Table of Contents

WIZ550WEB Users' Guide

Overview

WIZ550web provides the lightweight webserver operating. It controls digital output or monitors digital and analogue input through web browser. Example web pages is stored in micro SD memory card.

Features

- HTTP Server + Demo pages to control all I/O port
- Guarantee system stability and reliability by using the H/W TCP/IP chip W5500
- Support extra “Configuration Tool Program” through network.
- Support serial configuration like AT commands.
- 10/100 Mbps Ethernet Interface
- RoHS Compliant

2014/10/28 09:25 · bongbong

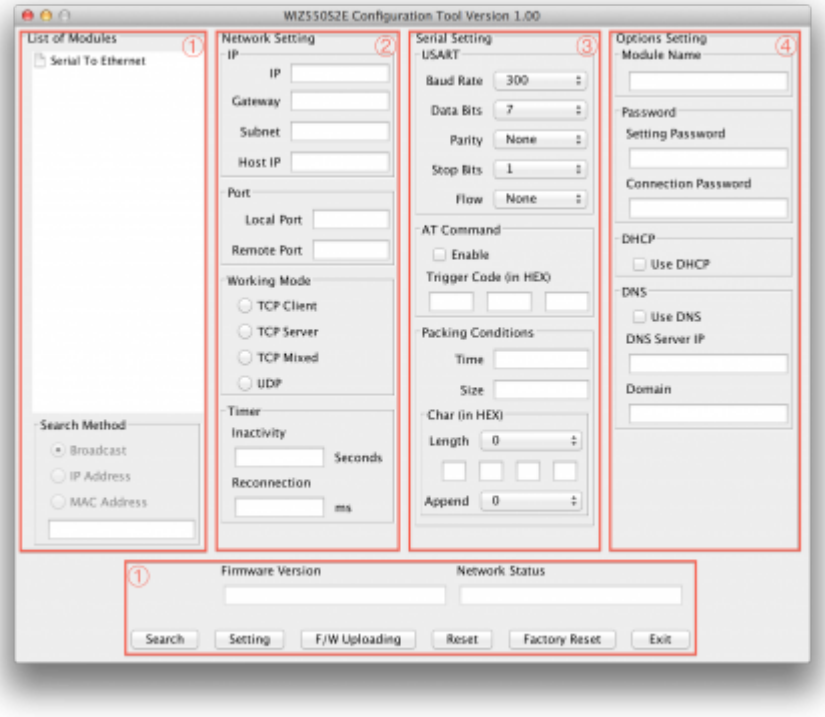
TFTP Guide Document

TFTP Guide Document

2016/11/29 16:01 · Jin Hee Ahn

Configuration Tool

Description



WIZnet Configuration tool is an application program which is based on java and can be used in most OS platforms including Windows, MAC OS and Linux. Please download .jar file and execute it over Java Virtual Machine.

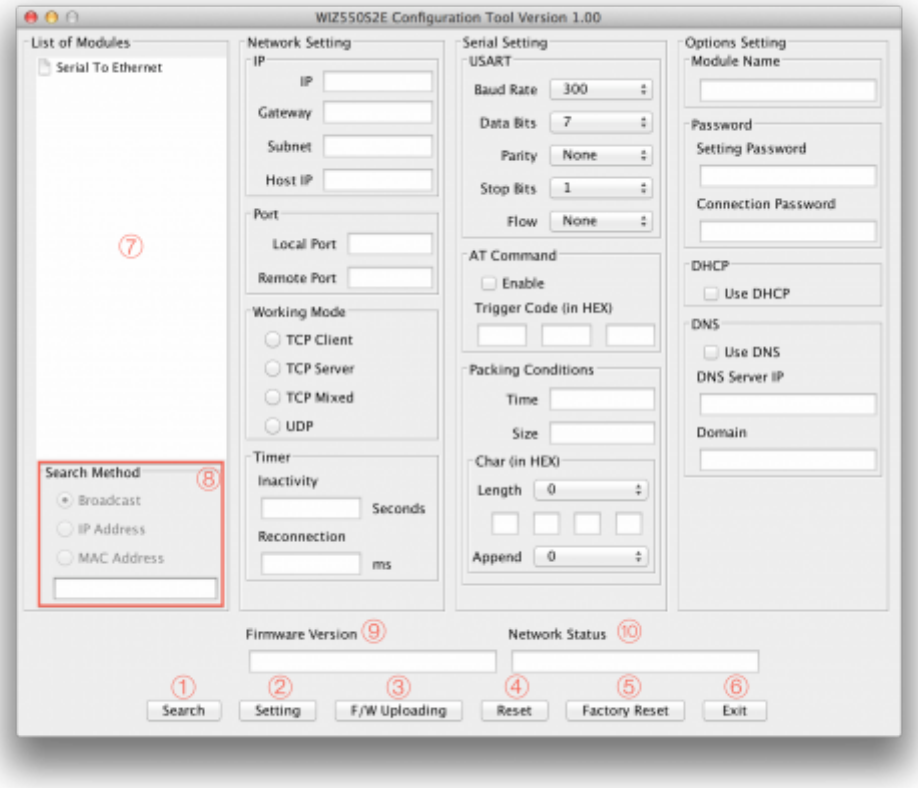
WIZnet Configuration tool consists four sections

1. Common Configuration Section
2. Network Configuration Section
3. Serial Configuration Section
4. Option Configuration Section

You can set the Common Configuration, “IP” of Network configuration with “Module Name” and “Setting Password” of Option configuration section for WIZ550WEB.

Notice) You can set WIZ550WEB module through WIZnet Configuration Tool version 1.02 or the latest version.

Common Configurations

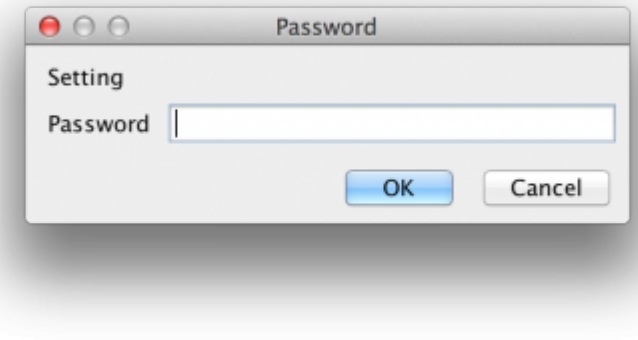


Search

The Search function is used to search for all existing modules on the same LAN. By using UDP broadcast, it finds all modules on the same subnet, and found devices will be listed in the “Serial to Ethernet” tree(Search Window) with its MAC address.

Setting

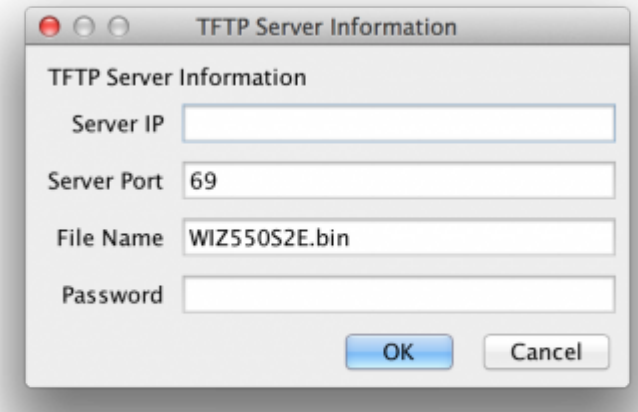
This function is used to apply your configurations.
When you select the MAC address from the “Search Window”, the default value of the module will be displayed. Modify your configurations and click “Setting” button to apply your settings. The module will re-initialize and save the changed configurations.
Users can change the configurations by following the steps below.




```
1. Select the MAC address of the device which you would like to modify in the
   “Search Window”
2. Modify the settings according to your needs
3. Click the “Setting” button and then "Password Input Windows" pop up
  - Default Password is "WIZnet"
4. Input "Setting Password" and Click "OK" button
5. The module will be initialized by a re-booting process
6. To verify your settings, please click ‘Search’ button and view your new
   settings
```

F/W Uploading

Firmware will be uploaded through TFTP. Click “F/W Uploading” Button and a popup window will shows as follow.




```
Server IP : TFTP Server IP
Server Port : TFTP Server Port (TFTP default Port : 69)
File Name : Firmware File Name
Password : Setting Password
```

☞ WIZnet Configure tool does not supported TFTP server. So please use TFTP program separately.

Reset

This is the function which makes Module reboot. This requires password to reboot.

Factory Reset

All setting value is initialized to factory default, if the “Factory Reset” button is clicked. Factory default values of Module are listed below.

Category	Item	Value
Network	Local IP	192.168.11.100
	Local Gateway	192.168.11.1
	Local Subnet	255.255.255.0

Exit

Close the configuration tool program window.

Search Window

If you click the “Search” button, all MAC addresses on the same subnet will be displayed.

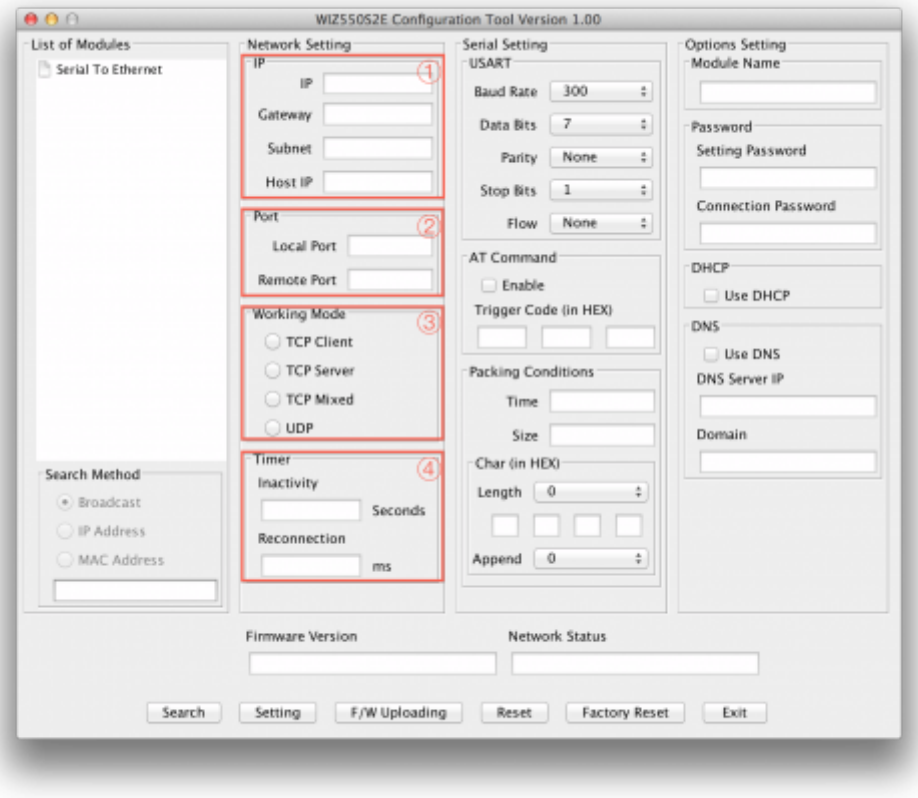
Search Method

Reserved

Firmware Version

It displays the firmware version.

Network Configurations



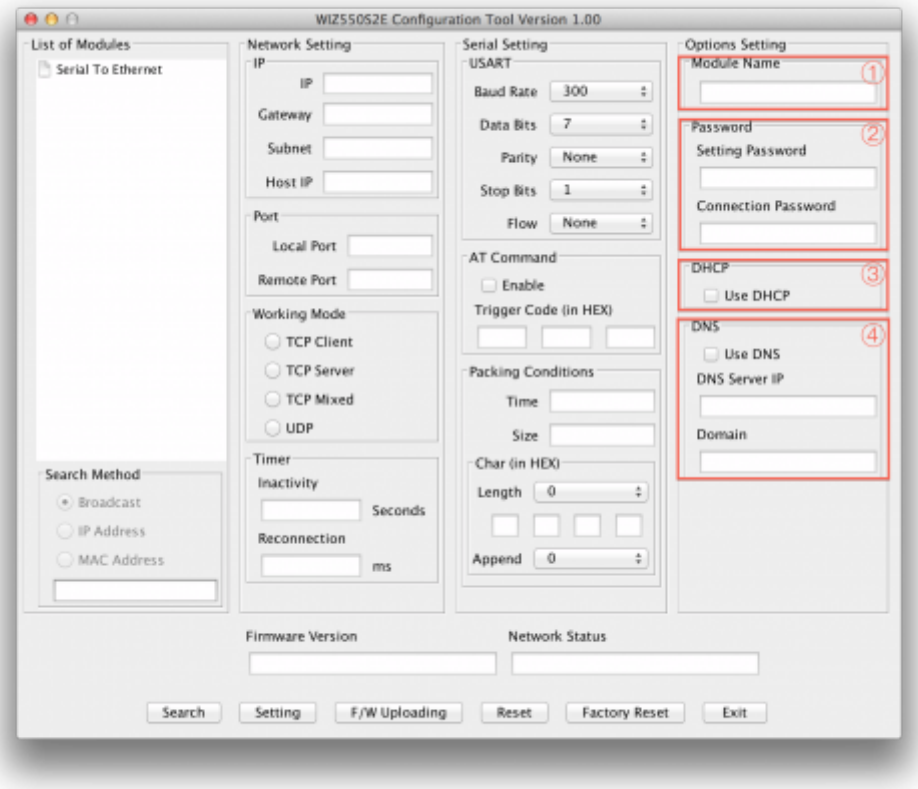
IP

This section is for setting Module mode's network information

```
IP:
    Module's IP Address
Gateway:
    Module's Gateway Address
Subnet mask:
    Module's Subnet Mask
```

☞ If you are unclear about your Local IP, Subnet Mask, Gateway information, you have to get this information from your network administrator. If the IP address is not correct, IP collision or network problems may occur.

Options Configurations



Module Name

The device name is displayed in this area.
User can use this name to distinguish this module with others

Password

Currently active for WIZ550web: Field “Setting Password”.

In this area the password for updating the settings can be changed. Critical functions like “Setting”, “Firmware Upload”, “Reset” and “Factory Reset” need this password to try issued action and avoid unauthorized users' command. If the user wants to replace the current or default Setting Password by a new one, this field should be updated with the new one before clicking on the “Setting” button and entering the old Setting Password.

Please note: the default Setting Password is “**WIZnet**” (without the brackets).

2014/10/24 15:44 · ywkwon

WIZ550WEB AT Command Set

This section provides a list of WIZ550WEB AT commands and their functions. Users can input commands and parameters through USART line. Every command starts with “AT”. Any other initial character will cause an error in return. Commands and parameters are all ASCII characters, i.e. when you input 'AT+NSTAT', you should input ASCII characters 'A', 'T', '+', 'N', 'S', 'T', 'A', 'T' and 'Enter Key' which is CR, LF (0x0d, 0x0A).

All commands should be terminated with CR(0x0D), LF(0x0A)

Some parameters are mandatory and others are optional. Parameters must be entered in the order of format column given by the command tables. Although the optional parameter is not used, the comma delimiters ',' must still be included in the command. In most cases, valid commands return the character [S] and invalid inputs return [F]. The possible responses sent from WIZ550WEB to the user are described as Responses. Below are examples of user input. As you can see, WIZ550WEB return “\r\n” back instead of “\r”, which means user (host system) always handle '\r\n' as the only delimiter.

Input by User	AT\r\n (0x61 0x74 0x0d 0x0a)
Output from WIZ550WEB	[S]\r\n (0x5b 0x53 0x5d 0x0d 0x0a)

Responses

Response Format

```
[(Type),(Id),(Param1),(Param2),(Param3),(Param4),(Param5),(Param6)]↓(Data)↓
```

- (Type): Type of response. It can be one of **S**, **D**, **F**, **W**, **R** and **V**.
- (Id): Socket Identifier. This is the mandatory in Async mode.
- (Param1) ~ (Param6):): These are included in case of commands retrieving module's setting value.
- ↓: This means 'Enter' key as delimiter and CR, LF(0x0d, 0x0a) are its real value.
- (Data): When huge data are needed, 'Data' will be followed in case of Type of response, D and R.

Responses are listed below.

Response	**Description**
Success Response	[S,(Id),(Param1),(Param2),(Param3),(Param4),(Param5),(Param6)]↓ Command Request Success, outputs with param when it's needed.
Success Dump Response	[D,(Id),(Size)]↓(Data)↓ Command Request Success, Outputs large data include 'Enter key' value.
Fail Response	[F,(Id),(ErrorCode),(ErrorParam)]↓ Command Request Fail, outputs with param when it's needed.
Wait Response	[W,(Id)]↓ Command is started with ID in Async mode.
Data Receive Response	[R,(SockId),(ReceivedSize),(SrcIP),(SrcPort)]↓(Data)↓ Outputs the received data.
Event Response	[V,(Id),(EventCode)]↓ Event occurred.

- (Id): 0 - System ID or 0~n - Socket Number
- (Size): Byte size of the output data
- (ErrorCode): Error Code
- (ErrorParam): Value of description for Error Code
- (SockId): Socket Identifier of the socket which received data
- (ReceivedSize): Byte size of received data
- (SrcIP): Sender's IP address. This is mandatory in case of UDP & TCP Client. In case of TCP Server this is omitted.
- (SrcPort): Sender socket's port number. This is mandatory in case of UDP & TCP Client. In case of TCP Server this is omitted.
- (EventCode): Indication of which event happened.

Error Code

General Error Code

Code	Error Name	Description
0	ERR_Undefined	Undefined Error
1	ERR_WrongOperator	Wrong Operator
2	ERR_WrongCommandSign	Wrong Command Sign
3	ERR_WrongArguments	Wrong Arguments
4	ERR_OurofRange	Parameter is out of Range
5	ERR_FuncDisabled	This function is disabled
6	ERR_NotAllowed	Not Allowed
7	ERR_CommandBusy	Command Busy
8	ERR_CommandTimeout	Command Timeout

Socket Error Code

Code	Error Name	Description
10	ERR_SockNotAvail	Socket Not Available
11	ERR_SockClosed	Socket Closed
12	ERR_SockPortNumNotAvail	Port Not Available
13	ERR_SockNotConnected	Not Connected
14	ERR_SockWrongAddr	Wrong Address
15	ERR_SockDataNotAvailable	Data Not Available

Other Error Code

Code	Error Name	Description
20	ERR_NoFreeMem	No Free Memory

Event Code

Socket Event Code

Code	Socket Event Name	Description
0	EVENT_SockConnected	Connected. Socket transition from Listen state to established state
1	EVENT_SockDisconnected	Disconnected. Socket transition from established state to disconnected state
2	EVENT_SockClosed	Closed. Socket transition to closed state
3	EVENT_SockDataRcvd	Data Received. The corresponding socket received data from its peer

---

Network Commands

Command	Prop.	Input Parameter	Response
AT+NSET	None or ?		[S,,S,(IP),(SN),(GW),(DNS)]
			[S,,D]
	=	S,(IP),(SN),(GW),(DNS)	[S]
		D	[S]
	-	*num*,Param	[S]
AT+NSTAT	None or ?		[S,,S/D,(IP),(SN),(GW),(DNS)]

Command	Prop.	Input Parameter	Response
AT+NMAC	None or ?		[S,,(MAC)]
	=	(MAC)	[S]

====AT+NSET====

- **Format:**

```
AT+NSET=<DHCP>,<IP>,<SN>,<GW>,<DNS>
```

- **Meaning:** Network Configuration

  <DHCP>: Static/DHCP

Parameter	Meaning
S	DHCP Off, Static
D	DHCP On, DHCP Client

  <IP>: IP Address (Optional)
  <SN>: Subnet Mask (Optional)
  <GW>: Gateway Address (Optional)
  <DNS>: DNS Address (Optional)

- **Response:**

```
[S]
```

- **Example 1:**

```
AT+NSET\r\n
```

```
AT+NSET?\r\n
```

- **Meaning:** Get Current Network Setting

  Note that <IP>,<SN>,<GW>,<DNS> address of response are not actual addresses, but addresses stored in the memory. So when DHCP is on, they are usually different from actual addresses.

- **Response:**

```
[S,,S,192.168.11.100,255.255.255.0,192.168.11.1,8.8.8.8]
```

```
[S,,D]
```

- ***Example 2:***

```
AT+NSET-2,192.168.11.110\r\n
```

- ***Meaning:*** *Update Second Parameter*

- ***Response:***

```
[S]
```

====AT+NSTAT====

- **Format:**

```
AT+NSTAT
```

```
AT+NSTAT?
```

- **Meaning:** Display Current Network Status
- **Response:**

```
[S,,<DHCP>,<IP>,<SN>,<GW>,<DNS>]
```

- ***Example 1:***

```
AT+NSTAT\r\n
```

```
AT+NSTAT?\r\n
```

- ***Meaning:*** *Display Current Network Status*
- ***Response:***

```
[S,,S,192.168.11.100,255.255.255.0,192.168.11.1,8.8.8.8]
```

```
[S,,D]
```

====AT+NMAC====

- **Format:**

```
AT+NMAC
```

```
AT+NMAC?
```

- **Meaning:** Get MAC Address
- **Response:**

```
[S,,<MAC>]
```

- ***Example 1:***

```
AT+NMAC=00:08:dc:1d:bb:8b\r\n
```

- ***Meaning:*** *Set MAC Address*
- ***Response:***

```
[S]
```

- ***Example 2:***

```
AT+NMAC\r\n
```

```
AT+NMAC?\r\n
```

- ***Meaning:*** *Get MAC Address*
- ***Response:***

```
[S,,00:08:dc:1d:bb:8a]
```

Management Commands

Command	Prop.	Input Parameter	Response
AT	None		[S]
	?		[D,,(Size)]↓(Data)
AT+MSTAT	None or ?		[S,,(Version)]
AT+MUSART1	None or ?		[S,,(BR),(W),(P),(S),(F)]
	=	(BR),(W),(P),(S),(F)	[S]
	-	*num*,Param	[S]
AT+MUSART2	None or ?		[S,,(BR),(W),(P),(S),(F)]
	=	(BR),(W),(P),(S),(F)	[S]
	-	*num*,Param	[S]
AT+MSAVE	None		[S]
AT+MRST	None		[S]
	=	F	[S]

====AT====

- **Format:**

```
AT
```

- **Meaning:** Terminal Check
- **Response:**

```
[S]
```

====AT+MSTAT====

- **Format:**

```
AT+MSTAT
```

```
AT+MSTAT?
```

- **Meaning:** Get Current Version
- **Response:**

```
[S,,<Version>]
```

====AT+MUSART1====

- **Format:**

```
AT+MUSART1=<BR>,<W>,<P>,<S>,<F>
```

- **Meaning:** Serial Interface(USART1) Configuration

: Baud rate

Parameter	Meaning
300	300bps
600	600bps
1200	1200bps
2400	2400bps
4800	4800bps
9600	9600bps
19200	19200bps
38400	38400bps
57600	57600bps
115200	115200bps
230400	230400bps

<W>: Word length

Parameter	Meaning
7	7 bits
8	8 bits

<P>: Parity bit

Parameter	Meaning
N	NONE
O	ODD
E	EVEN

<S>: Stop bit

Parameter	Meaning
1	1 bits
2	2 bits

<F>: Flow Control

Parameter	Meaning
0	NONE
1	RTS/CTS
2	RS422
3	RS485

- **Response:**

```
[S,,<BR>,( <W>, <P>, <S> ) <F>]
```

- **Example1:**

```
AT+MUSART1
```

```
AT+MUSART1?
```

- **Meaning:** Get Serial Interface(USART1) Information
- **Response:**

```
[S,,<BR>,( <W>, <P>, <S> ) <F>]
```

- **Example2:**

```
AT+MUSART1=,,E,,0
```

- **Meaning:** Set Serial Interface(USART1) Information
- **Response:**

```
[S]
```

====AT+MUSART2====

- **Format:**

```
AT+MUSART2=<BR>,<W>,<P>,<S>,<F>
```

- **Meaning:** Serial Interface(USART2) Configuration

: Baud rate

Parameter	Meaning
300	300bps
600	600bps
1200	1200bps
2400	2400bps
4800	4800bps
9600	9600bps
19200	19200bps

Parameter	Meaning
38400	38400bps
57600	57600bps
115200	115200bps
230400	230400bps

<W>: Word length

Parameter	Meaning
7	7 bits
8	8 bits

<P>: Parity bit

Parameter	Meaning
N	NONE
O	ODD
E	EVEN

<S>: Stop bit

Parameter	Meaning
1	1 bits
2	2 bits

<F>: Flow Control

Parameter	Meaning
0	NONE
1	RTS/CTS
2	RS422
3	RS485

- **Response:**

```
[S,,<BR>,( <W>, <P>, <S> ) <F>]
```

- **Example1:**

```
AT+MUSART2
```

```
AT+MUSART2?
```

- **Meaning:** Get Serial Interface(USART2) Information

- **Response:**

```
[S,,<BR>,( <W>, <P>, <S> ) <F>]
```

- **Example2:**

```
AT+MUSART2=,,E,,0
```

- **Meaning:** Set Serial Interface(USART2) Information
- **Response:**

```
[S]
```

====AT+MSAVE====

- **Format:**

```
AT+MSAVE
```

- **Meaning:** Save configuration data to flash
- **Response:**

```
[S]
```

====AT+MRST====

- **Format:**

```
AT+MRST
```

- **Meaning:** Reset Module
- **Response:**

```
[S]
```

Function Commands

Command	Prop.	Input Parameter	Response
AT+FIODIR	=	(PIN)	[S,,(Direction)]
	=	(PIN),(Direction)	[S]
AT+FIOVAL	=	(PIN)	[S,,(VAL)]
	=	(PIN),(VAL)	[S]

====AT+FIODIR====

- **Format:**

```
AT+FIODIR=<PIN>
```

```
AT+FIODIR=<PIN>,<Direction>
```

- **Meaning:** Read/Write the Status of GPIO Pin Direction

<PIN> : GPIO Pin Number (1 ~ 16)
<Direction> : GPIO Pin Direction

Parameter	Meaning
0	Not Used (Read Only)
1	Input
2	Output

- **Response:**

```
[S,,<Direction>]
```

```
[S]
```

- **Example1:**

```
AT+FIODIR=1\r\n
```

- **Meaning:** Read the Direction of GPIO Pin 1

- **Response:**

```
[S,,1]
```

- **Example2:**

```
AT+FIODIR=1,2\r\n
```

- **Meaning:** Write the Direction of GPIO Pin 1 to Output

- **Response:**

```
[S]
```

====AT+FIOVAL====

- **Format:**

```
AT+FIOVAL=<PIN>
```

```
AT+FIOVAL=<PIN>,<VAL>
```

- **Meaning:** Read/Write the Input/Output Value of GPIO Pin

  <PIN> : GPIO Pin Number (1 ~ 16)
  <VAL> : GPIO Pin Output Value

Parameter	Meaning
0	Low
1	High

- **Response:**

```
[S,,<VAL>]
```

```
[S]
```

- **Example1:**

```
AT+FIOVAL=1\r\n
```

- **Meaning:** Read the Value of GPIO Pin 1

- **Response:**

```
[S,,1]
```

- **Example2:**

```
AT+FIOVAL=1,1\r\n
```

- **Meaning:** Write the Value of GPIO Pin 1 to High(Output Only)

- **Response:**

```
[S]
```

2014/10/24 16:02 · ywkwon