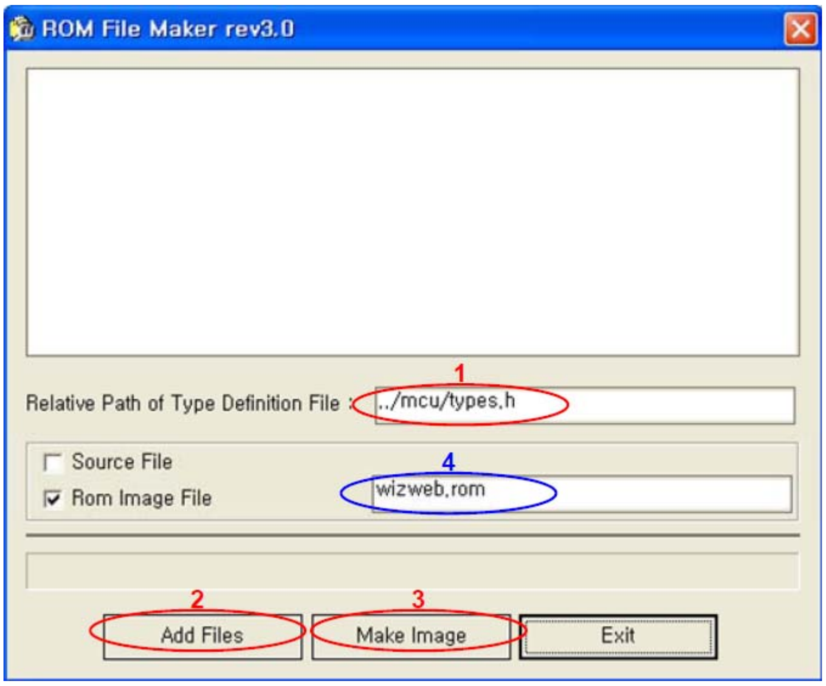
dout.htm
dout.htm.bak
index.html
ipconfig.htm
led_of.gif
led_on.gif
left.htm
main.htm
main.htm.bak
main.jpg
ROMFILEMaker.exe
style.css
wiz_logo.gif
wiz_logo_old.gif
wizweb.rom

Index.html is the webpage displayed by default. The files, such as main.htm,main,jpg,led_of.gif, are other necessary files in this webserver. ROMFILEMaker.exe is used to convert all the wepage files (.htm,.html,.gif,.jpg and .css) to host MCU readable rom file(wizweb.rom)

Figure 3.

3.3 Making ROM file

Example web pages must be combined together as one Romfile, and be written in the STM32F103C8's flash memory with the HTTP server implementation code. The provided program, ROMFILEMaker.exe, is for combining example web pages into one Romfile. This program can be downloaded from WIZnet's homepage -> download center (http://www.wiznet.co.kr/en/). The picture shown below is a captured screen of ROMFILEMaker.exe running.


ROM File Maker rev3.0

Relative Path of Type Definition File : ../mcu/types.h

☐ Source File

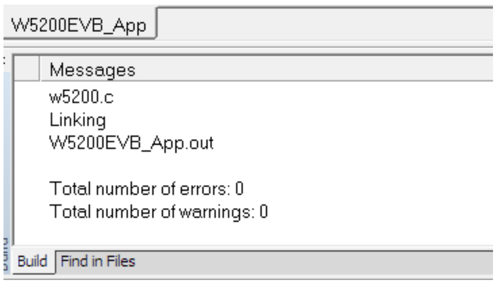
☑ Rom Image File: wizweb.rom

Figure 4. ROMFILEMaker program

Assign the relative path of types.h file from the W5200 code in area #1. Click the Add Files button, highlighted #2, to add web pages that should be combined. If too many web pages are all added at the same time, an error will occur; therefore, if user plans to combine numerous web pages, repeat the adding step couple of times. Currently, the web pages exist in the same folder. If user clicks the Make Image button, highlighted #3, the combined files are created as the file name in area #4. Do not change the Rom Image file name, highlighted #4, since a change in that will cause the user to change the batch file command, that is used for combining the program binary and a web page.

3.4 Program binary making

Use the IAR 5.41 to compile files that is attached in the HTTP server application note. If successful, the message like shown below will appear in the Output Window. And then find the binary file named W5200EVB_App.bin in \Work\App\Debug\Exe. Right click this bin file and get its size from prosperities option. Finally change its size to hex value and write this hex value in following macro define.(For example: 19940 byte = 4DE4 byte)



```
Messages
w5200.c
Linking
W5200EVB_App.out

Total number of errors: 0
Total number of warnings: 0
```

Figure 5.

```
#define FLASH_ROMFILE_START_ADDRESS  0x08004DE4 //This value is determined by the
size of f/w
#define MAX_WRITABLE_FLASH_ADDRESS   0x0801FFFF //STM32F103C8 has 64K flash
(0x08000000~0x0801FFFF)
```

Note: If the source code is modified, the size of the output bin file is modified too, therefore, the size of this definition "FLASH_ROMFILE_START_ADDRESS" must be modified every time after the compilation.

3.5 Combine bin and rom file

In order to combine the wizweb.rom file and W5200EVB_App.bin file into one, move the two files to the folder which allbin.bat file and allbin.exe file exists (\binhex_util). Then run the allbin.bat file which is used for creating a bin file by adding wizweb.rom at the end of W5200EVB_App.bin. The merged file is created in All_XXXXXX.bin form. The current date will be shown in the part XXXXXX. Load this file in the STMicroelectronics flash loader.exe program, and wirte the merged file in STM32F103C8's flash.

3.6 System Flash Map

Assumption: W5200EVB_App.bin is 19940 byte (0x4DE4 byte)

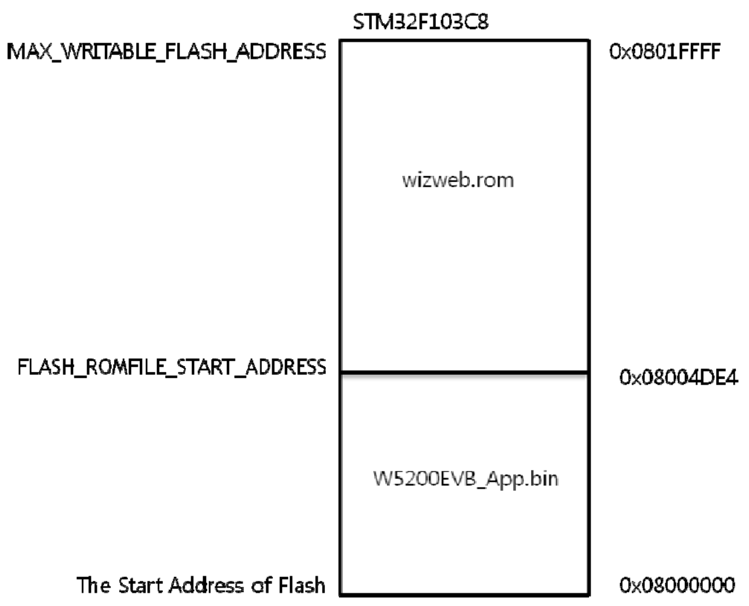
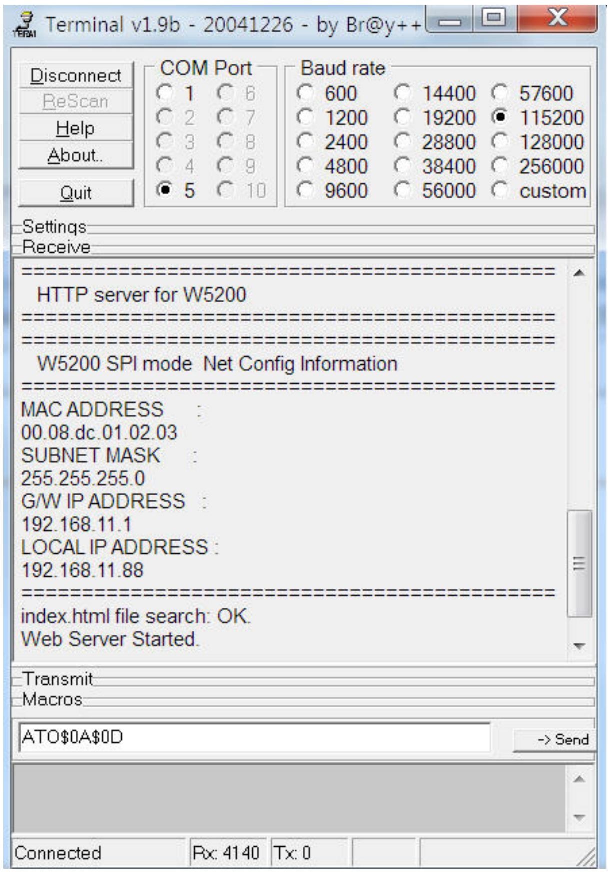

STM32F103C8
MAX_WRITABLE_FLASH_ADDRESS
0x0801FFFF
wizweb.rom
FLASH_ROMFILE_START_ADDRESS
0x08004DE4
W5200EVB_App.bin
The Start Address of Flash
0x08000000

Figure 6.

4. HTTP Server Demonstration

If the combined file for HTTP server is successfully created as mentioned in section 3.5, use STMicroelectronics flash loader.exe program to write the bin file to W5200E01-M3 board. And then, switch the SW2 from [PROG] to [RUN] and press the SW1 to reset the board. After resetting, a Hyper terminal message will appear as shown below.



COM Port: [✓ 5]
Baud rate: [✓ 115200]

```
============================================
  HTTP server for W5200
============================================
============================================
  W5200 SPI mode  Net Config Information
============================================
MAC ADDRESS      :
00.08.dc.01.02.03
SUBNET MASK      :
255.255.255.0
G/W IP ADDRESS   :
192.168.11.1
LOCAL IP ADDRESS :
192.168.11.88
============================================
index.html file search: OK.
Web Server Started.
```

ATO$0A$0D

Connected | Rx: 4140 | Tx: 0

Figure 7.

The IP address of W5200E01-M3 can be checked from the message above. Run the HTTP client program and connect with the board’s IP. After entering the IP, the web page as shown below will appear.

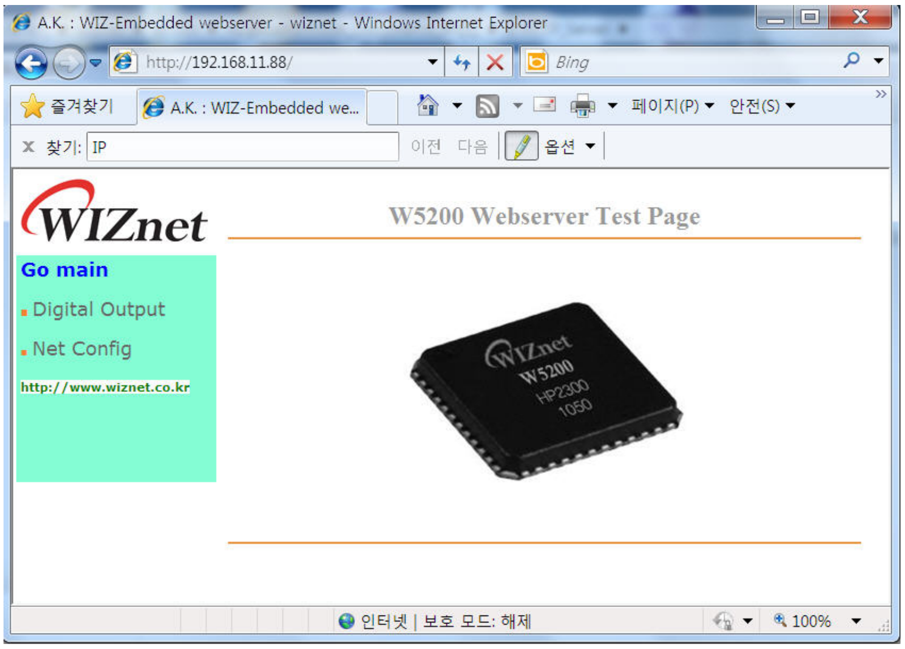

A.K. : WIZ-Embedded webserver - wiznet - Windows Internet Explorer
http://192.168.11.88/
W5200 Webserver Test Page
Go main
Digital Output
Net Config
http://www.wiznet.co.kr

Figure 8.

"Go main" menu on the left top of is for the main page of this the HTTP server. "Digital Output" menu is for monitoring and controlling the LED3 and LED4 of W5200E01-M3. "Net Config" menu is for checking and setting the network parameters of W5200E01-M3.

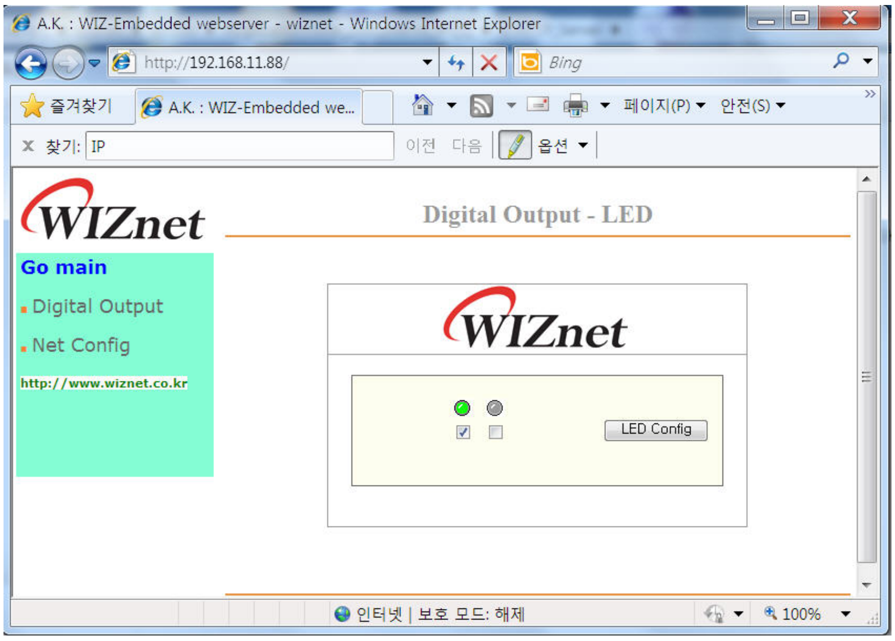
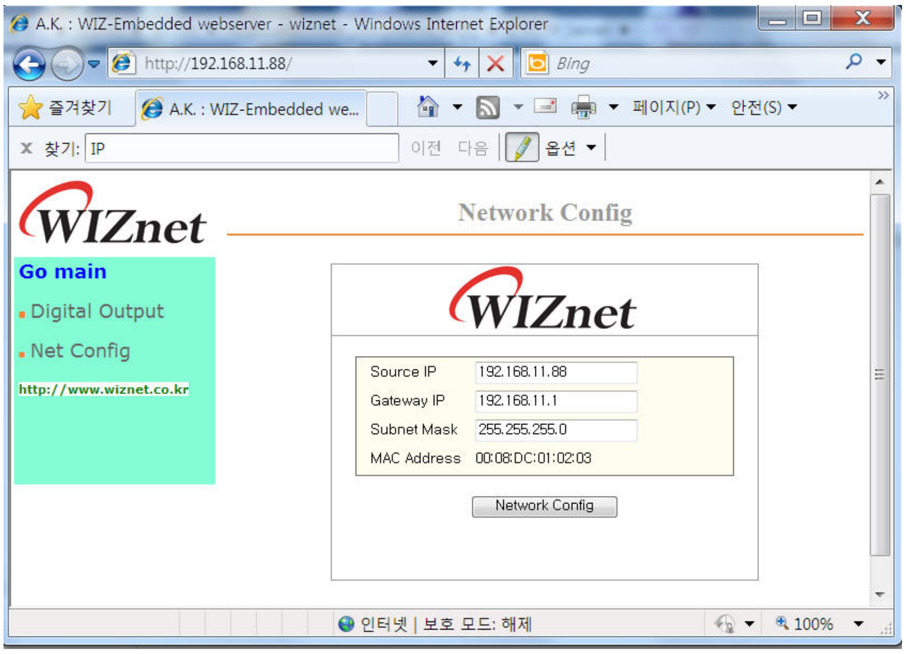


Figure 9.



Source IP 192.168.11.88
Gateway IP 192.168.11.1
Subnet Mask 255.255.255.0
MAC Address 00:08:DC:01:02:03

Figure 10.

Document History Information

Version	Date	Descriptions
Ver. 0.9βeta	Sep, 2011	