

# WIZnet S2E (Serial-to-Ethernet) Device's Configuration Tool Programming Guide

Rev 0.3.0

This document describes how to make your own Configuration Tool for WIZ100SR, WIZ105SR and WIZ110SR of WIZnet. And the detailed communication protocols and message formats will also be introduced. For the other S2E devices, the communication port numbers and setting message are different from these products. If you are using other S2E devices of WIZnet and you want to make you own configuration program, please contact us to get more detailed information for programming.

## 1. Search Device

There are two type search methods: one is broadcast using UDP, the other is uni-cast using TCP. The broadcast method can find all connected S2E devices in the same subnet; as well the uni-cast method can only find the S2E device which has the input IP address. The two methods use the same command that is shown in Fig. 1. Please, note that the remote UDP port is 1460 and local UDP port is 5001, and the remote TCP port for direct searching is 1461.

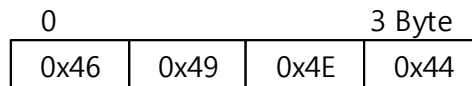
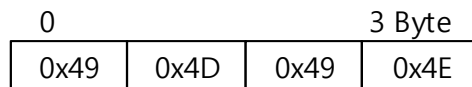
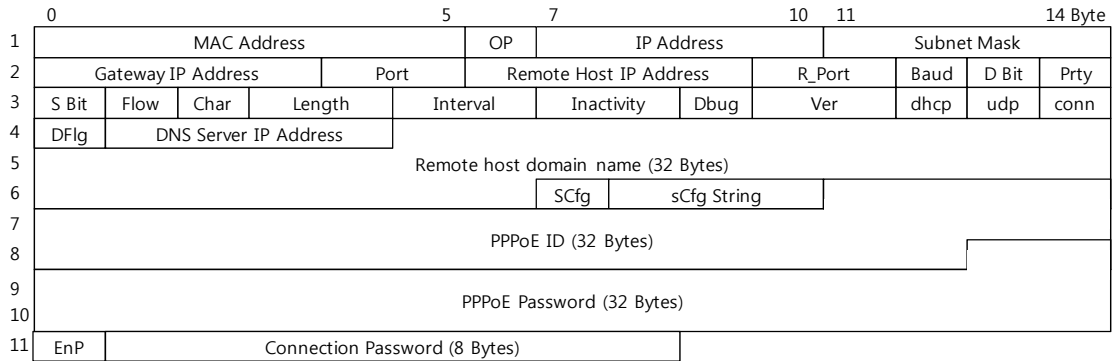


Fig. 1 Search request command format ("FIND")

Once the S2E device receives the 'Search command' from one host, it will response the host with the command and detailed current setting message as shown in Fig. 2 (a) and (b), respectively.



(a) Searching response command ("IMIN")



(b) Setting message

Fig. 2 Searching response command (a) and setting message (b)

The total length of the response message is 153 Bytes (includes command), and each field is introduced as follows.

- MAC address (6 Bytes)  
This field is read only which indicates the MAC address of the S2E device.
- OP (1 Byte)  
Indicate the operation mode  
0x00: client mode  
0x01: server mode  
0x02: mixed mode
- IP address (4 Byte)  
S2E device's IP address
- Subnet mask (4 Byte)  
S2E device's subnet mask
- Gateway IP address  
Gateway IP address of S2E device
- Port (2 Byte)  
Opened TCP or UDP port of S2E device
- Remote host IP address (4 Byte)  
Remote host IP address for S2E device to connect (Client mode)
- R\_port (2 Byte)  
TCP or UDP port of remote host
- Baud (1 Byte)  
Indicate the current serial baud rate of the S2E device  
0xA0: 1200  
0xD0: 2400

0xE8:	4800
0xF4:	9600
0xFA:	19200
0xFD:	38400
0xFE:	57600
0xFF:	115200
0xBB:	230400

- D\_Bit (1 Byte)

The current serial port data bits of the S2E device

0x07: 7 bits

0x08: 8 bits

- Prty (1 Byte)

Indicate the serial port parity settings of the S2E device

0x00: None

0x01: Odd

0x02: Even

- S\_Bit (1 Byte)

Indicate the current serial port stop bits of the S2E device

0x01: 1 bit

- Flow (1 Byte)

Indicate the serial port flow control settings of the S2E device

0x00: None

0x01: Xon/Xoff

0x02: CTS/RTS

- Char (1 Byte)

The field defines the special character that the input serial data will not be sent out until the defined character received. 0 means this packing condition is not used.

- Length (2 Bytes)

Input serial data packing length. When the input serial data length reaches to the value, it will be sent out. Please, note that only the low 1 byte is used and the maximum length is 0x00FF. 0 means the packing condition is disabled.

- Interval (2 Bytes)

Input serial data packing interval, which indicates how long to send the input serial data once. 0 means the condition is not used and 0xFFFF is the maximum interval in millisecond.

- Inactivity (2 Bytes)  
This field indicates that the TCP connection will be closed when there is no input serial data in the defined inactivity interval in second. 0 disables the condition.
- Dbug (1 Byte)  
This field determines if the debug message is printed out or not via serial port.  
0x00: Enable output debug message  
0x01: Disable output debug message
- Ver (2 Bytes)  
Firmware version. Low byte is major version and high byte is minor version (e.g. 0x0309 means version 3.9).
- dhcp (1 Byte)  
This field defines how the device gets its IP address.  
0x00: Using static IP address  
0x01: Using DHCP dynamic IP address  
0x02: Using PPPoE gets dynamic IP address
- udp (1 Byte)  
This field defines the transmission protocol, TCP or UDP.  
0x00: Using TCP  
0x01: Using UDP
- conn (1 Byte)  
This field is read only which indicates the remote host connection status.  
0x00: The device is disconnected with remote host  
0x01: The device is connected with remote host
- DFlg (1 Byte)  
This flag indicates the remote host using IP address or domain name.  
0x00: IP address  
0x01: Domain name
- DNS Server IP Address (4 Bytes)  
This field defines the DNS server IP address of the device.
- Remote host domain name (32 Bytes)  
This field stores the remote host domain name.
- sCfg (1 Byte)  
This field defines the serial configuration mode trigger status.  
0x00: Disabled  
0x01: Enabled

- sCfg String (3 Bytes)  
Serial configuration mode trigger command. (if the device receives the defined 3 bytes via serial port, it will go to configuration mode)
- PPPoE ID (32 Bytes)  
User's PPPoE ID.
- PPPoE Password (32 Bytes)  
User's PPPoE password.
- EnP (1 Byte)  
This byte defines remote host TCP connection password is used or not. This function can only be used in server mode. If this function is enabled, the remote host must send the defined password to the device when connection built.  
0x00: Disabled  
0x01: Enabled
- Connection Password (8 Bytes)  
The connection password defined for the remote host.

Fig. 3 shows the message exchange timing sequence between configuration tool and a S2E device.

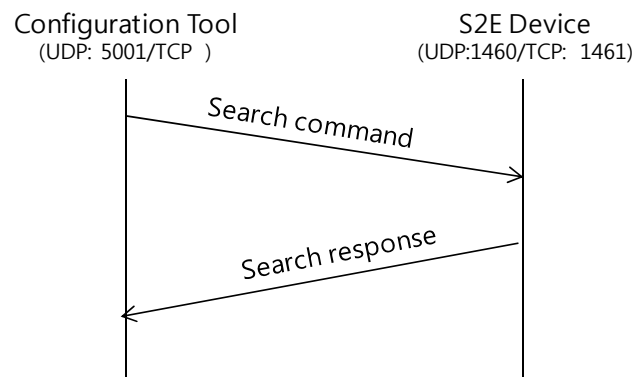


Fig. 3 Searching message exchange timing sequence diagram

## 2. Setting Device

If some S2E devices have been listed after searching, you can select one device to set its parameters using the following command. Here, the remote UDP port is 1460 and local UDP port is 5001, and TCP port 1461 is opened in the S2E device side for directly

connection oriented setting. You should send the setting command as shown in Fig. 4 with the setting message to the selected device.

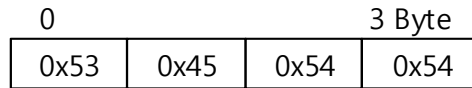


Fig. 4 Setting request command ('SETT')

After the device receiving the setting request command and message successfully, it will response the host with a setting response command as shown in Fig. 5 with the same setting message.

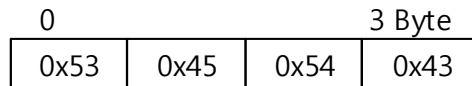


Fig. 5 Setting response command ('SETC')

If the host receives the response command from the S2E device, it means that the new setting message has been processed successfully. And now the S2E device will run with new parameter values. The whole setting procedure is shown in Fig. 6.

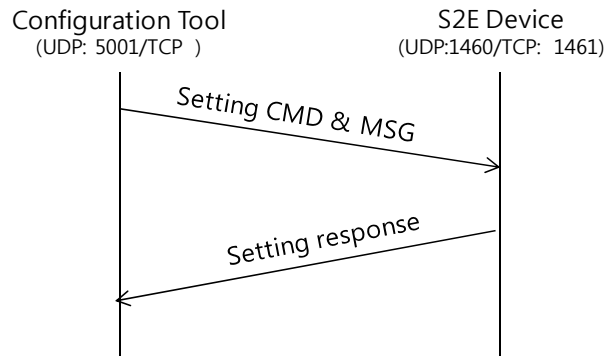
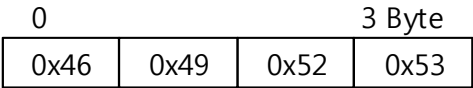


Fig. 6 Setting message exchange timing sequence diagram

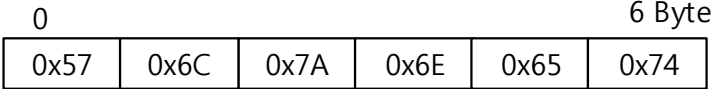
### 3. Upload Firmware

The S2E's firmware can also be updated via network. A command is needed to make the S2E device jump to boot code to receive data of the new firmware. The command is shown in Fig. 7. You also can broadcast the command with the above introduced setting message via local UDP port 5001 to remote UDP port 1460. If the direct searching is checked, the command with setting message should be sent out by TCP via

port 1461. Please, note the created binary file should be attached a 6 bytes header and 2 bytes checksum in the end of the file. The header is shown as Fig. 8. After the S2E received the command, it will jump to boot and listen on the TCP port 1470. So, another TCP socket is needed to connect the predefined port. After the TCP connection established, the length of the generated binary file including header and checksum should be sent to the S2E firstly. The length is 4 bytes in little-endian. And now, you can transfer the binary file to the S2E via the established connection.



**Fig. 7 Firmware uploading command ('FIRS')**



**Fig. 8 Firmware header format ('Wiznet')**

After the S2E received the whole new firmware, it will reboot again, and the new firmware will be applied. Please, note if error occurs during file transferring, the version of firmware will be set to 82.0. If you find a device has this version number, you have to upload the firmware again. Please note that only the firmware provided by WIZnet can be uploaded to your device. The following Fig. 9 shows a whole firmware uploading procedure.

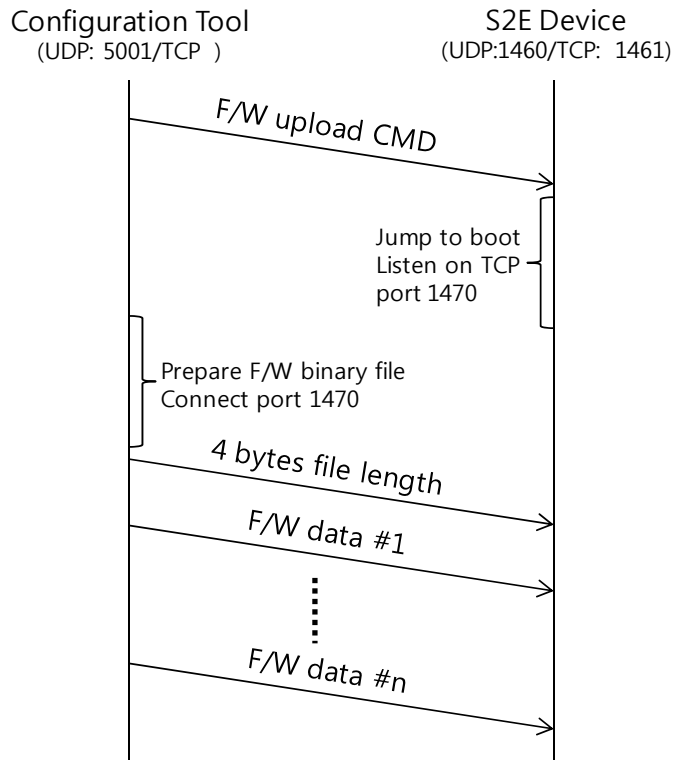


Fig. 9 Firmware uploading timing sequence diagram