# Getting Started Guide with WIZ550web

WIZ550web is an embedded Web server module based on WIZnet's W5500 hardwired TCP/IP chip, Users can control & monitor the 16-configurable digital I/O and 4-ADC inputs on module via web pages. WIZ550web provides the firmware & web page examples for user's customization.

**This page describes the following topics:**

- **Product Preview**
- **Hello world**
  - Product contents
  - SD card initialization
  - Data flash initialization
  - Serial debug message
- **WIZ550web Basic operations and CGI**
- **Basic Demo Webpage**
- **Examples for WIZ550web customization**

**Users can download the following source codes from the 'WIZ550web GitHub Repository'**

- Firmware source code (The projects for Eclipse IDE)
  - Application / Boot
- Demo webpage

**WIZ550web GitHub Repository**

https://github.com/Wiznet/WIZ550web

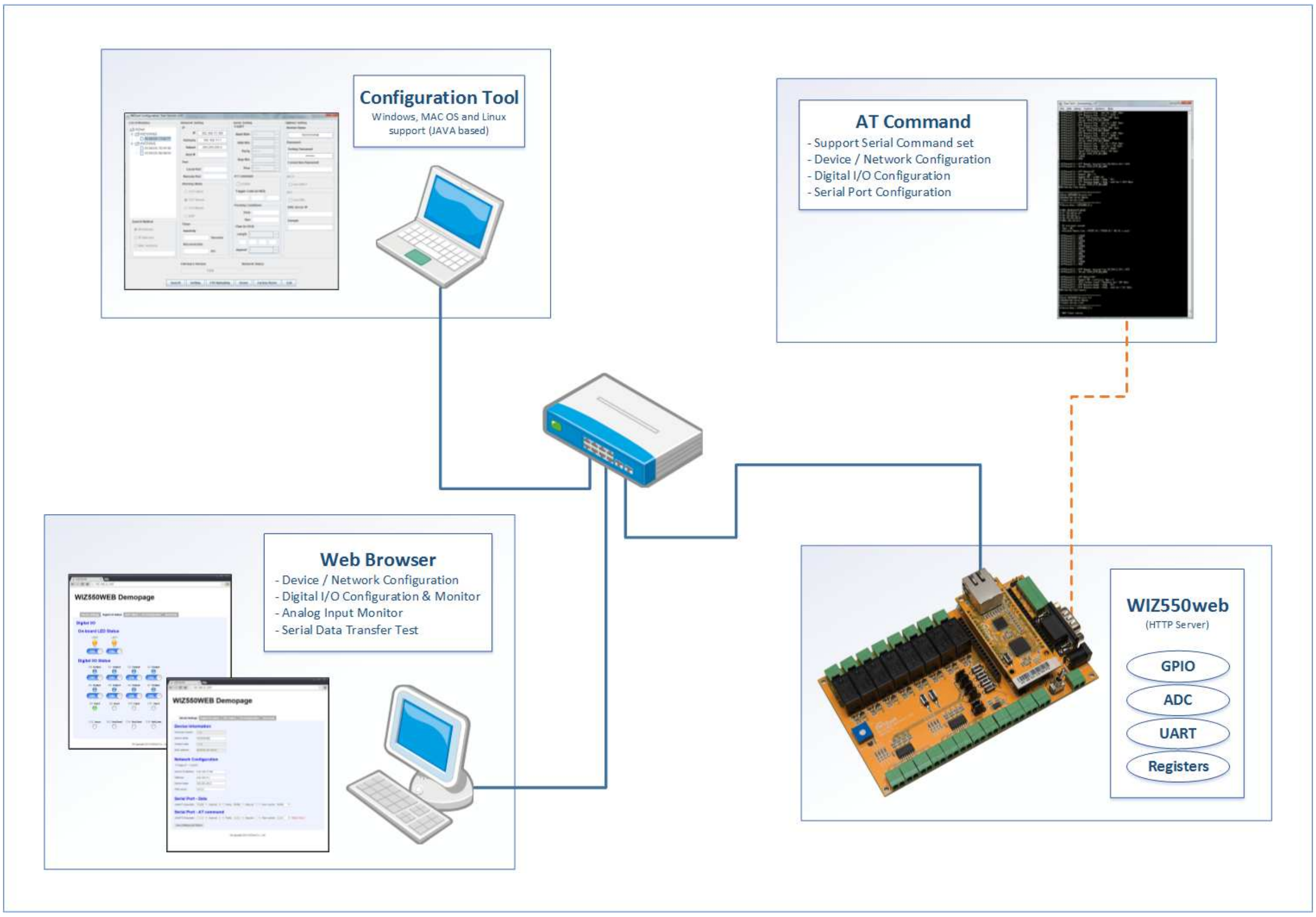
**WIZ550web GitHub Page**

http://wiznet.github.io/WIZ550web

**Develop Environment**

- Eclipse IDE for C/C++ Developers, Kepler Service Release 2
- ARM GCC 4.8.3 (2014q1)

# Product Preview


Configuration Tool
Windows, MAC OS and Linux support (JAVA based)
AT Command
- Support Serial Command set
- Device / Network Configuration
- Digital I/O Configuration
- Serial Port Configuration
Web Browser
- Device / Network Configuration
- Digital I/O Configuration & Monitor
- Analog Input Monitor
- Serial Data Transfer Test
WIZ550WEB Demopage
WIZ550WEB Demopage
WIZ550web
(HTTP Server)
GPIO
ADC
UART
Registers

# Hello World

## Product Contents

### Ordering Part No: WIZ550web

- WIZ550web module x 1

### Ordering Part No: WIZ550web-EVB

- WIZ550web module x 1
- WIZ550web baseboard x 1

- LAN cable x 1
- Serial cable x 1
- 12V Power adapter x 1

**SD card is option for both WIZ550web and WIZ550web-EVB**

Refer to recommended lists of SD card.

| Vendor | Capacity(Bytes) | Type | Class |
|---|---|---|---|
| Sandisk | 2G | SD | n/a |
| | 4G | SDHC | 4 |
| | 8G | SDHC | 4 |
| Samsung | 4G | SDHC | 6 |
| Transcend | 4G | SDHC | 4,10 |

## SD card Initialization

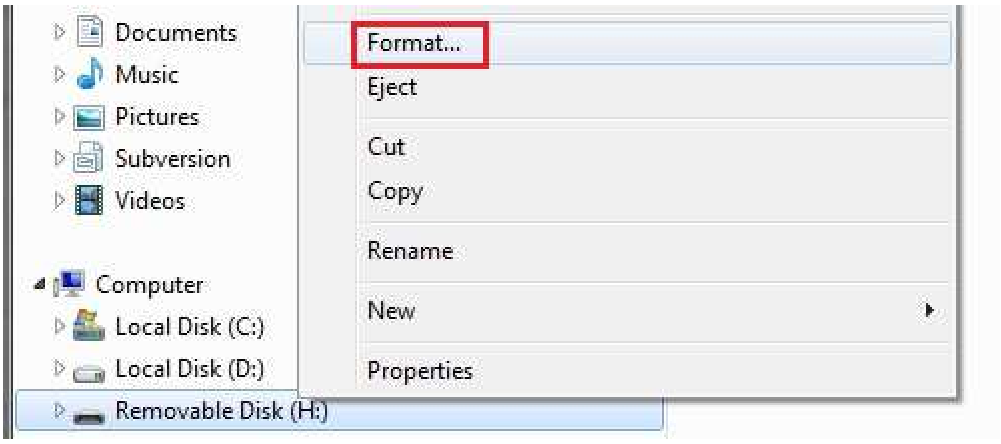
WIZ550web uses Micro SD card as a storage for web content and SD card is not included as default. SD card supports **FAT**[1] file system format and we recommend one use FAT32.

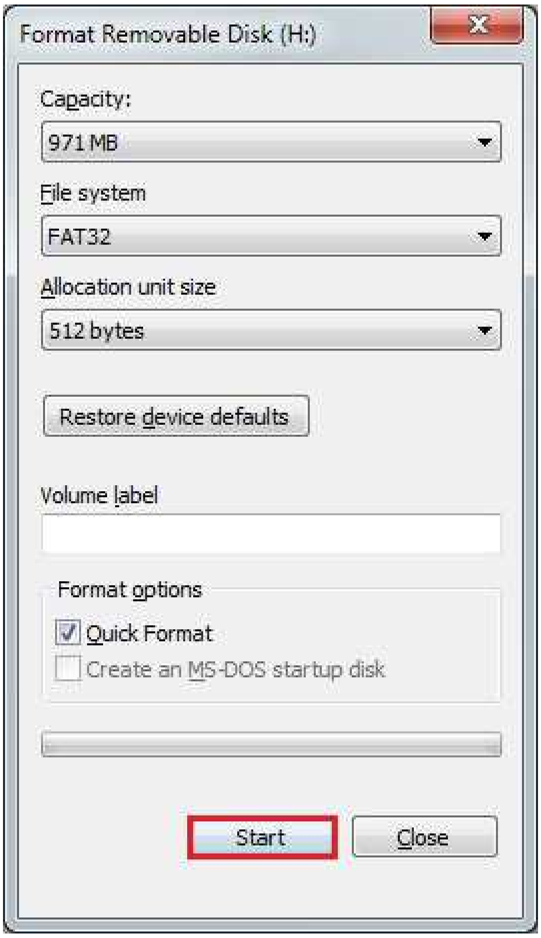
The process of using SD card as a storage for web content of WIZ550web is like the below.

### » The process of using SD card (for Windows)

1. Formatting a ready Micro SD card as FAT32 file system

Insert Micro SD card into SD card reader and Do format it on Windows explorer after insertion to PC





Format Removable Disk (H:)

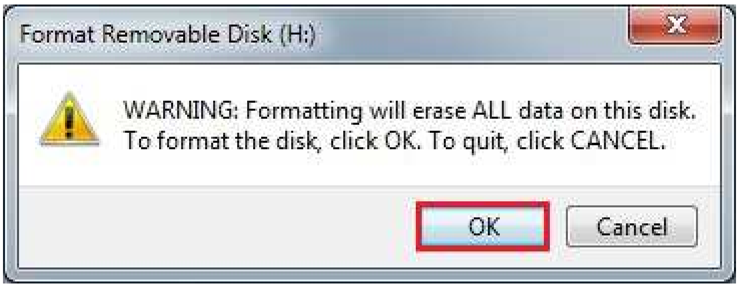
Capacity:
971 MB

File system
FAT32

Allocation unit size
512 bytes

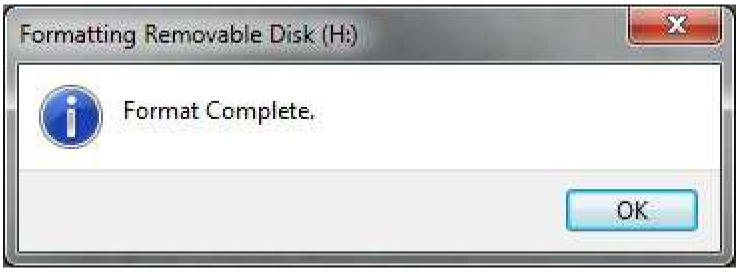
Volume label

Format options
☑ Quick Format
□ Create an MS-DOS startup disk



Format Removable Disk (H:)

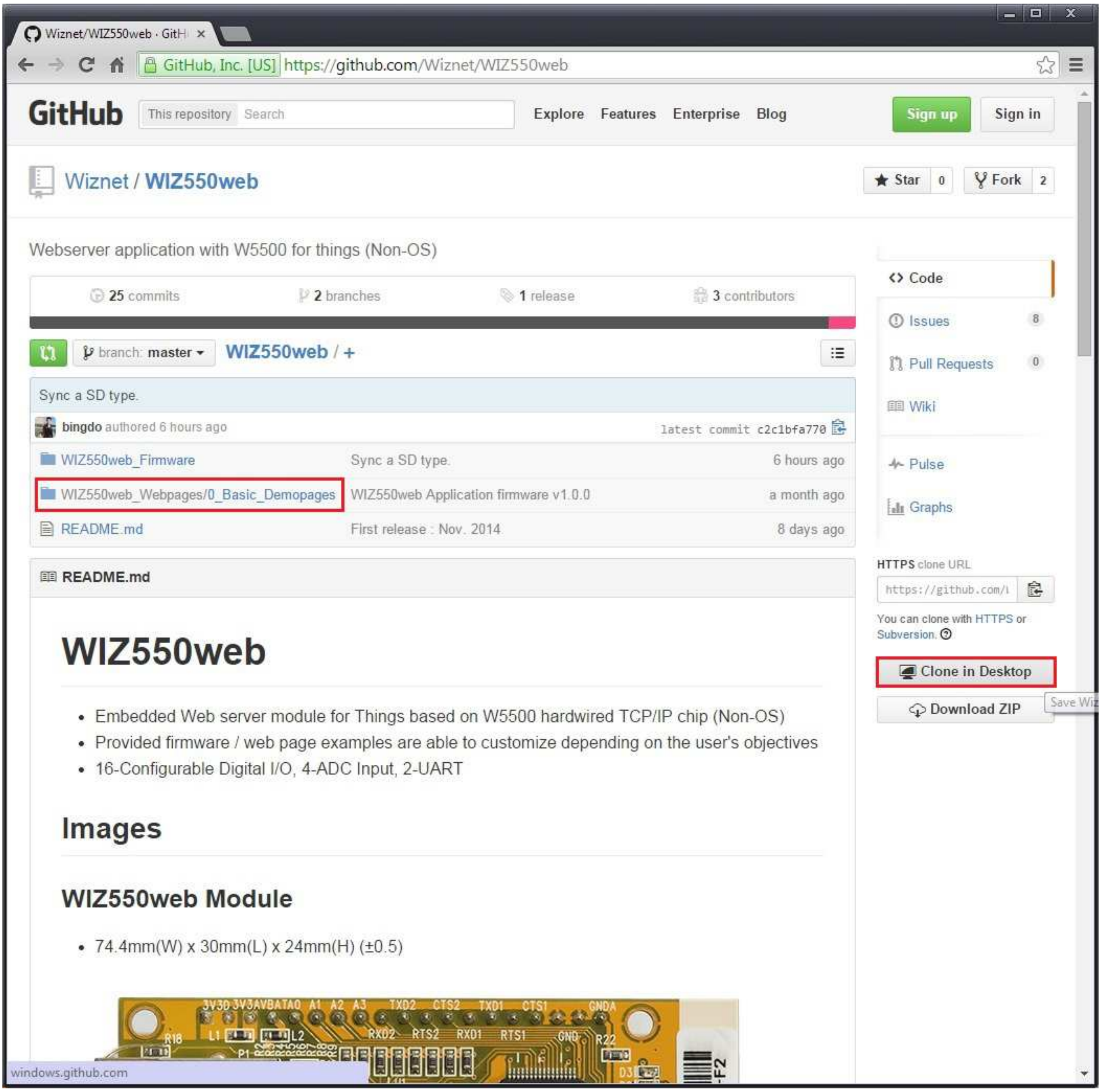
WARNING: Formatting will erase ALL data on this disk.
To format the disk, click OK. To quit, click CANCEL.



Formatting Removable Disk (H:)

Format Complete.

2. Downloading demo web page from WIZ550web GitHub repository



Webserver application with W5500 for things (Non-OS)

Sync a SD type.

bingdo authored 6 hours ago latest commit c2c1bfa770

| | | |
|---|---|---|
| WIZ550web_Firmware | Sync a SD type. | 6 hours ago |
| WIZ550web_Webpages/0_Basic_Demopages | WIZ550web Application firmware v1.0.0 | a month ago |
| README.md | First release : Nov. 2014 | 8 days ago |

You can clone with HTTPS or Subversion.

README.md

# WIZ550web

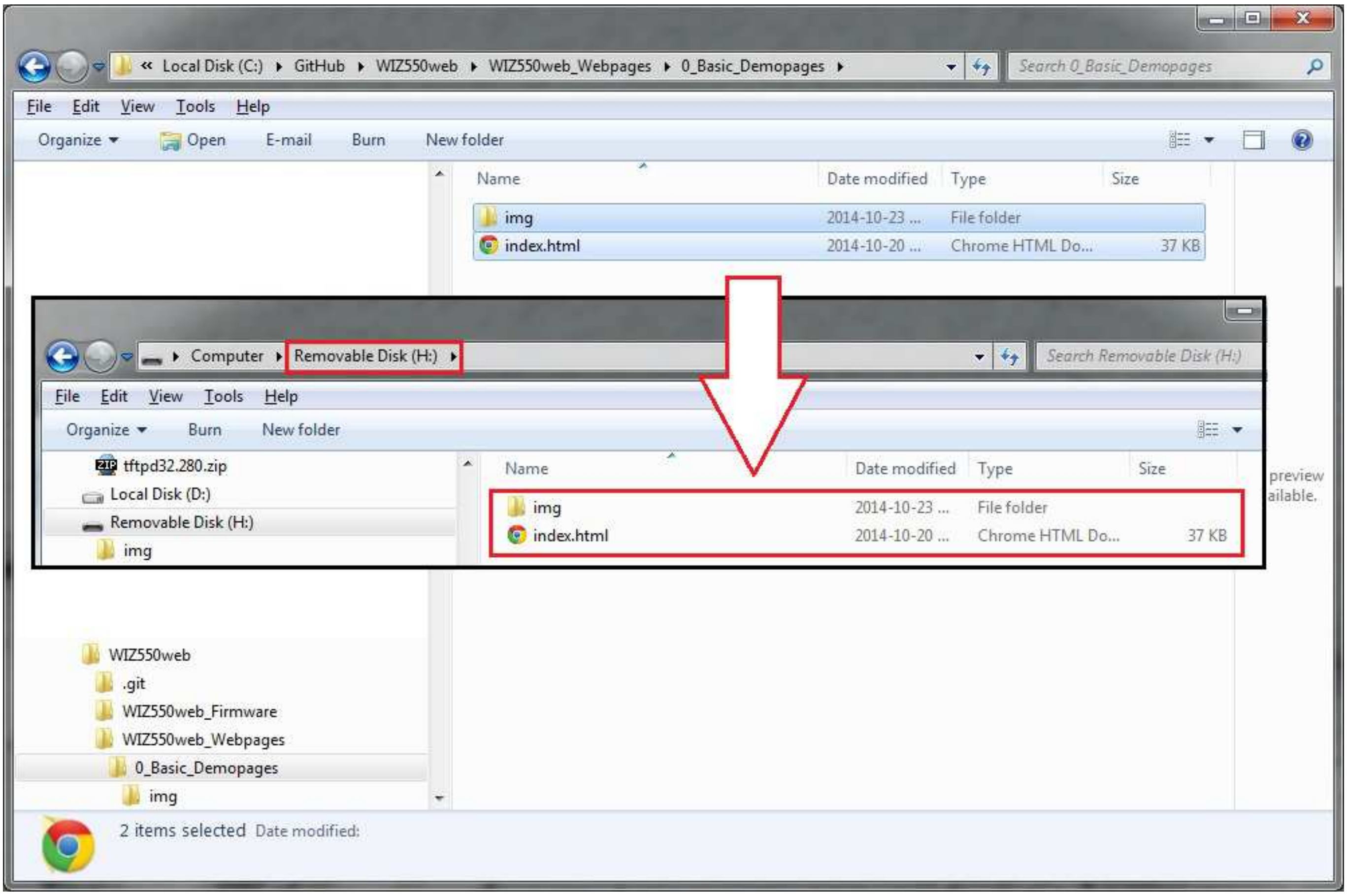
- Embedded Web server module for Things based on W5500 hardwired TCP/IP chip (Non-OS)
- Provided firmware / web page examples are able to customize depending on the user's objectives
- 16-Configurable Digital I/O, 4-ADC Input, 2-UART

## Images

### WIZ550web Module

- 74.4mm(W) x 30mm(L) x 24mm(H) (±0.5)

3. Copying demo Web page to Micro SD card


« Local Disk (C:) ▸ GitHub ▸ WIZ550web ▸ WIZ550web_Webpages ▸ 0_Basic_Demopages ▸
Search 0_Basic_Demopages
File Edit View Tools Help
Organize ▾ Open E-mail Burn New folder
WIZ550web
.git
WIZ550web_Firmware
WIZ550web_Webpages
0_Basic_Demopages
img
2 items selected Date modified:
Computer ▸ Removable Disk (H:) ▸
Search Removable Disk (H:)
File Edit View Tools Help
Organize ▾ Burn New folder
tftpd32.280.zip
Local Disk (D:)
Removable Disk (H:)
img
preview
ailable.

| Name | Date modified | Type | Size |
|---|---|---|---|
| img | 2014-10-23 ... | File folder | |
| index.html | 2014-10-20 ... | Chrome HTML Do... | 37 KB |

4. Insertion Micro SD card into SD card slot in WIZ550web

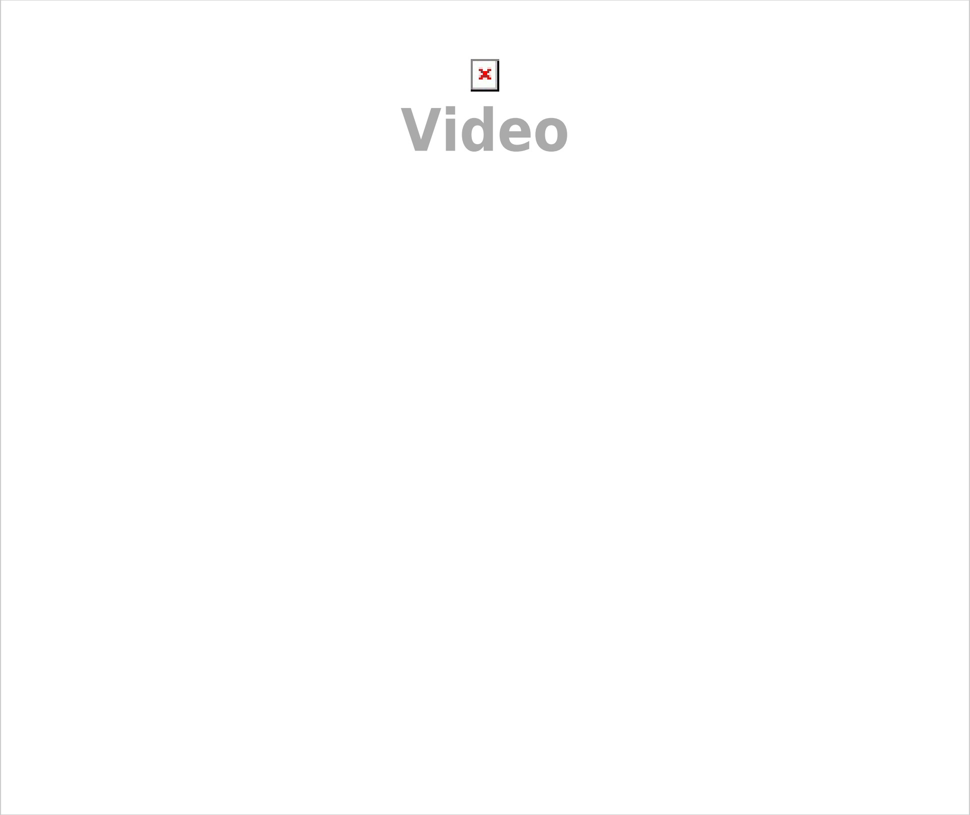
5. Done!

## Data flash Initialization

**We support a storage of data flash as well as SD card above v1.1.0 release.** WIZ550web firmware bin v1.1.0 Develop (zip) https://github.com/Wiznet/WIZ550web/releases

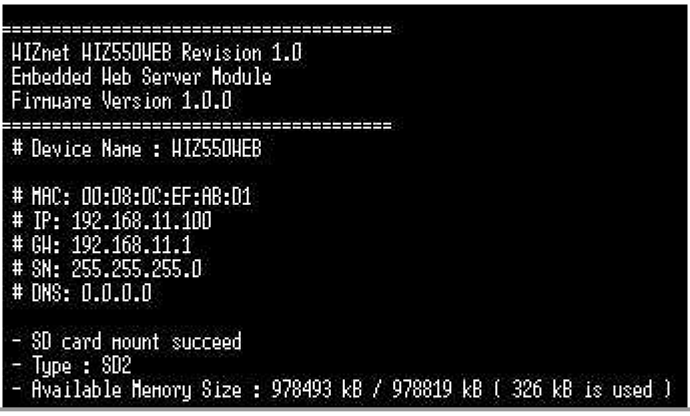
1. You can use one of a SD card and a data flash. The mount priority of a SD card is higher than a data flash.
2. If you wish to use a SD card, you must copy the web page to a SD card and insert a SD card into a slot.
3. If there is no SD card after detecting during about 3 seconds, you can use a data flash.
    1. You must have the initialization process of data flash at least once.
    2. When SW1 and SW2 are pressed at the same time, the data flash is initialized by FatFs. And reset a target.
    3. You must copy the web page to a data flash by FTP client tool.(ALFTP)

Refer to WIZ550web+FatFS+FTPServer Project Tutorial. (http://youtu.be/XtnT2_CNgaY)



## Serial Debug Message

When power is applied to the WIZ550web module, the serial debug message is print out via UART1 port. This message contains information such as the device name, network configurations, and SD card info.



```
======================================
WIZnet WIZ550WEB Revision 1.0
Embedded Web Server Module
Firmware Version 1.0.0
======================================
# Device Name : WIZ550WEB

# MAC: 00:08:DC:EF:AB:D1
# IP: 192.168.11.100
# GW: 192.168.11.1
# SN: 255.255.255.0
# DNS: 0.0.0.0

- SD card mount succeed
- Type : SD2
- Available Memory Size : 978493 kB / 978819 kB ( 326 kB is used )
```

The module will not work during the initialization routine of the WIZ550web firmware if either the **PHY link status** or **SD Card mount** is not operational.

**Please check the following if the module is not working.**

1. **LAN cable connection and Link status (Act LED of the LAN port)**
2. **SD card mount status (insertion)**

The factory default setting of UART1 is as below and the user can change settings through the demo web pages & AT commands.

- **115200-8-N-1**
  - Baud Rate : 115200
  - Data Bit : 8
  - Parity : NONE
  - Stop Bit : 1
  - Flow Control : NONE

# WIZ550web Web Server

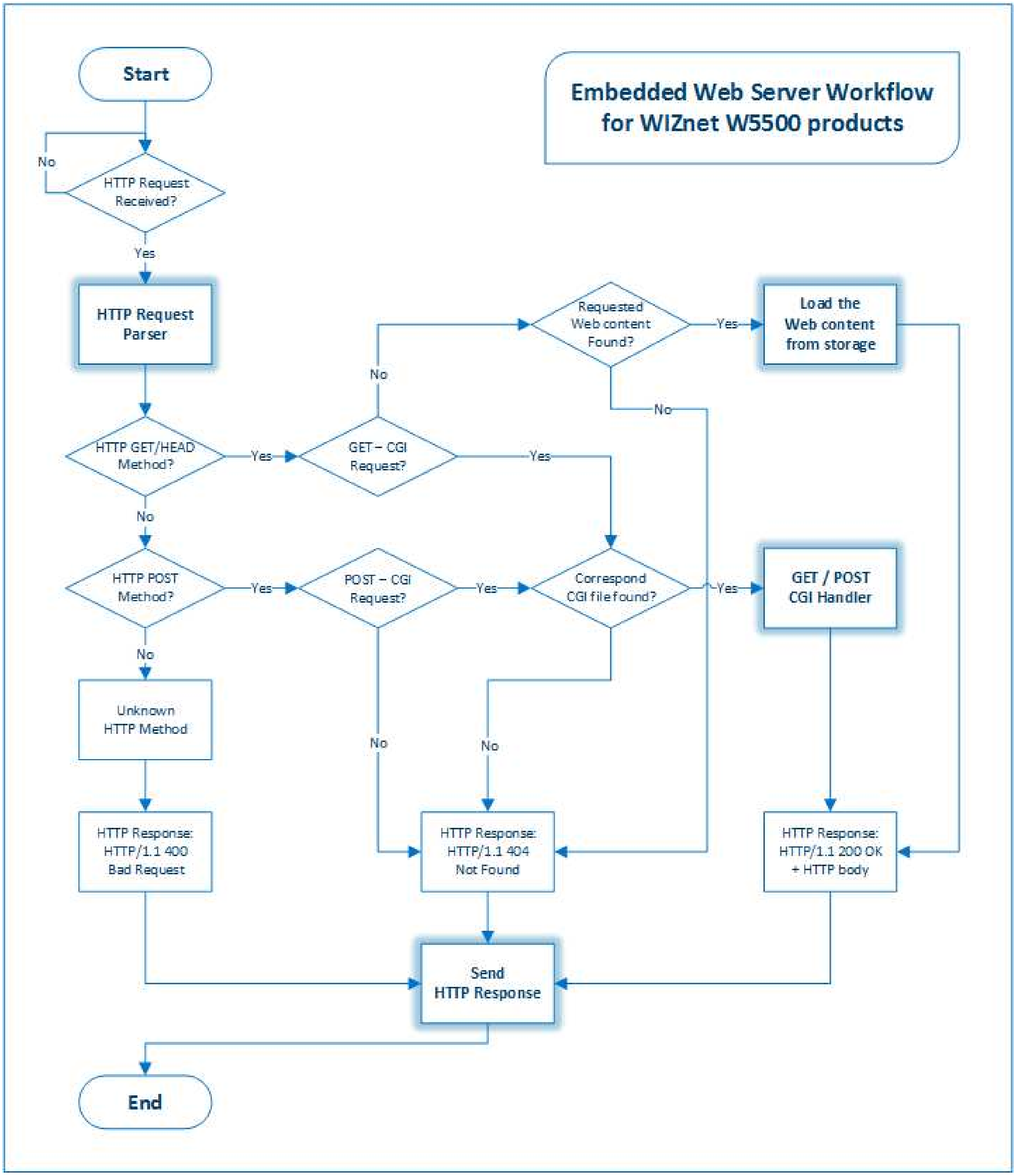
## Web Server Basic Operations

Common web servers and browsers communicate as follows.

1. Web browser (user's Webpage) sends HTTP request to web server
2. Web server processes the user's request and prepare the results
3. Web server sends HTTP response to web browser
4. Repeat step 1 ~ 3 above

Therefore, Web server (HTTP server) of WIZ550web is consists the following processes and operates as the flowchart below.

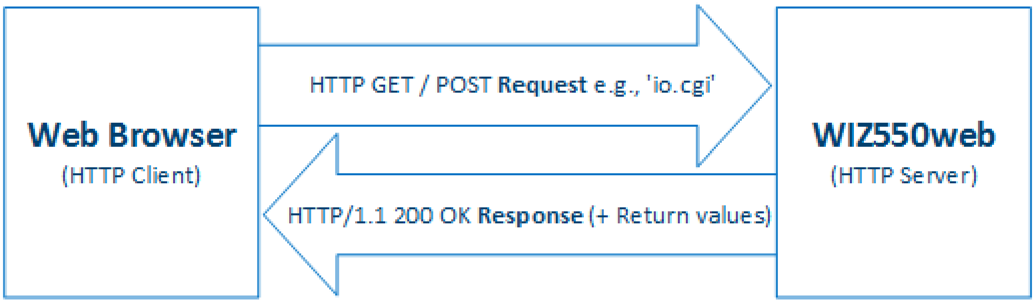
- **HTTP Request Parser**
  - Parse the HTTP Request
- **Web Content (web resources) Loader**
  - Find the requested web content from the web content storage (e.g., SD card) and prepare the response
- **HTTP Response Sender**
  - Send the HTTP response to web browser
  - Response vary depending on the processing results
    - Successful HTTP requests : HTTP/1.1 200 OK
    - The requested resource could not be found : HTTP/1.1 404 Not Found
- **CGI Handler**
  - Process the **User's requests** from web browser
  - CGI is the used to set the I/O status or to get the I/O status information


Embedded Web Server Workflow for WIZnet W5500 products
Start
HTTP Request Received?
No
Yes
HTTP Request Parser
HTTP GET/HEAD Method?
Yes
GET – CGI Request?
No
Requested Web content Found?
Yes
Load the Web content from storage
No
Yes
No
HTTP POST Method?
Yes
POST – CGI Request?
Yes
Correspond CGI file found?
Yes
GET / POST CGI Handler
No
Unknown HTTP Method
No
No
HTTP Response: HTTP/1.1 400 Bad Request
HTTP Response: HTTP/1.1 404 Not Found
HTTP Response: HTTP/1.1 200 OK + HTTP body
Send HTTP Response
End

## CGI for WIZ550web

WIZ550web uses CGI to configure the network or monitor and control the I/O.

CGI is an acronym for 'Common Gateway Interface’ and it operates the server's program and receives results of the client via web. CGI is not dependent on a particular language or platform because only the configuration and policies between the web page and the server needs to be satisfied.


Web Browser
(HTTP Client)
HTTP GET / POST Request e.g., 'io.cgi'
HTTP/1.1 200 OK Response (+ Return values)
WIZ550web
(HTTP Server)

CGI for WIZ550web consists the 'Request name + .cgi' using HTTP GET/POST request method. The CGI for each HTTP methods are handled as below.

**GET**

- The method for getting the values from web server
- Passed in the form of a **JavaScript callback function parameters**
  - Same structures as **JSON**[2)]
  - The function name in the HTTP response body must be the same on the Web page's JavaScript Callback function name
  - e.g., If the 'function IoStatusCallback' is Javascript function name in the Web page, web server must pass the data as next; `IoStatusCallback({“din”:[{“v”:“1”}, {“v”:“0”}], “led”:[{“v”:“1”}, {“v”:“0”}]})`

**POST**

- The method for putting the changed values to web server
- Values are passing by the **Web form element**
- **Key-value pairs**; Each pair is separated by '&' and the Key/value of a pair is represented by '='
  - e.g., 'Pin : 1, Val : 1' ⇒ 'Pin=1&Val=1'

Each requests are handled by **AJAX**[3)] techniques for efficient data processing in the limited resources embedded web server. The advantage is that the server avoids the full web page reload and reduces the load on the server.

# Basic Demo Web Pages

The Screenshots of Basic Demo Web Page were captured from **Google Chrome Browser version 37**

Provided basic demo web page of WIZ550web is in tab style of one HTML page (index.html). The configuration and purpose of each tabs are as below.

- **Device Settings**: Device information, Network / Serial Port configurations
  - Device Information
  - Network Configuration

  - Serial Port (for Data & Debug / AT commands)

- **Digital I/O Status**: Indicates Digital I/O's Direction and Status
  - On-board LED Status
  - Digital I/O Direction / Status: When the I/O is output direction, users can be set On/Off

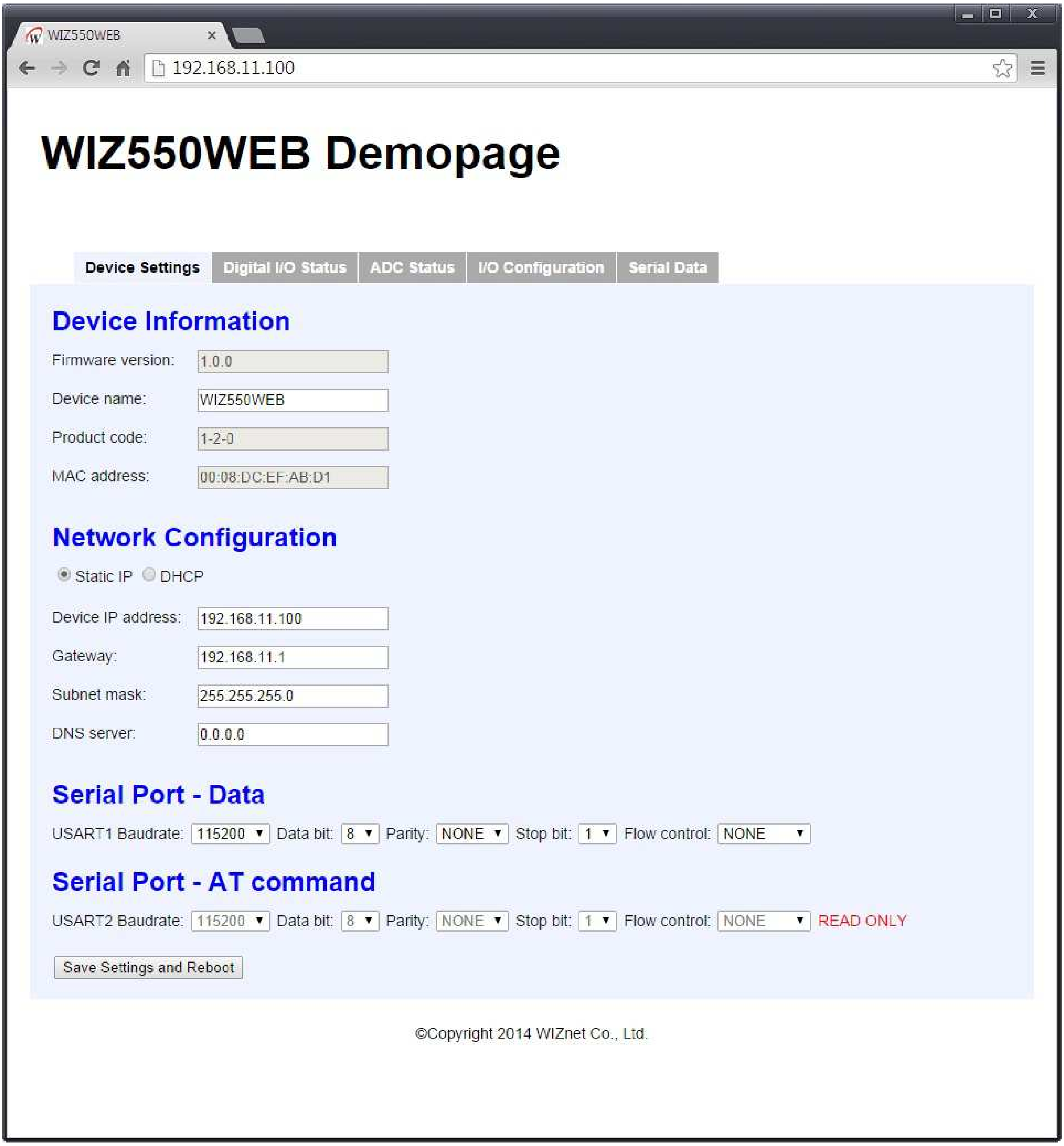
- **ADC Status**: Displays the Analog input value changes into a bar graphs
  - Analog Input Signals

- **I/O Configuration**: Digital I/O's Direction and Alias settings
  - Digital I/O State and Alias Configuration
  - Digital I/O Settings Factory Default

- **Serial Data**: Serial data transfer examples via Webpage to UART1.
  - Serial Tx (Web to UART1) / Rx (UART1 to Web)

## Device Settings

The 'Device Settings' tab is configured as below.



WIZ550WEB Demopage

Device Information

Firmware version: 1.0.0

Device name: WIZ550WEB

Product code: 1-2-0

MAC address: 00:08:DC:EF:AB:D1

Network Configuration

● Static IP ○ DHCP

Device IP address: 192.168.11.100

Gateway: 192.168.11.1

Subnet mask: 255.255.255.0

DNS server: 0.0.0.0

Serial Port - Data

USART1 Baudrate: 115200 Data bit: 8 Parity: NONE Stop bit: 1 Flow control: NONE

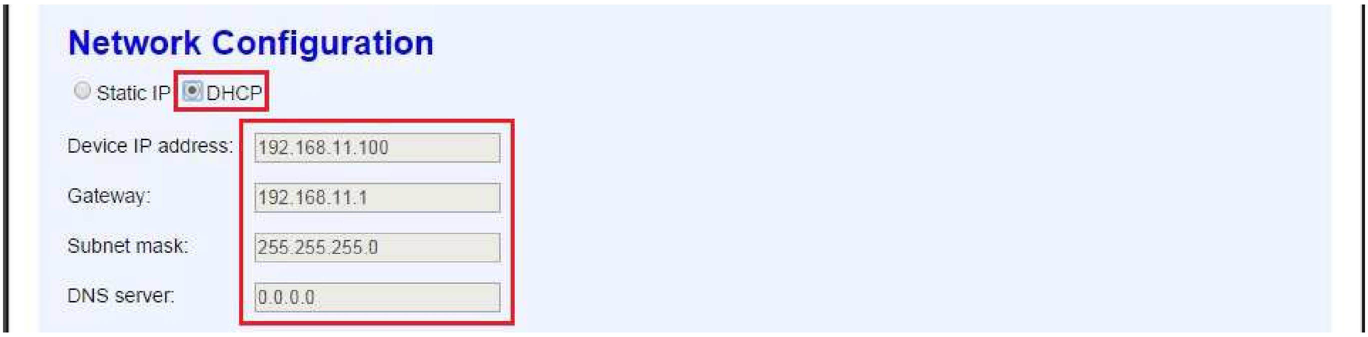
Serial Port - AT command

USART2 Baudrate: 115200 Data bit: 8 Parity: NONE Stop bit: 1 Flow control: NONE READ ONLY

©Copyright 2014 WIZnet Co., Ltd.

The user can check the device information like MAC address / Firmware version and set the network information (IP address, DHCP options and etc.), serial port configuration through the 'Device Settings' tab. After adjusting the settings, if the **[Save Settings and Reboot]** button is pressed, the configurations is saved and the WIZ550web module reboots. Please note that the module is rebooted after the settings.

If the 'DHCP' option enabled (DHCP Enable: IP address allocated from DHCP server), Network information HTML input tag is disabled and attribute changed to disabled. If the user successfully allocated the IP address from DHCP server after module reboot, the modified debug message is output to terminal as below.



Network Configuration

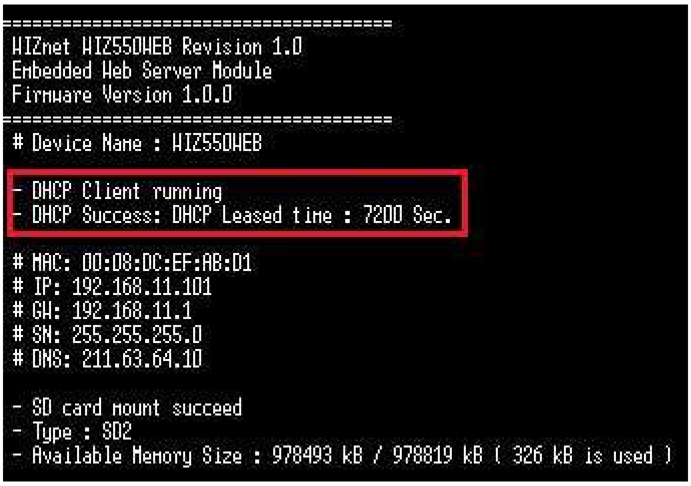
○ Static IP ● DHCP

Device IP address: 192.168.11.100

Gateway: 192.168.11.1

Subnet mask: 255.255.255.0

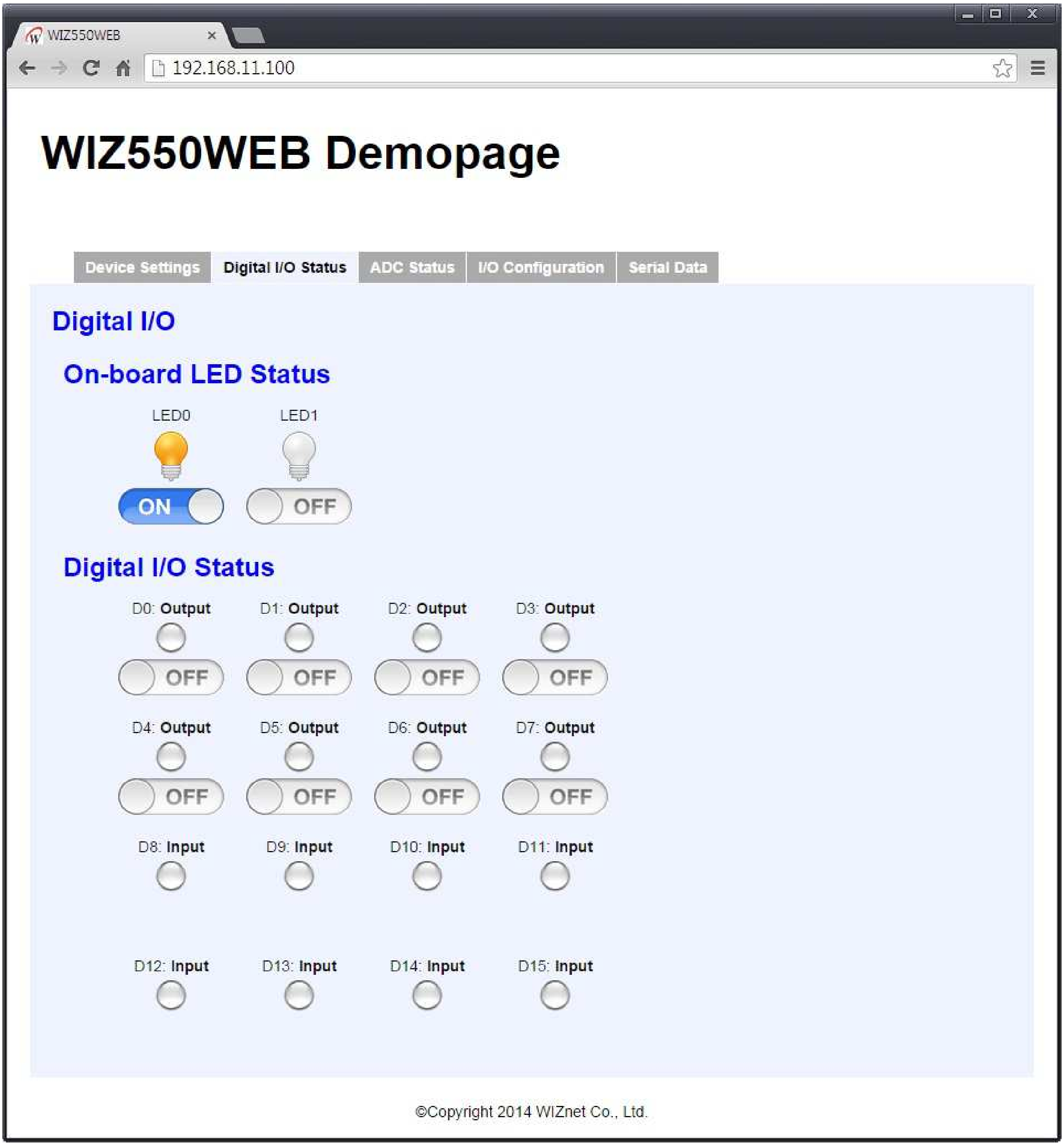
DNS server: 0.0.0.0



```
======================================
WIZnet WIZ550WEB Revision 1.0
Embedded Web Server Module
Firmware Version 1.0.0
======================================
# Device Name : WIZ550WEB

- DHCP Client running
- DHCP Success: DHCP Leased time : 7200 Sec.

# MAC: 00:08:DC:EF:AB:D1
# IP: 192.168.11.101
# GW: 192.168.11.1
# SN: 255.255.255.0
# DNS: 211.63.64.10

- SD card mount succeed
- Type : SD2
- Available Memory Size : 978493 kB / 978819 kB ( 326 kB is used )
```

## Digital I/O Status

The 'Digital I/O Status' tab is configured as below.



WIZ550WEB Demopage

Digital I/O

On-board LED Status

LED0: ON
LED1: OFF

Digital I/O Status

D0: Output OFF
D1: Output OFF
D2: Output OFF
D3: Output OFF
D4: Output OFF
D5: Output OFF
D6: Output OFF
D7: Output OFF
D8: Input
D9: Input
D10: Input
D11: Input
D12: Input
D13: Input
D14: Input
D15: Input

©Copyright 2014 WIZnet Co., Ltd.

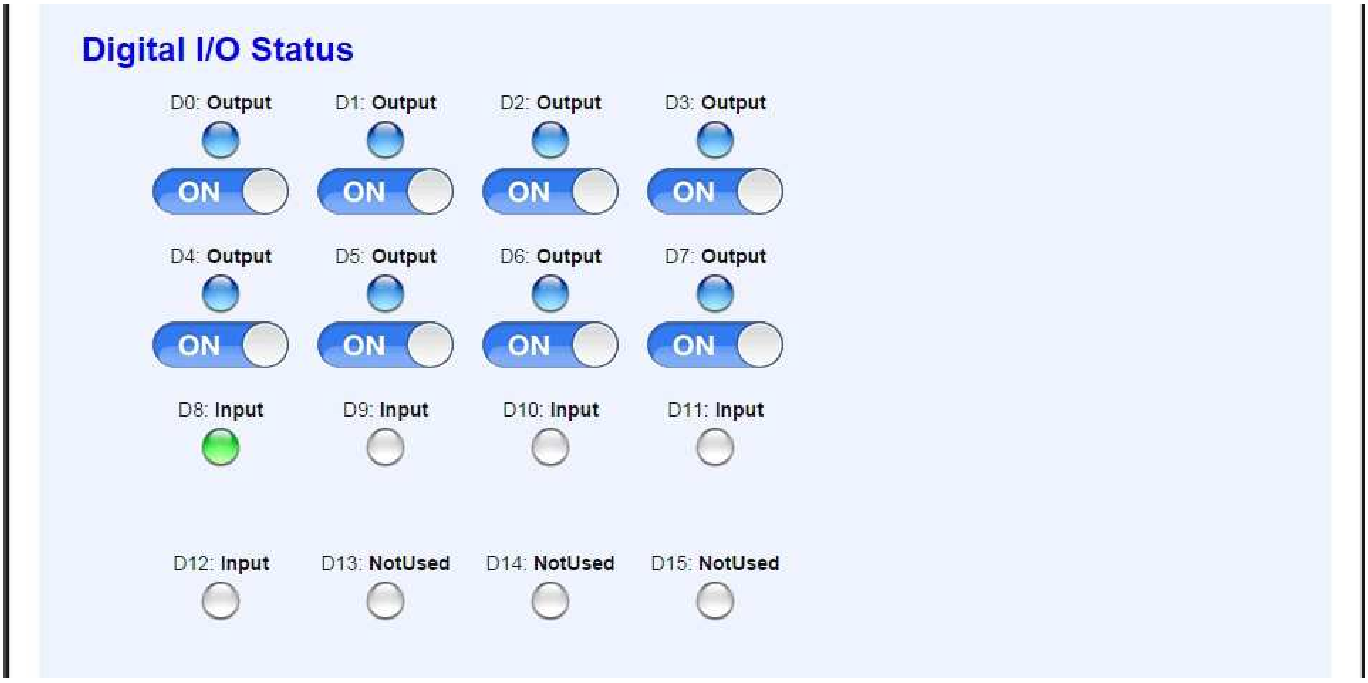
[On-board LED] WIZ550web's LEDs can be control by On/Off button in web page.
[Digital I/O Status] Digital I/O Status indicates the current status. When the I/O is output, users can set On/Off. Each I/O information are configured as below.

- **Pin number** (e.g., D0)**: Direction** (Input / Output / NotUsed)
- **Status LED Images** (🔵 Blue LED / 🟢 Green LED / ○ Off)
- **On/Off Buttons** (for Output directions)

For 'Input',
- On/Off button is not indicated. If the input signal is 'high', the 'green LED' is turned on.
For 'Output',
- On/Off button is indicated. If the output signal is 'high', the 'blue LED' is turned on.

For 'NotUsed',
- On/Off button is not indicated. turned off the LED (off LED).

Users can set each I/O directions in 'I/O configuration' tab. See below for digital I/O status setting examples.



Digital I/O Status

D0: Output D1: Output D2: Output D3: Output

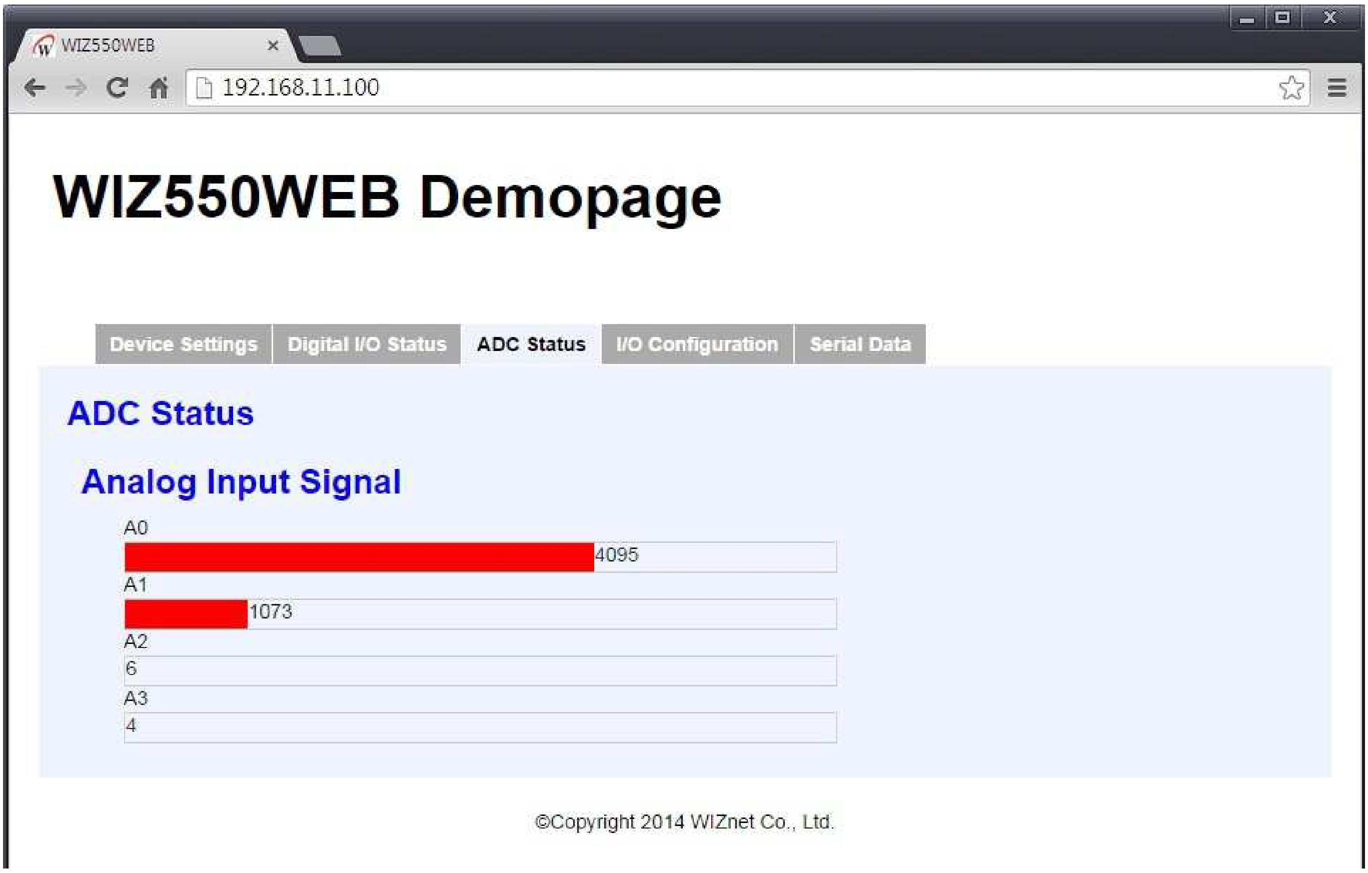
D4: Output D5: Output D6: Output D7: Output

D8: Input D9: Input D10: Input D11: Input

D12: Input D13: NotUsed D14: NotUsed D15: NotUsed

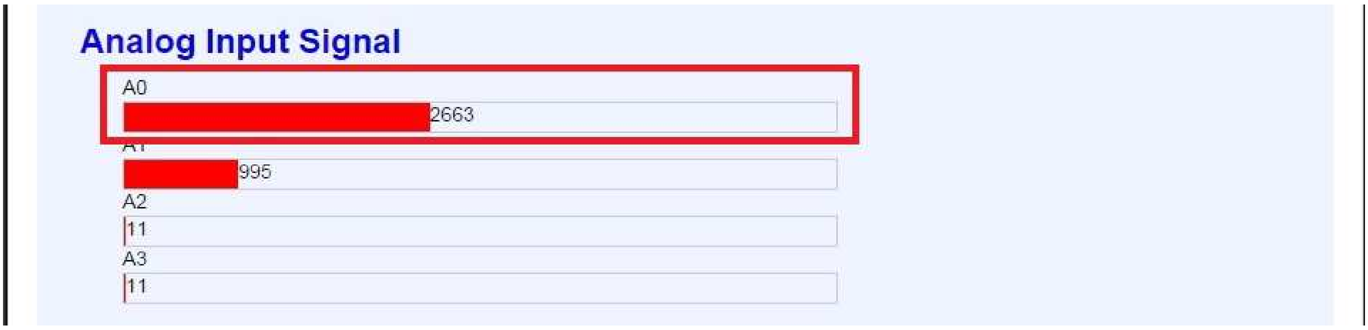
## ADC Status

The 'ADC Status' tab is configured as below.



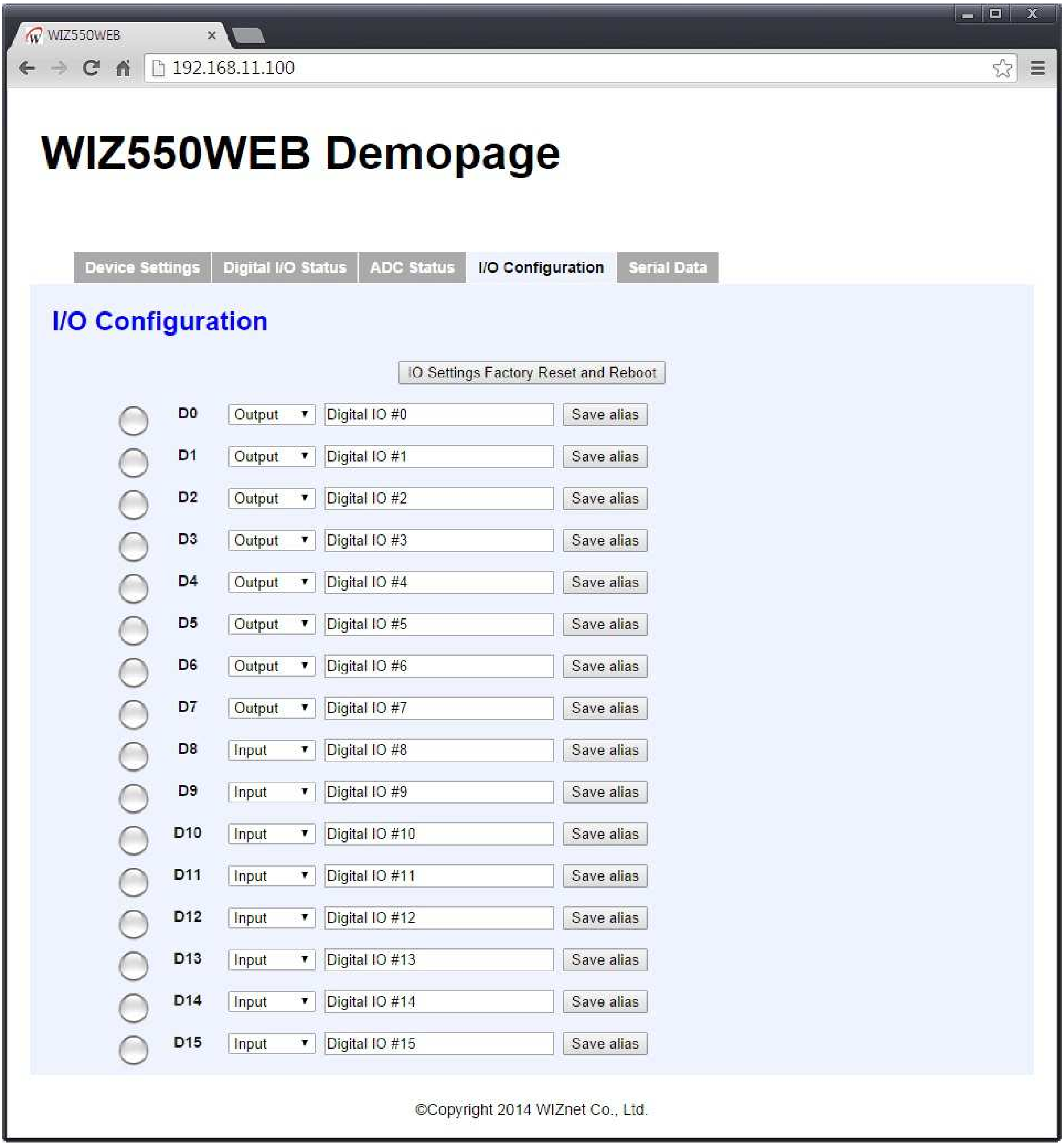
The Analog input signal through ADC changes into a bar graphs is displayed in real-time.

WIZ550web baseboard is equipped with Potentiometer and Temperature sensor on A0 / A1. For example, by adjusting the value of the Potentiometer (A0), users can see the changes in the graph as below.



## I/O Configuration

The 'I/O Configuration' tab is configured as below.



WIZ550WEB Demopage

I/O Configuration

IO Settings Factory Reset and Reboot

| | | | | |
|---|---|---|---|---|
| ○ | D0 | Output | Digital IO #0 | Save alias |
| ○ | D1 | Output | Digital IO #1 | Save alias |
| ○ | D2 | Output | Digital IO #2 | Save alias |
| ○ | D3 | Output | Digital IO #3 | Save alias |
| ○ | D4 | Output | Digital IO #4 | Save alias |
| ○ | D5 | Output | Digital IO #5 | Save alias |
| ○ | D6 | Output | Digital IO #6 | Save alias |
| ○ | D7 | Output | Digital IO #7 | Save alias |
| ○ | D8 | Input | Digital IO #8 | Save alias |
| ○ | D9 | Input | Digital IO #9 | Save alias |
| ○ | D10 | Input | Digital IO #10 | Save alias |
| ○ | D11 | Input | Digital IO #11 | Save alias |
| ○ | D12 | Input | Digital IO #12 | Save alias |
| ○ | D13 | Input | Digital IO #13 | Save alias |
| ○ | D14 | Input | Digital IO #14 | Save alias |
| ○ | D15 | Input | Digital IO #15 | Save alias |

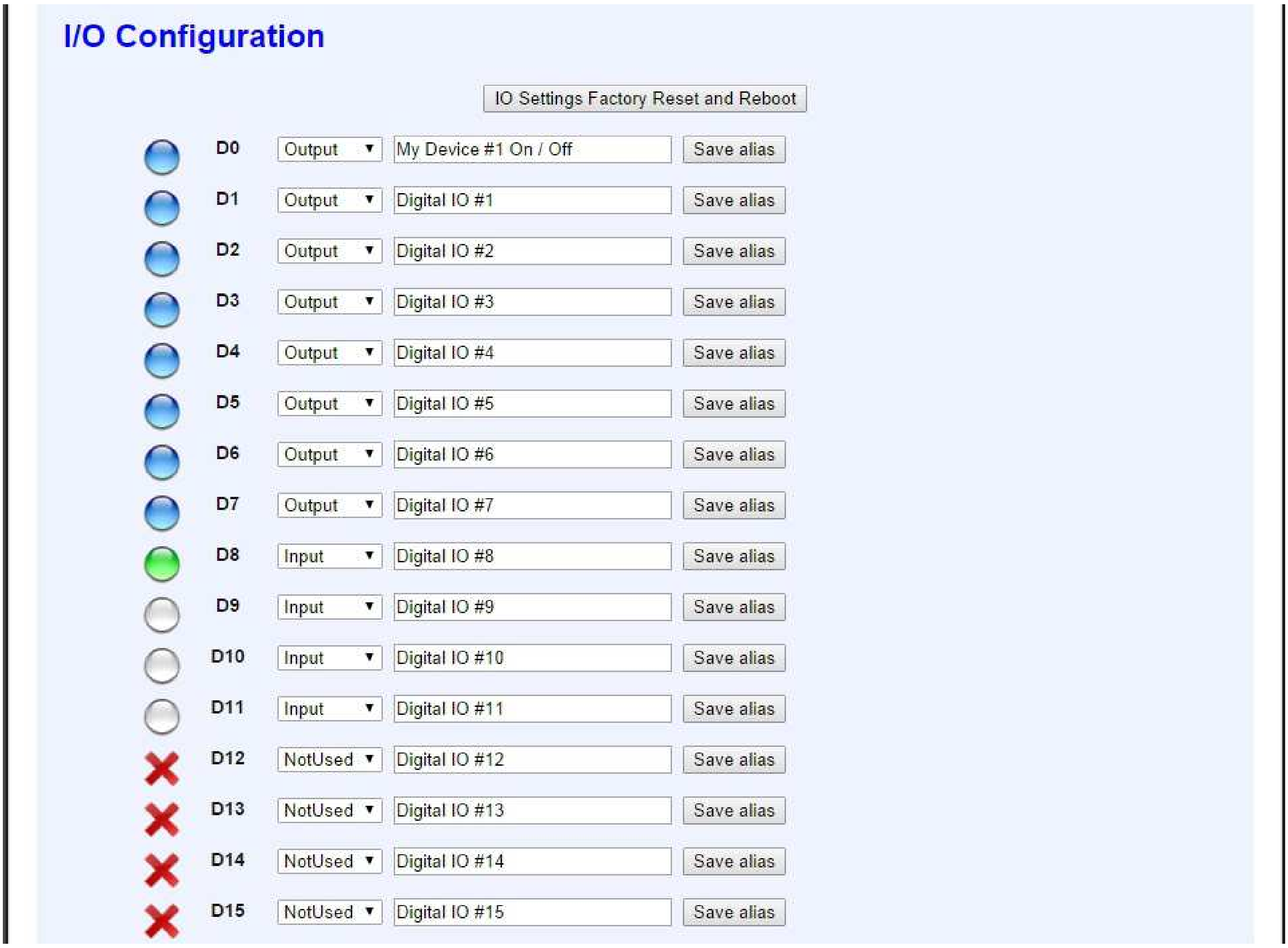
©Copyright 2014 WIZnet Co., Ltd.

Change of the I/O Directions and Aliases setting will be reflected immediately. Each I/O's LED images are the same as I/O status page (Blue / Green / Off).

Only when the 'Not Used' option is selected, the red 'X' is displayed for visibility.

When you press the **[IO Settings Factory Reset and Reboot]** button, the WIZ550web module factory reset is executed, and is rebooted. Please note that the module is rebooted with the setting default value.

If you change the various settings, it is displayed as below.



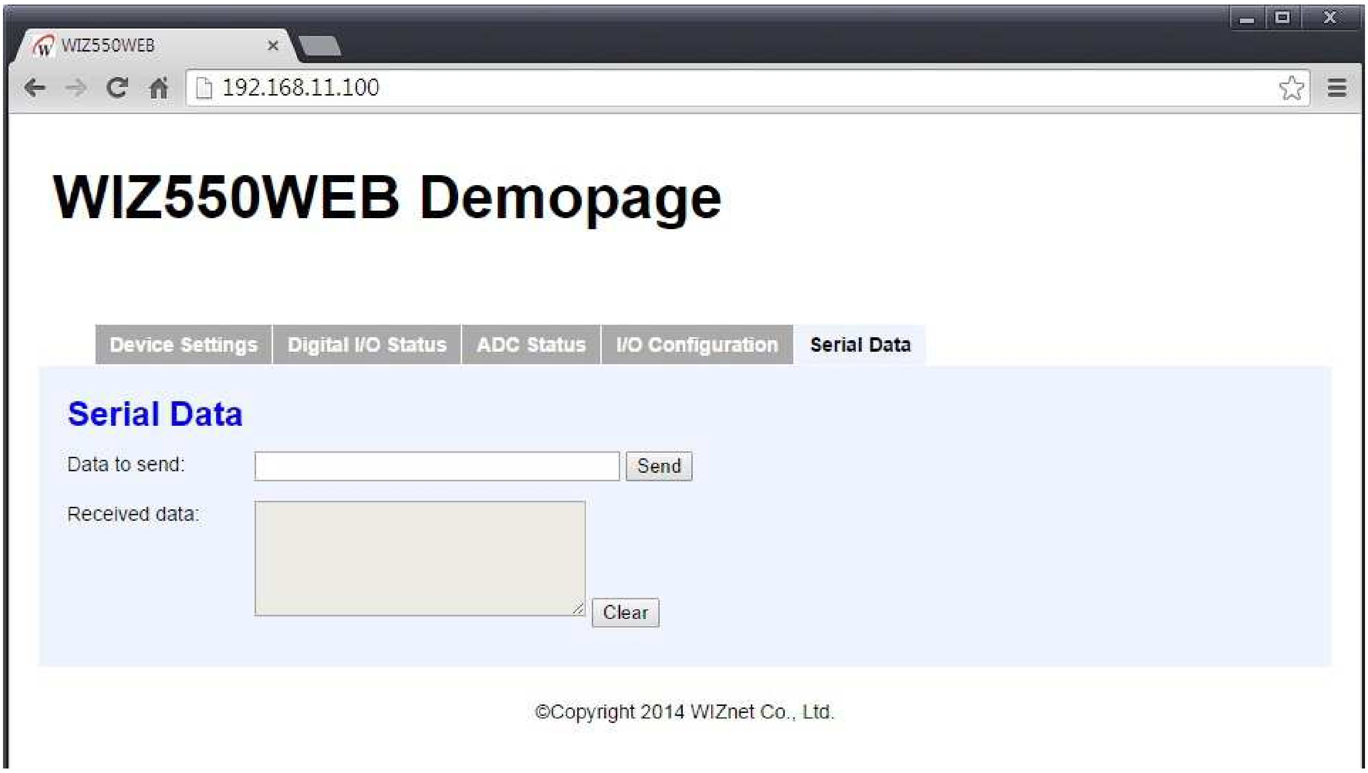
I/O Configuration

IO Settings Factory Reset and Reboot

| | | | | |
|---|---|---|---|---|
| 🔵 | D0 | Output ▼ | My Device #1 On / Off | Save alias |
| 🔵 | D1 | Output ▼ | Digital IO #1 | Save alias |
| 🔵 | D2 | Output ▼ | Digital IO #2 | Save alias |
| 🔵 | D3 | Output ▼ | Digital IO #3 | Save alias |
| 🔵 | D4 | Output ▼ | Digital IO #4 | Save alias |
| 🔵 | D5 | Output ▼ | Digital IO #5 | Save alias |
| 🔵 | D6 | Output ▼ | Digital IO #6 | Save alias |
| 🔵 | D7 | Output ▼ | Digital IO #7 | Save alias |
| 🟢 | D8 | Input ▼ | Digital IO #8 | Save alias |
| ○ | D9 | Input ▼ | Digital IO #9 | Save alias |
| ○ | D10 | Input ▼ | Digital IO #10 | Save alias |
| ○ | D11 | Input ▼ | Digital IO #11 | Save alias |
| ❌ | D12 | NotUsed ▼ | Digital IO #12 | Save alias |
| ❌ | D13 | NotUsed ▼ | Digital IO #13 | Save alias |
| ❌ | D14 | NotUsed ▼ | Digital IO #14 | Save alias |
| ❌ | D15 | NotUsed ▼ | Digital IO #15 | Save alias |

## Serial Data

The 'Serial Data' tab is configured as below.



WIZ550WEB Demopage

Serial Data

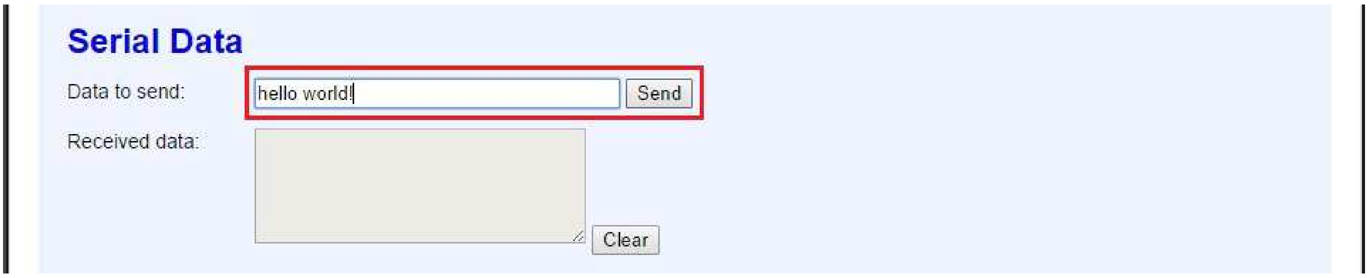
Data to send: Send

Received data: Clear

©Copyright 2014 WIZnet Co., Ltd.

In this page, user can test the 'Serial data input/output through Webpage to UART1' (Serial data Tx/Rx) functions.

Type 'hello world!' in the **'Data to send'** input box and then press the **[Send]** button.

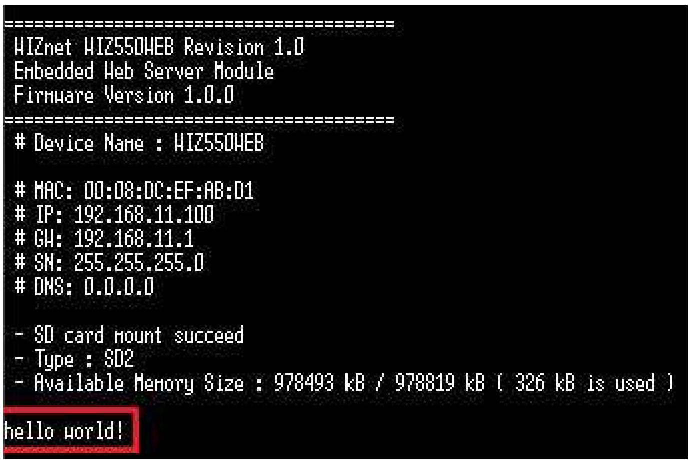


Serial Data

Data to send: hello world!

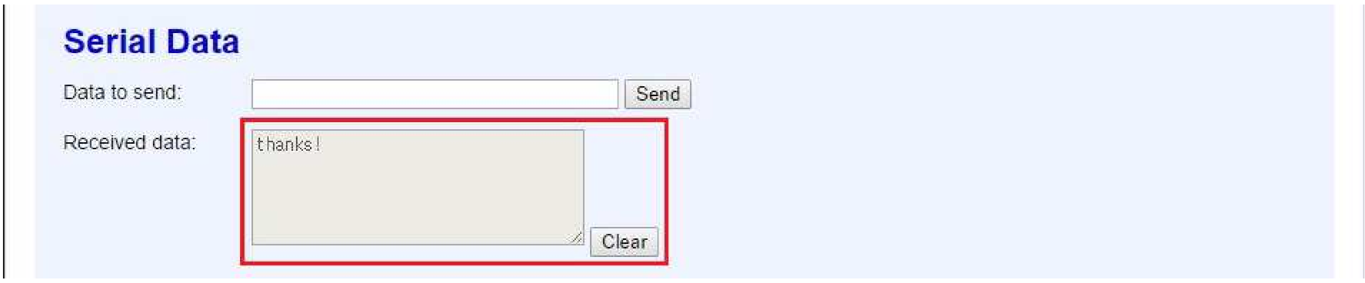
Received data:

Serial terminal shows the string from web page through UART1.



```
=======================================
 WIZnet WIZ550WEB Revision 1.0
 Embedded Web Server Module
 Firmware Version 1.0.0
=======================================
 # Device Name : WIZ550WEB

 # MAC: 00:08:DC:EF:AB:D1
 # IP: 192.168.11.100
 # GW: 192.168.11.1
 # SN: 255.255.255.0
 # DNS: 0.0.0.0

 - SD card mount succeed
 - Type : SD2
 - Available Memory Size : 978493 kB / 978819 kB ( 326 kB is used )

hello world!
```

On the contrary, if the user types the string on serial terminal, the **'Received data'** box in web page shows the typed string.



Serial Data

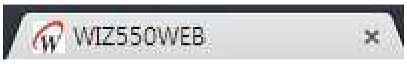
Data to send:

Received data: thanks!

## Note

### Favicon

Favicon is a complex word of 'favorites' and 'icon', and is displayed as a small icon in the address bar / tabs of web browser (e.g., Google Chrome, MS Internet Explorer). By using this, the user is able to know the provider of the web page. The example basic demo web pages include 'favicon'.



User can add the 'Favicon' as the following steps

1. Create the target image to 16×16 sized icon and save in the name of 'favicon.ico'
2. Move the 'favicon.ico' to web content storage
3. Insert the link code at the top of the 'index.html' page

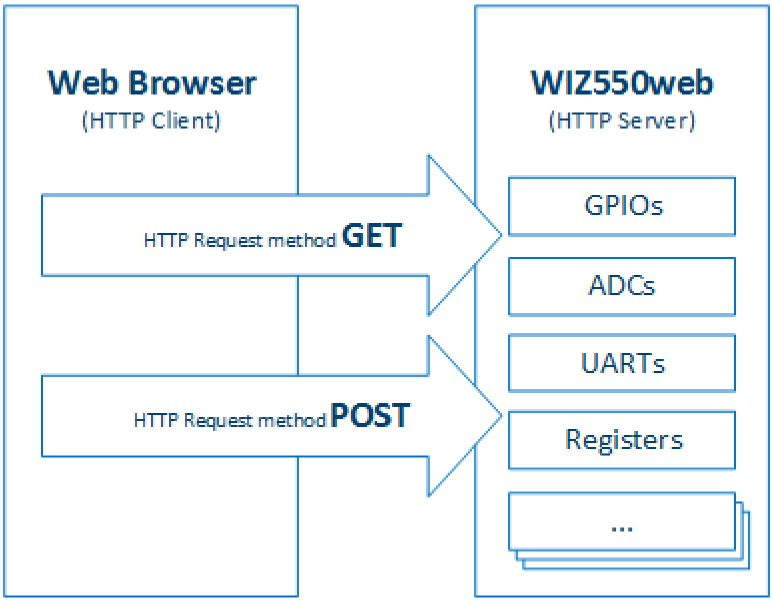
e.g., `<link rel='shortcut icon' href='/favicon.ico' type='image/x-icon'>`

# WIZ550web Customizing

## Basics

WIZ550web provides basis for user customized web server. Users can implement an embedded web server which can control the I/O by modifying the firmware source code and web pages.

Function of the Web server can be a variety of combinations depending on the user's application; I/O monitor / control, transfer the string to UART, MCU and W5500 chip register control, Etc.



Therefore, users need to modify the firmware source code (C language) and web pages (HTML and JavaScript).

In this page, we provide a description on how to get the I/O's status (value) of WIZ550web module and set the new value through examples of the basic demo web page.

Users can learn how to control the H/W (I/O, UART, Registers and Etc.) using the web page. If users need to add additional functions, use the same method described on this page.

## WIZ550web Web I/O Examples

This Section is **UNDER CONSTRUCTION**.
Check back soon. — *Jim 2015/01/20 11:25*

[1] **FAT: File Allocation Table**
File system format which is used in majority of memory cards and computers in popular. There are FAT12/16/32. FAT32 in WIZ550web supports up to 4G Byte for one file.

[2] **JSON: JavaScript Object Notation**

[3] **AJAX: Asynchronous JavaScript And XML**
AJAX is a group of interrelated Web development techniques used on the client-side to create asynchronous Web applications. With Ajax, Web applications can send data to and retrieve from a server asynchronously (in the background) without interfering with the display and behavior of the existing page.JavaScript and the XMLHttpRequest object provide a method for exchanging data asynchronously between browser and server to avoid full page reloads.

From:
http://wizwiki.net/wiki/ -

Permanent link:
**http://wizwiki.net/wiki/doku.php?id=products:wiz550web:wiz550webgsg_en**

Last update: **2015/02/09 13:05**