Secure Terminal Server

STS Series

User Guide

Version 1.4.1

2006-06-09

Copyright Information

Copyright 1998-2006, Sena Technologies, Inc. All rights reserved.

Sena Technologies reserves the right to make any changes and improvements to its product without providing prior notice.

Trademark Information

HelloDevice™ is a trademark of Sena Technologies, Inc.

Windows® is a registered trademark of Microsoft Corporation.

Ethernet® is a registered trademark of XEROX Corporation.

Notice to Users

Proper back-up systems and necessary safety devices should be utilized to protect against injury, death or property damage due to system failure. Such protection is the responsibility of the user.

This device is not approved for use as a life-support or medical system.

Any changes or modifications made to this device without the explicit approval or consent of Sena Technologies will void Sena Technologies of any liability or responsibility of injury or loss caused by any malfunction.

Technical Support

Sena Technologies, Inc.

210 Yangjae-dong, Seocho-gu

Seoul 137-130, Korea

Tel: (+82-2) 573-5422

Fax: (+82-2) 573-7710

E-Mail: support@sena.com

Website: http://www.sena.com

Revision history

Revision	Date	Name	Description
V1.0.2	**2003-12-3**	**O.J. Jung**	Initial Release
V1.1.0	**2004-01-12**	**O.J. Jung**	Revision with release of version 1.1.0
V1.1.1	**2004-01-30**	**O.J. Jung**	Typographical errors are fixed
V1.2.0	**2004-06-11**	**O.J. Jung**	Revision with release of version 1.2.0
V1.3.0	**2004-10-11**	**O.J. Jung**	Revision with release of version 1.3.0
V1.3.1	**2004-10-15**	**O.J. Jung**	Added Appendix 6
V1.3.2	**2005-05-18**	**O.J. Jung**	Appendix 7 is added, PC Card List is updated, The description about DSR behavior is corrected. Typographical errors are fixed.
V1.3.3	**2005-07-26**	**O.J. Jung**	Descriptions for Modem Emulation mode command and IP Statistics changed
V1.4.0	**2006-02-03**	**O.J. Jung**	Typo on AT command in modem emulation mode is corrected. Revision with release version 1.4.0
V1.4.1	**2006-06-06**	**O.J. Jung**	Revision with release version 1.4.1

Contents

1. Introduction

1.1. Overview

The STS Series is a secure terminal server (or device server) that makes your legacy serial devices manageable by industry-standard Ethernet network. Based on open network protocols such as TCP/IP and UDP, it gives you ultimate flexibility to your serial devices. With PPPoE (PPP-over-Ethernet) connection feature of the STS Series, the RS232 serial devices could be managed over DSL-based broadband network.

With the rich broadband network connectivity protocols such as DHCP, PPPoE and Dynamic DNS, you could easily manage the legacy serial devices over broadband Internet by using DSL or cable modem connection. The built-in Dynamic DNS protocol of the STS Series enables you to access the serial devices with their domain names.

The STS Series also provides you with full-featured system management functionality of system status display, firmware upgrade, remote reset and system log display by using various ways such as telnet, serial console port or web.

You could easily configure and administrate the STS Series, with the full-featured management functions of status monitor, remote reset, error log monitor and firmware upgrade by using Telnet and serial console port under the password protection support.

For critical applications of secure data communication, the STS Series supports SSLv2, SSLv3 and TLSv1 for data encryption. In addition, IP address filtering function is provided for protecting unintentional data streams to be transmitted to the STS Series.

Typical application areas of the STS Series are:

- Industrial automation
- Network management
- Retail/Point of sale
- Remote metering
- Remote display
- Building automation
- Security/Access control systems
- General data acquisition application
- Medical application

The STS Series gives you ideal remote management capability of control, monitoring, diagnosis and data gathering over RS232 serial devices.

Please note that this manual assumes user knowledge of Internetworking protocols and serial communications.

1.2. Package Check List

- STS Series external box
- External 110V or 230V power supply or power cord
- CAT5 cable
- Console cable kit
- Quick Start Guide
- CD-ROM, including the Serial/IP Com Port Redirector, HelloDevice-IDE, HelloDevice Manager and manuals

1.3. Product Specification

	STS400	STS800	STS1600
Serial Interface	4-port	8-port	16-port
	Serial speeds 75bps to 230Kbps		
	Flow Control: Hardware RTS/CTS, Software Xon/Xoff		
	RJ45 connector		
	Signals: RS232 Rx, Tx, RTS, CTS, DTR, DSR, DCD, GND		
	Modem controls: DTR/DSR and RTS/CTS		
Network Interface	10/100 Base-Tx Ethernet with RJ45 Ethernet connector		
	Supports static and dynamic IP address		
Protocols	- ARP, IP/ICMP, TCP, UDP, Telnet, SSH v1 & v2, - SSL v2 & v3, TLS v1 - DNS, Dynamic DNS, HTTP, HTTPS, - SMTP with/without Authentication, pop-before SMTP, - DHCP client, NTP, PPPoE, SNMP v1 & v2		
PCMCIA	Supports one of the following PC cards: ATA flash memory card 802.11b Wireless LAN card 10/100 Base-TX LAN Card Modem card		
Security	User ID & Password		
	HTTPS		
	Secure terminal interface: SSH		
	Data Encryption: SSLv2/v3, TLS v1, 3DES and RC4		
	IP address filtering		
	SCP		
Modem emulation	Full support for AT commands		
Management	Web, Telnet or Serial console port or HelloDevice Manager		
	O/S support: Windows 98/ME/NT/2000/XP		
	System log Automatic email delivery of error log		
	System statistics Full-featured system status display		
	Firmware Stored in Flash memory and upgradeable via serial console, telnet or web		
Diagnostic LED	Power Ready 10/100 Base Link, Act Serial InUse/Rx/ Tx for each port PC Card		
Environmental	Operating temperature: 5'C to 50'C Storage temperature: –40'C to 66'C		
Power	5VDC, 1.5A @ 5VDC		110 ~ 240VAC
Dimension L x W x H (mm)	245 x 153 x 30 (mm)		432 x 193 x 44.5
	DIN-rail mount option		19 in. rack mountable
Weight (kg)	1.5		2.8
Certification	FCC(A), CE(A), MIC		

Warranty	5-year limited warranty

1.4. Terminologies and acronyms

This section will define commonly used terms in this manual. These terms are related to Internetworking, and defined in regards to their use with STS Series.

- **MAC address**

On a local area network or other network, the MAC (Media Access Control) address is the computer's unique hardware number. (On an Ethernet LAN, it is the same as the Ethernet address.)

It is a unique 12-digit hardware number, which is composed of 6-digit OUI (Organization Unique Identifier) number and 6-digit hardware identifier number. The STS Series has the following MAC address template: 00-01-95-xx-xx-xx. The MAC address can be found on the bottom of the original package.

- **Host**

A user's computer connected to the network

Internet protocol specifications define "host" as any computer that has full two-way access to other computers on the Internet. A host will have a specific "local" or "host number" that, together with the network number, forms its unique IP address.

- **Session**

A series of interactions between two communication end points that occur during the span of a single connection

Typically, one end point requests a connection with another specified end point. If that end point replies, agreeing to the connection, the end points take turns exchanging commands and data ("talking to each other"). The session begins when the connection is established at both ends and terminates when the connection is ended.

- **Client/Server**

Client/server describes the relationship between two computer programs in which one program, the client, makes a service request from another program, the server, which fulfills the request.

A server is a computer program that provides services to other computer programs on one or many computers. The client is the requesting program or user in a client/server relationship. For example, the user of a Web browser is effectively making client requests for pages from servers all over the Web. The browser itself is a client in its relationship with the computer that is getting and returning the requested HTML file. The computer handling the request and sending back the HTML file is a server.

Table 1-1 Acronym Table

ISP	Internet Service Provider
PC	Personal Computer
NIC	Network Interface Card
MAC	Media Access Control
LAN	Local Area Network
UTP	Unshielded Twisted Pair
ADSL	Asymmetric Digital Subscriber Line
ARP	Address Resolution Protocol
IP	Internet Protocol
ICMP	Internet Control Message Protocol
UDP	User Datagram Protocol
TCP	Transmission Control Protocol
DHCP	Dynamic Host Configuration Protocol
SMTP	Simple Mail Transfer Protocol
FTP	File Transfer Protocol
PPP	Point-To-Point Protocol
PPPoE	Point-To-Point Protocol over Ethernet
HTTP	HyperText Transfer Protocol
DNS	Domain Name Service
DDNS	Dynamic Domain Name Service
SNMP	Simple Network Management Protocol
RADIUS	Remote Access for Dial-In User Service
SSH	Secure Shell
NTP	Network Time Protocol
UART	Universal Asynchronous Receiver/Transmitter
Bps	Bits per second (baud rate)
DCE	Data Communications Equipment
DTE	Data Terminal Equipment
CTS	Clear to Send
DSR	Data Set Ready
DTR	Data Terminal Ready
RTS	Request To Send
DCD	Data Carrier Detect

2. Getting Started

This chapter describes how to set up and configure the STS Series.

- *2.1 Panel Layout* explains the layout of the panel and LED indicators.
- *2.2 Connecting the Hardware* describes how to connect the power, the network, and the equipment to the STS Series.
- *2.3 Accessing the Web Browser Management Interface* describes how to access the console port using a serial console or a Telnet or Web menu from remote location.

The following items are required to get started.

- One power cable (included in the package)
- Console and Ethernet cables (included in the package)
- Cable kit (included in the package)
- One PC with Network Interface Card (hereafter, NIC) and/or one RS232 serial port.

2.1. Panel Layout

2.1.1. STS400/800 Panel Layout

The STS400/800 has three groups of LED indicator lamps to display the status, as shown in *Figure 2-1* and *Figure 2-2* (i.e. System, Ethernet and Serial ports). The first three lamps on the left side indicate Power, Ready and PC Card interface. The next three lamps are for Ethernet 100Mbps, Link and Act. Next lamps indicate InUse, Receive and Transmit of the serial ports.

Table 2-1 describes the function of each LED indicator lamp. The rear panel shows the serial ports with RJ45 connector, Ethernet port, the STS400/800 console port and the power socket.

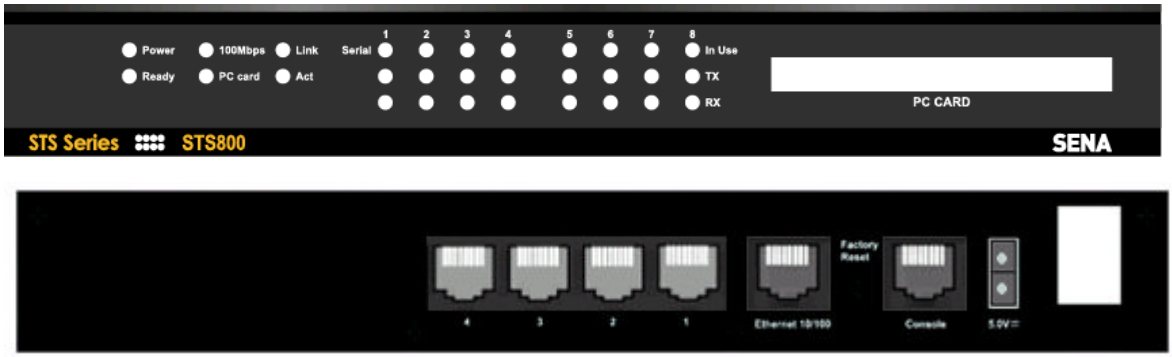


Figure 2-1 The panel layout of the STS800

Table 2-1 LED indicator lamps of the STS Series

Lamps		Function
System	**Power**	Turned on if power is supplied
	Ready	Turned on if system is ready to run
	PC card	Turned on if a PCMCIA device is running
Ethernet	**100Mbps**	Turned on if 100Base-TX connection is detected
	LINK	Turned on if connected to Ethernet network
	Act	Blink whenever there is any activities such as incoming or outgoing packets through the STS Series Ethernet port
Serial port	**InUse**	Turned on if the serial port is in use (Port buffering enabled or port access in use)
	Rx/Tx	Blink whenever there is any incoming or outgoing data stream through the serial port of the STS Series

2.1.2. STS1600 Panel Layout

The STS1600 has three groups of LED indicator lamps to display the status, as shown in *Figure 2-2* (i.e. System, Ethernet and Serial ports). The first three lamps on the left side indicate Power, Ready and PCMCIA interface. The next three lamps are for Ethernet 100Mbps, Link and Act. Next lamps indicate InUse, Receive and Transmit of the serial ports.
Table 2-1 describes the function of each LED indicator lamp.

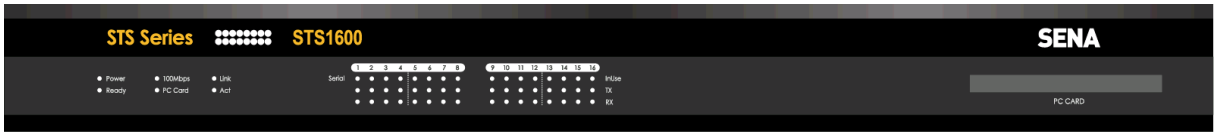


Figure 2-2 The panel layout of the STS1600

2.2. Connecting the Hardware

This section describes how to connect the STS Series to the equipment for initial testing.

- Connect a power source to the STS Series
- Connect the STS Series to an Ethernet hub or switch
- Connect the device

2.2.1. Connecting the power

Connect the power cable to the STS Series. If the power is properly supplied, the [Power] lamp will light up green.

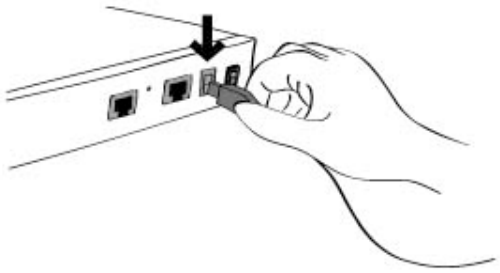


Figure 2-3 Connecting the power to the STS400/800

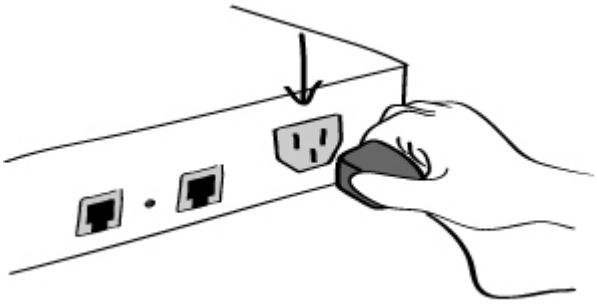


Figure 2-4 Connecting the power to the STS1600

2.2.2. Connecting to the network

Plug one end of the Ethernet cable to the STS Series Ethernet port. The other end of the Ethernet cable should be connected to a network port. If the cable is properly connected, the STS Series will have a valid connection to the Ethernet network. This will be indicated by:

The [Link] lamp will light up green.

The [Act] lamp will blink to indicate incoming/outgoing Ethernet packets

The [100Mbps] lamp will light up green if the STS Series is connected to 100Base-TX network

The [100Mbps] lamp will not turn on if the current network connection is 10Base-T.

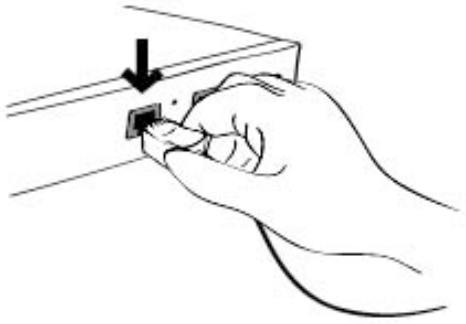


Figure 2-5 Connecting a network cable to the STS400/800/1600

2.2.3. Connecting to the device

Connect the console cable to the STS Series serial port. To connect to the console port of the device, the user needs to consider the type of console port provided by the device itself. In the STS Series cable kit package, plug-in adapters are provided for the easier connectivity to the user's devices. Please refer to the *Appendix 1 Connections* for details.

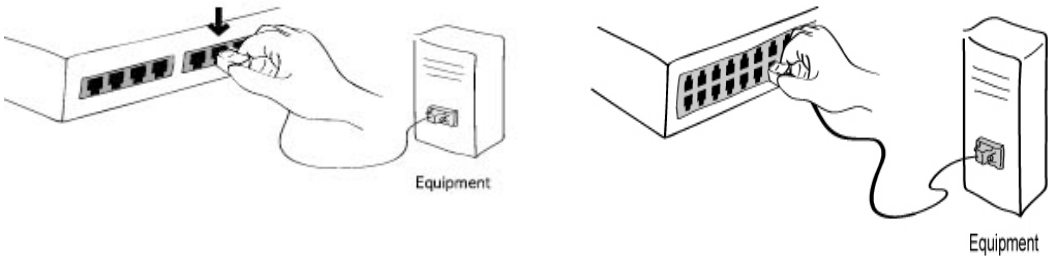


Figure 2-6 Connecting a equipment to the STS400/800(Left) / STS1600(Right)

2.2.4. Accessing the System Console

There are several ways to access the STS Series. These methods are dependent on whether the user is located at a local site or a remote site, or whether s/he requires a menu-driven interface, graphic menu system or CLI (Command Line Interface).

- **System console:**

  Local users can connect directly to the system console port of the STS Series using the console/Ethernet cable with the corresponding adapter.

- **Remote console:**

  Remote users who require a menu-driven interface can utilize Telnet (port 23) connections to the STS Series using terminal emulator.

- **Web:**

  Remote users who want to use a web browser to configure the STS Series can connect to the STS Series using conventional web browsers, such as Internet Explorer or Netscape Navigator.

The above methods require the user authentication by the STS Series system.

2.2.5. Using the System console

1) Connect one end of the console/Ethernet cable to the console port on the STS Series.

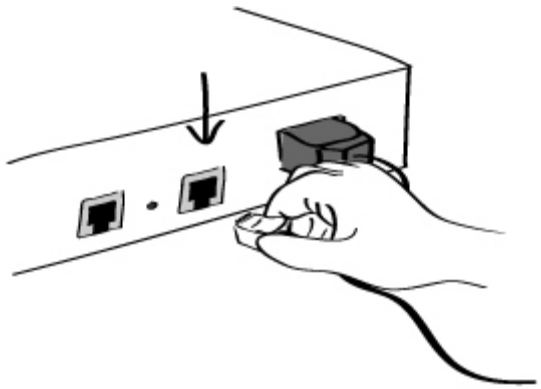


Figure 2-7 Connecting a system console cable to the STS Series

2) Connect to the user's computer with the RJ45-DB9 female adapter.
3) Connect the other end of the cable to the serial port of the user's computer.
4) Run a terminal emulator program (i.e. HyperTerminal). Set up the serial configuration parameters of the terminal emulation program as follows:
   - **9600 Baud rate**
   - **Data bits 8**
   - **Parity None**
   - **Stop bits 1**
   - **No flow control**
5) Press the [ENTER] key.
6) Enter your user name and password to log into the STS Series. The factory default user

settings are as follows.

Login: root Password: root

Login: admin Password: admin

```
192.168.161.5 login: root
Password:****
root@192.168.161.5:~#
```

7) Upon authentication, the CLI are initially provided for configuration. For details on the CLI, refer to the chapter *8 CLI guide.*

8) "ss.edit" command will allow you to enter the text-menu driven interface and the menu screen in *Figure 2-8* is displayed.

```
root@192.168.161.5:~#ss.edit

------------------------------------------------------------------------------
 Welcome to STS-800 configuration page
 Current time: 08/22/2003 21:52:36     F/W REV.: v1.0.1
 Serial No.: STS800438349-42944        MAC address: 00-01-95-04-19-5a
 IP mode: DHCP                         IP address: 192.168.14.7
------------------------------------------------------------------------------
 Select menu:
  1. Network configuration
  2. Serial port configuration
  3. PC Card configuration
  4. System administration
  5. Save changes
  6. Exit without saving
  7. Exit and apply changes
  8. Exit and reboot
  <Enter> Refresh
 ----->
```

Figure 2-8 The main menu screen (STS800)

From the main menu screen, the user may select the menu item for the configuration of the STS Series parameters by typing the menu number and pressing the [ENTER] key. In the submenu screen, users can configure the required parameters guided by online comments. All the parameters are stored into the non-volatile memory space of the STS Series, and it will not be stored until users select menu "*5.Save changes*". All the configuration change will be effective after selecting the menu "*7. Exit and apply changes*" or "*8. Exit and reboot*".

2.2.6. Using Remote console

The IP address of the STS Series must be known before users can access the STS Series using the Remote console (see chapter *3 Network Configuration* for details). The default IP address of STS

Series is **192.168.161.5**.

The Remote console access function can be disabled in the remote host access option (*3.5 IP Filtering* for details).

The following instructions will assist in setting up the Remote Console functionality:

1) Run either a Telnet program or a program that supports Telnet functions (i.e. TeraTerm-Pro or HyperTerminal). The target IP address and the port number must match the STS Series. If required, specify the port number as 23. Type the following command in the command line interface of user's computer.

```
telnet 192.168.161.5
```

Or run a Telnet program with the following parameters:

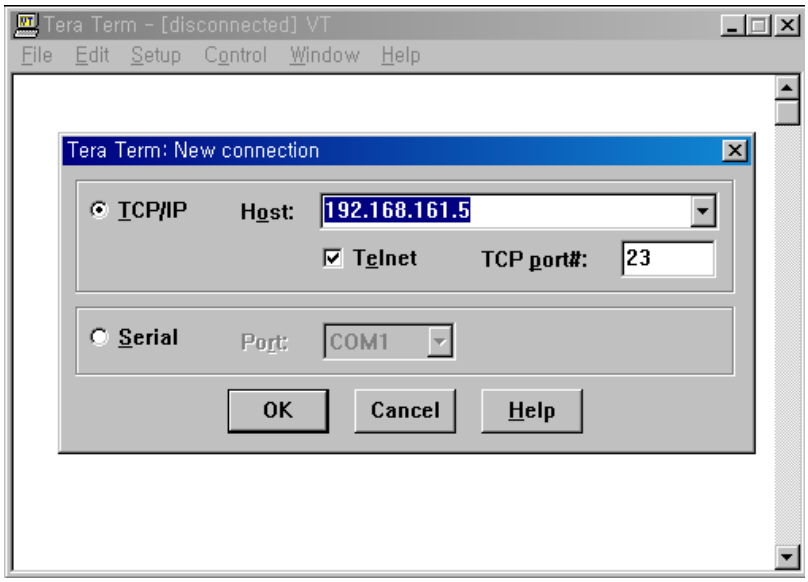


Figure 2-9 Telnet program set up example (TeraTerm Pro)

2) The user must log into the STS Series. Type the user name and password. A factory default setting of the user name and password are both **root** for the system root and **admin** for the system administrator.
3) Upon authentication by the STS Series, the CLI prompts or text menu screens are shown.

2.3. Accessing the Web Browser Management Interface

The STS Series supports both HTTP and HTTPS (HTTP over SSL) protocols. The STS Series also provides has its own Web management pages. To access the STS Series Web management page, enter the IP address or resolvable hostname of the STS Series into the web browser's URL/Location

field. This will direct the user to the STS Series login screen. The user must authenticate themselves by logging into they system with a correct user name and password. The factory default settings are:

Login: root Password: root

Login: admin Password: admin

Note: *Before accessing the* STS Series *Web management page, the user must check the IP address (or resolvable Hostname) of the STS Series and Subnet mask settings.*

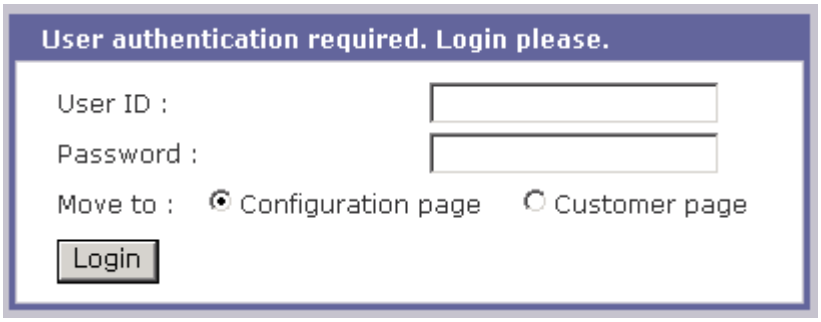


Figure 2-10 Login screen of the STS Series web management

Figure 2-10 shows Login screen of the STS Series web management. After login, select initial page where he want to move to after login. If user selects Configuration page, he can see the configuration homepage of the Super Series Web management interface shown on *Figure 2-11*. If user select Customer page, he can see the default Customer homepage of the STS Series Web management interface or his own homepage. For more detail information about user customization of Web UI, please refer to 9 User customization guide section.

Figure 2-11 shows the configuration homepage of the STS Series Web management interface. A menu bar is provided on the left side of the screen. The menu bar includes the uppermost configuration menu groups. Selecting an item on the menu bar opens a tree view of all the submenus available under each grouping. Selecting a submenu item will allow the user to modify parameter settings for that item. Every page will allow the user to [Save to flash], [Save & apply] or [Cancel] their actions. After changing the configuration parameter values, the users must select [Save to flash] to save the changed parameter values to the non-volatile memory. To apply all changes made, the user must select [Apply Changes]. This option is available on the bottom of the menu bar. Only when the user selects [Apply changes] will the new parameter values be applied to the STS Series configuration. The user also can select [Save & apply] to save parameters and apply changes in one step.

If the user does not want to save the new parameter values, the user must opt to [Cancel]. All changes made will be lost and the previous values restored.

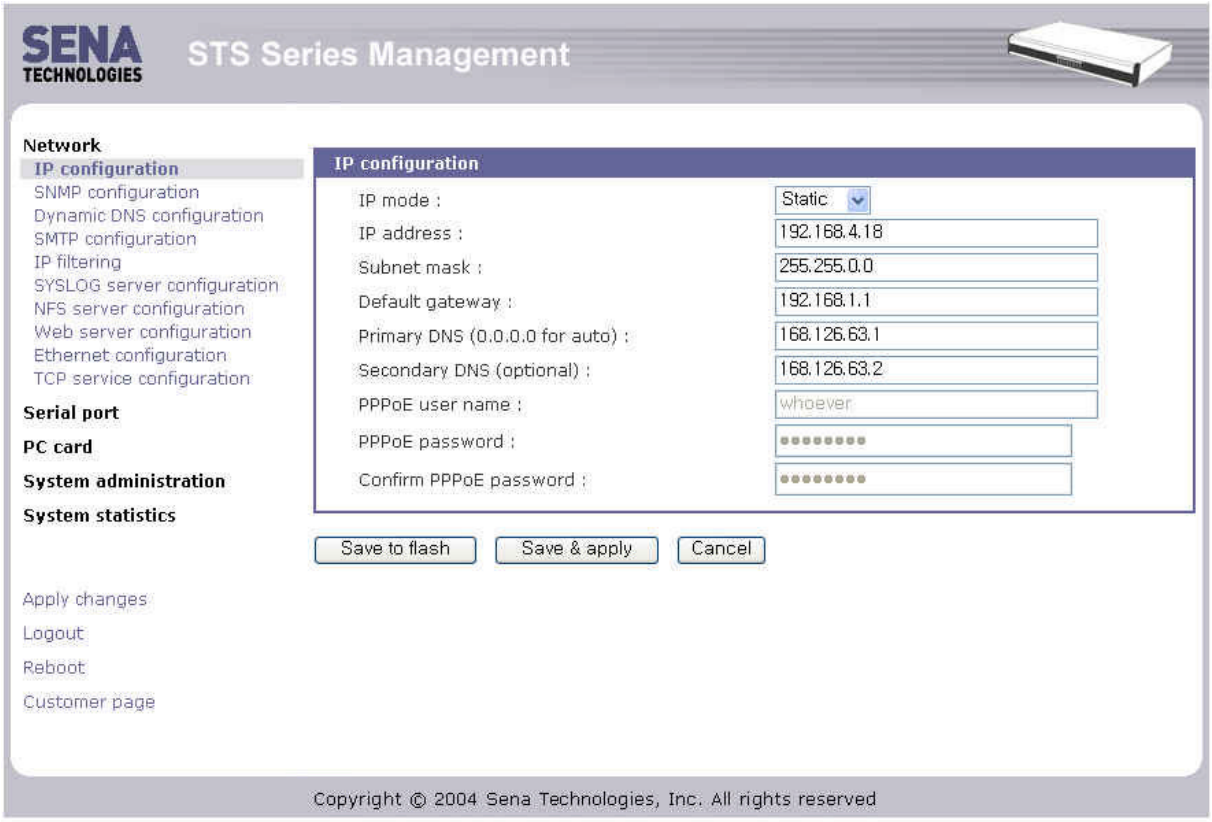


Figure 2-11 The STS Series web management screen

3. Network Configuration

3.1. IP Configuration

The STS Series requires a valid IP address to operate within the user's network environment. If the IP address is not readily available, contact the system administrator to obtain a valid IP address for the STS Series. Please note that the STS Series requires a unique IP address to connect to the user's network.

The users may choose one of three Internet protocols in setting up the STS Series IP address: i.e.,

- **Static IP**
- **DHCP** (Dynamic Host Configuration Protocol)
- **PPPoE** (Point-to-Point Protocol over Ethernet)

The STS Series is initially defaulted to **STATIC** mode, with a static IP address of **192.168.161.5**. *Table 3-1* shows the configuration parameters for all three IP configurations. *Figure 3-1* shows the actual web-based GUI to change the user's IP configuration.

Table 3-1 IP configuration Parameters

Static IP	IP address
	Subnet mask
	Default gateway
	Primary DNS/ Secondary DNS
DHCP	Primary DNS/ Secondary DNS (Optional)
PPPoE	PPPoE Username
	PPPoE Password
	Primary DNS/ Secondary DNS (Optional)

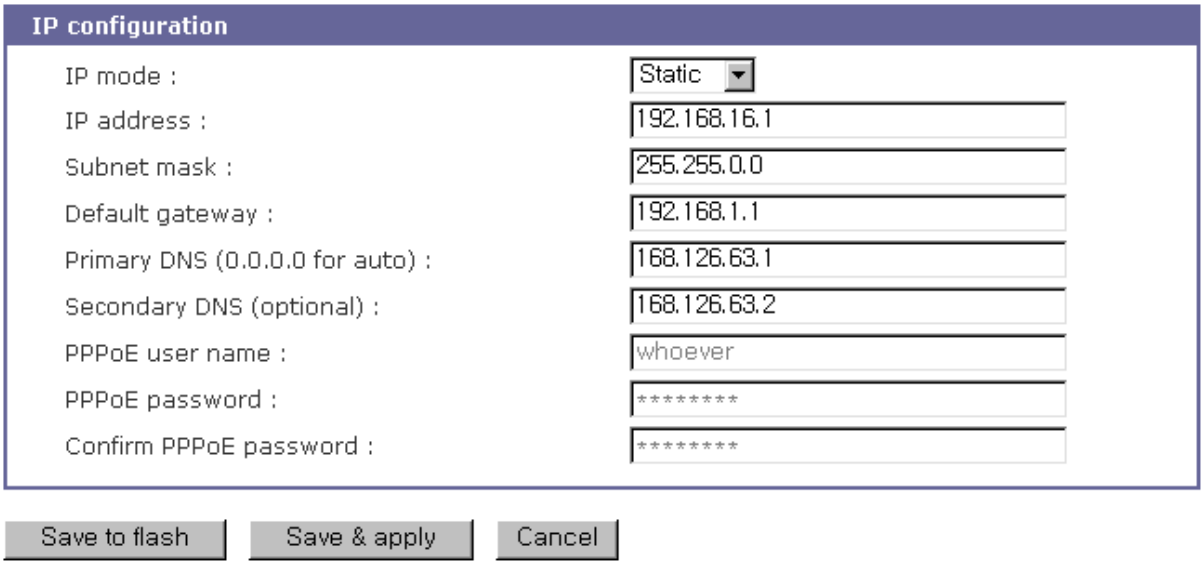


Figure 3-1 IP Configuration

3.1.1. Using a Static IP Address

When using a **Static IP** address, the user must manually specify all the configuration parameters associated with the IP address of the STS Series. These include the IP address, the network subnet mask, the gateway computer and the domain name server computers. This section will look at each of these in more detail.

Note: *The STS Series will attempt to locate all this information every time it is turned on. .*

- **IP address**

A Static IP address acts as a "static" or permanent identification number. This number is assigned to a computer to act as its location address on the network. Computers use these IP addresses to identify and talk to each other on a network. Therefore, it is imperative that the selected IP address be both unique and valid in a network environment.

Note*: 192.168.1.x will never be assigned by and ISP (Internet Service Provider). IP addresses using this form are considered private. Actual applications of the* STS Series *may require access to public network, such as the Internet. If so, a valid public IP address must be assigned to the user's computer. A public IP address is usually purchased or leased from a local ISP.*

- **Subnet mask**

A subnet represents all the network hosts in one geographic location, such as a building or local area network (LAN). The STS Series will use the subnet mask setting to verify the origin of all packets. If the desired TCP/IP host specified in the packet is in the same geographic location (on the local

network segment) as defined by the subnet mask, the STS Series will establish a direct connection. If the desired TCP/IP host specified in the packet is not identified as belonging on the local network segment, a connection is established through the given default gateway.

- **Default gateway**

A gateway is a network point that acts as a portal to another network. This point is usually the computer or computers that control traffic within a network or a local ISP (Internet service provider). The STS Series uses the IP address of the default gateway computer to communicate with hosts outside the local network environment. Refer to the network administrator for a valid gateway IP address.

- **Primary and Secondary DNS**

The DNS (Domain Name System) server is used to locate and translate the correct IP address for a requested web site address. A domain name is the web address (i.e. **www.yahoo.com**) and is usually easier to remember. The DNS server is the host that can translate such text-based domain names into the numeric IP addresses for a TCP/IP connection.

The IP address of the DNS server must be able to access the host site with the provided domain name. The STS Series provides the ability to configure the required IP addresses of both the Primary and Secondary DNS servers addresses. (The secondary DNS server is specified for use when the primary DNS server is unavailable.)

3.1.2. Using DHCP

Dynamic Host Configuration Protocol (DHCP) is a communications protocol that lets network administrators manage and automate the assignment of IP addresses centrally in an organization's network. DHCP allows the network administrator the ability to supervise and distribute IP addresses from a central point and automatically send a new IP address when a computer is plugged into a different network location.

When in static IP mode, the IP address must be entered manually at each computer. If a computer is moved to another network location, a new IP address must be assigned. DHCP allows all the parameters, including the IP address, subnet mask, gateway and DNS servers to be automatically configured when the IP address is assigned. DHCP uses a “lease” concept in assigning IP addresses to a computer. It limits the amount of time a given IP address will be valid for a computer. All the parameters required to assign an IP address are automatically configured on the DHCP server side, and each DHCP client computer receives this information when the IP address is provided at its boot-up.

Each time a computer is reset, the STS Series broadcasts a DHCP request over the network. The reply generated by the DHCP server contains the IP address, as well as the subnet mask, gateway

address, DNS servers and the “lease” time. The STS Series immediately places this information in its memory. Once the “lease” expires, the STS Series will request a renewal of the “lease” time from the DHCP server. If the DHCP server approves the request for renewal, the STS Series can continue to work with the current IP address. If the DHCP server denies the request for renewal, the STS Series will start the procedure to request a new IP address from the DHCP server.

***Note**: While in DHCP mode, all network-related parameters for the* STS Series *are to be configured automatically, including the DNS servers. If the DNS server is not automatically configured, the user may manually configure the settings by entering the primary and secondary DNS IP addresses. To force an automatic configuration of the DNS address, set the primary and secondary DNS IP addresses to 0.0.0.0 (recommended).*

A DHCP sever assigns IP addresses dynamically from an IP address pool, which is managed by the network administrator. This means that the DHCP client, i.e. the STS Series, receives a different IP address each time it boots up. The IP address should be reserved on the DHCP server side to assure that the user always knows the newly assigned STS Series address. In order to reserve the IP address in the DHCP network, the administrator needs the MAC address of the STS Series found on the label sticker at the bottom of the STS Series.

3.1.3. Using PPPoE

PPPoE (Point-to-Point Protocol over Ethernet) is a specification for connecting multiple computer users on an Ethernet LAN (local area network) to a remote site through a modem or similar device. PPPoE can be used to multiple users the ability to share ADSL, cable modem, or wireless connection to the Internet.

To use the STS Series in PPPoE mode, users require a PPPoE account and the necessary equipment for PPPoE access (i.e. an ADSL modem). Since the STS Series provides a PPPoE protocol, it can access the remote host on the Internet over an ADSL connection. The user will have to set up the user name and password of the PPPoE account for the STS Series.

The STS Series negotiates the PPPoE connection with the PPPoE server whenever it boots up. During the negotiation, the STS Series receives the information required for an Internet connection, such as the IP address, gateway, subnet mask and DNS servers. If the connection is established, the STS Series will maintain the connection for as long as possible. If the connection is terminated, the STS Series will attempt to make a new PPPoE connection by requesting a new connection.

***Note**: While in PPPoE mode, all network-related parameters for the STS Series are to be configured automatically, including the DNS servers. If the DNS server is not automatically configured, the user may manually configure the settings by entering the primary and secondary DNS IP addresses. To*

force an automatic configuration of the DNS address, set the primary and secondary DNS IP addresses to 0.0.0.0 (recommended).

3.2. SNMP Configurations

The STS Series has the SNMP (Simple Network Management Protocol) agent supporting SNMP v1 and v2 protocols. Network managers like NMS or SNMP Browser can exchange information with STS Series, as well as access required functionality.

SNMP protocols include GET, SET, GET–Next, and TRAPs. With these functions, a manager can be notified of significant events (TRAPs), query a device for more information (GET), and make changes to the device state (SET). SNMPv2 adds a GET–Bulk function for retrieving tables of information and security functions.

With the SNMP configuration panel, the user can configure MIB-II System objects, access control settings and TRAP receiver settings. The manager configured in this menu can perform both information exchange and action control. *Figure 3-2* shows a SNMP configuration screen via a web interface.

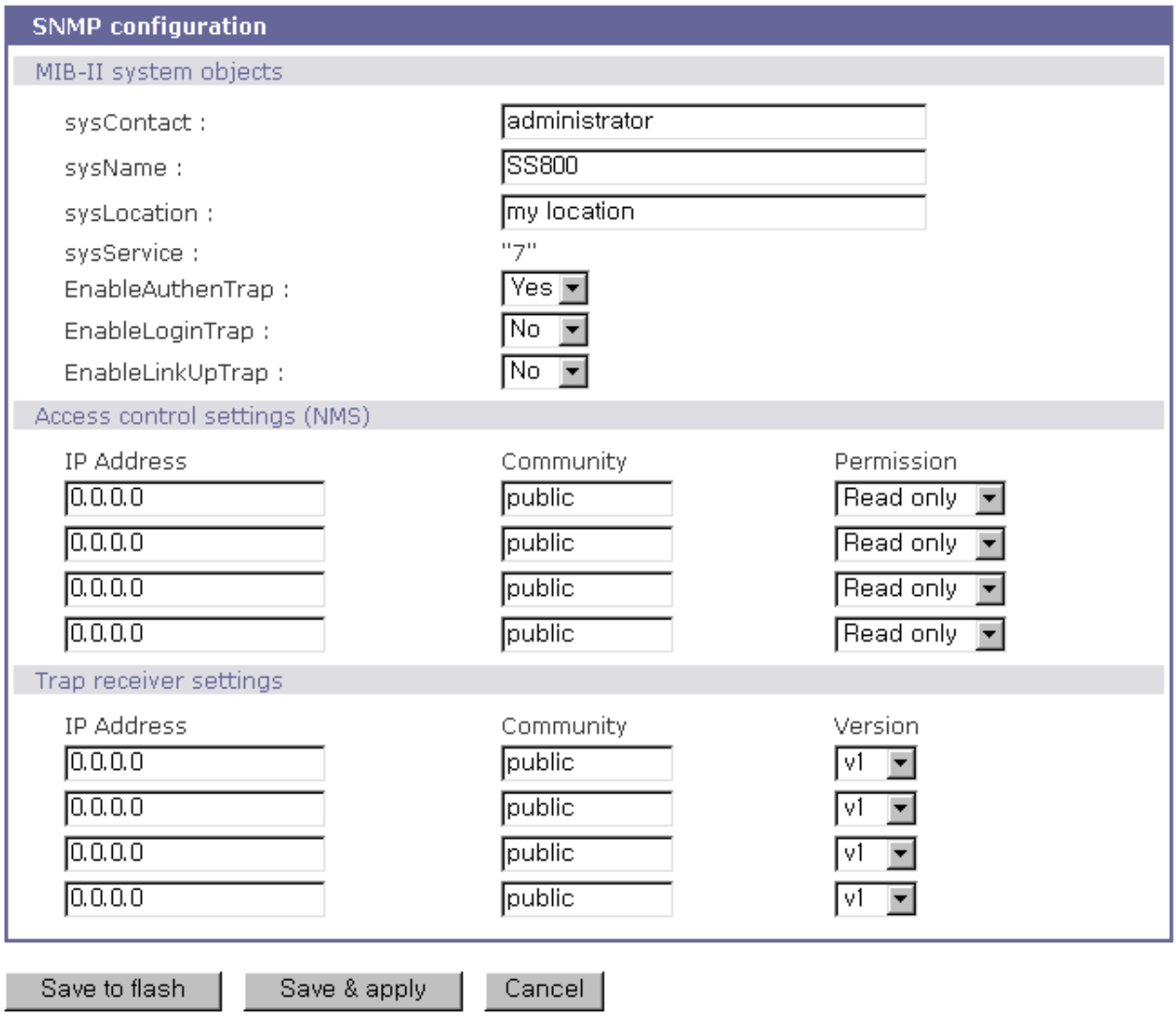


Figure 3-2 SNMP Configuration

3.2.1. MIB-II System objects Configuration

MIB–II System objects configuration sets the System Contact, Name, Location, and Authentication-failure traps used by the SNMP agent of the STS Series. These settings provide the values used for the MIB-II sysName, sysContact, sysLocation, sysService and enableAuthenTrap.

Brief descriptions of each object are as follows,

- sysContact: Identification of the contact person for the managed system (STS Series), and a description of how to contact the person.
- sysName: Name used to identify the system. By convention, this is the fully qualified domain name of the node.
- sysLocation: The physical location of the system (e.g., Room 384, Operations Lab, etc.).
- sysService(Read Only) : A series of values, separated by commas, that indicate the set of

services that the system provides. By default, STS Series only supports an Application(7) service level.

- EnableAuthenTrap: Indicates whether the SNMP agent process is permitted to generate authentication-failure traps. The value of this object overrides any configuration information; as such, it provides a means whereby all authentication-failure traps may be disabled..
- EnableLinkUpTraps: Indicates whether the SNMP agent process is permitted to generate Ethernet-link traps
- EnableLoginTrap: Indicates whether the SNMP agent process is permitted to generate system login traps.

If users need support for adding or modifying MIBs, please contact Sena technical support.

For more information about the MIBs and SNMP, see the RFCs 1066, 1067, 1098, 1317, 1318 and 1213.

3.2.2. Access Control Configuration

Access Control defines accessibility of managers to the STS Series SNMP agent. Only the manager set in this menu can access STS Series SNMP agent to exchange information and control actions. If there is no specified IP address (all IP address are defaulted to 0.0.0.0), a manager from any host can access the STS Series SNMP agent.

3.2.3. Trap Receiver Configuration

The Trap receiver defines managers, which can be notified of significant events(TRAP) from the STS Series SNMP agent.

3.2.4. Management using SNMP

The STS Series can be managed through the SNMP protocol using NMS (Network Management System) or SNMP Browser. Before using the NMS or SNMP Browser, the user must set the access control configuration properly so that the STS Series permits host access where the NMS or SNMP Browser is executed. *Figure 3-3* shows a screen shot of a typical SNMP browser with MIB-II OIDs of the STS Series SNMP agent.

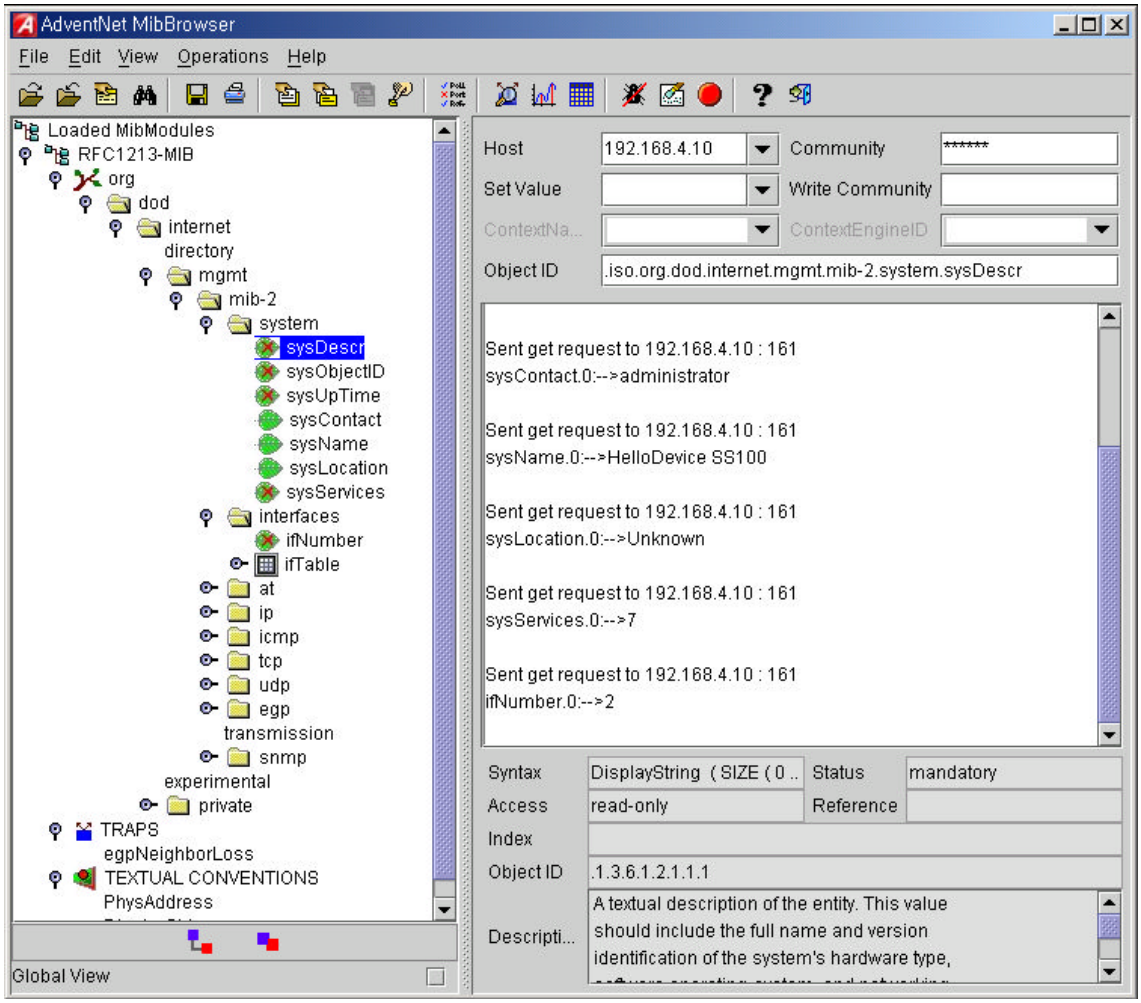


Figure 3-3 Browsing MIB-II OIDs of STS Series SNMP agent using SNMP Browser (AdventNet MibBrowser)

3.3. Dynamic DNS Configuration

When users connect the STS Series to a DSL line or use a DHCP configuration, the IP address might be changed whenever it reconnects to the network. It can therefore be very difficult to post all related contacts for each new IP address. In addition, if the administrator only has access through the remote console, there is no way to know if an IP address has changed, or what the new IP address is.

A Dynamic DNS service is provided by various ISPs or organizations to deal with the above issue. By using the Dynamic DNS service, users can access the STS Series through the hostname registered in the Dynamic DNS Server regardless of any IP address change.

By default, the STS Series only supports Dynamic DNS service offered at Dynamic DNS Network Services, LLC (www.dyndns.org). Contact Sena technical support for issues regarding other Dynamic DNS service providers.

To use the Dynamic DNS service provided by Dynamic DNS Network Services, the user must set up an account in their Members' NIC (Network Information Center - http://members.dyndns. org). The

user may then add a new Dynamic DNS Host link after logging in to their Dynamic DNS Network Services Members NIC.

After enabling the Dynamic DNS service in the Dynamic DNS Configuration menu, the user must enter the registered Domain Name, User Name, and Password. After applying the configuration change, users can access the STS Series using only the Domain Name.

Figure 3-4 shows the Dynamic DNS configuration web interface.

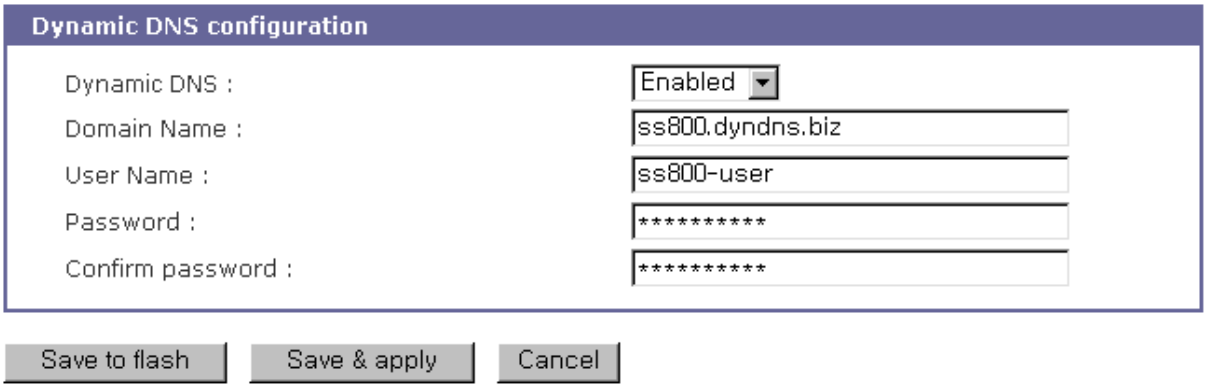


Figure 3-4 Dynamic DNS Configuration

3.4. SMTP Configuration

The STS Series can send an email notification when the number of system log messages reaches to certain value and/or when an alarm message is created due to an issue with serial port data. The user must configure a valid SMTP server send these automatically generated emails. The STS Series supports three SMTP server types:

- SMTP without authentication
- SMTP with authentication
- POP-before-SMTP

These examples can be seen in *Figure 3-6*. Required parameters for each SMTP configuration include:

- SMTP server IP address
- SMTP user name
- SMTP user password
- Device mail address

The device mail address specifies the sender's email address for all log and alarm delivery emails. SMTP servers often check only the sender's host domain name of the email address for validity.

Consequently, the email address set for the device can use an arbitrary username with a registered hostname (i.e. *arbitrary_user@yahoo.com* or *anybody@sena.com*).

The SMTP user name and SMTP user password are required when either SMTP with authentication or POP-before-SMTP mode is selected.

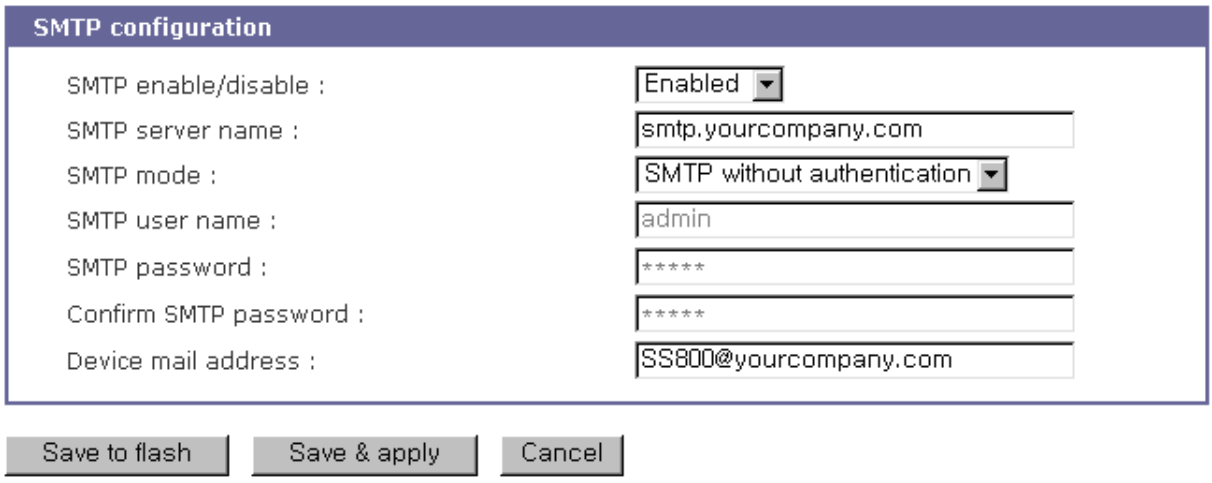


Figure 3-5 SMTP Configurations

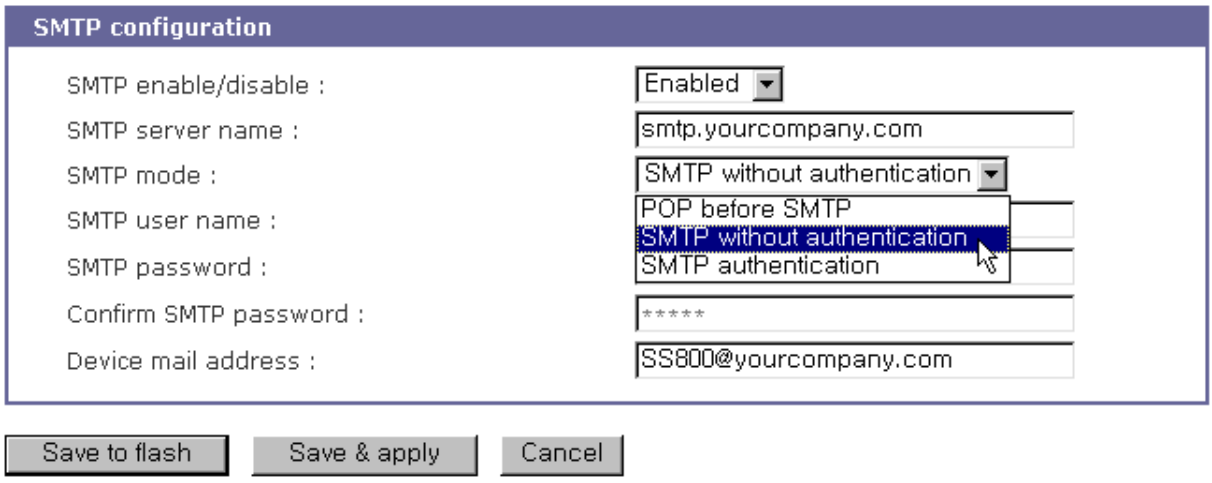


Figure 3-6 SMTP mode selection in SMTP configuration

3.5. IP Filtering

The STS Series prevents unauthorized access using either an IP address based filtering method or through the management web page of the STS Series. The users can allow one of the following scenarios by changing the parameter settings:

- Only one host of a specific IP address can access the STS Series
- Hosts on a specific subnet can access the STS Series
- Any host can access the STS Series

The IP filtering feature for access to Telnet console, SSH console or Web server may be enabled or disabled. The factory default of the filtering feature is “Enabled”.

The user may allow a host or a group of hosts to access the STS Series for configuration. The user must then enter the IP address and subnet of access. Any user on a remote host must stay in the specified subnet boundary to have the configuration access.

To allow only a specific host to have configuration access to the STS Series, enter the IP address of the specific host and just give 255.255.255.255 for the subnet

To allow any hosts to have access to the STS Series, give 0.0.0.0 for both of the IP address and subnet. Refer to *Table 3-2* for more details. The device’s default settings for allowed remote hosts for configuration is “Any”.

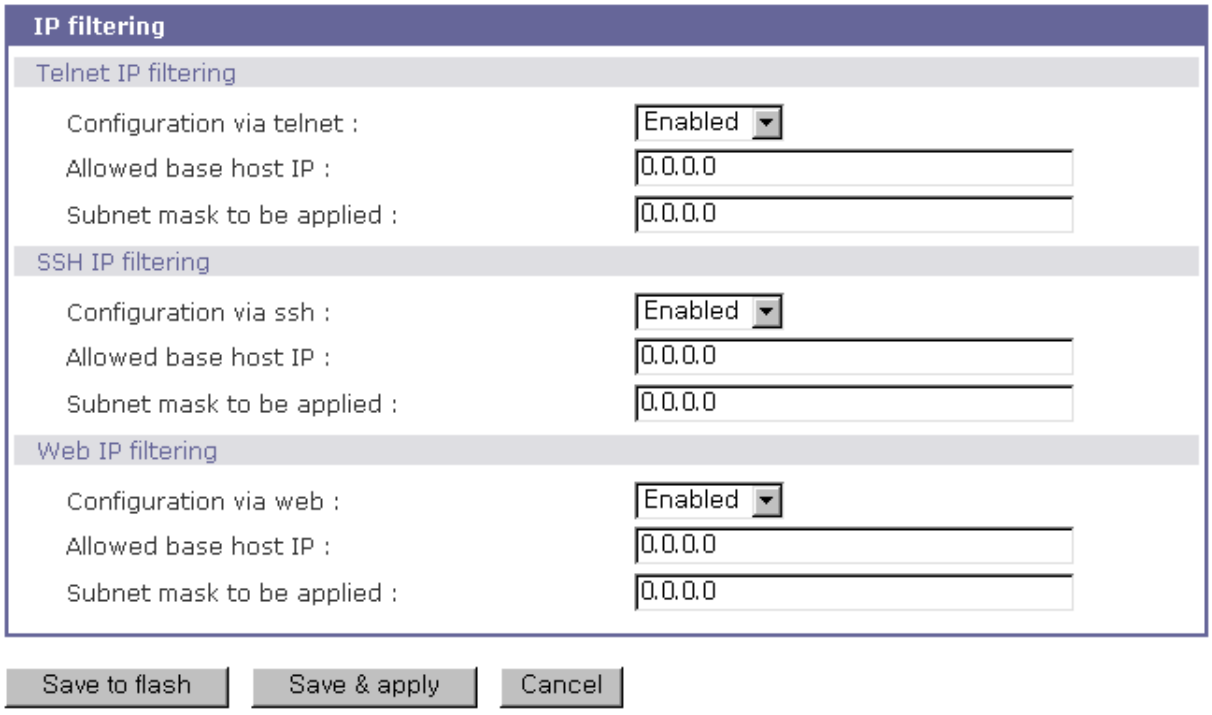


Figure 3-7 IP filtering Configuration

Table 3-2 Input examples of allowed remote hosts

Allowable Hosts	Input format	
	Base Host IP address	Subnet mask
Any host	0.0.0.0	0.0.0.0
192.168.1.120	192.168.1.120	255.255.255.255
192.168.1.1 ~ 192.168.1.254	192.168.1.0	255.255.255.0
192.168.0.1 ~ 192.168.255.254	192.168.0.0	255.255.0.0
192.168.1.1 ~ 192.168.1.126	192.168.1.0	255.255.255.128
192.168.1.129 ~ 192.168.1.254	192.168.1.128	255.255.255.128

Web access also uses the IP filtering function, which can be enabled or disabled. The factory default setting is “Enabled”. When enabled, the user can specify a web host or hosts allowed to access the STS Series for configuration.

3.6. SYSLOG server configuration

The STS Series supports a remote message logging service, SYSLOG service for the system and port data logging. To use the remote SYSLOG service, the user must specify the SYSLOG server’s IP address and the facility to be used. *Figure 3-8* shows the SYSLOG server configuration page of the supplied Web interface.

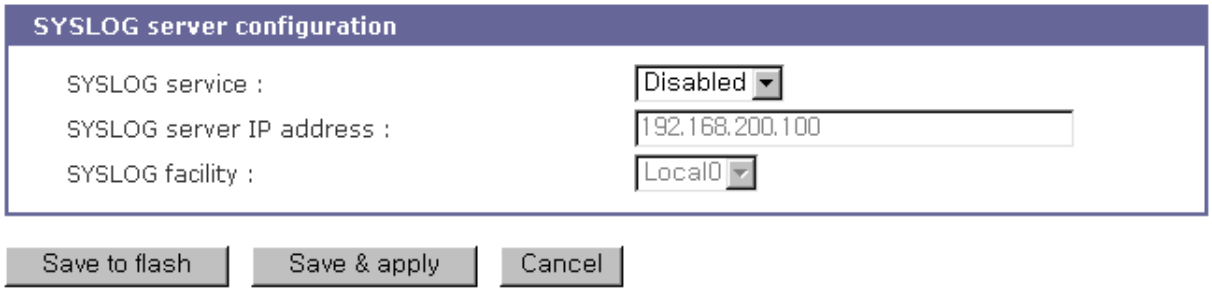


Figure 3-8 SYSLOG server configuration

To receive log messages from the STS Series, the SYSLOG server must be configured as “remote reception allowed”. If there is a firewall between the STS Series and the SYSLOG server, there must be a rule that allows all outgoing and incoming UDP packets to travel across the firewall.

The STS Series supports SYSLOG facilities from `local0` to `local7`. The user can employ these facilities to save messages from the STS Series separately in the SYSLOG server.

If the SYSLOG service is enabled and the SYSLOG server configuration is properly set up, the user may configure the storage location for the system log or port data log of the STS Series as SYSLOG server. For more information about the configuration of port/system log storage location, please refer to section, *4.2.11 Port Logging* and *6.2 System Logging.*

3.7. NFS server configuration

The STS Series supports NFS **(Network File System)** service for system or port data logging functions. To use this service, the user must specify the IP address of a NFS server and the mounting path on the NFS server. *Figure 3-9* shows the web based NFS server configuration page.

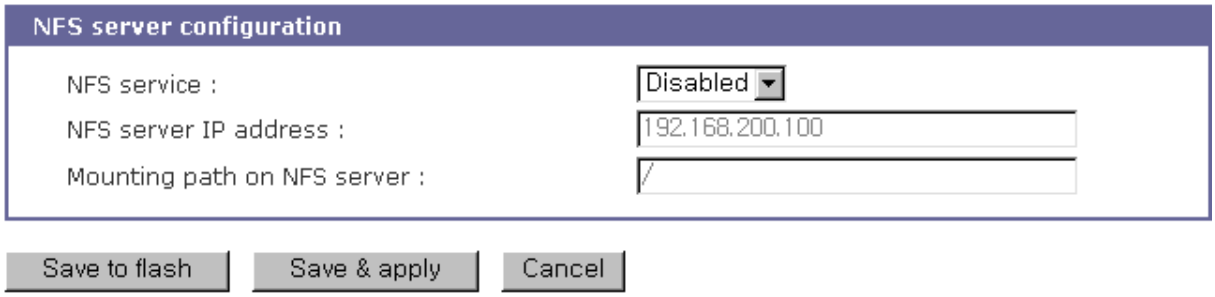


Figure 3-9 NFS server configuration

To store the STS Series log data to the NFS server, the NFS server must be configured as “read and write allowed”. If there is a firewall between the STS Series and the NFS server, there must be a rule that allows all outgoing and incoming packets to travel across the firewall.

If the NFS service is enabled and the NFS server configuration is properly set up, the user may configure the storage location for the system log or port data log of the If there is a firewall between the STS Series and the SYSLOG server, there must be a rule that allows all outgoing and incoming UDP packets to travel across the STS Series as the NFS server. For more information about the configuration of the port/system log storage location, please refer to section, *4.2.11 Port Logging* and *6.2 System Logging*.

3.8. Ethernet configuration

The STS Series supports several types of Ethernet modes:

- Auto Negotiation
- 100 BaseT Half Duplex
- 100 BaseT Full Duplex
- 10 BaseT Half Duplex
- 10 BaseT Full Duplex

After changing the Ethernet mode, the user must reboot the system. The factory default value of the Ethernet mode is Auto Negotiation. With most network environments, Auto Negotiation mode works fine and is recommended. Invalid Ethernet mode configuration may make the STS Series not work in the network environment.

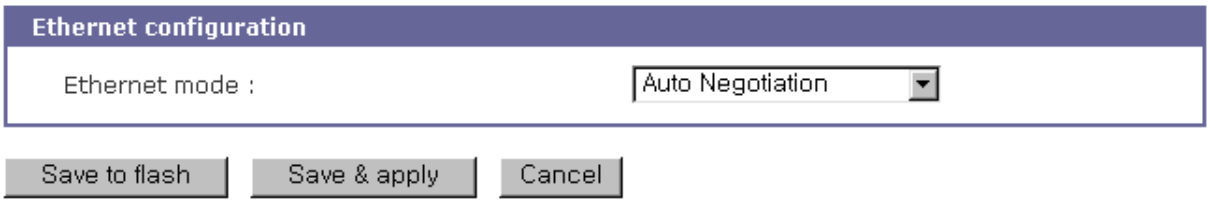


Figure 3-10 Ethernet mode configuration

3.9. Web server configuration

The Web server supports both HTTP and HTTPS (HTTP over SSL) services simultaneously. The user can opt to enable or disable each individually. *Figure 3-11* shows the Web server configuration page.

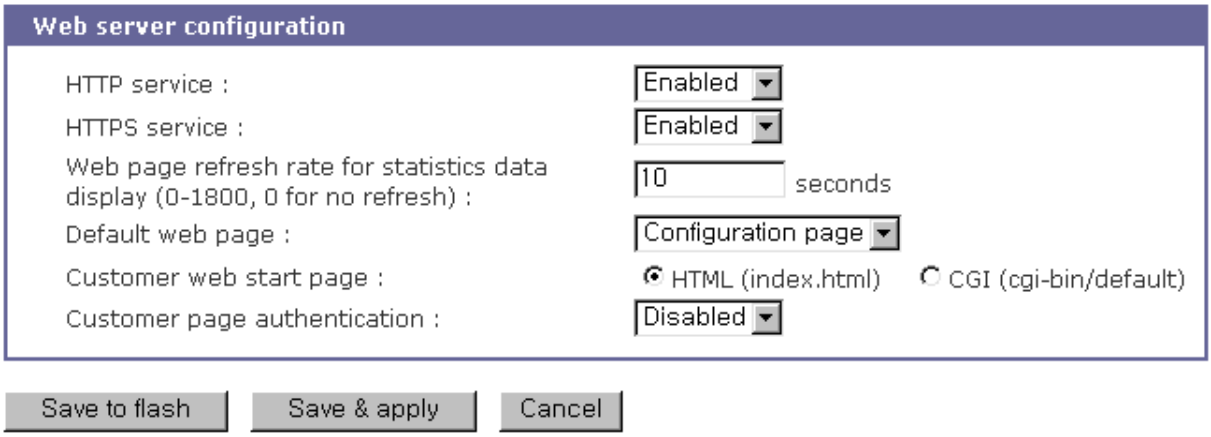


Figure 3-11 Web server configurations

The Web page refresh rate can be also adjusted in this configuration page. The refresh rate is only applicable to the system statistics pages, such as network interfaces, serial ports, IP, ICMP, TCP and UDP. Other pages in the Web interface are not refreshed automatically. For more information about the system statistics, please refer to section *7 System Statistics.*

On this configuration menu page, users can select default web page that will pop up after user logs in to the Web UI. Factory default is *Configuration page* and user can changes this to *Customer page.* After selecting Default web page as *Customer page,* user can set *Customer web start page* as one of HTML(index.html) or CGI(default) which are located under /usr2 directory. User can change or modify these files for his own purpose. For anonymous access to the customer page without user ID and password, disable the customer page authentication feature. For more information about customizing user web page, please refer to 9.3 *User defined web pages* section.

3.10. TCP service configuration

If a TCP session is established between two hosts, it should be closed (normally or abnormally) by either of the hosts to prevent the lock-up of the corresponding TCP port. To prevent this type of lock-up situation, the STS Series provides a TCP "keep-alive" feature. The STS Series will send packets back and forth through the network periodically to confirm that the network is still alive. The corresponding TCP session is closed automatically if there's no response from the remote host.

To use the TCP "keep-alive" feature with the STS Series, the users should configure three parameters as follows:

- **TCP keep-alive time:**
  This represents the time interval between the last data transmission and keep-alive packet submissions by the STS Series. These "keep-alive" messages are sent to the remote host to confirm that the session is still open. The default time value is 15 sec.
- **TCP "keep-alive" probes:**
  This represents how many "keep-alive" probes will be sent to the remote host, until it decides that the connection is dead. Multiplied with the "TCP 'keep-alive' intervals", this gives the time that a link is forced to close after a "keep-alive" packet has been sent for the first time. The default is 3 times
- **TCP keep-alive intervals:**
  This represents the waiting period until a "keep-alive" packet is retransmitted due to no acknowledgement by the original Chinatown. The default value is 5 seconds.

By default, the STS Series will send the keep-alive packets 3 times with 5 seconds interval after 15 seconds have elapsed since the time when there's no data transmitted back and forth.

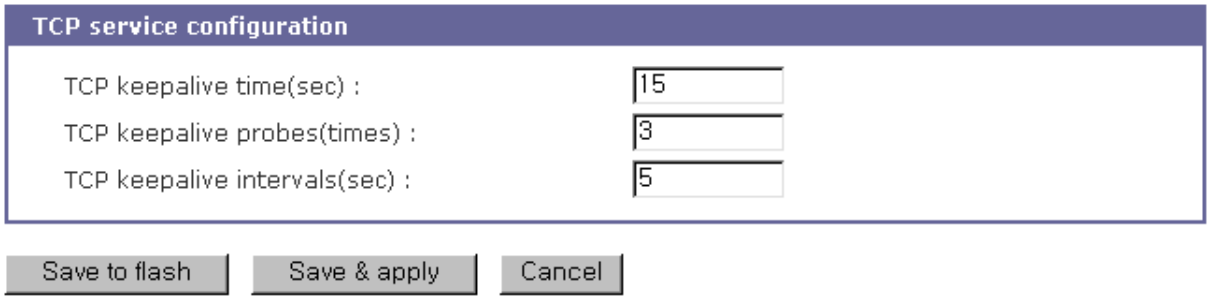


Figure 3-12 TCP keep-alive configuration

4. Serial Port Configuration

4.1. Overview

The serial port configuration capability allows the user to configure the host mode of each port, serial communication parameters, cryptography, port logging parameters and other related parameters.

The serial port's host mode can be set as any of the following:

- **TCP** :
  The STS Series operates as a TCP server and client. If the connection is not established, it accepts all incoming connections from any registered remote hosts and connects to the registered remote hosts if there is any data from the serial devices. Otherwise, it will send data back and forth. In summary, the STS Series will work as if it is virtually connected to the remote host.
- **UDP** :
  The UDP mode operation is similar to that of TCP mode except that it is based on UDP protocol.
- **Modem emulation** :
  Select this mode when the serial device already supports modem AT commands or users want to perform the session control by using AT commands. Only TCP session is supported.
- **Virtual COM** :
  This feature will be added later.

With the **port-logging** feature while in console server mode, the data sent through the serial port is transferred to **MEMORY, SYSLOG server, NFS server's storage** or an **ATA/IDE fixed disk card** using a PC Card slot. The user can also define keywords for each serial port that will trigger an email or SNMP trap notification. This will enable the user to monitor the data from the attached device.

Using **MEMORY** to store data will result in loss of all information when the STS Series is turned off. Use the **NFS server** or **ATA/IDE fixed disk card** to preserve the serial port log data.

The serial ports can be configured individually or all at once. *Table 4-1* summarizes the configuration parameters related to the serial port configuration.

Table 4-1 Serial port configuration parameters

All serial ports setting **Or** **Individual serial port**	Port Enable/Disable		
	Port title		
	Apply all port settings (Individual serial port setting only)		
	Host mode	**TCP**	TCP listening port
			Telnet protocol
			Max allowed connection
			Cyclic connection

serial port setting #1~#8(1/4)			Inactivity timeout (0 for unlimited)
		UDP	UDP listening port
			Max allowed connection
			Accept UDP datagram from unlisted remote host or not
			Send to recent unlisted remote host or not
			Inactivity timeout (0 for unlimited)
		Modem emulation	
	Remote host[1]	Add or Edit a remote host[2] Primary host address Primary host port Secondary host address Secondary host port	
		Remove a remote host	
	Port IP filtering[3]	Allowed host IP	
		Subnet mask to be applied	
	Cryptography[4]	Encryption method None/SSLv2/SSLv3/SSLv3 rollback to v2/ TLSv1/3DES/RC4	
		Cipher suite selection	
		Verify client (server mode only)	
		Verify certificate chain depth	
		Check the certificate CN	
	Filter application	Filter application path	
		Filter application arguments	
	Serial Port Parameters	Type	
		Baud rate	
		Data bits	
		Parity	
		Stop bits	
		Flow control	
		DTR behavior	
		DSR behavior	
		Inter-character timeout (ms)	
	Modem	Enable/Disable modem	
		Modem init-string	
		DCD behavior	
	Port logging	Enable/Disable Port logging	
		Port log storage location	
		Port log buffer size	
		Display port log	
	Port event handling	Enable/Disable port event handling	
		Notification interval	
		Email notification	Enable/Disable Email notification
			Title of Email
			Recipient's Email address
		SNMP notification	Enable/Disable SNMP notification
			Title of SNMP trap
			SNMP trap receiver's IP address
			SNMP trap community
			SNMP trap version

[1] TCP/UDP mode only.
[2] A secondary remote host is available for connection-fail backup in TCP mode
[3] TCP/UDP mode only.
[4] TCP mode only

		Add/Edit a keyword Keyword string Email notification SNMP trap notification Port command
		Remove a keyword

Figure 4-1 shows the web-based serial port configuration screen. This serial port configuration main screen summarizes port information. In this summary page, user can find how host mode, encryption option, local port number and serial port parameters are configured at a time. In this page, the meaning of string on Host mode column is as follows,

Host mode	Description		
	Mode[5]	Encryption[6]	Telnet[7]
TCP	TCP	Disabled	Disabled
TCPs	TCP	Enabled	Disabled
TEL	TCP	Disabled	Enabled
TELs	TCP	Enabled	Enabled
UDP	UDP	*[8]	*
Modem Emulation	Modem Emulation	*	*

To select and configure a serial port individually, click the port number or title. To configure all of the serial ports at once, click [All] or [Port Title], located below the [All port configuration] label.

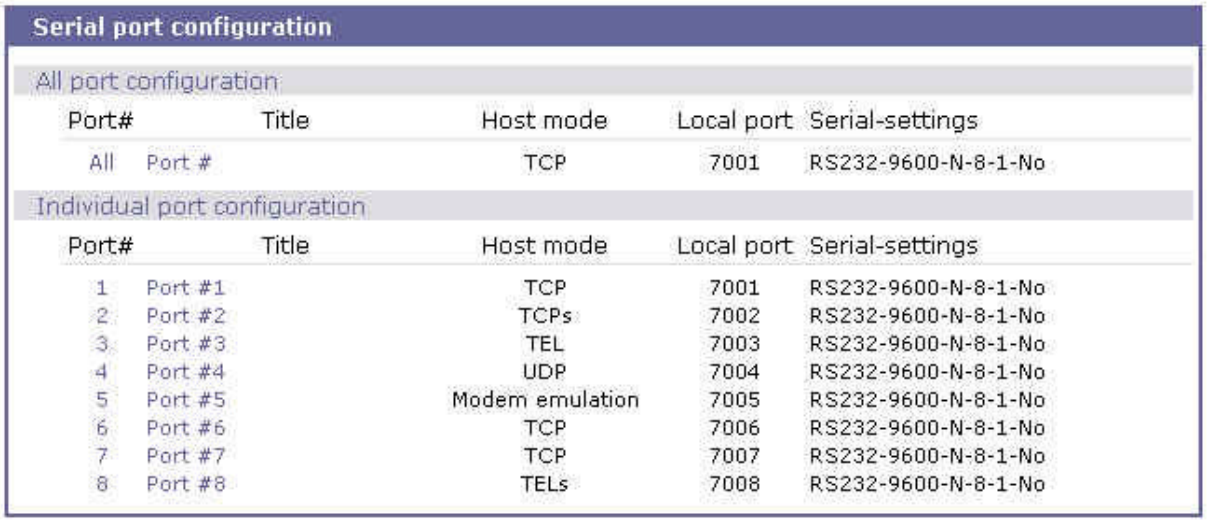


Figure 4-1 Serial port configuration main screen

[5] See section *4.2.4 Host Mode Configuration*
[6] See section *4.2.7 Cryptography configuration*
[7] See section *4.2.4.1 TCP mode*
[8] Not support

4.2. Individual Port Configuration

The STS Series allows serial ports to be configured either individually or all at once. The parameters for both **individual** and **all port configurations** are similar.

Individual Port Configurations are classified into nine (9) groups:

1. Port enable/disable
2. Port title
3. Apply all port settings
4. Host mode
5. Remote host: A*vailable only when the host mode is set to TCP or UDP mode*
6. Port IP filtering: A*vailable only when the host mode is set to TCP or UDP mode*
7. Cryptography: A*vailable only when the host mode is set to TCP mode and Modem Emulation mode*
8. Filter application
9. Serial port parameters
10. Modem configuration
11. Port logging
12. Port event handling: A*vailable only when the port-logging feature of the port is enabled*

Users can switch to another serial port configuration screen conveniently using the [--- Move to ---] list box at the right upper side of the individual port configuration screen.

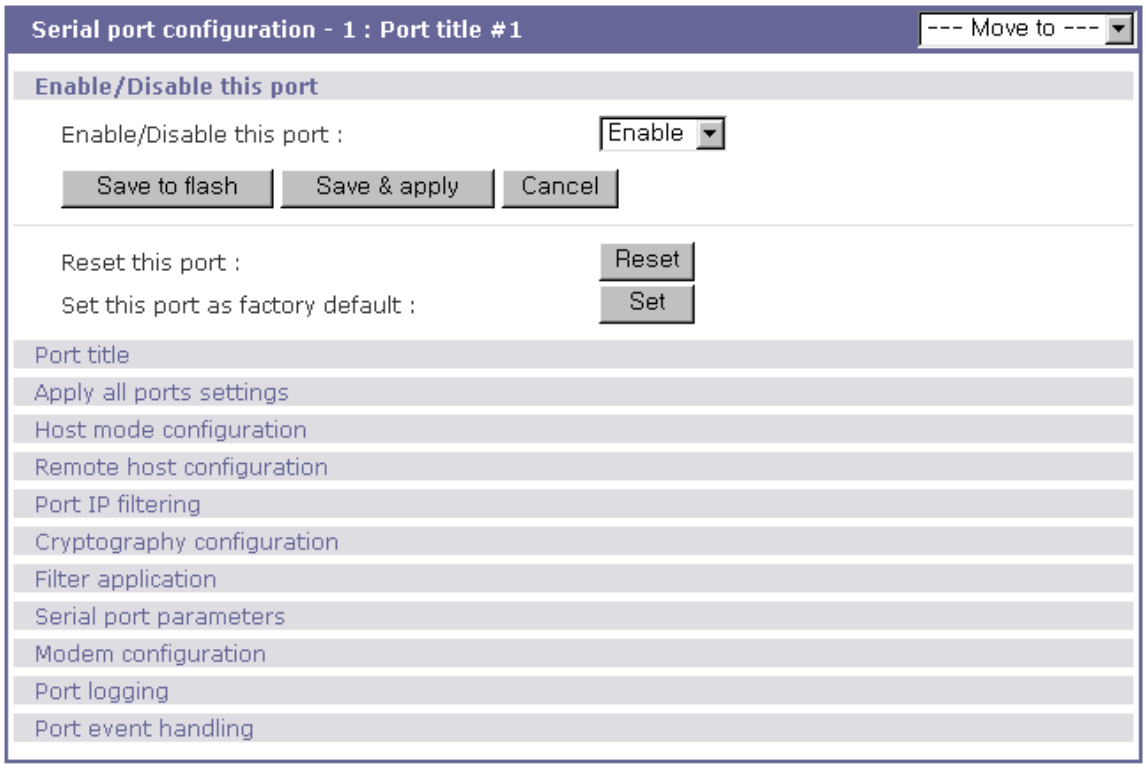


Figure 4-2 Serial port enable/disable

4.2.1. Port Enable/Disable

Each serial port can be enabled or disabled. If a serial port is disabled, users cannot access the serial port. *Figure 4-2* shows the serial port enable/disable screen.

By clicking on the [Reset] button, users can reset a stuck or deadlocked serial port. Click on the [Set] button to set the port as factory default.

4.2.2. Port Title

Users can enter descriptive information for each port based on the device attached to it. This can include the device type, vendor, and/or location. The port title is helpful in the configuration process,

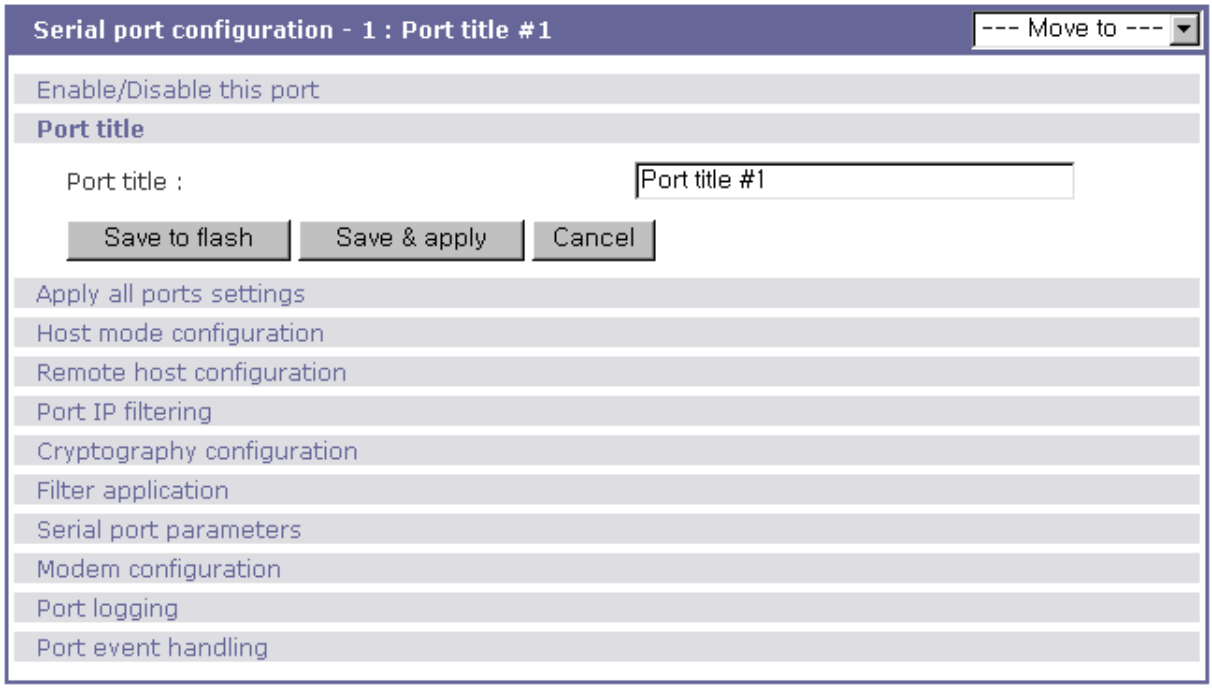


Figure 4-3 Port title configuration

4.2.3. Apply All Port Settings

To prevent the possibility of the user inadvertently selecting to change all port settings at the same time, the STS Series provides the ability to enable or disable this function at an individual serial port level. Changes made when using the “change all port parameters at once” function will not be applied to an individual serial port if the function has been disabled (See *Figure 4-4*. This shows the **[apply all port setting]** configuration screen.

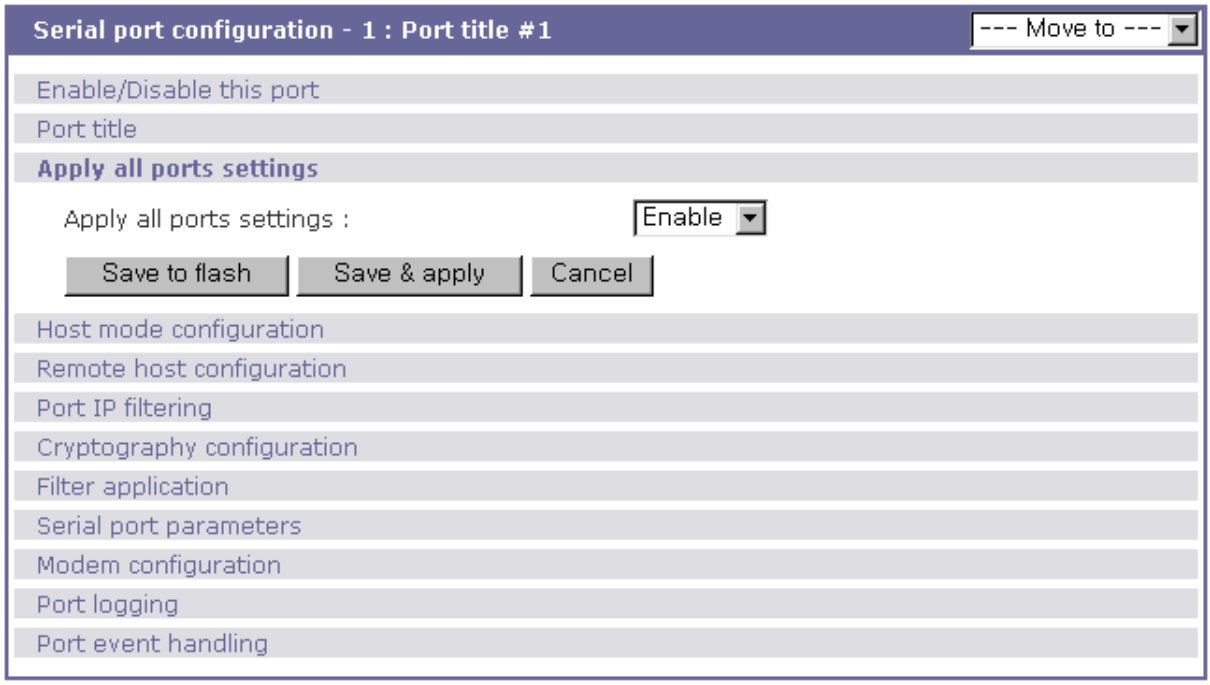


Figure 4-4 Apply all port setting configuration.

4.2.4. Host Mode Configuration

The STS Series operating mode is called the "host mode." Three host modes are available: **TCP mode**, **UDP mode, Modem emulation mode**.

TCP mode

The STS Series works as both TCP server and client. This mode works for most applications, since it will transfer the data either from serial port or from TCP port. If there is no connection established on a TCP port, the TCP port accepts a connection request from any registered remote hosts and relays the transmitted data to the coupled serial port. If there is any data from the serial port, it connects to the registered remote hosts and redirects the data.

UDP mode

The UDP mode operation is similar to that of TCP mode except that it utilizes UDP protocol

Modem emulation mode

Select this mode when the serial device already supports modem AT commands or users want to perform the session control by using AT commands. Only TCP session is supported.

Figure 4-5 shows the main workspace screen for the host mode configuration.

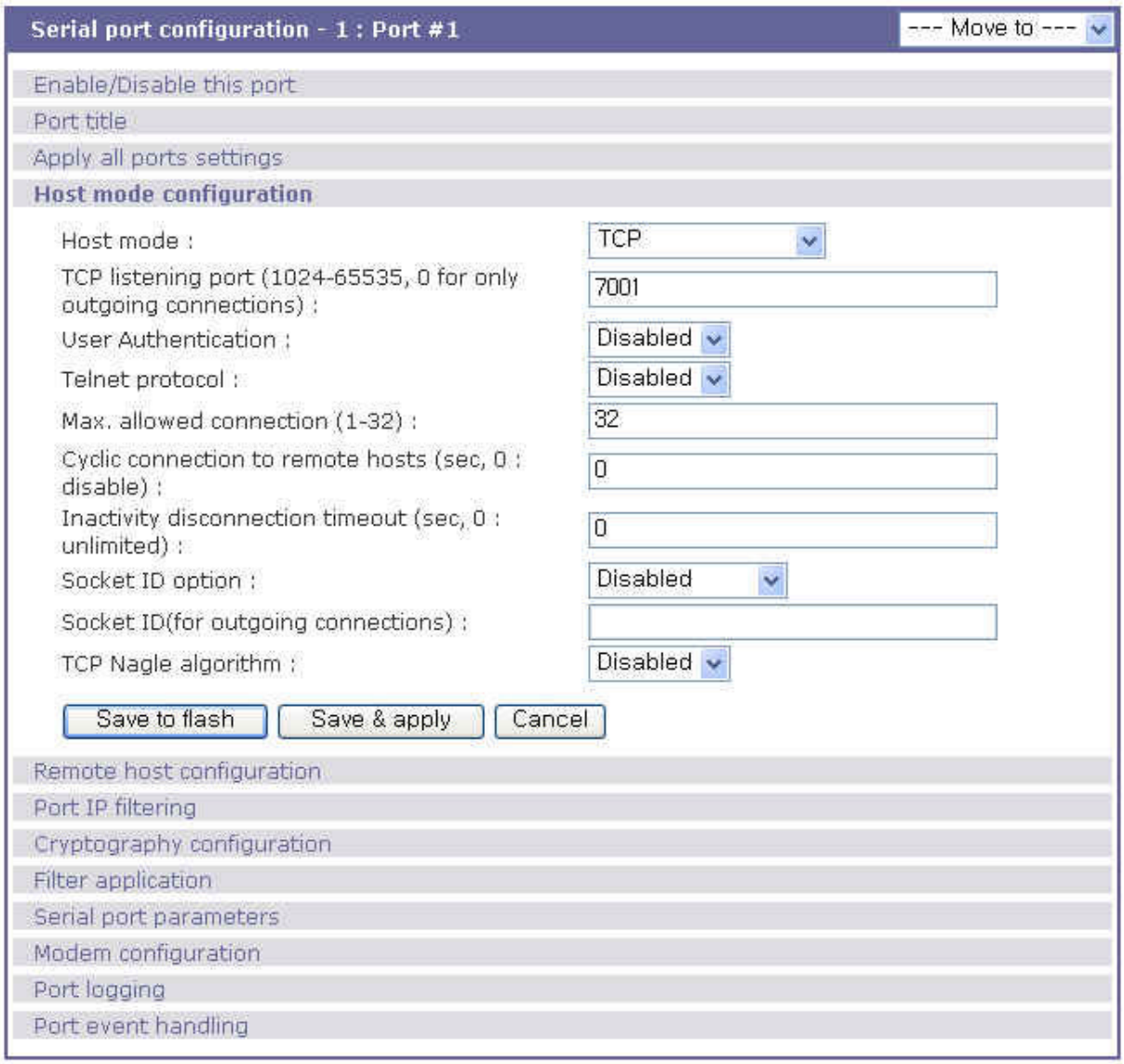


Figure 4-5 Host mode configuration

4.2.4.1. TCP mode

For easier understanding of TCP modes, a simplified **State Transition Diagram** is often used. And to help users understand the diagram, the TCP state of the STS Series is briefly described as follows.

- [Listen]

It represents “a waiting for a connection request from any registered remote host”. It is a default start-up mode when it is set as TCP mode.

- [Closed]

It means “no connection state”. If the data transfer between a remote host and the STS Series is completed, the state is changed to this state as a result that either of the remote host or the STS Series sent a disconnection request. After this, the state is automatically changed to [Listen] mode.

- [Sync-Received]

The state is changed from [Listen] to [Sync-Received] if one of the remote hosts has sent a connection request. If the STS Series accepts the request, the state is changed into [Established].

- [Sync-Sent]

If the STS Series has sent a connection request to a remote host, the state is changed from [Closed] to [Sync-Sent]. This state is maintained until the remote host accepts the connection request.

- [Established]

It represents “an open connection”. If one of the hosts, the remote host or the STS Series, accepts a connection request from the other, the connection is opened and state is changed into [Established].

- [Data]

When it is in [Established] state, data from a host will be transferred to the other one. For easier understanding of the TCP session operation, we called the state as [Data] state when actual data transfer is performed. Actually, the [Data] mode is a part of [Established] state as is described in the RFC 793 [Transmission Control Protocol]. This is a normal state for the data transfer phase of the connection.

The STS Series works as either TCP server or client according to the situation. This will be the typical mode for most applications, since it will transfer the data either from serial port or from TCP port. The default TCP state is [Listen] which is the same as that of *TCP server* mode.

1) Typical State Transition

[Listen] - -> [Sync-Received] - -> [Established] - -> [Data] - -> [Closed] - -> [Listen]
Or
[Listen] - -> [Sync-Sent] - -> [Established] - -> [Data] - -> [Closed] - -> [Listen]

The initial state is [Listen]. If there are data coming from the serial port, it will connect to the remote host as a TCP client and then transfer data through the TCP port. If there is incoming connection request from the remote host, it will accept the connection as a TCP server, and then transfer data through the serial port. Thus, users can assume that the STS Series is always connected to the specified remote host.

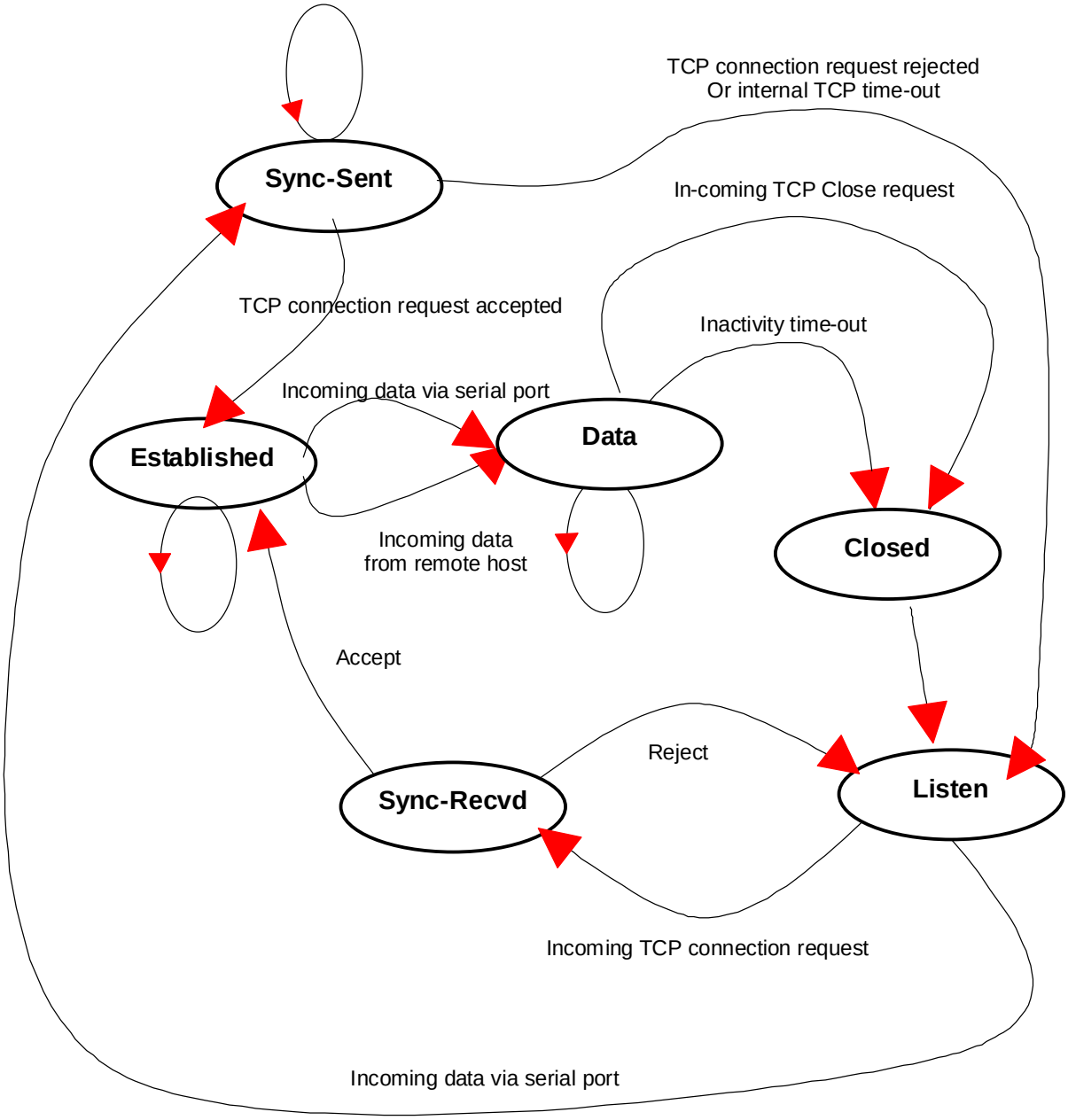


Figure 4-6 State Transition Diagram of TCP mode

2) Operations

Serial data transfer

Whenever the serial device sends data through the serial port of the STS Series, data will be accumulated on the serial port buffer of the STS Series. If the buffer is full or the time gap reaches the *inter-character timeout* (See *Options* in section 4.4 for details on *inter-character timeout*), the STS Series connect to the registered remote host(s). If a TCP session has not been established yet. If the STS Series succeeds in connecting to the remote host, the data in the serial port buffer will be transferred to the host. Otherwise, all the data stored in the buffer will be cleared.

Session disconnection

The connected session will be disconnected when the remote host sends disconnection request or when no data transfer activity is found through the serial port for certain amount of time, which is "*Inactivity timeout*" (See *Options* in section 4.4 for details on *Inactivity timeout*). All the data remained in the serial port buffer will be cleared when it is disconnected.

Connection request from remote host

All the incoming TCP connection requests will be rejected in *TCP client* mode.

3) Parameters

TCP listening port

This is the TCP port number through which remote host can connect a TCP session, and, send and receive data. Incoming connection request to the ports other than *TCP Listening Port* will be rejected. The STS Series does restrict the port number from 1024 to 65535 and if it is set as 0 only outgoing connection is permitted. (TCP server mode)

User Authentication

In TCP mode, Super Series support user authentication for port access. If this option is enabled, user should enter the user ID and password before accessing the port. (For user administration, please refer to the section *6.6 User Administration*)

Telnet protocol

In TCP mode, STS Series support Telnet Com Port Control Option (RFC2217 compliant) so that user can control serial parameters like baud rate, data bits and flow control option using his local RFC2217-compliant Telnet client program. (Please refer to section *4.2.9 Serial port parameters* for more detail information about serial parameters)

Usually this option is used with the RFC2217-compliant COM port redirector so that user can control parameters of serial ports of STS Series using his serial port application program.

For this purpose, SENA OEM version of Serial/IP from Tactical Software, LLC is bundled with STS Series. Please refer to documentations of Serial/IP for more detail information about using the COM port redirector. (Please refer to section Appendix 6Using STS Series with Serial/IP for more detail information)

Max. allowed connection

The STS Series supports multiple connections from external host(s) to a serial port up to 32. But if there are remote host connections by the *remote host list configuration* already, possible number of connection is reduced to (Max. allowed connection - remote host(s) connected already). For example, if user set *Max. allowed connection* as 32 and if there are 3 connections from STS Series to remote hosts, which are configured in the remote host list, then maximum number of

connection from external hosts to a serial port will be reduced to 29 (=32 – 3). For more detail information for *remote host list configuration*, please refer to 4.2.5 remote host list configuration section.

Cyclic Connection

If *Cyclic Connection* function is enabled, the STS Series will make an attempt to connect to the user-defined remote host(s) at a given interval even if there's no incoming serial data from the device connected to that serial port. If there is data on the remote host(s) to be sent to serial device, it can be transferred to the serial device via STS Series's serial port after the connection is established. Eventually, users can monitor the serial device periodically by making the remote host send the serial command to the STS Series whenever it is connected to the remote host. This option is useful when users need to gather the device information periodically even if the serial device does not send its data periodically. Figure 4-6 shows the State Transition Diagram of the session operations in *TCP* mode.

Inactivity Timeout

When *Inactivity Timeout* function is enabled, connection between remote host(s) and STS Series will be closed automatically if there is no data transmission during the value which is set in *Inactivity Timeout* configuration.

Socket ID

When Super Series connects remote host(s), sometimes it is needed to identify the device using the string. In this case, if user specifies specific strings in *Socket ID*, Super Series send these strings first before start the data transmission. User can specifies either Serial Numebr or specific strings up to 49 characters as Socket ID.(This option can be changed only by root user) In TCP mode, specified *Socket ID* strings are sent once at the time of establishing TCP connection.

TCP Nagle algorithm

Modern TCP implementations include a mechanism, known as the Nagle algorithm, which prevents the unnecessary transmission of a large number of small packets. This algorithm has proved useful in protecting the Internet against excessive packet loads. However, some applications suffer performance problems as a result of the traditional implementation of the Nagle algorithm.(An interaction between the Nagle algorithm and TCP's delayed acknowledgement policy can create especially severe problems, through a temporary "deadlock.") TCP Nagle algorithm can be disabled or enabled through this option.

4.2.4.2. UDP mode

The UDP mode operation is similar to that of *TCP* mode except that it is based on UDP protocol and only one pre-defined remote host is able to communicate with the STS Series. Users do not have to configure *cyclic connection,* since UDP is a connectionless protocol.

1) Operations

If a remote host sends a UDP datagram to the one of UDP *Local port* of the STS Series, STS Series first checks whether it is from one of the hosts configured on *remote host configuration.* If the remote host is one of the hosts configured on *remote host configuration*, then STS Series transfers the data through the serial port. Otherwise, the STS Series discards the incoming UDP datagram. But user can force STS Series accept all incoming UDP datagram regardless *remote host configuration* by setting *Accept UDP datagram from unlisted remote host* parameter as 'Yes'. If there is any incoming data from the serial port, the STS Series transfers the data to the remote host defined on *remote host configuration*. If the remote port is not opened, the STS Series will not transfer the data.

2) Parameters

UDP receiving port

The concept is the same as *TCP listening port*. See ***TCP mode parameters*** in the section 4.2.4.1 for details.

Max. allowed connection

The concept is the same as that of TCP communication. ***TCP mode parameters*** in the section 4.2.4.1 for details.

Accept UDP datagram from unlisted remote host

If *Accept UDP datagram from unlisted remote host* function is set as 'No', STS Series will accept only incoming UDP datagram from the remote host(s) configured on *remote host configuration*. On the contrary if *Accept UDP datagram from unlisted remote host* function is set as 'Yes', STS Series will accept all incoming UDP datagram regardless *remote host configuration.*

Send to recent unlisted remote host

If *Send to recent unlisted remote host* function is set as 'Yes', STS Series sends data to the remote host, which has connected STS Series recently. Recent unlisted remote host is a remote host, which has accessed a corresponding serial port of STS Series but is not configured on *remote host configuration.* Surely, STS Series also send data to the hosts, which are configured on *remote host configuration.* If *Send to recent unlisted remote host* function is set as 'No', STS Series sends data only to the host(s) which are configured on *remote host configuration.* STS Series maintains a recent unlisted remote host during the *Inactivity Timeout*.

Inactivity Timeout

In UDP mode, *Inactivity Timeout* is used in maintaining recent unlisted remote host. If there is no data transmission between unlisted remote host and serial port of STS Series during *Inactivity Timeout,* STS Series will not send data from a serial port to the recent unlisted remote host again. Namely, *Inactivity Timeout* in UDP mode is the time maintained recent unlisted remote host list by STS Series. If user set *Inactivity Timeout* as 0 in UDP mode, STS Series does not send any data from serial port to unlisted remote host.

Socket ID

When Super Series connects remote host(s), sometimes it is needed to identify the device using the string. In this case, if user specifies specific strings in *Socket ID*, Super Series send these strings first before start the data transmission. User can specifies either Serial Numebr or specific strings up to 49 characters as Socket ID.(This option can be changed only by root user) In UDP mode, specified *Socket ID* strings are added to every packet.

4.2.4.3. Modem emulation mode

In modem emulation mode, the serial port process acts as if it is a modem attached to the serial device. It accepts AT modem commands and answers to them, as modems would do. It also handles the modem signals correctly. Modem emulation mode is useful in the following cases.

- There already exists a modem attached to the users' serial device.

If users' serial device already has a modem for phone-line connection, it can be just replaced by the STS Series for Ethernet connection. What users need to do is to use an IP address (or domain name) instead of phone number as a parameter of ATA/ATDT commands.

- It is required to send serial data to the multiple remote hosts.

If the serial device should send data to the multiple hosts, modem emulation mode is required. For example, the first data from the serial device can be sent to the first data acquisition server and the second to the second server. What user device has to do is to change the IP address (or domain name) parameter whenever the device sends ATD(T) XXX command.

By using the modem emulation mode of the STS Series, users can have their serial device connected to the Ethernet network easily, which is cheaper than using phone line modem. *Table 4-2* is a summarized AT command table which is supported by the STS Series. *Figure 4-7* shows the typical case of the serial port command flow when ATDA command is used to connect to the Ethernet network.

Table 4-2 AT commands supported in the STS Series

Command	**Internal Operation**	**Response [9] (Verbose Code)**
+++	Return to command input mode	None
ATD(T) [remote IP or domain name]:[remote port] [CR][LF] or ATD(T) [remote IP][remote port] [CR][LF]	Set TCP mode as TCP client mode. And then, try to connect to the specified remote host. e.g. atdt192.168.1.9:1002 e.g. atdt1921680010091002 Connect to IP address, 192.168.1.9, port 1002 (Port Number is permitted from 1 to 65534) e.g. atdtss.sena.com:1002 Connect to domain address ss.sena.com, port 1002	If successful, CONNECT [CR][LF] If failure in connection, NO CARRIER [CR][LF] If other errors, ERROR [CR][LF]
AT or ATZ [CR][LF]	Initialize TCP socket and serial port	If successful, OK [CR][LF] If failure, ERROR [CR][LF] If successful, OK [CR][LF] If failure, ERROR [CR][LF]
ATA/ [CR][LF]	Repeat last command	
ATA [Local port number] [CR][LF]	Set TCP mode as TCP server mode. And then, set TCP state as [Listen]. -. If the command parameter, Local port number is not specified, the TCP session parameter, Local Port is used instead.	
ATEn [CR][LF]	E, E0: Disable echo E1: Enable echo	
ATOn [CR][LF]	O, O0: Turn to data mode	
ATQn [CR][LF]	Q, Q0: Response display on (default) Q1: Response display off	
ATVn [CR][LF]	V, V0: Response = <numeric code> [CR][LF] V1 (default): Response = <verbose code> [CR][LF]	
AT&Dn [CR][LF]	D, D0: ignore DTR(PC) signal D2(default): disconnect TCP session	
AT&Fn [CR][LF]	F, F0, F1: Restore default modem settings	
AT&Kn [CR][LF]	K, K0: No flow control K3: RTS/CTS flow control (default) K4: Xon/Xoff (if supported)	
AT&Sn [CR][LF]	S, S0: DSR(PC) always high S1: DSR(PC) shows TCP connection	
ATHn [CR][LF]	H, H0: Disconnect current TCP connection All the data will be cleared H1: Keep the current TCP connection	OK [CR][LF]
ATIn [CR][LF]	I, I0 : display “Sena Technologies, Inc.” I3 : display model number Others : display “OK”	<=
AT\Tn [CR][LF]	Set inactivity timer to n minutes \T, \T0: inactivity timer disabled (default)	OK [CR][LF]
ATBn, ATCn, ATLn, ATMn, ATNn, ATP, ATT, ATYn, AT%Cn, AT%En, AT&Bn, AT&Gn, AT&In, AT&Qn, AT&V, ATMn, AT\An, AT\Bn, AT\Nn, ATXn	none	OK [CR][LF]

[9] If *Echo* mode is enabled, the command will be sent back first. And then, corresponding response will be sent. If disabled, only response will be sent.

ATS?, ATSn=x, AT&Cn, AT&Wn, AT&Zn=x	none	ERROR [CR][LF]
ATFn [CR][LF]	None	If n=1 OK [CR][LF] If others, ERROR [CR][LF]
ATWn	None	If n=0 OK [CR][LF] If others, ERROR [CR][LF]

Table 4-3 AT commands Response Code

Verbose Code (After "ATV1" command executed)	**Numeric Code** (After "ATV0" command executed)	**Description**
OK	0	Command executed
CONNECT	1	Modem connected to line
RING	2	A ring signal has been detected
NO CARRIER	3	Modem lost carrier signal
ERROR	4	Invalid command

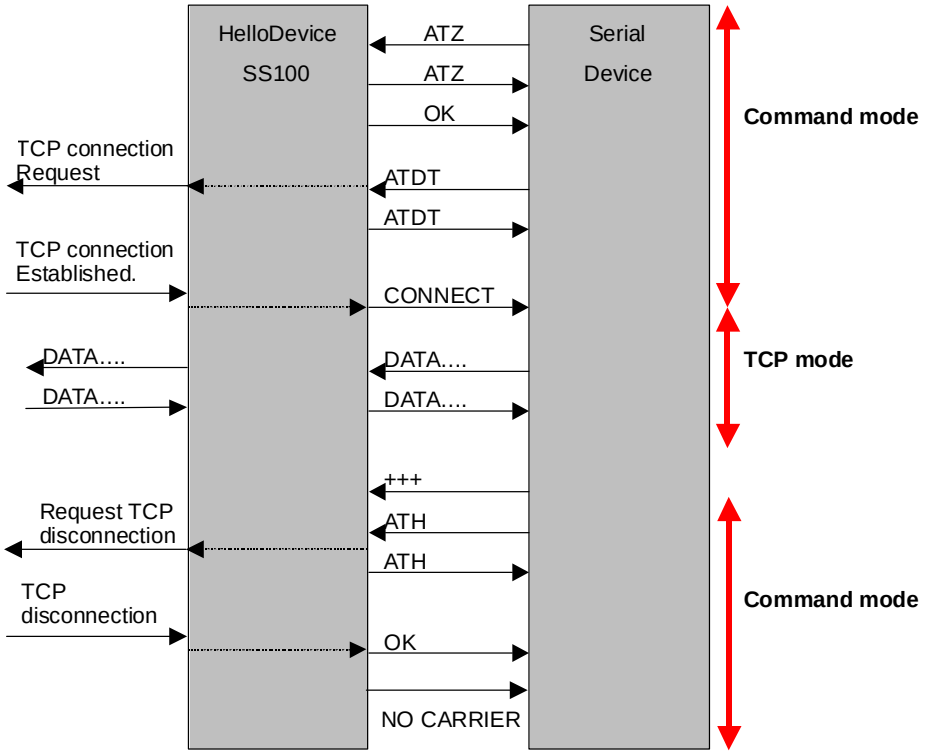


Figure 4-7 Typical case of command/data flow of modem emulation mode

4.2.5. Remote host configuration

Remote host configuration is the list of hosts that will receive data from serial port of STS Series when

there is data transmission from a serial port of STS Series.

In TCP mode, user can also configure secondary remote host that will receive data from serial port if STS Series fails to connect to primary remote host. But if connection to primary remote host can be made, STS Series dose not send data to secondary remote host until connection to primary remote host failed. And the maximum possible number of primary remote host is limited up to 16.

In UDP mode, user can configure only primary remote host because there is no way for STS Series to check status of primary remote host, so secondary remote host is meaningless.

Figure 4-8 shows Remote host configuration pages of the Web UI. (TCP mode)

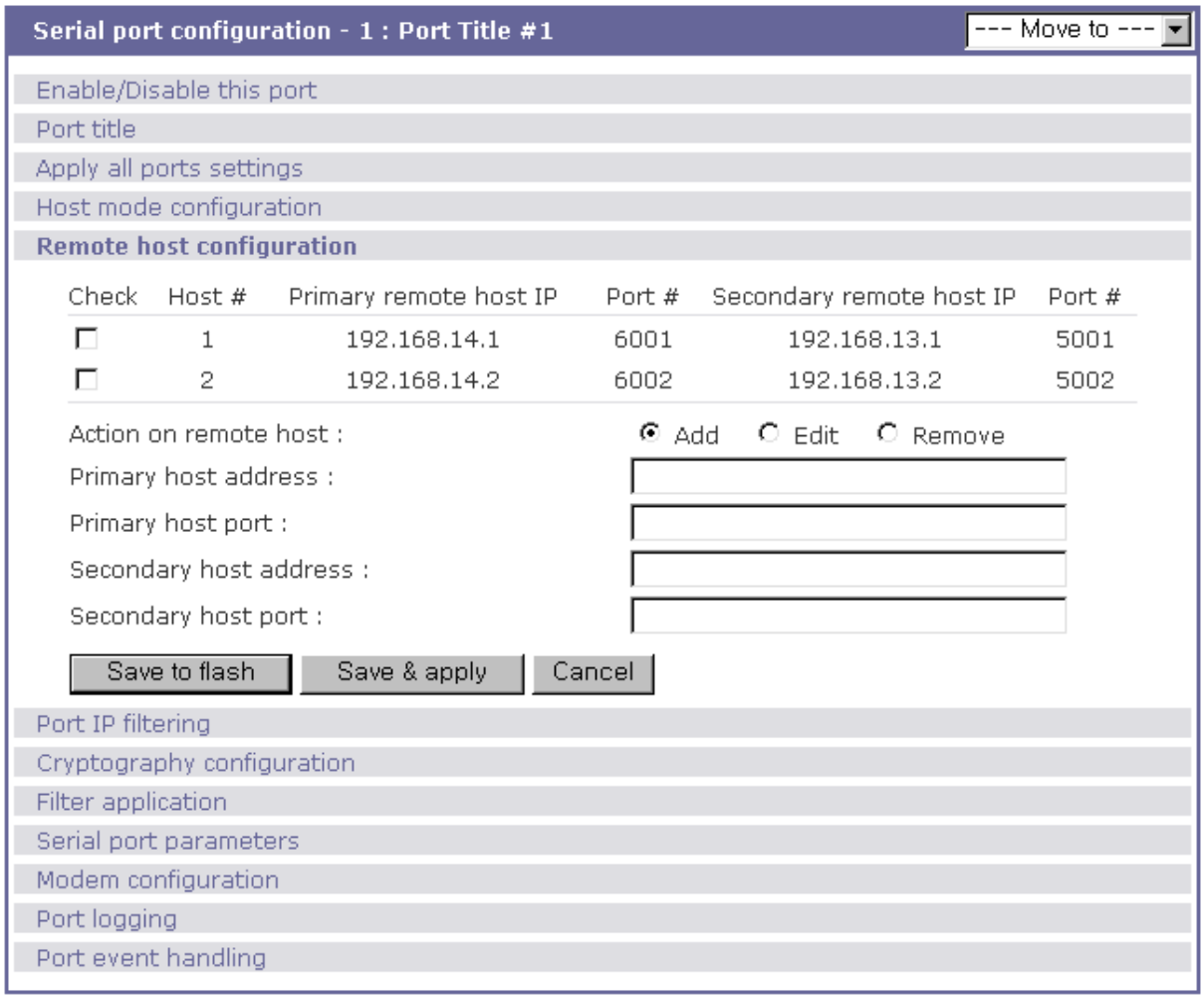


Figure 4-8 Remote host configuration

4.2.6. Port IP filtering configuration

The remote hosts that are allowed to access the STS Series serial ports can be specified based on the IP address filtering rules. The user may allow specific hosts to access the STS Series serial ports by providing a valid IP address or network address and its subnet mask. Please refer to section *3.5 IP Filtering* for more details.

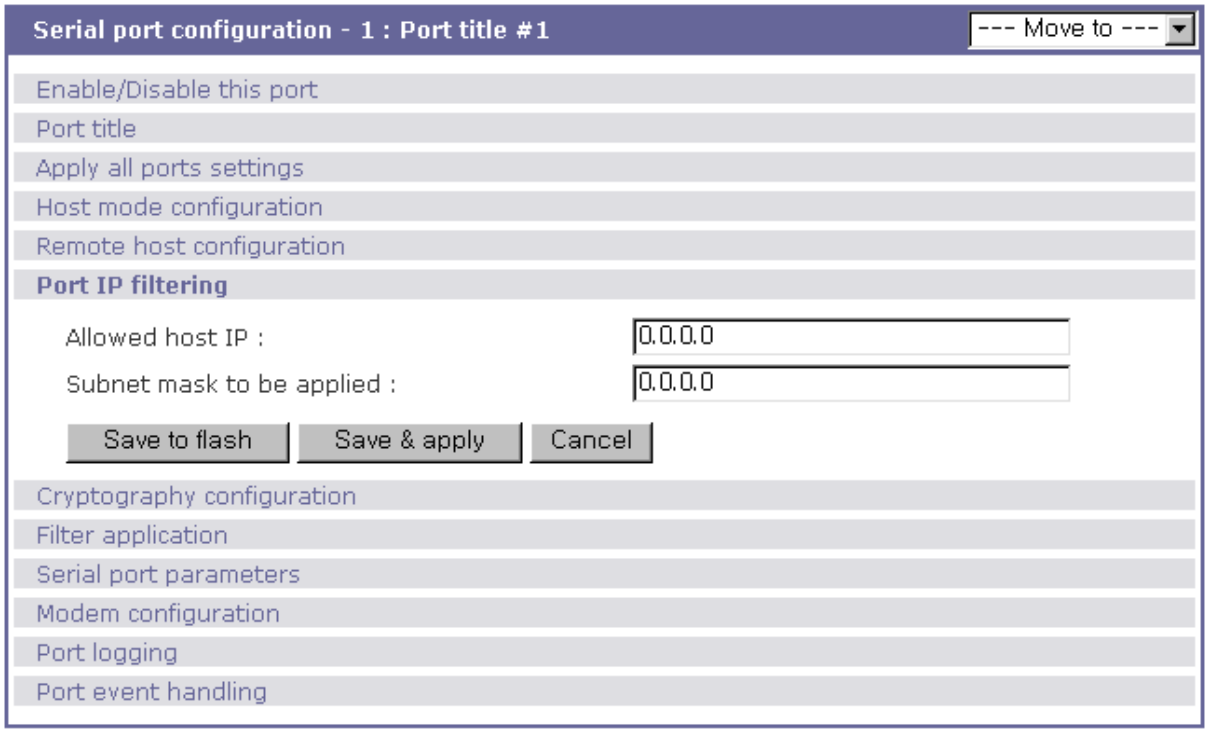


Figure 4-9 Port IP filtering for serial ports

4.2.7. Cryptography configuration

The STS Series supports encrypted sessions for only TCP mode including modem emulation mode (not UDP mode).

4.2.7.1. Secure Sockets Layer(SSL) and Transport Layer Security(TLS) cryptography method

By setting the cryptography method as one of SSLv2, SSLv3, SSLv3 rollback to v2 or TLSv1, the STS Series can communicate with other device supporting SSL/TLS cryptography method in encrypted sessions.

SSL was developed by Netscape for use between clients and servers. SSL layers on top of any transport protocol and can run under application protocols such as HTTP. SSL aims to be secure, fast, and adaptable to other Web protocols. SSL provides data security for applications that communicate across networks. SSL is a transport-layer security protocol layered between application protocols and TCP/IP.

TLS is an updated version of SSL. The protocol is specified in an Internet RFC, developed under the auspices of the Internet Engineering Task Force (IETF). TLS is an evolution of SSL and it specifies a mechanism for falling back to SSL if either client or server does not support the newer protocol, so a transition to TLS is relatively painless.

To initiate SSL/TLS sessions, exchange of messages called the SSL handshake is required between two devices (Server and Client). The SSL/TLS protocol uses a combination of public-key and symmetric key encryption. Symmetric key encryption is much faster than public-key encryption, but public-key encryption provides better authentication techniques. The handshake allows the server to authenticate itself to the client using public-key techniques, and then allows the client and the server to cooperate in the creation of symmetric keys used for rapid encryption, decryption, and tamper detection during the session that follows. The details of handshake process step involved can be summarized as follows:

1. The client sends the server the client's SSL/TLS version number, cipher settings, randomly generated data, and other information the server needs to communicate with the client using SSL/TLS.
2. The server sends the client the server's SSL/TLS version number, cipher settings, randomly generated data, and other information the client needs to communicate with the server over SSL/TLS. The server also sends its own certificate and, if the client is requesting a server resource that requires client authentication, requests the client's certificate.
3. The client uses some of the information sent by the server to authenticate the server. If the server cannot be authenticated, the user is warned of the problem and informed that an encrypted and authenticated connection cannot be established. If the server can be successfully authenticated, the client goes on to next step.
4. Using all data generated in the handshake so far, the client (with the cooperation of the server, depending on the cipher being used) creates the premaster secret for the session, encrypts it with the server's public-key (obtained from the server's certificate, sent in step 2), and sends the encrypted premaster secret to the server. SSL/TLS differ in the way this "shared" master secret is created
5. If the server has requested client authentication (an optional step in the handshake), the client also signs another piece of data that is unique to this handshake and known by both the client and server. In this case the client sends both the signed data and the client's own certificate to the server along with the encrypted premaster secret.
6. If the server has requested client authentication, the server attempts to authenticate the client. If the client cannot be authenticated, the session is terminated. if the client can be successfully authenticated, the server uses its private key to decrypt the premaster secret, then performs a series of steps (which the client also performs, starting from the same premaster secret) to generate the master secret.
7. Both the client and the server use the master secret to generate the session keys, which are symmetric keys used to encrypt and decrypt information exchanged during the SSL/TLS session and to verify its integrity--that is, to detect any changes in the data between the time it was sent and the time it is received over the SSL/TLS connection.

8. The client sends a message to the server informing it that future messages from the client will be encrypted with the session key. It then sends a separate (encrypted) message indicating that the client portion of the handshake is finished.
9. The server sends a message to the client informing it that future messages from the server will be encrypted with the session key. It then sends a separate (encrypted) message indicating that the server portion of the handshake is finished.
10. The SSL/TLS handshake is now complete, and the SSL/TLS session has begun. The client and the server use the session keys to encrypt and decrypt the data they send to each other and to validate its integrity.

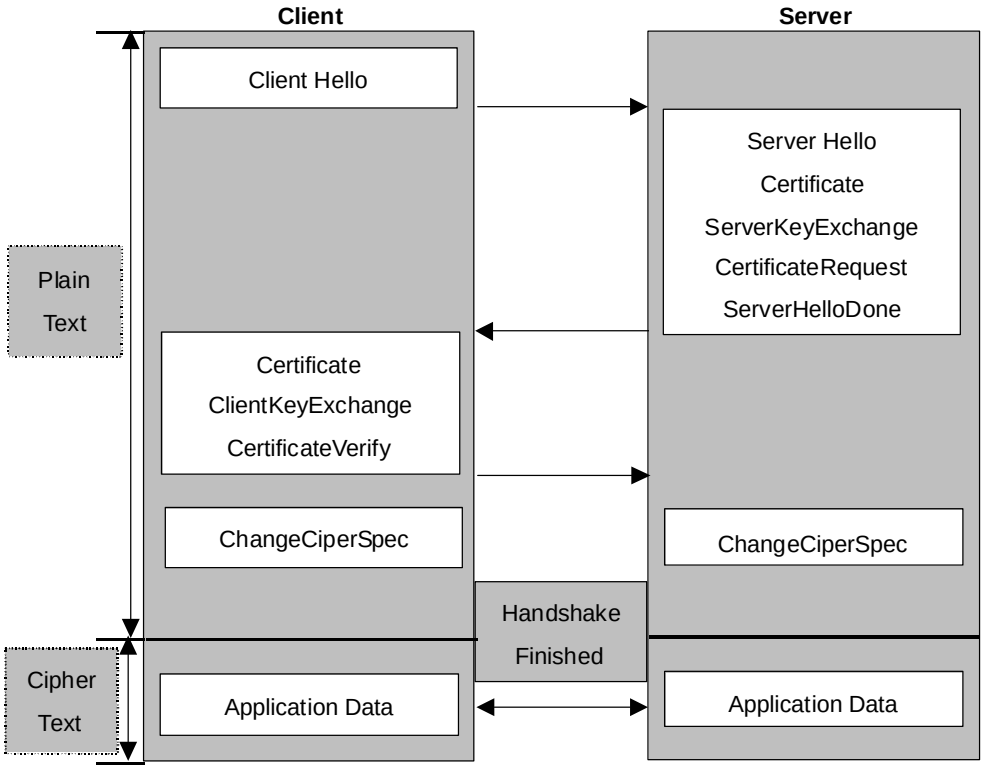


Figure 4-10 Typical SSL/TLS Handshake Process

The STS Series can act as a SSL/TLS server or as a SSL/TLS client depending on status of TCP mode. If TCP connection with SSL/TLS is initiated from remote host first, STS Series acts as a SSL/TLS server during the SSL handshake process. On the contrary, if TCP connection with SSL/TLS is initiated from serial port of STS Series first, STS Series acts as a SSL/TLS client during the SSL handshake process.

When user uses SSL/TLS cryptography method, user can configure following parameters.

- **Enable/Disable cipher suites**

  A cipher suite is an object that specifies the asymmetric, symmetric and hash algorithms that are

used to secure an SSL/TLS connection. The asymmetric algorithm is used to verify the identity of the server (and optionally, that of the client) and to securely exchange secret key information. The symmetric algorithm is used to encrypt the bulk of data transmitted across the SSL/TLS connection. The hash algorithm is used to protect transmitted data against modification during transmission. The length of the keys used in both the symmetric and asymmetric algorithms must also be specified.

When a client makes an SSL/TLS connection to a server, it sends a list of the cipher suites that it is capable of and willing to use. The server compares this list with its own supported cipher suites and chooses the first cipher suite proposed by the client that it is capable of and willing to use. Both the client and server then use this cipher suite to secure the connection.

Choice of cipher suite(s) depends on environment and security requirements. The RSA-based cipher suites are the most widely used and may also give some advantages in terms of speed.

The STS Series support various cipher suites and user can select each cipher suite by enabling or disabling corresponding cipher suite.

- **Verify client (server mode only)**

  If user selects *Verify client* option as Yes, STS Series will request the client's certificate while in SSL handshaking process (Step 2). On the contrary, if user selects *Verify client* option as No, STS Series does not request the client's certificate while in SSL handshaking process (Step 2).

- **Verify certificate chain depth**

  A certificate chain is a sequence of certificates, where each certificate in the chain is signed by the subsequent certificate. The purpose of certificate chain is to establish a chain of trust from a its own(peer) certificate to a trusted CA certificate. The CA vouches for the identity in the peer certificate by signing it. If the CA is one that user trusts (indicated by the presence of a copy of the CA certificate in user's root certificate directory), this implies user can trust the signed peer certificate as well. In STS Series, user can restrict number of certificate chain depth so that STS Series does not search a trusted CA certificate infinitely in a certificate chain.

- **Check the certificate CN**

  If user selects *Check the certificate CN* option as Yes, STS Series will check whether the host name is matched with Common Name(CN) in the certificate, and if they do not matched, STS Series will close connection request to the remote host. On the contrary, if user selects *Check the certificate CN* option as No, STS Series does not check whether the host name is matched with Common Name(CN) in the certificate.

  STS Series checks Common Name(CN) only if it acts as SSL/TLS client.

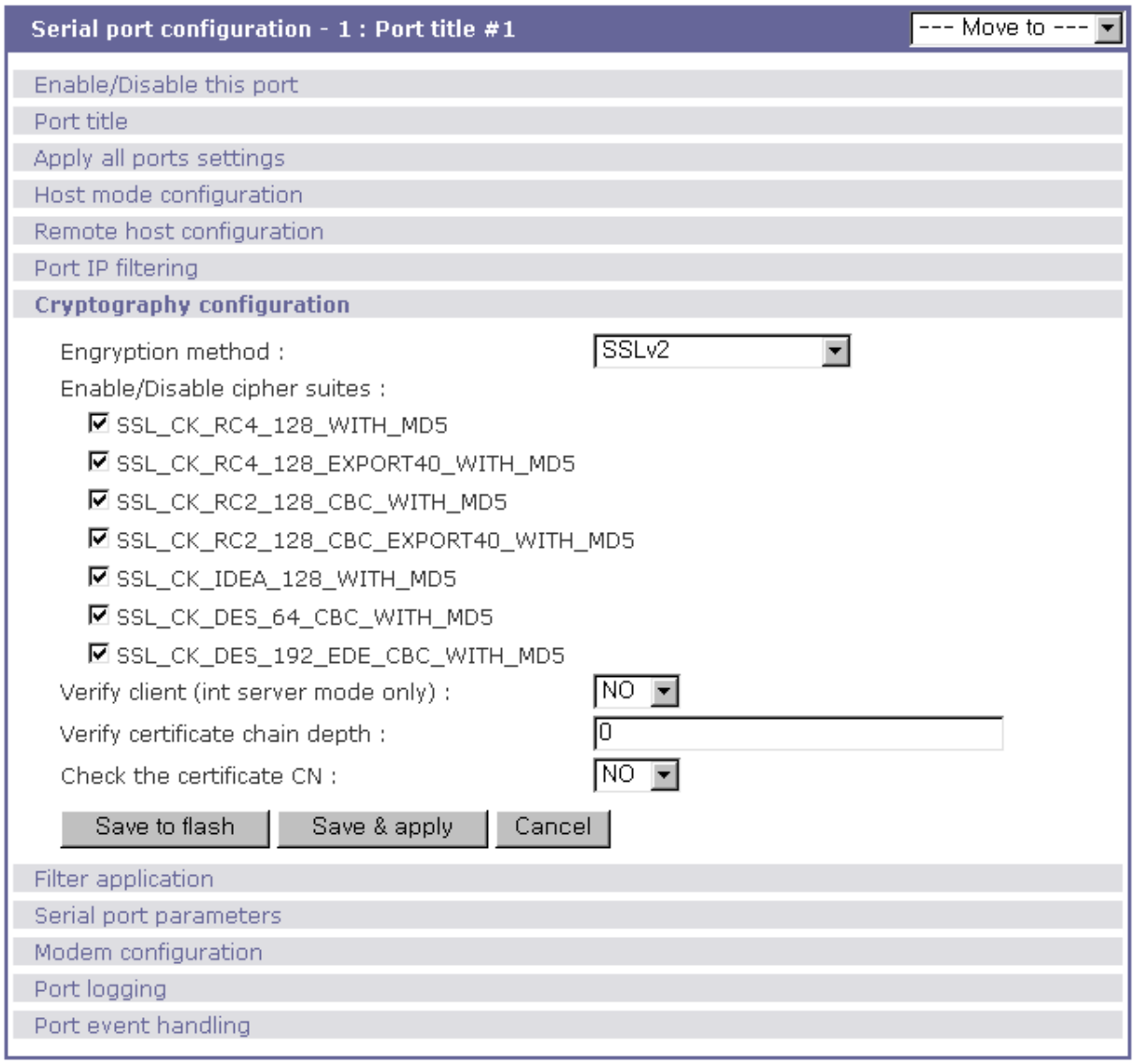


Figure 4-11 Cryptography configuration

4.2.7.2. 3DES cryptography method

By setting the cryptography method as *3DES*, the STS Series can communicate with other STS Series device or HelloDevice Pro Series in 3DES(168 bits) encrypted sessions.

Figure 4.12 shows record format of 3DES packet where meanings of each field are as follows,

Length	Data	Padding

Figure 4-12 Record Format of 3DES packet

- **Length**

  The length is 8-bits number. The length is the length of content (data and padding). 3DES is a 64-

bit block cipher algorithm, and then the length must be a multiple of 8(64/8).

- **Padding**

  The padding is a standard block cipher. The pad value is the total number of pad bytes in the padding(1~8).

In 3DES algorithm in STS Series, key and initial vector, which are used in generating encrypted data packet, is derived from key block. And key block is generated by using user configured key string. *Figure 4-13* shows key derivation process.

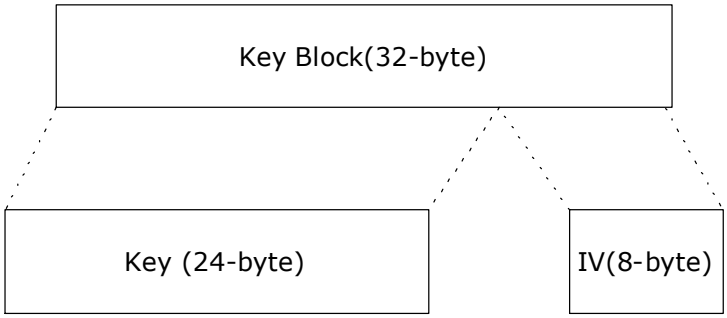


Figure 4-13 Key derivation

The key block is defined as:

Key_Block = MD5(KEY_STRING) + MD5(MD5(KEY_STRING)+KEY_STRING)

= (16 bytes) + (16 bytes)

Key = first 24bytes of Key Block

IV(Initial Vector) = last 8 bytes of Key block

4.2.7.3. RC4 cryptography method

By setting the cryptography method as RC4, the STS Series can communicate with other STS Series device in RC4 encrypted sessions. In RC4 encryption mode, STS Series will encrypt/decrypt all the TCP stream with the user configured key string, and there is no header and no padding. RC4 is faster than 3DES.

4.2.8. Filter application

The STS Series supports user manipulation of raw data between remote host and serial device connected serial port. Filter application configuration can be used for this purpose. If user makes his own filter application program, he can upload it to STS Series and configure name of the program and its arguments through filter application configuration menu. For more detail information about making

filter application, please refer to 9.4 Making and running user's own code section.

Note :

File uploading is supported only in console menu. For more information about file uploading, please refer to 6.10 User File Uploading section.

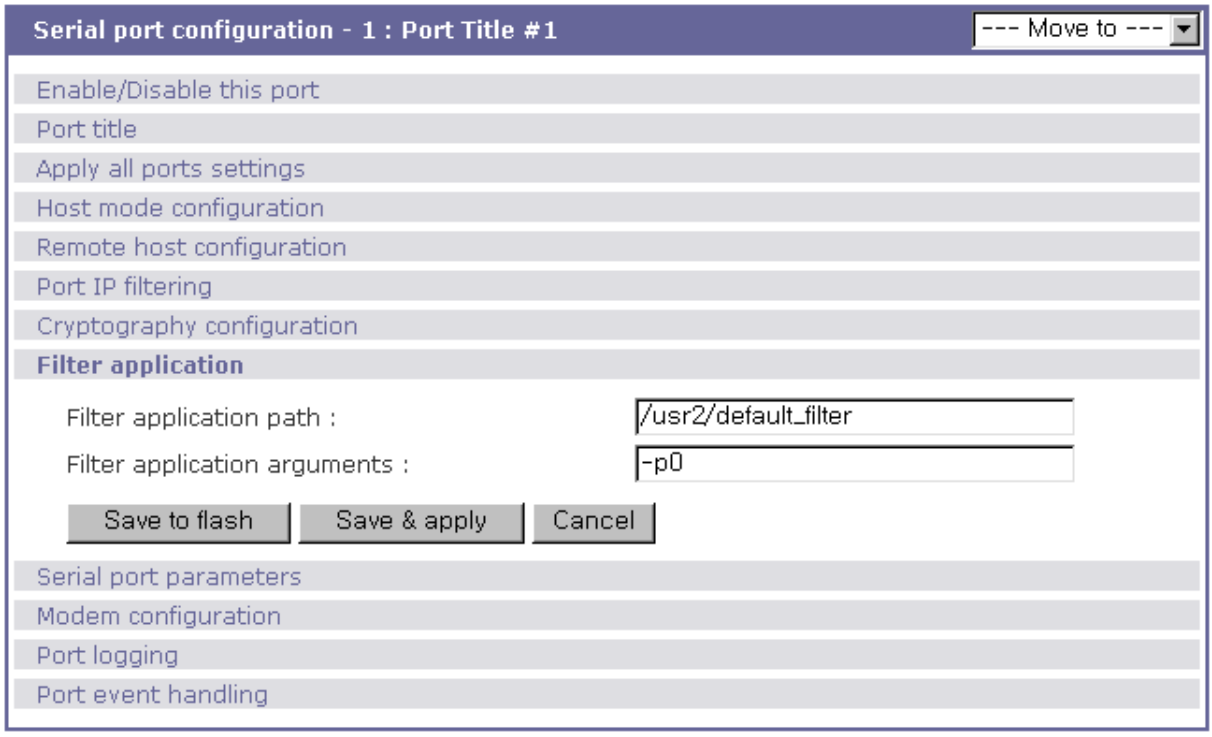


Figure 4-14 Filter application

4.2.9. Serial port parameters

To connect the serial device to the STS Series serial port, the serial port parameters of the STS Series should match exactly to that of the serial device attached. The serial port parameters are required to match this serial communication. The parameters required for the serial communication are: UART type, baud rate, data bits, parity, stop bits, flow control DTR/DSR behavior and inter-character timeout.

First of all, the STS Series and the serial device must agree on the serial communication type, which is RS232 mode. For more information about pin out of serial port and wiring diagram, please refer to *Appendix 1 Connections* section.

- **Baud rate**

  The valid baud rate for the STS Series is as follows:

  75, 150, 200, 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 38400, 57600, 115200, and 230400

The factory default setting is 9600.

- **Data bits**

  Data bits can be between 7 bits and 8 bits. The factory default setting is 8 bits.

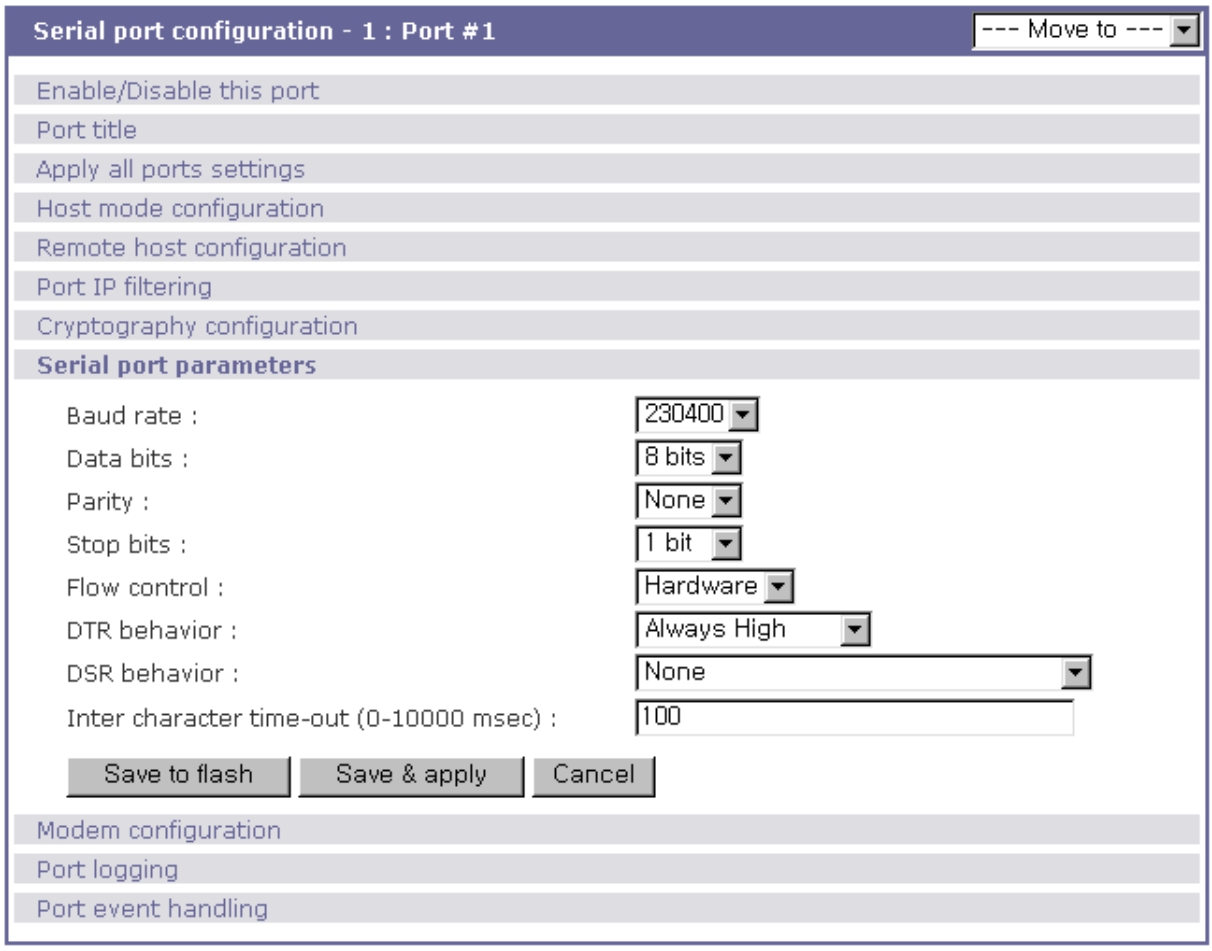


Figure 4-15 UART configuration

- **Parity**

  Parity can be **none, even** or **odd**. The factory default setting is none.

- **Stop bits**

  Stop bits can be between 1 bit and 2 bits. The factory default setting is 1 bit.

- **Flow control**

  The factory default setting of the flow control is *None*. Software Flow Control using XON/XOFF and hardware flow control using RTS/CTS are supported by the STS Series.

  Software flow control method controls data communication flow by sending special

characters XON/XOFF(0x11/0x13) between two connected devices. And hardware flow control method controls data communication flow by sending signals back and forth between two connected devices.

Note:

Flow control is supported only in RS232 mode. RS422 and RS485 mode do not support any kind of flow control method in hardware or software.

- **DTR/DSR behavior**

  The purpose of the DTR/DSR pin is to emulate modem signal control or to control TCP connection state by using serial port signal. The DTR is a write-only output signal, whereas the DSR is a read-only input signal in the STS Series side.

  The DTR output behavior can be set to one of three types: *always high, always low or high when open*. If the DTR behavior is set to *high when open*, the state of the DTR pin will be maintained high if the TCP connection is established.

  The DSR input behavior can be set to one of two types: *none or allow TCP connection only by high*. If user sets the DSR input behavior as *Allow TCP connection only by HIGH,* TCP connection to remote host from STS Series is made only when the DSR status is changed from low to high. And TCP connection to remote host is disconnected when the DSR status is changed from high to low. And also STS Series accepts TCP connection from the remote host only when the DSR status is high. In case of UDP mode, STS Series receives UDP data from the remote host only when the DSR status is high.

  In modem emulation mode, the connection to the remote host will be disconnected regardless of the current DSR input behavior option if the DSR status goes to low.

  Note:

  DTR/DSR behavior menu will not be shown when the modem is enabled.

- **Inter-character timeout**

  This parameter defines the interval that the STS Series fetches the overall serial data from its internal buffer. If there is incoming data through the serial port, the STS Series stores data into the internal buffer. The STS Series transfers data stored in the buffer via TCP/IP, only if the internal buffer is full or if the inter-character time interval reaches to the time specified as *inter-character timeout*. If *inter-character timeout* is set as 0, then data stored in the internal buffer will be transferred immediately without any delay.

  Optimal inter-character timeout would be different according to your application but at least it must be larger than one character interval within specified baud rate. For example, assume that the serial port is set to 1200 bps, 8 Data bits, 1 stop bit, and no parity. In this case, the total

number of bits to send a character is 10 bits and the time required to transfer one character is

10 (bits) / 1200 (bits/s) * 1000 (ms/s) = 8.3 ms.

Therefore, you have to set *inter-character timeout* to be larger than 8.3 ms. The *inter-character timeout* is specified in milliseconds.

If users want to send the series of characters into a packet, serial device attached to the STS Series should send characters without time delay larger than *inter-character timeout* between characters and the total length of data must be smaller than or equal to the STS Series internal buffer size. The serial communication buffer size of STS Series is 256 bytes.

4.2.10. Modem configuration

The STS Series supports direct modem connection to the serial port of it. When user wants to connect modem to a serial port, he must configure Modem init-string and DCD behavior on modem configuration page. The STS Series supports modem connection only when host mode is set as *TCP mode*.

- **Enable/Disable modem**

  By enabling this menu, user can attach a modem directly to the serial port of STS Series. If this parameter is enabled, STS Series considers this port will be used for modem use exclusively.

- **Modem init-string**

  User can specify modem initialization string for his modem in *Modem init-string* parameter. When a serial port is set as modem mode by setting *Enable/Disable modem* parameter as Enabled, STS Series sends modem initialization string to the serial port whenever rising edge of DTR pin is detected or parameter related with serial port configuration is changed.

- **DCD behavior**

  If *DCD behavior* is set as *Allow TCP connection only by HIGH,* STS Series permits a connection from the remote host only when the DCD status of serial port is high. This feature is useful when user want to use a serial port only for dial-in modem mode. In this case, if there is no connection through modem already, STS Series dose not permit TCP side connection.

- **Automatic release modem connection**

  If *Automatic release modem connection* is set as *Enable,* modem connection will be closed by STS Series if all TCP connections are closed once at least one TCP connection is opened. If this option is set as *Disable,* modem connection will not be closed by STS series even if all TCP connections are closed. Please note that actual phone line connection will be closed if one of

modems closes connection, regardless of this option. That is, this option can be used for STS Series to disconnect modem connection by itself when all TCP connections are closed.

If user want to use dial-out function, he should set *DCD behavior* as *None* because he must be able to access modem connected to a serial port to send dial out command to the modem first.

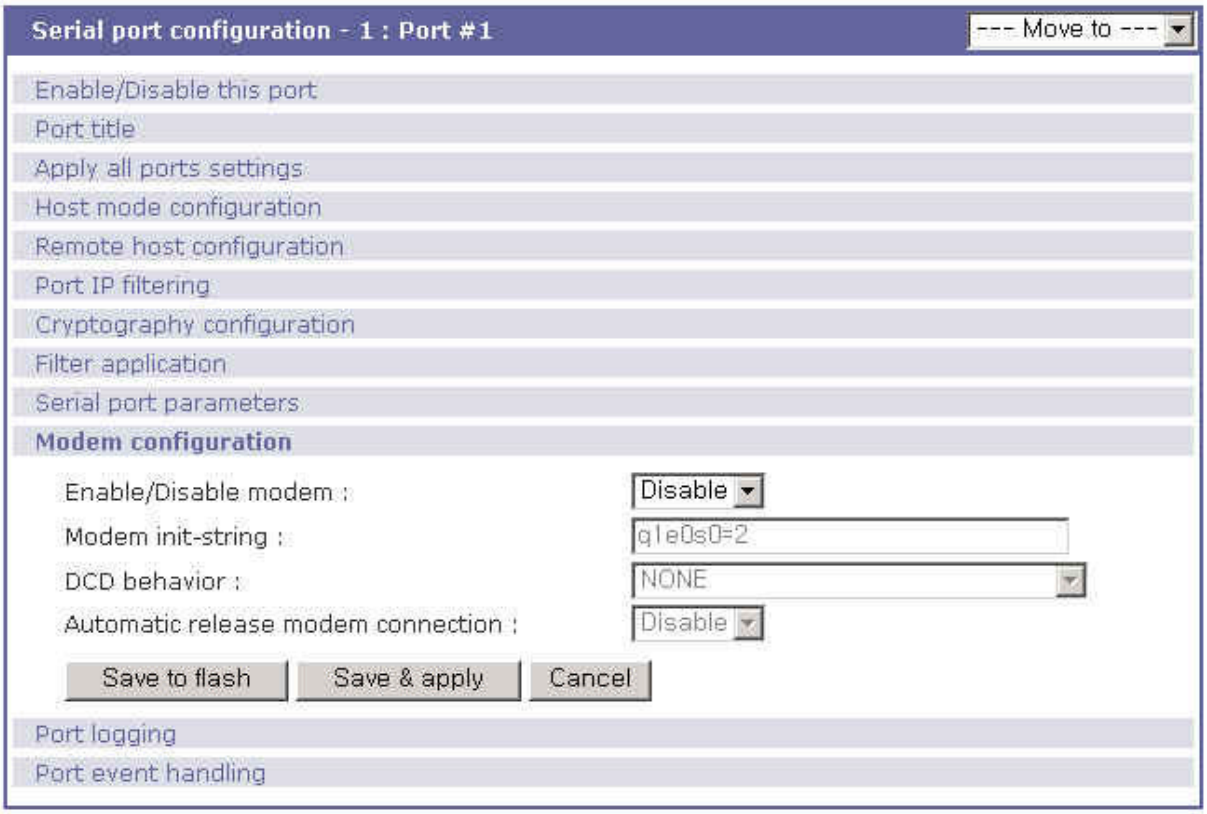


Figure 4-16 Modem configuration

4.2.11. Port Logging

With the port logging feature, the data sent through the serial port is stored to MEMORY, an ATA/IDE fixed disk card, a SYSLOG server or a mounting point on an NFS server.

- **Enable/disable port logging**
  This parameter defines whether to enable or disable the port-logging feature. The factory default setting is [disabled].

- **Port log storage location**
  The port log data can be stored to the STS Series internal memory, an ATA/IDE fixed disk card inserted in PCMCIA slot, the mounting point on an NFS server or the SYSLOG server. If the

internal memory is used to store port log data, the port log data will be cleared when the STS Series is turned off. To preserve the serial port log data, set the storage location to be the ATA/IDE fixed disk card, SYSLOG server or NFS server. To do this, the user must configure the corresponding media in advance. Unless the media is properly set up, the user will not be able to select a storage location from the interface.

- **Port log buffer size**

  This parameter defines the maximum amount of port log data to be logged. When using internal memory to store the log data, the total size of the port buffer cannot exceed 3200 Kbytes (i.e. sum of all port buffer size of each serial port should be smaller than or equal to 3200 Kbytes). The factory default setting is 4 Kbytes.

  When using an ATA/IDE fixed disk card to store log data, the maximum port buffer size is dependent upon the card capacity.

  When using an NFS server to store log data, the maximum port buffer size is unlimited. The user should configure the NFS server to ensure that the port logging system works properly.

  When using the SYSLOG server to store log data, the user cannot set the port log buffer size.

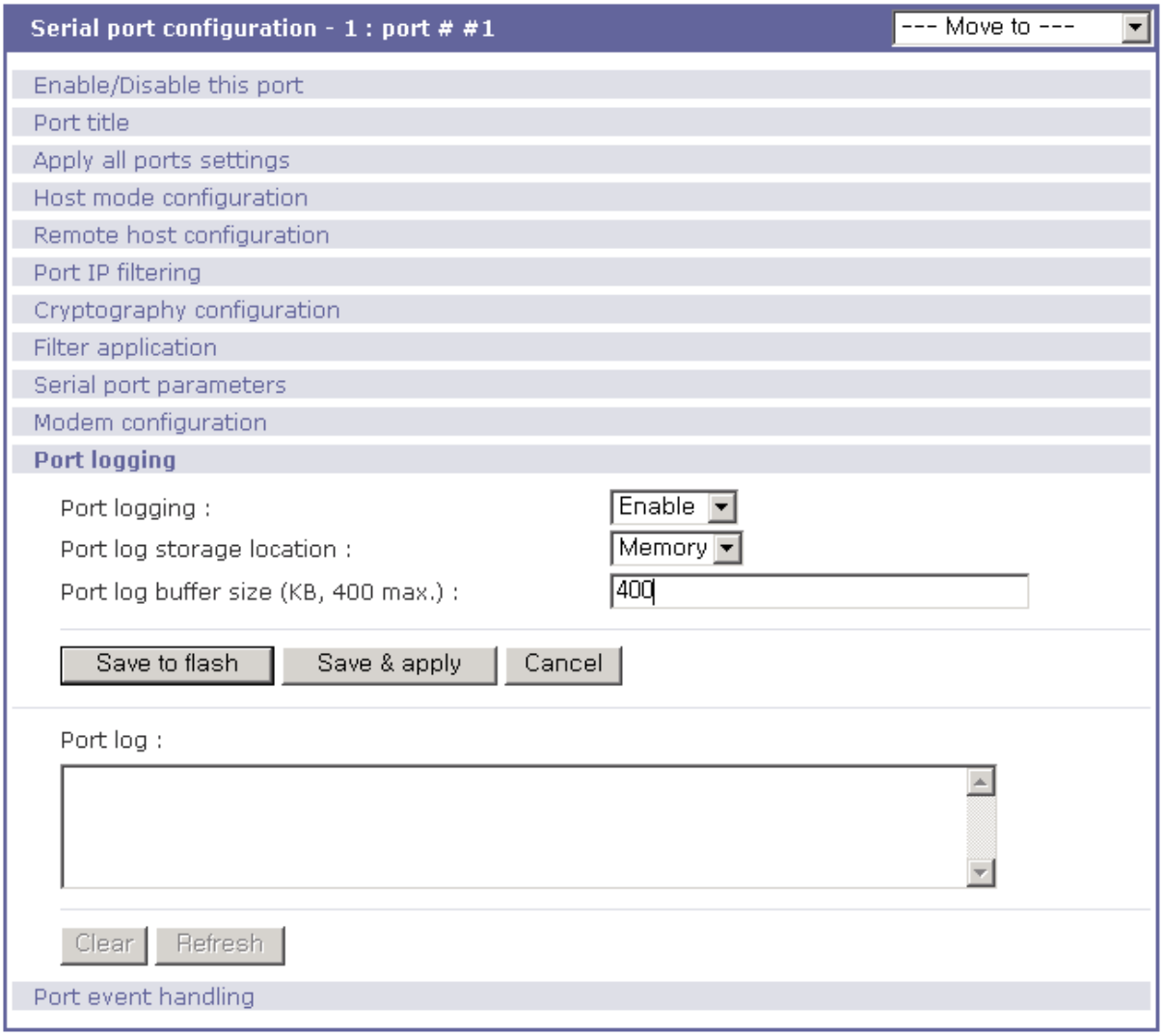


Figure 4-17 Port logging configuration

4.2.12. Port event handling configurations

The STS Series provides a user for a means of monitoring or reacting to data from serial device attached to a serial port of it through *Port event handling* configuration. Namely, user can define keywords for each serial port that will trigger the email/SNMP notification or command sent to the serial port directly on *Port event handling* configuration. And this will enable the user to monitor the data from the attached device or to manage/control a device attached serial port directly when pre-defined keywords are detected. At the same time, the status of the connection between the STS Series and the serial device and the status of the TCP connection between the STS Series and remote hosts could be monitored and managed in the same way of the port keywords as well.

Each reaction can be configured individually upon each keyword. Reaction can be an email delivery, SNMP trap sending, command sending or either combination of all reactions.

- **Port event handling**

  If the user wants to enable *port event handling* feature, set *Port event handling* as enable. . This is a global parameter so if this feature is disabled, the STS Series does not take any actions on port events.

- **Notification interval**

  To prevent STS Series from being trapped in handling port event, there is a *Notification interval* parameter. STS Series will send notification email or SNMP trap every *Notification* interval even it detect predefined keyword within *Notification interval.* The smaller value of this parameter will result in immediate response for predefined keyword and heavy usage of system resources. The largest value accepted by user is recommended to prevent system resource usage minimization.

- **Email notification**

  This parameter enables or disables Email notification feature of STS Series. When STS Series sends Email notification, it used SMTP server configured in SMTP server configuration. If the SMTP server is not configured correctly or disabled, Email feature gets disabled also. For details of SMTP server configurations and descriptions, please refer to section 3.4 *SMTP Configuration*.

- **Title of Email**

  This parameter set Title of Email that will be sent by STS Series when pre-defined keyword is detected.

- **Recipient's Email address**

  This parameter set mail recipient who will receive notification mail when pre-defined keyword is detected.

- **SNMP notification**

  This parameter enables or disables SNMP notification feature of STS Series.

- **Title of SNMP trap**

  This parameter set Title of SNMP trap that will be sent by STS Series when pre-defined keyword is detected.

- **SNMP trap receiver IP**

  This parameter set IP address of SNMP trap receiver that will receive SNMP trap notification when pre-defined keyword is detected.

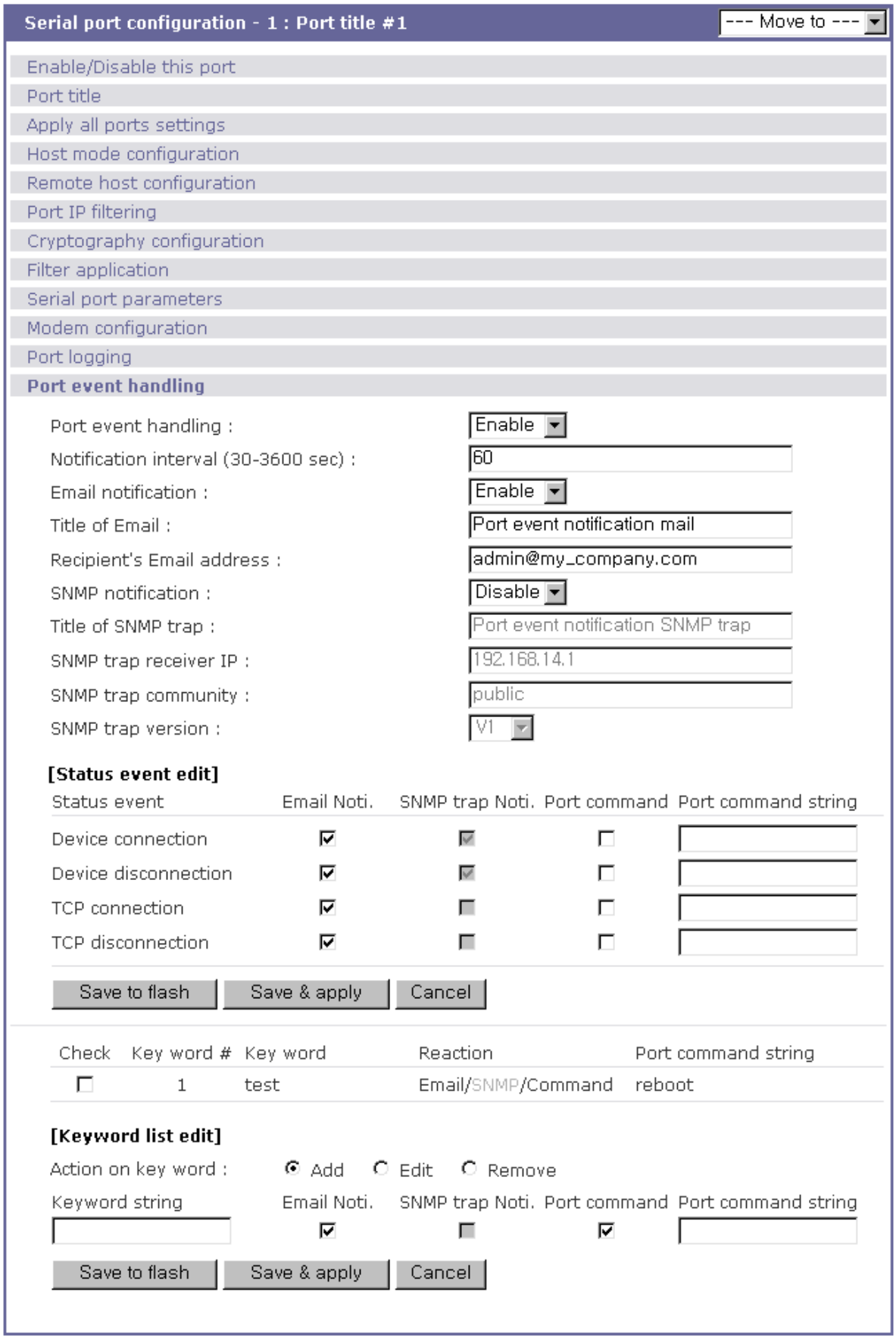


Figure 4-18 Port event handling configurations

- **SNMP trap community**

  This parameter set a community that will be included in SNMP trap message when pre-defined keyword is detected.

- **SNMP trap version**

  This parameter set a version of SNMP trap, which will be sent when pre-defined keyword is detected.

[Status event edit]

- **Device connection/disconnection**

  Fill in the check boxes of the preferred actions that are to be taken on the event of serial device connection or disconnection.

- **TCP connection/disconnection**

  Fill in the check boxes of the preferred actions that are to be taken on the event of TCP connection or disconnection from remote hosts.

[Keyword list edit]

- **Action on key word**

  User can select “Add” or “Remove” for the action on keyword selected.

- **Keyword string**

  User can specify any word, which he/she wants to set as a keyword.

- **Email notification**

  User can select enable or disable for the Email notification action on keyword selected.

- **SNMP trap notification**

  User can select enable or disable for the SNMP trap notification action on keyword selected.

- **Port command**

  User can select enable or disable for the port command action on keyword selected.

- **Port command string**

  STS Series supports direct reaction to a device attached to serial port when pre-defined keyword is detected. User can specify command or string, which will be sent to a serial port on this menu.

4.3. All Port Configurations

If modifications are being made to all serial ports are similar or the same, changes can be made to the serial port configuration for all serial ports simultaneously. With the **all port configuration** function, the configuration will be applied to all the serial ports; unless an individual ports "**apply all port setting"** option is disabled.

"All port configuration" parameters can be grouped into the following groups:

1. Port enable/disable
2. Port title
3. Host mode
4. Remote host configuration
5. Port IP filtering
6. Cryptography configuration (*Only valid and visible if host mode set to TCP or Modem Emulation mode*)
7. Filter application
8. Serial port parameters
9. Modem configuration (*Only valid and visible if host mode set to TCP mode*)
10. Port logging
11. Port event handling

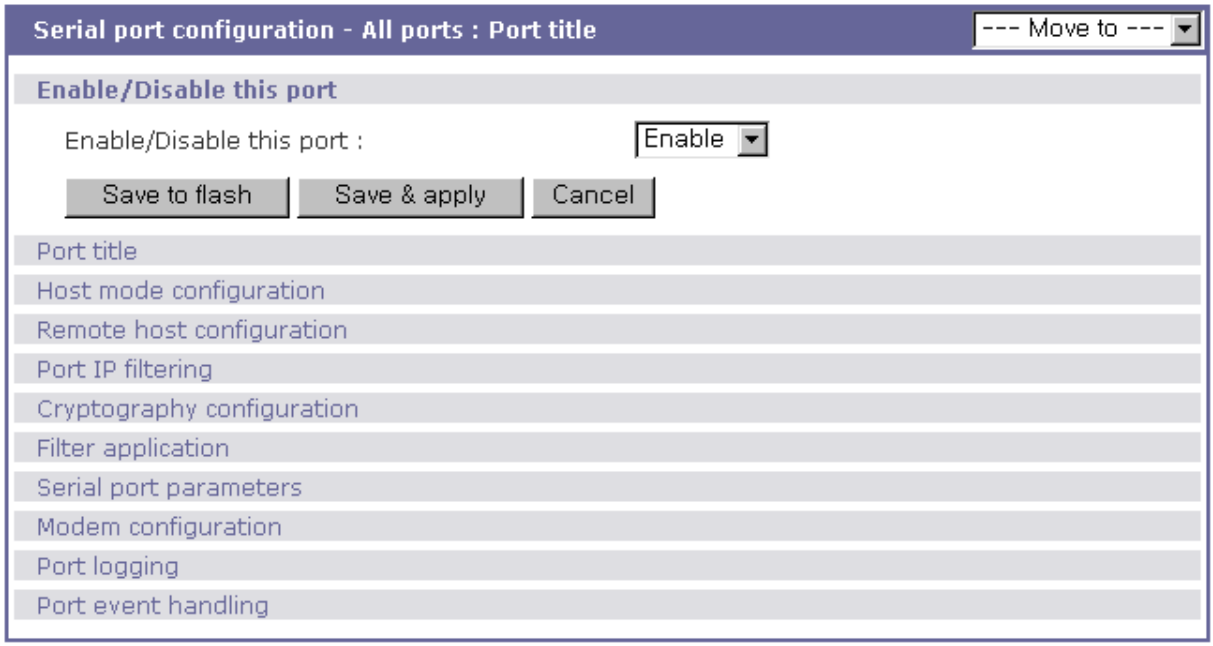


Figure 4-19 All port configuration

- **Port enable/disable**

  This parameter enables or disables port function.

- **Port title**

  If this parameter is set with a certain string, the port title of each serial port will be set with a combination of this string and the port number. For example, if the port title is set with “my server”, the port title of port 1 will be set with “my server #1”, the port title of port#2 will be “my server #2”, and so on.

- **Host mode**

  If the host mode is set to TCP or UDP mode, the listening port number of each serial port will be set with the following equation:

  (listening port number + serial port number - 1)

  Other parameters of each serial port will be set as the same value set in as “all port configuration”.

- **Remote host configuration, Port IP filtering, Cryptography configuration, Filter application, Serial port parameters, Modem configuration, Port logging, Port event handling**

  For the parameters of the groups above, the values set in an “all port configuration” will be set identically for all of the serial ports.

5. PC Card Configuration

The STS Series has one extra PC card slot for increased expandability. It supports four types of PC cards:

- Wireless LAN card
- Modem card
- ATA/IDE fixed disk card

The user can allow access via another network connection with either a LAN or wireless LAN card. The ATA/IDE fixed disk card allows the user the ability to store and carry system and serial port log data. Using the card slot for a modem card allows the user out-of-band access to the STS Series without a serial port to connect to an external modem.

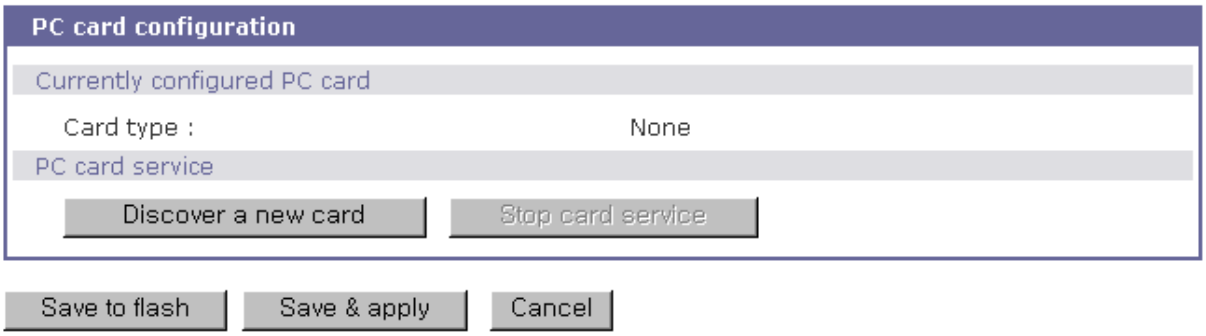


Figure 5-1 Initial PC card configuration menu screen

To use the PC card slot, the users must complete the following steps.

Step 1. Insert the PC card into the PC card slot.

Step 2. Select Discover a new card on the PC card configuration menu.

Step 3. The STS Series will use its plug and play functionality to discover the card type. It will then display the configuration menu screens. The user can now set card's operation parameters.

Step 4. Save the configuration settings by selecting Save to flash

Step 5. Select [**Apply changes**] from the menu to apply the newly configured settings.

If STS Series fails to discover the PC card, the following error message will be displayed on the menu screen.

Figure 5-2 Failure to detect error message

Refer to *Appendix B.PC Card supported by* STS Series to view a list of PC cards support by the STS Series.

To stop or remove the PC card, user must complete the following steps.

Step 1. Select [(**Ban- show the actual button)** Stop card service].

Step 2. Save the configuration changes by selecting [**Save to flash**].

Step 3. Apply changes by selecting [Apply changes] from the menu.

Step 4. Remove the PC card from the PC card slot.

Note: *Removing the PC card from the slot without following the above instructions may cause a system malfunction.*

5.1. LAN Card Configuration

A LAN card will create two network interfaces and two IP addresses. The users can assign a valid IP address to each serial port. The IP address must be valid in the STS Series built-in network interface or the environment of STS Series PC card LAN interface environment.

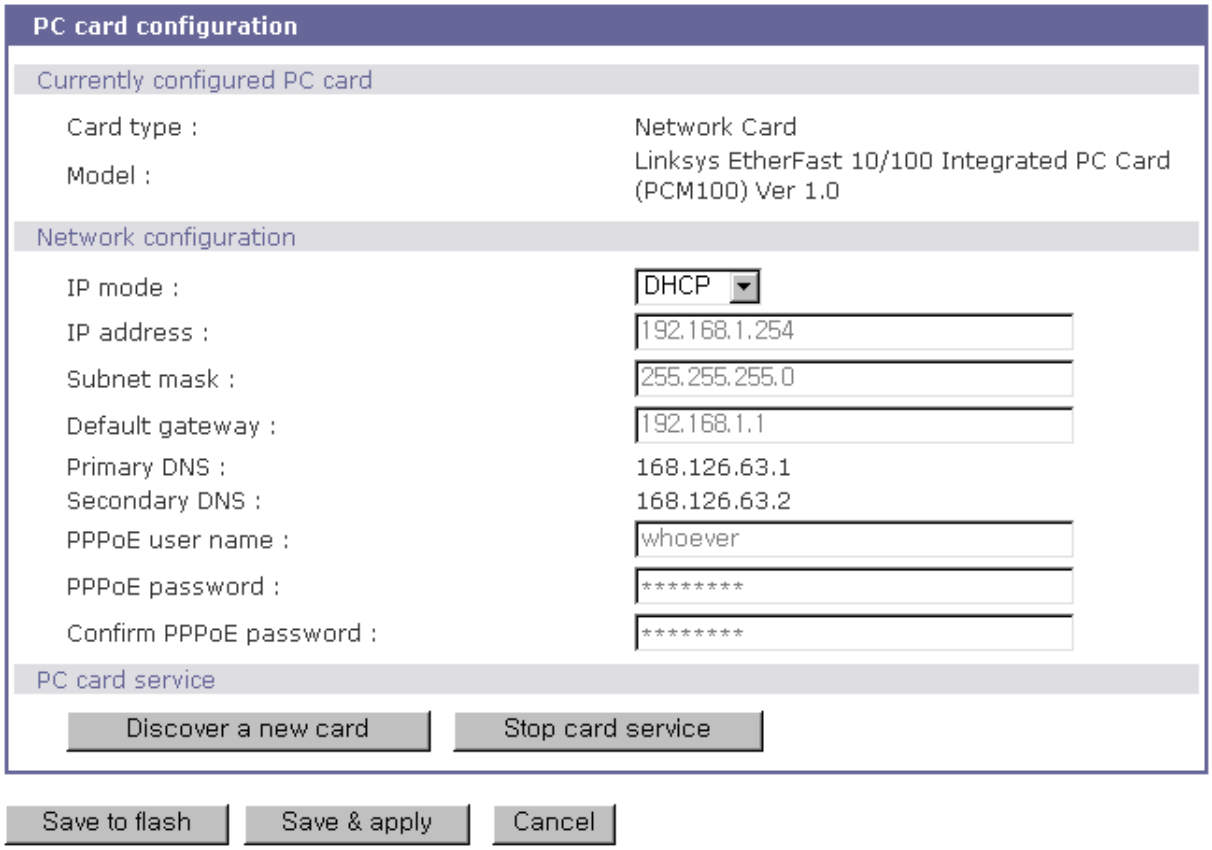


Figure 5-3 PC LAN card configuration

The user must manually select PC LAN card as the card type and set the primary and secondary DNS servers when configuring a PC LAN card. All other configuration steps are the same as detailed in Section *3.1 IP Configuration*.

Refer to *Appendix B.PC Card supported by STS Series to view* a list of LAN PC cards supported by the STS Series.

5.2. Wireless LAN Card Configuration

A wireless LAN card will result in two network interfaces and two IP addresses. The user can assign a valid IP address to each serial port. The IP address must valid in the STS Series built-in network interface or in the wireless LAN interface environment.

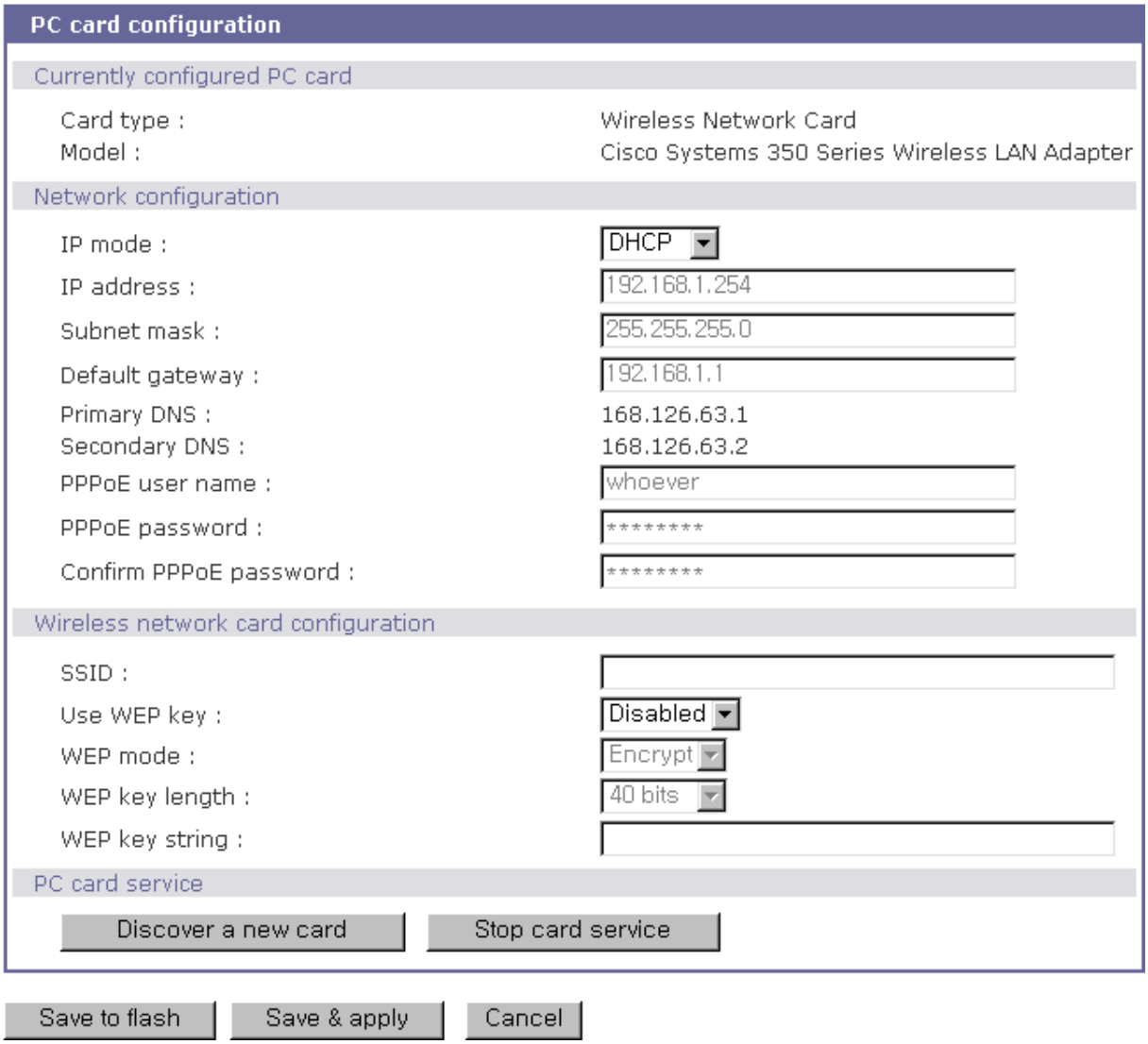


Figure 5-4 PC wireless LAN card configuration

The user must manually select WIRELESS LAN CARD as the card type and set the primary and secondary DNS servers when configuring a PC LAN card. All other configuration steps are the same as detailed in Section *3.1 IP Configuration*.

The STS Series supports SSID(Service Set Identifier) and WEP(Wired Equivalent Privacy) key features for the wireless LAN configuration. The user may configure the SSID to specify an AP (Access Point). The user may also configure the WEP mode as either encrypted or shared. The WEP key length must be either 40 or 128 bits. The 40-bit WEP key length requires the user to enter 5 hexadecimal code sets without colons (:). The 128 bits WEP key length requires the user to enter 13 hexadecimal code sets without colons (:).

For example, to use the 128 bits WEP key length option, the user must enter 13 hexadecimal code sets as follows:

000F25E4C2000F25E4C2000F24

Refer to Appendix B.PC Card supported by STS Series to view a list of wireless LAN cards supported by the STS Series.

5.3. Serial Modem Card Configuration

Using the extra PC card slot as a modem will allow the user on-line access without tying up a serial port with an external modem. Most 56Kbps PC serial modem cards are compatible with the PC card slot. A complete catalog of modem cards supported by the STS Series is listed in *Appendix B*.

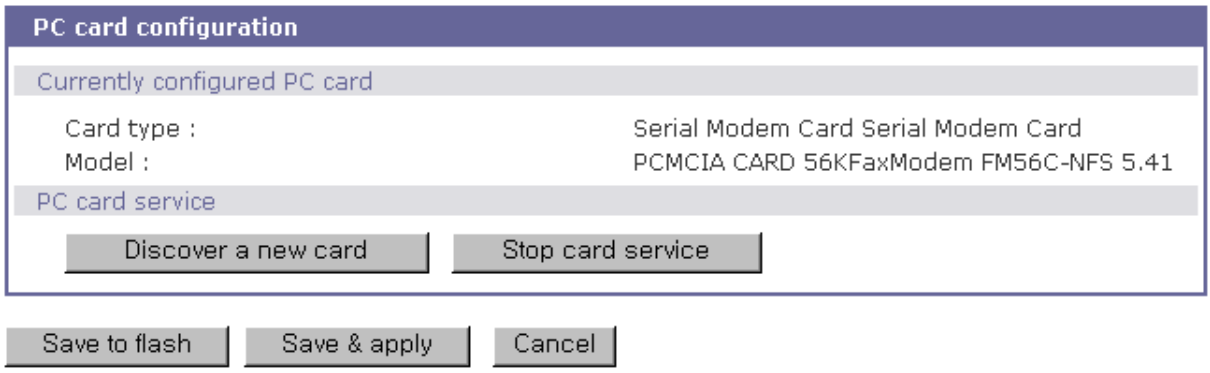


Figure 5-5 PC serial modem card configuration

5.4. ATA/IDE Fixed Disk Card Configuration

The user must configure the total data size required to use the PC ATA/IDE fixed disk card to store the system and serial port log. The STS Series will automatically locate the total storage size and the disk space available on the disk.

The user may delete all the files currently on the card by selecting Delete.

The user may select Format to format the card. The STS Series supports both **EXT2** and **VFAT** file systems for the disk card.

The user may store or retrieve the STS Series configuration files to/from the disk by exporting/importing the STS Series configuration.

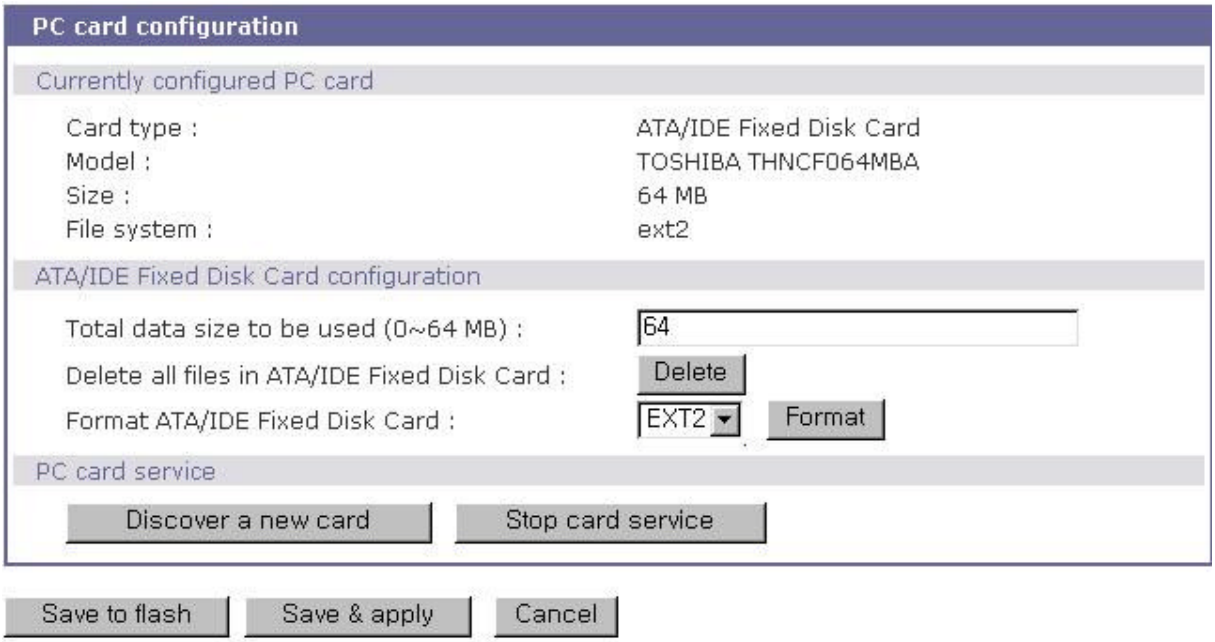


Figure 5-6 PC ATA/IDE fixed disk card configuration

6. System Administration

The STS Series display the system status and the log data via a Status Display Screen. This screen is to be used for management purposes. System status data includes the model name, serial number, firmware version and the network configuration of the STS Series. The STS Series can also be configured to deliver log data automatically via email to a specified recipient with the system-logging feature.

The users can configure the STS Series's device name, date and time settings, and reload factory default settings in this menu group. The users can also upgrade the firmware of the STS Series using the web interface, remote consoles or serial console.

6.1. System Status

Figure 6-1 System status display

6.2. System Logging

The STS Series provides both the system logging feature and the system log status display. The user may configure the STS Series to enable or disable the system logging process, the system log buffer size, as well as select the log storage location.

- **System log storage location**

  The system log can be stored in the **STS Series internal memory**, the **ATA/IDE fixed disk card** inserted in PCMCIA slot, the **mounting point on an NFS server** or the **SYSLOG server**. If the internal memory is used to store system log data, the log data will be cleared when the STS Series is turned off. To preserve the system log data, set the storage location to be the ATA/IDE fixed disk card, SYSLOG server or NFS server. To do this, the user must configure the corresponding media in advance. Unless the media is properly set up, the user will not be able to select a storage location from the interface.

- **System log buffer size**

  This parameter defines the maximum amount of system log data that can be logged. When using internal memory to store data, the total size of the system log cannot exceed 300 Kbytes.

  When using an ATA/IDE fixed disk card to store log data, the maximum buffer size is dependent upon the card capacity.

  When using an NFS server to store logs data, the maximum buffer size is unlimited. The user should configure the NFS server to ensure that the port logging system works properly.

  When using the SYSLOG server to store log data, the user cannot set the buffer size.

  The STS Series can also be configured to send log data automatically if the number of logs unsent reaches a pre-defined number. If enabled, the user must set parameters to initiate the creation of a email. These parameters would include the number of logs required to trigger an email, the recipient email address, etc. *Figure 6-2* shows the configuration and system log view screen.

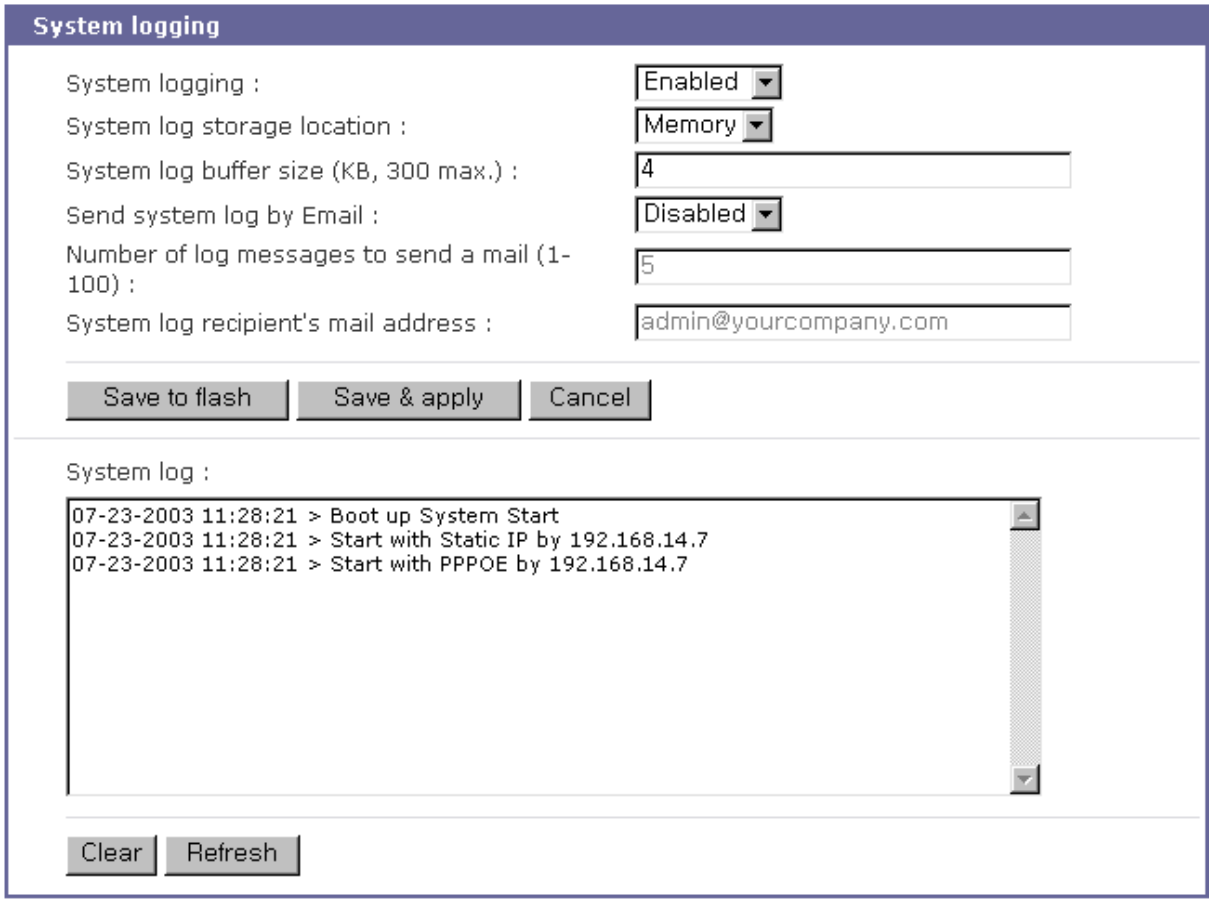


Figure 6-2 System log configuration and view

6.3. User Logged on List

This function allows a user to view current and historical user activity on the shell of STS Series.

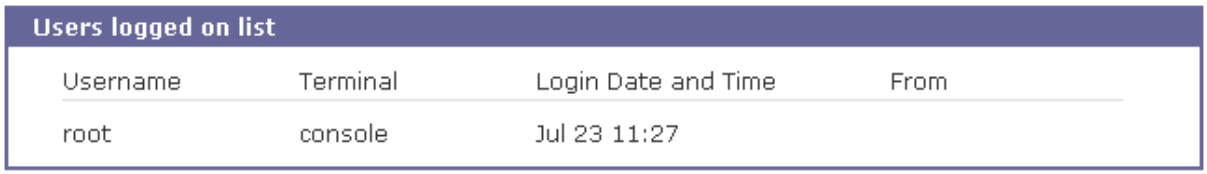


Figure 6-3 User logged on list

The list displays the following information for users who have logged into the system:

User name

Terminal type for the session

Time connected

IP address of the remote host

Note: Users access via the web will not appear on the list. Connections are not always made using HTTP/HTTPS protocol.

6.4. Change Password

The password for the administrative users of STS Series can be changed.

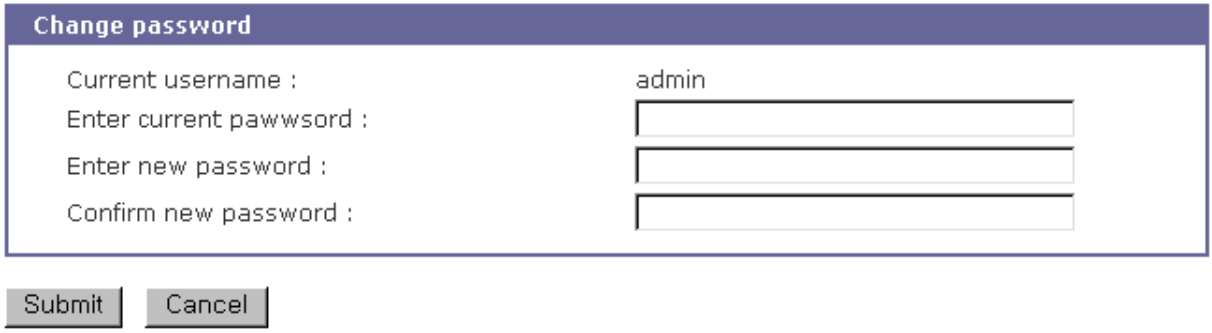


Figure 6-4 Changing the password

6.5. Device Name Configuration

The STS Series has its own name for administrative purposes. *Figure 6-5* shows the device name configuration screen. When user changes Device name, hostname of STS series shall be changed and then prompt on CLI also shall be changed to the corresponding one as follows,

```
root@SS800_Device:~#
```

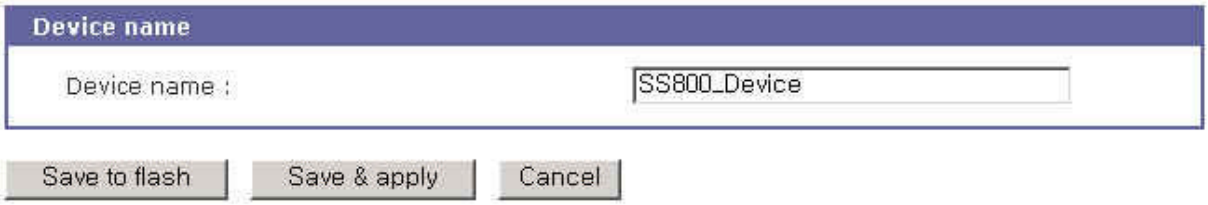


Figure 6-5 Device name configuration

Please note that user cannot set space character as one of device name. And If user sets blank as Device name then hostname is set as IP address of STS series automatically.

And also the device name is utilized for management program, HelloDevice Manager.

6.6. User Administration

The STS Series support user authentication for port access. The user for port access can be configured User Administration menu as shown on *Figure 6-6 User Administration*. When user is added, it is possible to assign ports for the user to be allowed to access. Please note that user authentication should be enabled to enable user authentication for port and user authentication is

applied only for TCP mode. (Please also refer to the descriptions for User Authentication in Section 4.2.4.1 TCP mode)

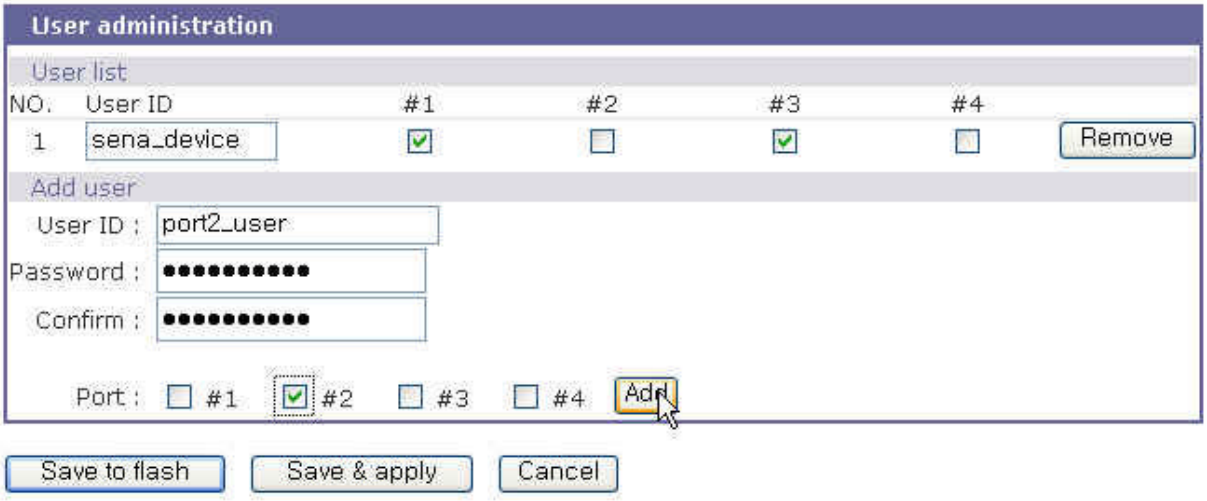


Figure 6-6 User Administration

6.7. Date and Time Settings

The STS Series maintains current date and time information. The STS Series clock and calendar settings are backed up by internal battery power. The user can change the current date and time, as shown in *Figure 6-7*.

There are two date and time settings. The first is to use the NTP server to maintain the date and time settings. If the NTP feature is enabled, the STS Series will obtain the date and time information from the NTP server at each reboot. If the NTP server is set to 0.0.0.0, the STS Series will use the default NTP servers. In this case, the STS Series should be connected from the network to the Internet. The user may also need to set the time offset from UTC depending on the users' location.

The second method is to set date and time manually without using the NTP server. This will allow the date and time information to be kept maintained by the internal battery backup.

The users may also need to set the timezone and the time offset from UTC depending on the users' location to set system date and time exactly. If the user uses daylight saving time, the user may need to set the daylight saving time properties such as the daylight saving timezone, the time offset from UTC, start data and time, end date and time. It allows the SS110/400/800 to calculate the exact system time.

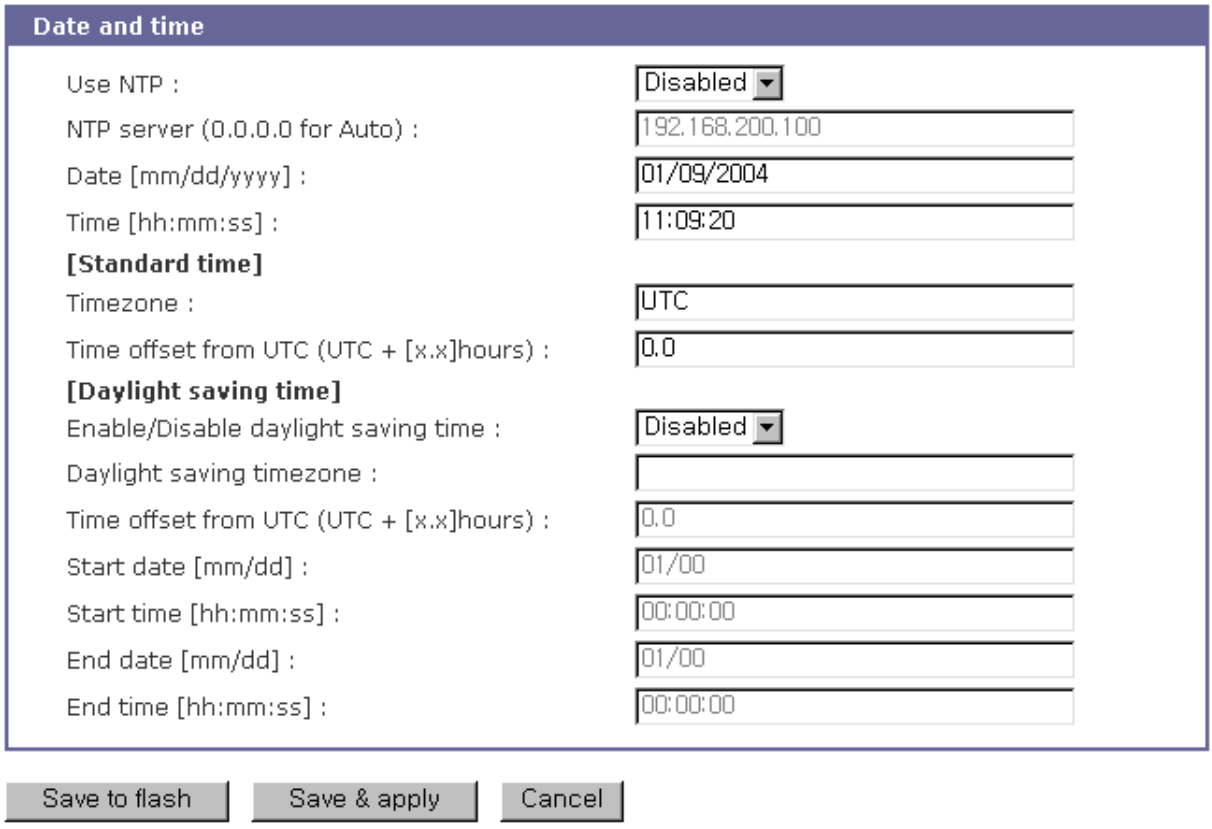


Figure 6-7 Date and time configuration

6.8. Configuration management

The user may export the current configurations to a file at such locations as CF card, NFS server, user space or local machine and import the exported configurations as current configurations from CF card, NFS server, user space or local machine.

The user may restore the factory default settings at any time by selecting “Factory default” at location property at the import part or by pushing the factory default reset switch on the back panel of the STS series. *Figure 6-8* shows the configuration management screen. The following parameters should be properly set up to export / import configurations:

Configuration export

Location : Location to export to.

Encrypt : Yes or No.

File name

Configuration import

Location : Location to import from. By selecting **Factory default**, the user may restore the factory settings.

Configuration selection : Determines what kinds of configurations are imported.

Encrypt : Yes or No. If location is Factory default, it has no effects.

File selection : List all the exported files satisfying the encrypting option at the selected location which is one of CF card, NFS server and user space.

Local : Helps to browse the exported file at local machine if location is local machine.

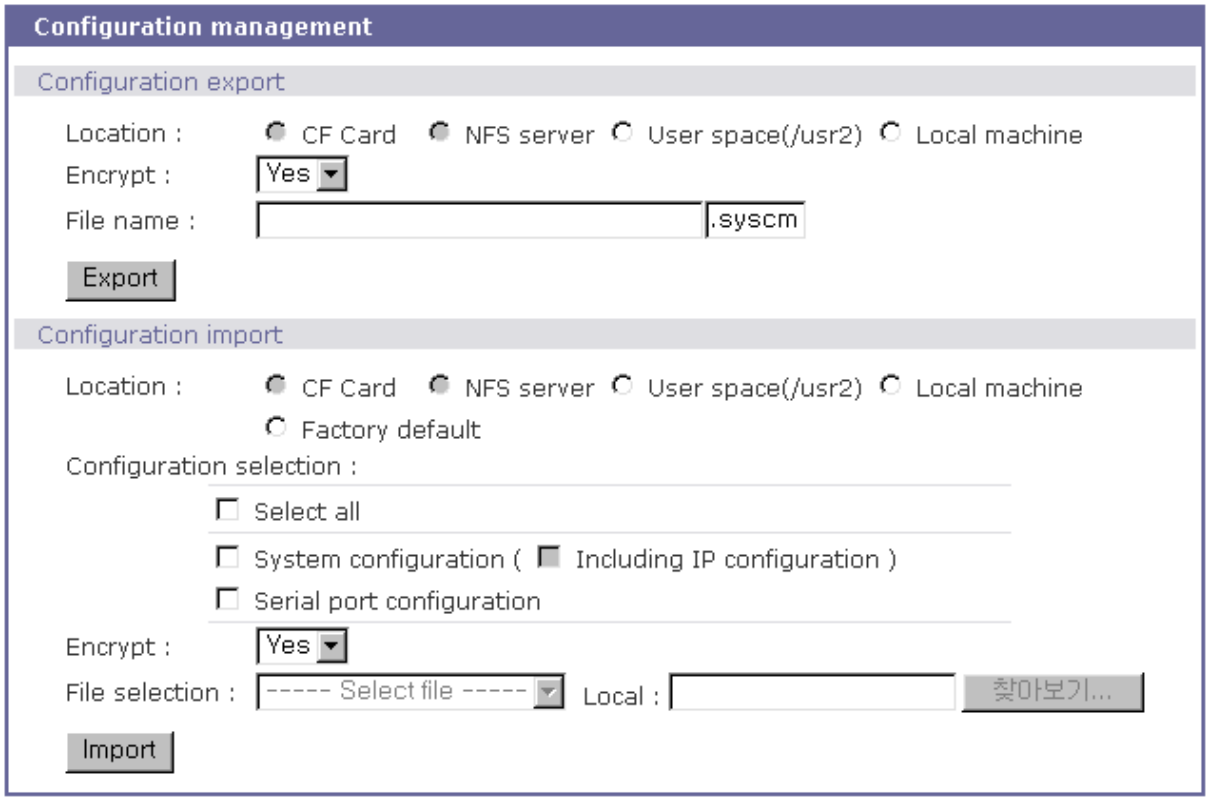


Figure 6-8 Configuration management

To export the current configurations, follow this:

1. Select the location to export to.
2. Select the encrypting option
3. Type the file name.
4. Click the [Export] button.

To import the exported configurations, follow this:

1. Select the location to import from.
2. Select the configurations to import.
3. Select the encrypting option.
4. Select the file to import from the file selection list box if location is not local machine nor factory default.
5. Select the file to import by clicking browse button if location is local machine.

6. Click the [Import] button.

6.9. Firmware Upgrade

Firmware upgrades are available via serial, remote console or web interface. The latest upgrades are available on the Sena web site at http://www.sena.com/support/downloads/.

Figure 6-9 shows the firmware upgrade web interface.

To upgrade firmware via the web:

1. Select the latest firmware binary by clicking browse button.
2. Select and upload the selected version.
3. Once the upgrade has been completed, the system will reboot to apply the changes.

Figure 6-9 Firmware upgrade

To use either a remote or serial console to upgrade your firmware, the TELENT/SSH or terminal emulation program must support Zmodem transfer protocol. After the firmware upgrade, the previous settings will be reset to the factory default settings, except the IP configuration settings.

To upgrade firmware via a remote console:

1. Obtain the latest firmware.
2. Connect the terminal emulation program using either TELENT/SSH or a serial console port.
   (TELNET or SSH is recommended since the process of firmware upgrade by serial console requires extremely long time.)
3. Select from the firmware upgrade menu as shown *Figure 6-10*.
4. Follow the online directions and transfer the firmware binary file using the Zmodem protocol as shown in *Figure 6-11*.
5. Once the upgrade has been completed, the system will reboot to apply the changes

6. If the firmware upgrade fails, the STS Series will display error messages as shown in *Figure 6-12*. It will also maintain the current firmware version.

```
 Login : admin
 Password : *****

--------------------------------------------------------------------------------
 Welcome to STS-800 configuration page
 Current time: 07/23/2003 15:04:07    F/W REV.:    v1.0.0
 Serial No.:   STS800438349-42944     MAC address: 00-01-95-04-19-5a
 IP mode:     Static IP            IP address:  192.168.14.7
--------------------------------------------------------------------------------
 Select menu:
  1. Network configuration
  2. Serial port configuration
  3. PC Card configuration
  4. System administration
  5. Save changes
  6. Exit without saving
  7. Exit and apply changes
  8. Exit and reboot
  <Enter> Refresh
---> 4
--------------------------------------------------------------------------------
 System administration
--------------------------------------------------------------------------------
 Select menu:
  1. System status
  2. System logging
  3. Device name: SS800 Device
  4. Date and time
  5. Change password
  6. User file upload
  7. Reload factory default settings
  8. Reload factory default settings except IP settings
  9. Firmware upgrade
  <ESC> Back, <Enter> Refresh
--->9
Do you want to upgrade firmware? (y/n): y
Transfer firmware by zmodem using your terminal application.
To escape, press Ctrl+X
**B0ff000005b157
```

Figure 6-10 Firmware upgrade using remote/serial console

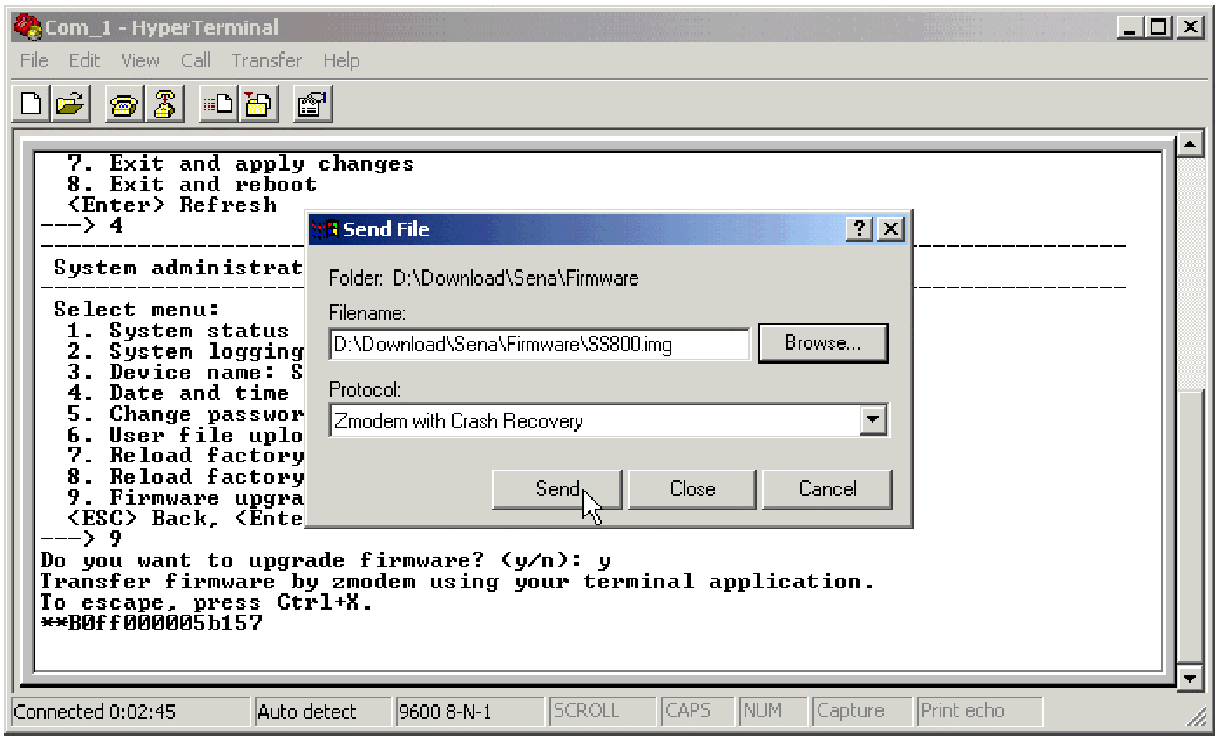


Figure 6-11 Transfer binary file by Zmodem (HyperTerminal)

```
--->9
Do you want to upgrade firmware? (y/n): y
Transfer firmware by zmodem using your terminal application.
To escape, press Ctrl+X
**B0ff000005b157
**B0ff000005b157
**B0ff000005b157
**B0ff000005b157
Firmware upgrade failed !
 Now reboot ...
```

Figure 6-12 Firmware upgrade failure message

6.10. User File Uploading

User can upload his own file to the STS Series. But file uploading feature is only supported in console menu. File uploading menu is located under "*4. System administration --> 6. User file upload*" of console menu as shown on *Figure 6-13*.

To upload user file to the SS110/400/800 using console menu, user must use a TELNET/SSH or terminal emulation program which supports Zmodem transfer protocol.

Uploading procedure is similar to firmware upgrade using console menu as follows,

1. Prepare user file to be uploaded.
2. Connect the terminal emulation program using either TELNET/SSH or a serial console port.

(TELNET or SSH is recommended since the process of firmware upgrade by serial console. Using a serial console port may take a long time.)

3. Select from the user file upload menu as shown *Figure 6-13*.
4. Follow the online directions and transfer the user file using the Zmodem protocol as shown in *Figure 6-11*.
5. Once the upload has been completed, the system will display success messages as shown in *Figure 6-13*.
6. If the upload fails, the STS Series will display error messages as shown in *Figure 6-14*.

Note :

User file uploading is permitted only under user space (/usr2) directory. For more information about file system inside STS Series, please refer to *8.2 Flash partition* section.

```
------------------------------------------------------------------------------
 Welcome to STS-800 configuration page
 Current time: 08/14/2003 11:56:13     F/W REV.    : v1.0.0
 Serial No.  : STS800438349-42944      MAC address: 00-01-95-04-d3-03
 IP mode     : DHCP                    IP address : 192.168.222.206
------------------------------------------------------------------------------
 Select menu:
  1. Network configuration
  2. Serial port configuration
  3. PC Card configuration
  4. System administration
  5. Save changes
  6. Exit without saving
  7. Exit and apply changes
  8. Exit and reboot
  <Enter> Refresh
---> 4
------------------------------------------------------------------------------
 System administration
------------------------------------------------------------------------------
 Select menu:
  1. System status
  2. System logging
  3. Device name: STS800 Device
  4. Date and time
  5. Change password
  6. User file upload
  7. Reload factory default settings
  8. Reload factory default settings except IP settings
  9. Firmware upgrade
  <ESC> Back, <Enter> Refresh
---> 6
 Do you want to upload a file to user space? (y/n): y
 Enter a filename: test.txt
 The file will be saved as /usr2/test.txt.
 Transfer a file by zmodem using your terminal application.
 To escape, press Ctrl+X.
**B01ff000005b157

Uploading a file is completed.
```

Figure 6-13 User file upload menu and success messages

```
Do you want to upload a file to user space? (y/n): y
 Enter a filename: test.txt
 The file will be saved as /usr2/test.txt.
 Transfer a file by zmodem using your terminal application.
 To escape, press Ctrl+X.
**B01ff000005b157

Uploading a file failed.
```

Figure 6-14 User file upload fail messages

7. System Statistics

The STS Series Web interface provides system statistics menus. The user can use the menus to access statistical data and tables stored in the STS Series memory. Network interfaces statistics and serial ports statistics display statistical usage of the link layer, **lo**, **eth** and serial ports. IP, ICMP, TCP and UDP statistics display usages of four primary components in the TCP/IP protocol suite.

7.1. Network Interfaces Statistics

Network interfaces statistics display basic network interfaces usage of the STS Series, **lo** and **eth0**. **lo** is a local loop back interface and **eth0** is a default network interface of STS Series.

Network interfaces statistics

Interface		lo	eth0
Receive	Bytes	680	7448861
	Packets	8	8057
	Errors	0	0
	Drop	0	0
	FIFO	0	0
	Frame	0	0
	Compressed	0	0
	Multicast	0	0
Transmit	Bytes	680	766794
	Packets	8	3991
	Errors	0	0
	Drop	0	0
	FIFO	0	0
	Frame	0	330
	Compressed	0	0
	Multicast	0	0

Figure 7-1 Network interfaces statistics

7.2. Serial Ports Statistics

Serial ports statistics display the usage history of 32 serial ports, baud rate configurations and each port's pin status. (: On : Off)

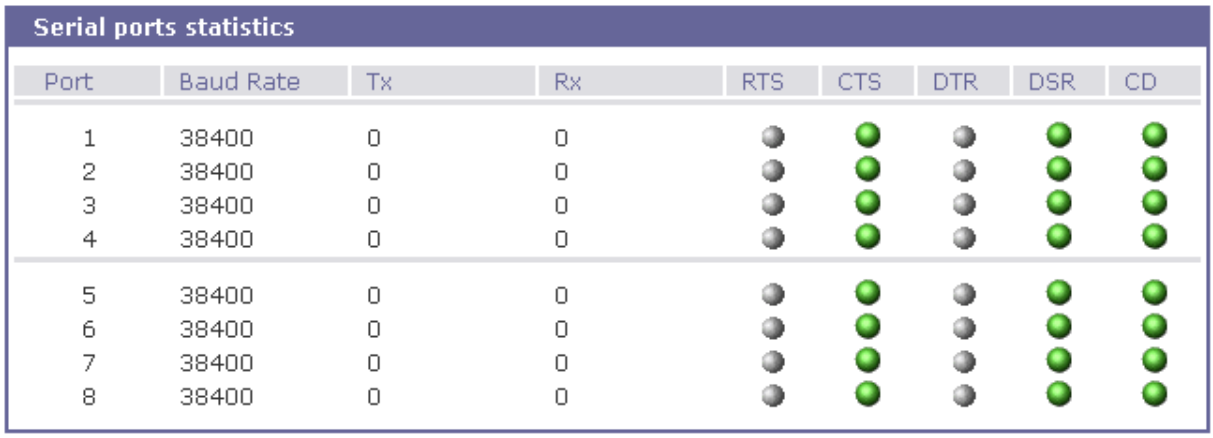


Figure 7-2 Serial ports status

7.3. IP Statistics

The IP Statistics screen provides statistical information about packets/connections using an IP protocol. Definitions and descriptions of each parameter are described below:

Forwarding :

Specifies whether IP forwarding is enabled or disabled.

DefaultTTL :

Specifies the default initial time to live (TTL) for datagrams originating on a particular computer.

InReceives :

Shows the number of datagrams received.

InHdrErrors :

Shows the number of datagrams received that have header errors. Datagrams Received Header Errors is the number of input datagrams discarded due to errors in their IP headers, including bad checksums, version number mismatch, other format errors, time-to-live exceeded, errors discovered in processing their IP options, etc.

InAddrErrors :

Specifies the number of datagrams received that have address errors. These datagrams are discarded because the IP address in their IP header's destination field was not a valid address to be received at this entity. This count includes invalid addresses (for example, 0.0.0.0) and addresses of unsupported Classes (for example, Class E).

ForwDatagrams :

Specifies the number of datagrams forwarded.

InUnknownProtos :

Specifies the number of locally addressed datagrams received successfully but discarded because of an unknown or unsupported protocol.

InDiscard :

Specifies the number of input IP datagrams for which no problems were encountered to prevent their continued processing, but which were discarded (for example, for lack of buffer space). This counter does not include any datagrams discarded while awaiting reassembly.

InDelivers :

Specifies the number of received datagrams delivered.

OutRequests :

Specifies the number of outgoing datagrams that an IP is requested to transmit. This number does not include forwarded datagrams.

OutDiscards :

Specifies the number of datagrams for which no route could be found to transmit them to the destination IP address. These datagrams were discarded. This counter includes any packets counted in Datagrams Forwarded that meet this "no route" criterion.

ReasmTimeout :

Specifies the amount of time allowed for all pieces of a fragmented datagram to arrive. If all pieces do not arrive within this time, the datagram is discarded.

ReasmReqds :

Specifies the number of datagrams that require reassembly.

ReasmOKs :

Specifies the number of datagrams that were successfully reassembled.

ReasmFails :

Specifies the number of datagrams that cannot be reassembled.

FragOKs :

Specifies the number of datagrams that were fragmented successfully.

FragFails :

Specifies the number of datagrams that need to be fragmented but couldn't be because the IP header specifies no fragmentation. For example, if the datagrams "Don't Fragment" flag was set, the datagram would not be fragmented. These datagrams are discarded.

FragCreates :

Specifies the number of fragments created.

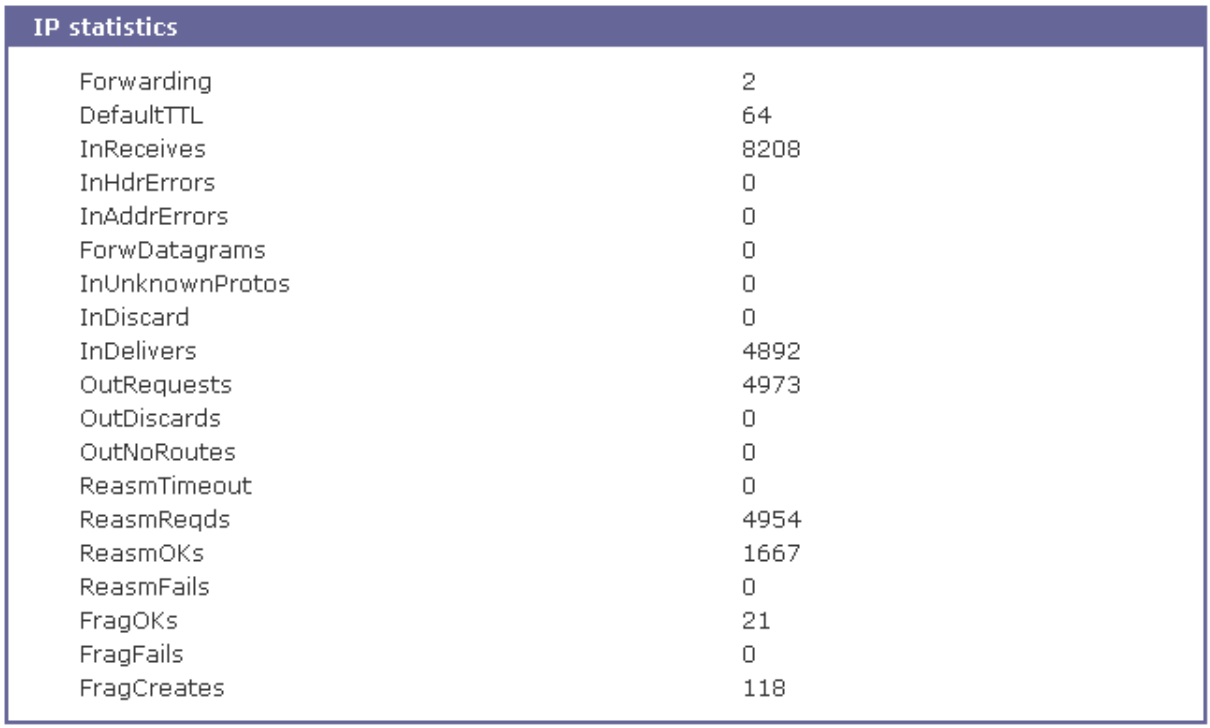


Figure 7-3 IP statistics

7.4. ICMP Statistics

The ICMP Statistics screen provides statistical information about packets/connections using an ICMP protocol. Definitions and descriptions of each parameter are described below:

InMsgs, OutMsgs :

Specifies the number of messages received or sent.

InErrors, OutErrors :

Specifies the number of errors received or sent.

InDestUnreachs, OutDestUnreachs :

Specifies the number of destination-unreachable messages received or sent. A destination-unreachable message is sent to the originating computer when a datagram fails to reach its intended destination.

InTimeExcds, OutTimeExcds :

Specifies the number of time-to-live (TTL) exceeded messages received or sent. A time-to-live exceeded message is sent to the originating computer when a datagram is discarded because the number of routers it has passed through exceeds its time-to-live value.

InParmProbs, OutParmProbs :

Specifies the number of parameter-problem messages received or sent. A parameter-problem message is sent to the originating computer when a router or host detects an error in a datagram's IP header.

InSrcQuenchs, OutSrcQuenchs :

Specifies the number of source quench messages received or sent. A source quench request is sent to a computer to request that it reduces its rate of packet transmission.

InRedirects, OutRedirects :

Specifies the number of redirect messages received or sent. A redirect message is sent to the originating computer when a better route is discovered for a datagram sent by that computer.

InEchos, OutEchos :

Specifies the number of echo requests received or sent. An echo request causes the receiving computer to send an echo reply message back to the originating computer.

NEchoReps, OutEchoReps :

Specifies the number of echo replies received or sent. A computer sends an echo reply in response to receiving an echo request message.

InTimestamps, OutTimestamps :

Specifies the number of time-stamp requests received or sent. A time-stamp request causes the receiving computer to send a time-stamp reply back to the originating computer.

InTimestampReps, OutTimestampReps :

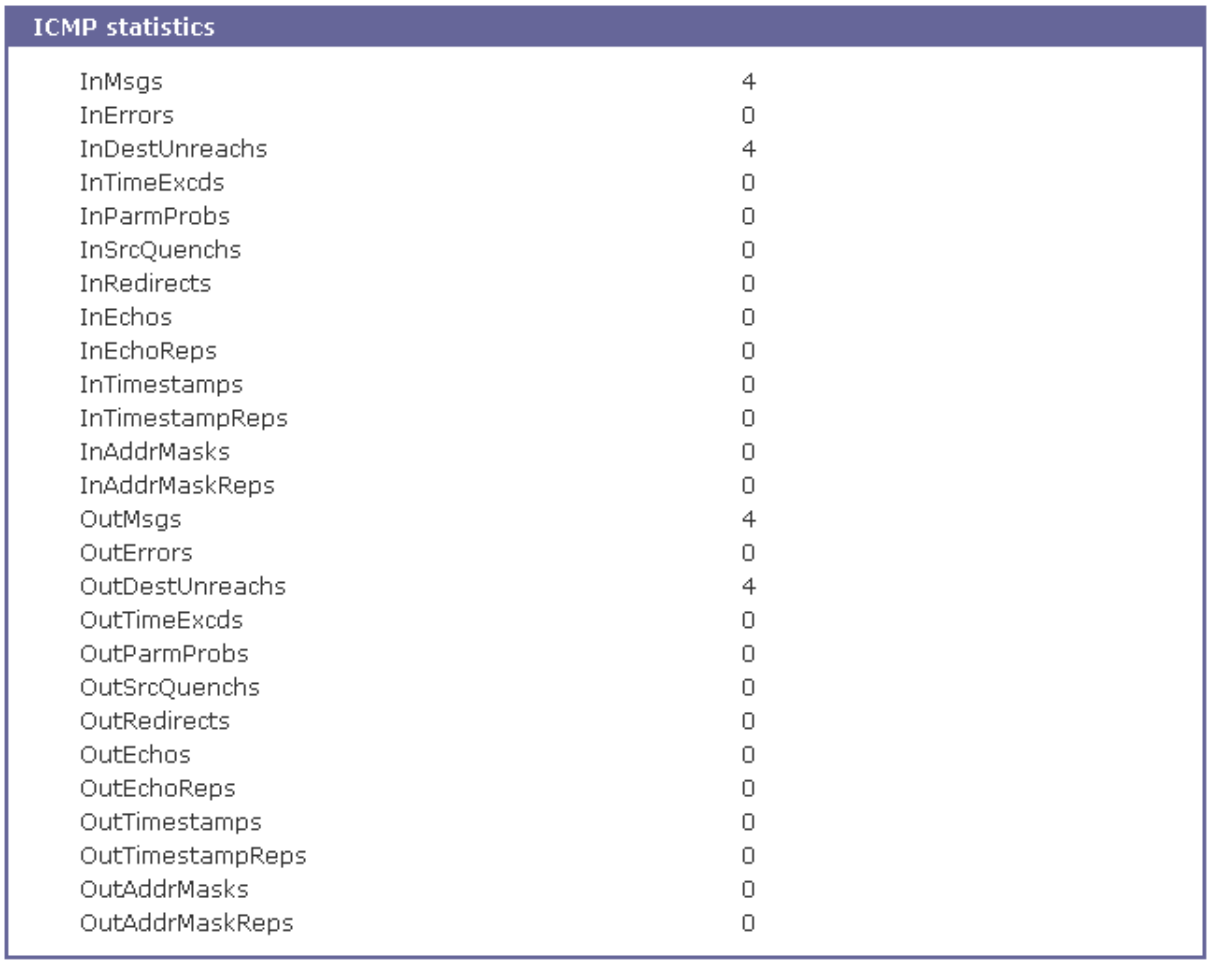
Specifies the number of time-stamp replies received or sent. A computer sends a time-stamp reply in response to receiving a time-stamp request. Routers can use time-stamp requests and replies to measure the transmission speed of datagrams on a network.

InAddrMasks, OutAddrMasks :

Specifies the number of address mask requests received or sent. A computer sends an address mask request to determine the number of bits in the subnet mask for its local subnet.

InAddrMaskReps, OutAddrMaskReps :

Specifies the number of address mask responses received or sent. A computer sends an address mask response in response to an address mask request.



ICMP statistics	
InMsgs	4
InErrors	0
InDestUnreachs	4
InTimeExcds	0
InParmProbs	0
InSrcQuenchs	0
InRedirects	0
InEchos	0
InEchoReps	0
InTimestamps	0
InTimestampReps	0
InAddrMasks	0
InAddrMaskReps	0
OutMsgs	4
OutErrors	0
OutDestUnreachs	4
OutTimeExcds	0
OutParmProbs	0
OutSrcQuenchs	0
OutRedirects	0
OutEchos	0
OutEchoReps	0
OutTimestamps	0
OutTimestampReps	0
OutAddrMasks	0
OutAddrMaskReps	0

Figure 7-4 ICMP statistics

7.5. TCP Statistics

The TCP Statistics screen provides statistical information about packets/connections using a TCP protocol. Definitions and descriptions of each parameter are described below:

RtoAlgorithm :

Specifies the retransmission time-out (RTO) algorithm in use. The Retransmission Algorithm can have one of the following values.

0 : CONSTANT - Constant Time-out

1: RSRE - MIL-STD-1778 Appendix B

2: VANJ - Van Jacobson's Algorithm

3: OTHER - Other

RtoMin :

Specifies the minimum retransmission time-out value in milliseconds.

RtoMax :

Specifies the maximum retransmission time-out value in milliseconds.

MaxConn :

Specifies the maximum number of connections. If is the maximum number is set to -1, the maximum number of connections are dynamic.

ActiveOpens :

Specifies the number of active opens. In an active open, the client is initiating a connection with the server.

PassiveOpens :

Specifies the number of passive opens. In a passive open, the server is listening for a connection request from a client.

AttemptFails :

Specifies the number of failed connection attempts.

EstabResets :

Specifies the number of established connections that have been reset.

CurrEstab :

Specifies the number of currently established connections.

InSegs :

Specifies the number of segments received.

OutSegs :

Specifies the number of segments transmitted. This number does not include retransmitted segments.

RetransSegs :

Specifies the number of segments retransmitted.

RetransSegs :

Specifies the number of errors received.

OutRsts :

Specifies the number of segments transmitted with the reset flag set.

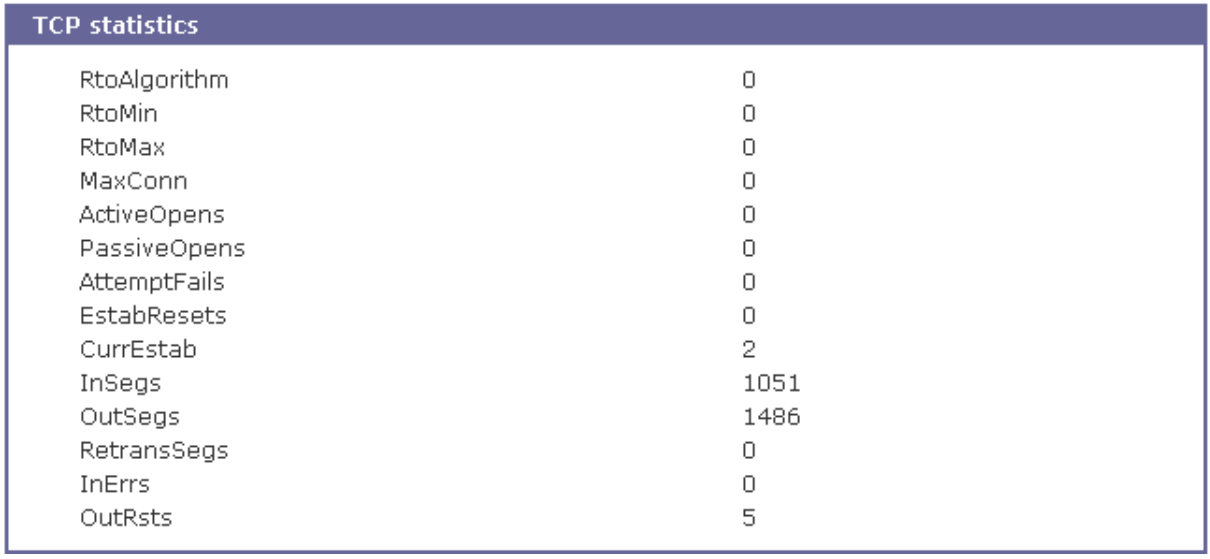


Figure 7-5 TCP statistics

7.6. UDP Statistics

The UDP Statistics screen provides statistical information about packets/connections using a UDP protocol. Definitions and descriptions of each parameter are described below:

InDatagrams :

Specifies the number of datagrams received.

NoPorts :

Specifies the number of received datagrams that were discarded because the specified port was invalid.

InErrors :

Specifies the number of erroneous datagrams that were received. Datagrams Received Errors is the number of received UDP datagrams that could not be delivered for reasons other than the lack of an application at the destination port.

OutDatagrams :

Specifies the number of datagrams transmitted.

Figure 7-6 UDP statistics

8. CLI guide

8.1. Introduction

The STS Series **root** or **System Administrator** (only **admin** account is added for this group user by factory default) can access the Linux console command line interface (CLI) of the STS Series via the serial console or TELENT/SSH. In the CLI, the authorized user can perform standard Linux commands to view the status of the STS Series, edit the configuration, apply configuration changes, define user scripts and transmit files between the STS Series and remote hosts.

The STS Series provides 1024 KB user space mounted in `/usr2` for read/write capabilities in its internal flash memory. Using the user space, the user can create his own scripts or executable binaries to customize the STS Series.

A **root** user will always have access to the CLI through the serial console on the STS Series back panel or by using a Telnet client from their workstation.

A **System Administrator** cannot have access to the CLI. He can only access console configuration menu or Web UI.

8.2. Flash partition

The STS Series internal flash is partitioned as shown in the table below. The users can freely access the Mtdblock5 which is mounted on /usr2 for their own needs. The user can also access files at /etc, /var, and /temp at their own risk. Simply accessing these files will not affect the STS Series after rebooting. However, if the user invokes the command saveconf, the changes in the configuration file will be committed to the internal flash memory area of the STS Series. This will result in the changes being kept after the reboot sequence. Invalid configuration changes can affect the STS Series behavior. At worst, it may cause the STS Series to be inoperable.

Block	Type	Mount point	Size (KB)
Mtdblock0	Bootloader	None	128
Mtdblock1	Kernel	None	768
Mtdblock2	CRAMFS (Read only)	/	6080
Mtdblock3	Ram disk image (4MB)	/etc, /var, /tmp	64
Mtdblock4	EXT2 (R/W)	/cnf (normally unmounted)	64
Mtdblock5	JFFS2 (R/W)	/usr2	1024
Mtdblock6	Reserved	None	64
Total			8192

8.3. Supported Linux Utilities

8.3.1. Shell & shell utilities:

sh, ash, bash, echo, env, false, grep, more, sed, which, pwd

8.3.2. File and disk utils:

ls, cp, mv, rm, mkdir, rmdir, ln, mknod, chmod, touch, sync,
gunzip, gzip, zcat, tar, dd, df, du, find, cat, vi, tail,
mkdosfs, mke2fs, e2fsck, fsck, mount, umount, **scp**

8.3.3. System utilities:

date, free, hostname, sleep, stty, uname, reset,
insmod, rmmod, lsmod, modprobe,
kill, killall, ps, halt, shutdown, poweroff, reboot, telinit, init,
useradd, userdel, usermod, whoami, who, passwd, id, su, who

8.3.4. Network utilities:

ifconfig, iptables, route, telnet, ftp, ssh, ping

8.4. Accessing CLI as root or system administrator

Serial console:

1) Connect the console port of the STS Series with the PC serial port
2) Run the PC terminal emulation program
3) Configure the PC serial port to: 9600-8-N-1 No flow control
4) Press <enter>
5) Login with the STS Series root or admin account

Telnet console:

1) `telnet` *`STS Series_ip_address`*

8.5. Editing STS Series configuration in CLI

8.5.1. Configuration file save/load mechanism:

1) While booting, the Super Series uncompresses `/cnf/cnf.tar.gz` to `/tmp/cnf/*` and unmounts `/cnf`
2) When changing the configuration, the STS Series changes the contents of the files in

/tmp/cnf

3) When the user saves the configuration, the STS Series mounts /cnf and compresses /tmp/cnf/* to /cnf/cnf.tar.gz (Web [Save to flash], or "saveconf" in CLI)

8.5.2. To change configuration in CLI:

To change the STS Series configuration in the CLI, run the menu-driven configuration utility "configmenu", or configure manually as follows:

1) Edit the configuration file manually using vi command
   (Please see *Appendix C. STS Series Configuration files* for detail descriptions of each parameter in the configuration files)
2) Save the configuration file to flash using the "saveconf" utility
3) Apply all changes using "applyconf" utility

```
root@192.168.0.117:~# configmenu

or

root@192.168.0.117:~# cd /tmp/cnf
root@192.168.0.117:/tmp/cnf# vi redirect.cnf
root@192.168.0.117:/tmp/cnf# saveconf
root@192.168.0.117:/tmp/cnf# applyconf
```

8.6. Running user defined scripts

Shell script /usr2/rc.user is automatically called when the STS Series is booting. Users can modify the rc.user file to run user defined script or binaries

```
#!/bin/bash
#
# rc.user : Sample script file for running user programs at boot time
#
#PATH=/bin:/usr/bin:/sbin:/usr/sbin
# Add shell command to execute from here

echo 'This is the welcome message defined by users'exit 0
```

8.7. File transmission

The users can use an ftp client for file transmission and use /usr2 directory for data read/write

```
root@192.168.0.117:~# cd /usr2
root@192.168.0.117:/usr2# ftp 192.168.2.3
Connected to 192.168.2.3.
220 lxtoo.senalab.co.kr FTP server (Version wu-2.6.1-16) ready.
Name (192.168.2.3:root): sena
331 Password required for sena.
```

```
Password:
230 User sena logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> get test.tgz
local: test.tgz remote: test.tgz
200 PORT command successful.
150 Opening BINARY mode data connection for test.tgz (350 bytes).
226 Transfer complete.
350 bytes received in 0.04 secs (9.6 kB/s)
ftp> bye
```

In addition to a regular FTP client, the user can copy files securely as encrypted using scp client program. If the user wants to copy a file from the STS Series(192.168.0.120) to user's PC, type a command on the user's PC as shown below:

```
[root@localhost work]# scp root@192.168.0.120:/usr2/rc.user /work
The authenticity of host '192.168.0.120 (192.168.0.120)' can't be established.
RSA key fingerprint is c1:70:ab:52:48:ab:e5:dc:47:9c:94:ed:99:6f:94:4f.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.0.120' (RSA) to the list of known hosts.
root@192.168.0.120's password:
rc.user           100% |*****************************|   173      00:00
[root@localhost work]#
```

8.8. Examples

8.8.1. Disabling the Telnet Port of the Unit

The STS Series unit does not support disabling the remote console port individually (port 22 for SSH or port 23 for Telnet to the box)

Currently, the user can only disable or enable all remote consoles together. This must be done using the UI or console configuration menu.

The user may bypass this and disable only one (Telnet or SSH) remote console by modifying the script 'rc.user'. Below are two examples of how this could be done.

Example1. Modify '`inetd.conf`'

Step 1 Modify `/etc/inetd.conf` (comment out or delete telnet service)

Step 2 Copy `inetd.conf` to `/usr2/inetd.conf`

Step 3 Edit `usr2/rc.user` script as follows:

```
#!/bin/bash
#
# rc.user : Sample script file for running user programs at boot time
#

#PATH=/bin:/usr/bin:/sbin:/usr/sbin

# Add shell command to execute from here
# Add shell command to execute from here
```

```
cp -a /usr2/inetd.conf /etc/inetd.conf
ps -ef
while killall inetd 2>/dev/null;
do sleep 1;
ps -ef
done
/usr/sbin/inetd
ps -ef

exit 0
```

The user may now disable the telnet service every time the system boots up.

Example 2. Run iptables rule

Step 1 Modify 'usr2/rc.user script as follows:

```
#!/bin/bash
#
# rc.user : Sample script file for running user programs at boot time
#

#!/bin/bash
#
# rc.user : Sample script file for running user programs at boot time
#

#PATH=/bin:/usr/bin:/sbin:/usr/sbin

# Add shell command to execute from here

# if user wants to disable telnet service from all host
iptables -A INPUT -p tcp -s --dport 23 -j DROP

# if user wants to enable telnet service only from specific hosts(192.168.0.0 ~
192.168.0.255)
#iptables -A INPUT -p tcp -s ! 192.168.0.1/255.255.255.0 --dport 23 -j DROP

exit 0
```

The user may now disable the telnet service every time the system boots up.
If the user resets the STS Series to the factory defaults, /usr2/rc.user script file will be renamed to /usr2/rc.user.old# file, and the default rc.user file will be restored.

8.8.2. Periodical program execution

User can use crontab to execute a specific program periodically. To enable periodical jobs using crontab, please complete following steps,

Step 1 Create a crontab file on /usr2 directory. Following sample crontab file generates

current_date file under /tmp directory and revise its contents every 2 minutes.

```
SHELL=/bin/bash
# Sample crontab job
# Run every two minutes
* * * * * echo `date` > /tmp/current_date
```

Step 2 Register crontab file using following command.

```
root@SS800_Device:/usr2# crontab samplecrontab_file
```

Step 3 To make cron job permanent for every system reboot, please use rc.user script as follows:

```
#!/bin/bash
#
# rc.user : Sample script file for running user programs at boot time
#

#PATH=/bin:/usr/bin:/sbin:/usr/sbin

# Add shell command to execute from here
crontab /usr/samplecrontab_file

exit 0
```

Please note that –e option (editing current crontab using editor) is not supported in STS. So user must user vi editor to change contents in crontab file.

For more information about the format of crontab file, please refer to Linux crontab manual(man 5 crontab).

9. User customization guide

9.1. Introduction

The STS Series supports various ways of customization so that user can fit STS Series for his own purpose. The STS Series provides following types of user customization methods,

- Periodical program execution
- User defined web pages
- Making and running user's own code.

9.2. Periodical program execution

User can use crontab to execute a specific program periodically. To enable periodical jobs using crontab, please complete following steps,

Step 1 Create a crontab file on /usr2 directory. Following sample crontab file generates current_date file under /tmp directory and revise its contents every 2 minutes.

```
SHELL=/bin/bash
# Sample crontab job
# Run every two minutes
* * * * * echo `date` > /tmp/current_date
```

Step 2 Register crontab file using following command.

```
root@SS800_Device:/usr2# crontab samplecrontab_file
```

Step 3 To make cron job permanent for every system reboot, please use rc.user script as follows:

```
#!/bin/bash
#
# rc.user : Sample script file for running user programs at boot time
#

#PATH=/bin:/usr/bin:/sbin:/usr/sbin

# Add shell command to execute from here
crontab /usr/samplecrontab_file

exit 0
```

Please note that –e option (editing current crontab using editor) is not supported in STS. So user must user vi editor to change contents in crontab file.

For more information about the format of crontab file, please refer to Linux crontab manual(man 5

crontab).

9.3. User defined web pages

STS Series supports user defined web pages. User can set user-defined page as a first page which will be popped up after user login to Web UI. For more information about changing default web page, please refer to *3.9 Web server configuration* section.

Once default web page is changed to *Customer page, Customer page* will be popped up after user logs in to Web UI. To change contents of *Customer page,* user must modify index.html or default CGI program. These two files are located under /usr2 directory of STS Series.

To change contents of index.html file, it is sufficient to change contents of index.html file. But to change contents of CGI program, user should compile source code of CGI program after modifying original source code. To compile source code of CGI program, user need cross development environment or SDK(Software Development Kit) for Super Series. Please contact Sena Technical Support to get SDK for Super Series or more information about cross development environment.

9.4. Making and running user's own code

To make user's own application code, cross development environment or SDK(Software Development Kit) for STS Series is needed. STS Series SDK is provided in the form of PC CF card. (Please contact Sena Technical Support to get SDK for Super Series or more information about cross development environment).

With STS Series SDK, user can make his own program in the CLI of Super Series.

For more detail information, please refer to Super Series customization guide.

With cross development environment, user can make his own program on his own Linux PC. And then he can upload his own program to STS Series. To run this program in STS Series, user can use user script file and/or crontab program. If the purpose of user program is manipulating of serial data, he can use filter application menu. For more information about configuring filter application, please refer to *4.2.8 Filter application* section.

Appendix 1. Connections

A 1.1. Ethernet Pin outs

The STS Series uses the standard Ethernet connector that is shielded connector compliant with AT&T258 specifications. *Table A-1* shows the pin assignment and wire color.

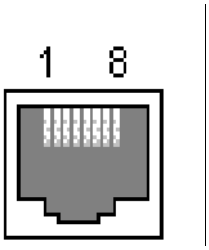


Figure A-1 Pin layout of the RJ45 connector

Table A-1 Pin assignment of the RJ45 connector for Ethernet

Pin	Description	Color
1	Tx+	White with orange
2	Tx-	Orange
3	Rx+	White with green
4	NC	Blue
5	NC	White with blue
6	Rx-	Green
7	NC	White with brown
8	NC	Brown

A 1.2. Console and Serial port pin-outs

The STS Series uses an RJ45 connector for console and serial ports. The pin assignment of the RJ45 connector for console and serial ports is summarized in *Table A-2*. Each pin has a function according to the serial communication type configuration.

Table A-2 Pin assignment of RJ45 connector for console and serial ports

Pin	RS232 (console and serial ports)
1	CTS
2	DSR
3	RxD
4	GND
5	DCD
6	TxD
7	DTR
8	RTS

A 1.3. Ethernet Wiring Diagram

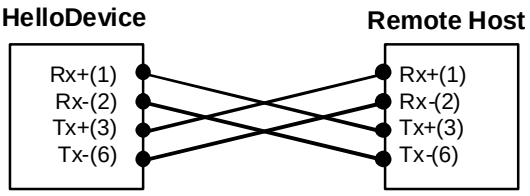


Figure A-2 Ethernet direct connection using crossover Ethernet cable

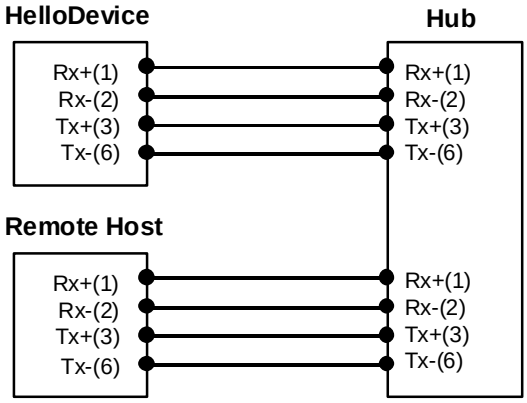


Figure A-3 Ethernet connection using straight through Ethernet cable

A 1.4. RS232 Serial Wiring Diagram

RJ45-DB9 female adapter

Using RJ45 to DB9(Female) ***Cross-over*** *Cable*

Description (RJ45)	Internal Cable Color	RJ45 Pin No.	DB9 Pin No.	Description (DB9)
CTS	Blue	1	7	RTS
DSR	Orange	2	4	DTR
RXD	Black	3	3	TXD
GND	Red	4	5	GND
DCD	Green	5	1	DCD
TXD	Yellow	6	2	RXD
DTR	Brown	7	6	DSR
RTS	White	8	8	CTS

RJ45-DB25 female adapter

Using RJ45 to DB25(Female) ***Cross-over*** *Cable*

Description (RJ45)	Internal Cable Color	RJ45 Pin No.	DB25 Pin No.	Description (DB25)
CTS	Blue	1	4	RTS
DSR	Orange	2	20	DTR
RXD	Black	3	2	TXD
GND	Red	4	7	GND
DCD	Green	5	8	DCD
TXD	Yellow	6	3	RXD
DTR	Brown	7	6	DSR
RTS	White	8	5	CTS

RJ45-DB25 male adapter

Using RJ45 to DB25(Male) ***Cross-over*** *Cable*

Description (RJ45)	Internal Cable Color	RJ45 Pin No.	DB25 Pin No.	Description (DB25)
CTS	Blue	1	4	RTS
DSR	Orange	2	20	DTR
RXD	Black	3	2	TXD
GND	Red	4	7	GND
DCD	Green	5	8	DCD
TXD	Yellow	6	3	RXD
DTR	Brown	7	6	DSR
RTS	White	8	5	CTS

RJ45-DB25 male adapter

Using RJ45 to DB25(Male) ***Straight*** *Cable*

Description (RJ45)	Internal Cable Color	RJ45 Pin No.	DB25 Pin No.	Description (DB25)
CTS	Blue	1	5	CTS
DSR	Orange	2	6	DSR
RXD	Black	3	3	RXD
GND	Red	4	7	GND
DCD	Green	5	8	DCD
TXD	Yellow	6	2	TXD
DTR	Brown	7	20	DTR
RTS	White	8	4	RTS

Appendix 2. PC card supported by STS

The following PC cards are supported by the STS Series series:

Table A-3 Network card

Manufacturer	Model/Name	STS probed Model name	Specification
3COM	3CXE589ET-AP	3Com Megahertz 589E TP/BNC LAN PC Card	10 Mbps LAN card
Linksys	Linksys EtherFast 10/100 Integrated PC Card (PCM100)	Linksys EtherFast 10/100 Integrated PC Card (PCM100) Ver 1.0	10/100 Mbps LAN card
Corega	FetherII PCC-TXD	corega K.K. corega FEtherII PCC-TXD	10/100 Mbps LAN card
Netgear	16bit PCMCIA Notebook Adapter FA411	NETGEAR FA411 Fast Ethernet	10/100 Mbps LAN card

Table A-4 Wireless Network card

Manufacturer	Model/Name	STS probed Model name	Specification
Cisco Systems	AIR-PCM340/Aironet 340	Cisco Systems 340 Series Wireless LAN Adapter	11 Mbps Wireless LAN Adapter
Lucent Technologies	PC24E-H-FC/Orinoco Silver	Lucent Technologies WaveLAN/IEEE Version 01.01	11 Mbps Wireless LAN Adapter
Lucent Technologies	PC24E-H-FC/Orinoco Gold	Lucent Technologies WaveLAN/IEEE Version 01.01	11 Mbps Wireless LAN Adapter
Agere Systems (Lucent Technologies)	Orinoco Classic Gold (PC24E-H-FC/Orinoco Gold)	Lucent Technologies WaveLAN/IEEE Version 01.01	11 Mbps Wireless LAN Adapter
Buffalo	AirStation (WLI-PCM-L11GP)	MELCO WLI-PCM-L11 Version 01.01	11 Mbps Wireless LAN Adapter

Table A-5 ATA/IDE Fixed Disk Card

Manufacturer	Model/Name	STS probed Model name	Specification
Advantech	CompactFlash	CF 48M	48 MB Storage card
SanDisk	SDP series	SunDisk SDP 5/3 0.6	64 MB Storage card
SanDisk	SDP series	SanDisk SDP 5/3 0.6	256 MB Storage card
Kingston	CompactFlash Storage Card	TOSHIBA THNCF064MAA	64 MB Storage card
Viking	CompactFlash	TOSHIBA THNCF064MBA	64 MB Storage card

Table A-6 Serial Modem Card

Manufacturer	Model/Name	STS probed Model name	Specification
Billionton Systems Inc.	FM56C series	PCMCIA CARD 56KFaxModem FM56C-NFS 5.41	Ambient (Intel) V.90 FAX/MODEM PC Card
Viking	PC Card Modem 56K	Viking V.90 K56flex 021 A	MODEM PC Card
KINGMAX	KIT PCMCIA 56K Fax/Modem Card	CIRRUS LOGIC 56K MODEM CL-MD56XX 5.41	V.90 FAX/MODEM PC Card
TDK	TDK DH6400	TDK DH6400 1.0	64Kbps
NTT DoCoMo	Mobile Card Triplex N	NTT DoCoMo Mobile Card Triplex N	64Kbps

Appendix 3. STS Configuration files

A 3.1. System.cnf

```
#
# system.cnf
#
#   system configuration which exist only one place on this file.
#

# kind of IP configuration mode
# 1 - static ip , 2 - dhcp , 3 - pppoe
ipmode = 1

# system ip addres
ipaddr = 192.168.161.5

# system subnet mask
subnet = 255.255.0.0

# system gateway
gateway = 192.168.1.1

# dns configuration
# 'p_dns' is a primary dns ip address and 's_dns' is a secondary dns ip address
# if you want to set dns authmatically in case of dhcp or pppoe,
# you can set 'bmanual_dns' to 0.
p_dns = 168.126.63.1
s_dns = 168.126.63.2

# pppoe configuration
# 'ppp_usr' is pppoe account name and 'ppp_pwd" is a password for that account
ppp_usr = whoever
ppp_pwd = pppoepwd

# Email logging configuration
# if you want to send log via E-mail, set 'emaillog' to 1
# 'emaillog_num' trigger sending email.
# The number of logs are greater than 'emaillog_num", then send it.
emaillog = 0
emaillog_num = 5

# SMTP configuration
# 'smtpsvr' is a SMTP server .
# 'sysmailaddr' is a sender address.
# 'recvmailaddr' is a receiver address.
# 'smtp_mode' means a SMTP server authentication mode.
#  1 - smtp w/o authentication , 2 - pop before smtp , 3 - smtp w/
authentication
# If 'smtp_mode" is 2 or 3, you need SMTP account information.
# 'smtp_user' is a SMTP account name and 'smtp_pwd" is a password.
bsmtp = 0
smtpsvr = smtp.yourcompany.com
sysmailaddr = SS800@yourcompany.com
recvmailaddr = admin@yourcompany.com
smtp_mode = 1
smtp_user = admin
smtp_pwd = admin

# 'device_name' mean a unit name assigned. A unit name will be a identifier
among PS products.
device_name = SS800 Device

# IP filtering configuration
```

```
# By setting 'btelnet' to 1, you can use remote console.
# Similarly by setting 'bweb' to 1, you can use remote console.
# 0 means that protect any access.
# 'enable_ip', 'enable_netmask' pair is a source rule specification for remote
console filtering.
# 'enable_webip', 'enable_webnetmask' pair is for web filtering.
btelnet = 1
bweb = 1
enable_ip = 0.0.0.0
enable_netmask = 0.0.0.0
enable_webip = 0.0.0.0
enable_webnetmask = 0.0.0.0

# dynamic DNS(DDNS) configuration
# dynamic dns can be enabled by setting 'bdyndns' to 1. 0 for disable.
# 'dyn_dn' is a domain name for your DDNS.
# 'dyn_user' is a account name for DDNS and 'dyn_pwd" is a password for it.
bdyndns = 0
dyn_dn = ss800.dyndns.biz
dyn_user = ss800-user
dyn_pwd = ss800-pwd

# NTP configuration
# 'ntp_enable' set to 1 for using NTP or set to 0.
# 'ntp_serverip' is the IP address of NTP server and 'ntp_offset' is a your
offset from UTC.
# If you don't know any NTP server IP, then set 'ntp_auto_conf' to 1.
ntp_enable = 0
ntp_auto_conf = 1
ntp_offset = 0.0
ntp_serverip = 192.168.200.100

# Log configuration
# system logging is enabled by 'log_enable' to 1.
# 'logbuf_size' is a variable for representing log buffer size by KB.
# 'log_stoloc' is a location to save log.
#  1 = memory 2 = CF card 3 = NFS 4 = SYSLOGD
# If you choose log location to SYSLOGD, 'logbuf_size' you've set will loose his
role - limiting log file size.
log_enable = 1
logbuf_size = 4
log_stoloc = 1

# syslog configuration
# You can run or kill syslogd by setting 'bsyslog_service' to 1 or 0.
# 'syslog_ip' is a IP addresss of a remote syslog server.
# 'syslog_2ndip' is a IP address of a secondary syslogd server which will get
the same logs.
# 'syslog_facility' specify what type of program is logging. 0 ~ 7 for LOCAL0 to
LOCAL7
bsyslog_service = 0
syslog_ip = 192.168.200.100
syslog_facility = 0

# NFS configuration
# You can mount or unmount NFS by setting 'bnfs_service'  to 1 or 0.
# 'nfs_ip' is a NFS server IP addresss and 'nfs_path' is a mount path.
bnfs_service = 0
nfs_ip = 192.168.200.100
nfs_path = /

# WEB configuration
# If you want to support HTTP, then set 'bweb_http' to 1. If not, set tot 0.
# 'bweb_https' is for HTTPS.
# 'web_refresh_rate' is for refresh the changing page when you see the system
status page.
bweb_http = 1
```

```
bweb_https = 1
web_refresh_rate = 10

# TCP configuration
# 'keepalive_time' is a time before keep alive takes place.
# 'keepalive_probes' is the number of allowed keep alive probes.
# 'keepalive_intvl' is a time interval between keep alive probes.
keepalive_time = 15
keepalive_probes = 3
keepalive_intvl = 5

# Ethernet configuration
# 'ethernet_mode' is a ethernet mode.
# 0 = Auto Negotiation,  1 = 100BaseT Half Duplex, 2 = 100BaseT Full Duplex,
# 3 = 10BaseT Half Duplex, 4 = 10BaseT Full Duplex
ethernet_mode = 0

# PCMCIA configuration
# 'pcmcia_card_type' shows a pcmcia card type.
# 0 for empty , -1 for unsupported card, 1 for CF card, 2 for Network card,
# 3 for Wireless Network card, 4 for Serial Modem card
pcmcia_card_type = 0

# PCMCIA ipconfiguration
# same with system ip configuration
pcmcia_ipmode = 2
pcmcia_ip = 192.168.1.254
pcmcia_subnet = 255.255.255.0
pcmcia_gateway = 192.168.1.1
pcmcia_ppp_usr = whoever
pcmcia_ppp_pwd = pppoepwd
pcmcia_bmanual_dns = 0

# In case of serial modem card, 'pcmcia_modem_initstr' means a modem init string.
pcmcia_modem_initstr = q1e0s0=2

# Wireless network card configuration
# To enable or disable Wired Equivalent Privacy(WEP), set 'pcmcia_wep_enb' to 1
or 0.
# 'pcmcia_web_mode' is a WEP mdoe. 1 for encrypted, 2 for shared
# 'pcmcia_wep_length' is a length for WEP. 1 for 40 bits, 2 for 128 bits
# 'pcmcia_wep_key_str' is a key string for WEP.
pcmcia_wep_enb = 0
pcmcia_wep_mode = 1
pcmcia_wep_length = 1

# 'pcmcia_cf_conf_max' is a maximum size to use in case of CF card.
pcmcia_cf_conf_max = 0
```

A 3.2. Redirect.cnf

```
#
# redirect.cnf
#
#   Port configuration is placed on this file.
#   Basically keys followed by 'port' key are data for those port.
#   Port number is zero base index and the maximum value for port is used as all
port configuration
#   Data followed by all port are default values and will NOT be applied.

# 'port' key notify the port data follow.
# If you want to activate the port, set 'benable' to 1. If not, set to 0.
# If you set 'bmanset' to 1, you don't want to change the port data by changing
```

```
all port configuration.
# If you want to change the port data by changing all port configuration, set to
0.
port = 0
benable = 0
bmanset = 0
port = 1
benable = 0
bmanset = 0
port = 2
benable = 0
bmanset = 0
port = 3
benable = 0
bmanset = 0
port = 4
benable = 0
bmanset = 0
port = 5
benable = 0
bmanset = 0
benable = 0
port = 6
bmanset = 0
benable = 0
port = 7
bmanset = 0
benable = 0

# As refered, maximum port (in case 8 port machine ,8) represents the
# defaults values for all port configuration.
port = 8
benable = 0
bmanset = 0

# Serial parameter configuration
# 'uarttype' is for UART type. But PS only support RS232.
# So set 'uarttype' to 0 and DO NOT CHANGE.
# 'baudrate' is for baudrate. From 1200 to 230400 is available.
# 'stopbits' is for stop bits. 1 for 1 bit, 2 for 2 bits
# 'databits' is for data bits. 7 for 7 bits, 8 for 8 bits.
# 'parity' is for parity. 0 for none, 1 for even , 2 for odd parity.
# 'flowcontrol" is for flow control. 0 for none, 1 for XON/XOFF,
#                                    2 for hardware flow control
# 'dtropt' is for DTR pin option.
#  1 = Always HIGH, 2 = Always LOW, 3 = High when open
# 'dsropt' is for DSR pin option.
#  0 = None, 1 = Allow TCP connection only by HIGH 2 = open/close TCP connection
# 'interchartimeout' is for inter-character timeout. It works ONLY FOR RAWTCP
# mode.
uarttype = 0
baudrate = 9600
stopbits = 1
databits = 8
parity = 0
flowcontrol = 0
dtropt = 0
dsropt = 0
interchartimeout = 100

# Host mode configuration
# 'hostmode' means a host mode.
#  0 = TCP mode, 1 = UDP mode, 2 = Mode emulation
hostmode = 0
# In TCP mode, 'localport' is a listening port.
localport = 0
# 'max_connection' is a maximum allowed number of remote host
```

```
max_connection = 32
# 'remotehost' is a remote host list
#                   (Primary IP address:port Secondary IP address:port)
remotehost = 192.168.0.135:7000 192.168.0.135:7001
# 'cyclictime ' is a cyclic connection time in seconds
cyclictime = 10
# 'inactivitytimeout' is a inactivity timeout in seconds.
inactivitytimeout = 100

# Cryptography Options
# 'encryptionmode' is encryption mode
# 0 = None, 1 = SSLv2, 2 = SSLv3, 3 = SSLV3 rollback v2, 4 = TLSv1
# 'encryptionkey' is encryption key file name
# 'key_password' is password for encryption key file
# 'cipher_suite' represents a combination of cipher suite.
# 'verify_client' is Verify client(server mode only) option
#  0 = No, 1 = Yes
# 'verify_chain_depth' is a number of chain depth to be searched
# 'verify_cn' is Compare the certificate CN and hostname option
#  0 = No, 1 = Yes
encryptionmode = 2
encryptionkey =
key_password = testing
cipher_suite = 524287
verify_client = 1
verify_chain_depth = 3
verify_cn = 1


# In UDP mode,
# 'accept_unlisted' is Accept UDP datagram from unlisted remote host option
# 0 = No, 1 = Yes
# 'send_to_unlisted' Send to recent unlisted remote host option
# 0 = No, 1 = Yes
accept_unlisted = 1
send_to_unlisted = 1

# IP filtering configuration
# 'allow_ip', 'allow_netmask' pair is a source rule specification for serial
port access filtering.
allow_ip = 0.0.0.0
allow_netmask = 0.0.0.0

# 'porttitle' is a port title.
porttitle = Port Title

# Mode configuration option
# 'modem_mode' is modem mode option
# 0 =Disable, 1 =Enable
# 'modem_initstr' is a modem initialization string
# 'modem_dcd_option' is modem DCD pin option
# 0 = None, 1 = Allow TCP connection only by HIGH
modem_mode = 0
modem_initstr =
modem_dcd_option = 0

# Event notification configuration
# Enable of disable Event notification by setting 'event_enable' to 1 or 0.
# 'notification_interval' is interval of event notification.
# 'bmail_handle' is a Enable/Disable E-mail notification option
# 0 = Disable, 1 = Enable
# 'mail_title' is a title of email notification.
# 'mail_address' is a mail recipient’s address
# 'bsnmp_handle' is a Enable/Disable SNMP notification option
# 0 = Disable, 1 = Enable
# 'snmp_title' is a title of SNMP trap notification.
# 'snmp_trap_receiver_ip' is a IP address of SNMP Trap receiver
```

```
# 'snmp_trap_receiver_community' is community of SNMP Trap
# 'snmp_trap_receiver_version' is SNMP trap version
# 0 = v1, 1 = v2c
event_enable = 1
notification_interval = 0
bmail_handle = 1
mail_title = jungoj@sena.com
mail_address = jung@sss.com
bsnmp_handle = 1
snmp_title = khfgj
snmp_trap_receiver_ip = 192.168.0.8
snmp_trap_receiver_community = public
snmp_trap_receiver_version = 0

# Event Keyword option
# 'keyword_index' is a index of keyword event
# 'keyword_str' is a event keyword
# 'snmp_enable' is a SNMP notification option for keyword
# 0 = Disable, 1 = Enable
# 'mail_enable' is a email notification option for keyword
# 0 = Disable, 1 = Enable
# 'command_enable' is a port command option for keyword
# 0 = Disable, 1 = Enable
# 'port_command' is a port command string for keyword
keyword_index = 0
keyword_str = test
snmp_enable = 1
mail_enable = 1
command_enable = 1
port_command = fghfgh

# Port buffering configuration
# Enable of disable port buffering by setting 'pb_enable' to 1 or 0.
# 'pb_size' is a maximum port buffering size. Maximum value are different by
location.
# 'pb_loc' is a location to store port buffer data.
#  1 = memory 2 = CF card 3 = NFS 4 = SYSLOGD
pb_enable = 0
pb_size = 4
pb_loc = 1
```

Appendix 4. Well-known port numbers

Port numbers are divided into three ranges: Well Known Ports, Registered Ports, and Dynamic and/or Private Ports. Well Known Ports are those from 0 through 1023. Registered Ports are those from 1024 through 49151. Dynamic and/or Private Ports are those from 49152 through 65535.

Well Known Ports are assigned by IANA, and on most systems, can only be used by system processes or by programs executed by privileged users. *Table A-7* shows some of the well-known port numbers. For more details, please visit the IANA website:
http://www.iana.org/assignments/port-numbers

Table A-7 Well-known port numbers

Port number	Protocol	TCP/UDP
21	FTP (File Transfer Protocol)	TCP
22	SSH (Secure SHell)	TCP
23	Telnet	TCP
25	SMTP (Simple Mail Transfer Protocol)	TCP
37	Time	TCP, UDP
39	RLP (Resource Location Protocol)	UDP
49	TACACS, TACACS+	UDP
53	DNS	UDP
67	BOOTP server	UDP
68	BOOTP client	UDP
69	TFTP	UDP
70	Gopher	TCP
79	Finger	TCP
80	HTTP	TCP
110	POP3	TCP
119	NNTP (Network News Transfer Protocol)	TCP
161/162	SNMP	UDP
443	HTTPS	TCP

Appendix 5. Guide to the Bootloader menu program

A 5.1. Overview

The bootloader menu provides a way to recover the STS Series unit using BOOTP/TFTP as a disaster recovery option and to diagnose the system hardware. If the user presses the <ESC> key within 3 seconds after the STS Series unit is powered up, he will enter the bootloader menu program. From this menu program, the user can set various system parameters, test system hardware, and perform firmware upgrades.

A 5.2. Main menu

After entering the bootloader menu program, the user will see following main menu page:

```
Bootloader 1.1.0 (May 23 2003 - 22:48:25)

CPU     :  XPC855xxZPnnD4 (50 MHz)
DRAM    :  64 MB
FLASH   :   8 MB
PC CARD  :  No card
EEPROM   :  A Type exist
Ethernet :  AUTO-NEGOTIATION
Autoboot Start:  0
-------------------------------------------------------------------------------
 Welcome to Boot Loader Configuration page
-------------------------------------------------------------------------------
 Select menu
 1. RTC configuration [ Feb 14 2003 - 11:00:26 ]
 2. Hardware test
 3. Firmware upgrade [S/W Version : v1.0.0]
 4. Exit and boot from flash
 5. Exit and reboot
  <ESC> Back, <ENTER> Refresh
 ----->
```

Figure A-4 Main Menu Page of Bootloader Menu

A 5.3. RTC configuration menu

Using the RTC configuration menu, the user can set the system time of the STS Series.

```
-------------------------------------------------------------------------------
 RTC configuration
-------------------------------------------------------------------------------
 Select menu
 1. Date(mm/dd/yy) : 02/14/03
 2. Time(hh:mm:ss) : 13:27:12
  <ESC> Back, <ENTER> Refresh
```

```
-----> 1
Enter Current Date (mm/dd/yy) : 02/15/03
press the ENTER key to continue

--------------------------------------------------------------------------------
RTC configuration
--------------------------------------------------------------------------------
Select menu
1. Date(mm/dd/yy) : 02/15/03
2. Time(hh:mm:ss) : 13:27:20
 <ESC> Back, <ENTER> Refresh
-----> 2
Enter Current Time (hh:mm:ss) : 13:25:00
press the ENTER key to continue

--------------------------------------------------------------------------------
RTC configuration
--------------------------------------------------------------------------------
Select menu
1. Date(mm/dd/yy) : 02/15/03
2. Time(hh:mm:ss) : 13:25:01
 <ESC> Back, <ENTER> Refresh
----->
```

Figure A-5 RTC configuration within Bootloader Menu Program

A 5.4. Hardware test menu

Using the Hardware test menu, the user can test hardware components. There are three hardware test modes:

- One time
- Looping (without External test in Auto test)
- Looping (with External test in Auto test)

If the user selects **One time**, an auto test or each component test is performed just once. In this mode, the `ping` test to the remote host (server IP address) and UART test are also performed once.

If the user selects **Looping** (without External test in Auto test), the auto test is performed repeatedly until the user presses the <ctrl-c> keys. In this mode, the `ping` test to the remote host (server IP address) and UART test are not performed.

If the user selects **Looping** (with External test in Auto test)', auto test is performed repeatedly until the user presses the <ctrl-c> keys. And, the `ping` test to the remote host (server IP address) and UART test are also performed repeatedly.

Note:

To perform the test on the Ethernet and UART properly, the user must connect an Ethernet cable to the Ethernet port of the STS Series and must plug the loopback connector to all the serial ports of the STS Series. There must exist a remote host with a valid IP address. The default server IP address is 192.168.0.128 and it can be changed using the [Firmware Upgrade] menu. Otherwise, the test may

not be performed properly.

```
--------------------------------------------------------------------------------
 Hardware Test
--------------------------------------------------------------------------------
 Select menu
 0. Test Mode - One time
 1. Auto test
 2. DRAM test
 3. FLASH test
 4. LED test
 5. EEPROM test
 6. UART test
 7. PC card test
 8. Ethernet test
  <ESC> Back, <ENTER> Refresh
 -----> 0

--------------------------------------------------------------------------------
 Hardware Test
--------------------------------------------------------------------------------
 Select menu
 0. Test Mode - Looping(without External test in Auto test)
 1. Auto test
 2. DRAM test
 3. FLASH test
 4. LED test
 5. EEPROM test
 6. UART test
 7. PC card test
 8. Ethernet test
  <ESC> Back, <ENTER> Refresh
 ----->0

--------------------------------------------------------------------------------
 Hardware Test
--------------------------------------------------------------------------------
 Select menu
 0. Test Mode - Looping(with External test in Auto test)
 1. Auto test
 2. DRAM test
 3. FLASH test
 4. LED test
 5. EEPROM test
 6. UART test
 7. PC card test
 8. Ethernet test
  <ESC> Back, <ENTER> Refresh
 ----->0

--------------------------------------------------------------------------------
 Hardware Test
--------------------------------------------------------------------------------
 Select menu
 0. Test Mode - One time
 1. Auto test
 2. DRAM test
 3. FLASH test
 4. LED test
 5. EEPROM test
 6. UART test
 7. PC card test
 8. Ethernet test
  <ESC> Back, <ENTER> Refresh
```

```
----->
```

Figure A-6 Hardware test menu within Bootloader Menu Program

When the user selects [Auto test], a test of all the hardware components is performed automatically.

```
--------------------------------------------------------------------------------
 Hardware Test
--------------------------------------------------------------------------------
 Select menu
 0. Test Mode - One time
 1. Auto test
 2. DRAM test
 3. FLASH test
 4. LED test
 5. EEPROM test
 6. UART test
 7. PC card test
 8. Ethernet test
  <ESC> Back, <ENTER> Refresh
 ----->1

*******  Hardware auto-detect and auto-test *******
[DRAM]
DRAM Test in progress ----------------------------------------[65536KB]
DRAM Test ----------------------------------------------------[SUCCESS]

[FLASH]
Flash Test Status---------------------------------------------[  100 %]
Flash Test ---------------------------------------------------[SUCCESS]

[FAN]
Fan Status ---------------------------------------------------[7020 RPM]

[LED]
SERIAL READY LED ON/OFF---------------------------------------3 time(s)

[EEPROM]
EEPROM   :  A Type exist
EEPROM Test ---------------------------------------------- [SUCCESS]

[UART]
<--Internal loop test-->
Port # 1 test in progressing(Read/Write)----------[SUCCESS]
Port # 2 test in progressing(Read/Write)----------[SUCCESS]
.
.
.
Port # 7 test in progressing(Read/Write)----------[SUCCESS]
Port # 8 test in progressing(Read/Write)----------[SUCCESS]

<--External loop test-->
Port # 1 test in progressing(Read/Write)----------[SUCCESS]
                      (RTS/CTS)-------------[SUCCESS]
                      (DTR/DSR)-------------[SUCCESS]
Port # 2 test in progressing(Read/Write)----------[SUCCESS]
                      (RTS/CTS)-------------[SUCCESS]
                      (DTR/DSR)-------------[SUCCESS]
.
.
.
Port # 7 test in progressing(Read/Write)----------[SUCCESS]
                      (RTS/CTS)-------------[SUCCESS]
                      (DTR/DSR)-------------[SUCCESS]
```

```
Port # 8 test in progressing(Read/Write)----------[SUCCESS]
                        (RTS/CTS)------------[SUCCESS]
                        (DTR/DSR)------------[SUCCESS]

[PCMCIA]
 5V CARD
 5.0V card found: Lucent Technologies WaveLAN/IEEE Version 01.01
          Network Adapter Card

[Ethernet]
Ethernet chip test-------------------------------------------[SUCCESS]
PING 192.168.0.135 from 192.168.161.5 : 64 bytes of ethernet packet.
64 bytes from 192.168.0.135 : seq=0 ttl=255 timestamp=11172879 (ms)
64 bytes from 192.168.0.135 : seq=1 ttl=255 timestamp=11173874 (ms)
64 bytes from 192.168.0.135 : seq=2 ttl=255 timestamp=11174875 (ms)
64 bytes from 192.168.0.135 : seq=3 ttl=255 timestamp=11175876 (ms)

      *******  Hardware auto-detect and auto-test  SUMMARY *******
1. DRAM Test  ----------------------------------------------[SUCCESS]
2. FLASH Test ----------------------------------------------[SUCCESS]
3. FAN Test   ----------------------------------------------[SUCCESS]
4. EEPROM Test---------------------------------------------[SUCCESS]
5. UART Test Summary
  Port NO |  exist status  |   exist status  |  exist status  | exist  status
-------------------------------------------------------------------------------
--
Port 01-04|  YES  SUCCESS  |   YES  SUCCESS  |  YES  SUCCESS  |  YES  SUCCESS
Port 05-08|  YES  SUCCESS  |   YES  SUCCESS  |  YES  SUCCESS  |  YES  SUCCESS

6.PC CARD Test Summary
 5V CARD
 5.0V card found: Lucent Technologies WaveLAN/IEEE Version 01.01
          Network Adapter Card
7. PING Test  ----------------------------------------------[SUCCESS]

PRESS any key to continue!!
```

Figure A-7 Hardware test screen within Bootloader Menu Program

For each hardware component test, the user can skip a test by pressing the <ESC> key.

```
-------------------------------------------------------------------------------
 Hardware Test
-------------------------------------------------------------------------------
 Select menu
 0. Test Mode - One time
 1. Auto test
 2. DRAM test
 3. FLASH test
 4. LED test
 5. EEPROM test
 6. UART test
 7. PC card test
 8. Ethernet test
  <ESC> Back, <ENTER> Refresh
 -----> 1

          *******  Hardware auto-detect and auto-test *******
[DRAM]
DRAM Test in progress --------------------------------------[  640KB]
DRAM Test --------------------------------------------------[SKIPPED]

[FLASH]
Flash Test Status------------------------------------------[   2 %]
FLASH Test -------------------------------------------------[SKIPPED]
```

Figure A-8 Skip the specific test using ESC key

If a failure occurs while **Auto Test** with looping mode is being performed, the test will stop and the serial **InUse** LEDs blink to indicate the hardware test has failed. In this case, the user must press the <ctrl-c> keys to return to the menu page.

A 5.5. Firmware upgrade menu

By using the 'Firmware upgrade' menu, the user can upgrade the firmware of the unit. Before firmware upgrade, the user can check the current firmware version by selecting menu item 3 from the Main menu page. The firmware upgrade menu program supports two protocols for remote firmware download: BOOTP and TFTP. The default protocol is BOOTP for DHCP environments. If the user selects TFTP, he must also set the IP address for the unit properly. The default IP address for the unit is 192.168.161.5.

For firmware upgrade, a firmware file configured as [Firmware File Name] on the server configured as [Server's IP address] must exist.

```
------------------------------------------------------------------------------
 Firmware upgrade
------------------------------------------------------------------------------
 Select menu
 1. Protocol [BOOTP]
 2. IP address assigned to Ethernet interface [192.168.161.5]
 3. Server's IP address [192.168.0.128]
 4. Firmware File Name [sts800.bin]
 5. Start firmware upgrade
  <ESC> Back, <ENTER> Refresh
 -----> 1
 Select protocol ( 1 = BOOTP, 2 = TFTP) : 2


------------------------------------------------------------------------------
 Firmware upgrade
------------------------------------------------------------------------------
 Select menu
 1. Protocol [TFTP]
 2. IP address assigned to Ethernet interface [192.168.161.5]
 3. Server's IP address [192.168.0.128]
 4. Firmware File Name [sts800.bin]
 5. Start firmware upgrade
  <ESC> Back, <ENTER> Refresh
 ----->
```

Figure A-9 Firmware upgrade menu within Bootloader Menu Program

If the user selects [Start firmware upgrade], a confirm message will be displayed on the screen. If the user enters 'y', the firmware upgrade process will start. This process cannot be stopped until it is finished.

```
------------------------------------------------------------------------------
 Firmware upgrade
------------------------------------------------------------------------------
```

```
Select menu
 1. Protocol [BOOTP]
 2. IP address assigned to Ethernet interface [192.168.161.5]
 3. Server's IP address [192.168.0.128]
 4. Firmware File Name [sts800.bin]
 5. Start firmware upgrade
 -----> 5
Firmware upgrade cannot be stopped until finished.
And all configuration parameters are restored to default values.
Do you really want to start firmware upgrade(y/n)?y
BOOTP broadcast 1
ARP broadcast 1
TFTP from server 192.168.0.128; our IP address is 192.168.161.5
Filename 'sts800.bin'.
Load address: 0x100000
Loading: ##################################################################
       ##################################################################
       ##################################################################
       ##################################################################
       ##################################################################
       ##################################################################
       ##################################################################
       ##################################################################
       ##################################################################
       ##################################################################
       ##################################################################
       ##################################################################
       ##################################################################
       ##################################################################
       ##################################
done
Bytes transferred = 5838194 (591572 hex)

Erase Flash Sectors 9-20 in Bank # 1
BLOCK 1 : Copy to Flash... done
Erase Flash Sectors 21-115 in Bank # 1
BLOCK 2 : Copy to Flash... done
Erase Flash Sectors 116-116 in Bank # 1
BLOCK 3 : Copy to Flash... done
BLOCK 4 : Erase Flash Sectors 117-117 in Bank # 1
Firmware upgrade is finished successfully.


-------------------------------------------------------------------------------
 Firmware upgrade
-------------------------------------------------------------------------------
Select menu
 1. Protocol [BOOTP]
 2. IP address assigned to Ethernet interface [192.168.161.5]
 3. Server's IP address [192.168.0.128]
 4. Firmware File Name [sts800.bin]
 5. Start firmware upgrade
  <ESC> Back, <ENTER> Refresh
 ----->
```

Figure A-10 Firmware upgrade process

After finishing the firmware upgrade process, the program will display the menu again along with a success message.

Appendix 6. Using STS Series with Serial/IP

A 6.1. STS Series vs. Serial/IP options

Table A-8 STS Series vs. Serial/IP option compatibility matrix table

Serial Port Configuration of STS Series			**Serial/IP Configuration**		
Host mode Configuration		Cryptography Configuration	Credentials	Connection Protocol	Security
Host mode	Telnet Protocol	Encryption Method			
TCP	Disabled	None	No login required	Raw TCP connection	Disable
TCP	Enabled	None	No login required	Telnet	Disable
TCP	Disabled	“SSLv2” or “SSLv3 rollback to v2”	No login required	Raw TCP connection	Negotiate SSLv3/TSLv1
TCP	Disabled	“SSLv3” or “SSLv3 rollback to v2”	No login required	Raw TCP connection	SSLv3
TCP	Disabled	“TLSv1” or “SSLv3 rollback to v2”	No login required	Raw TCP connection	TSLv1
TCP	Enabled	“SSLv2” or “SSLv3 rollback to v2”	No login required	Telnet	Negotiate SSLv3/TSLv1
TCP	Enabled	“SSLv3” or “SSLv3 rollback to v2”	No login required	Telnet	SSLv3
TCP	Enabled	“TLSv1” or “SSLv3 rollback to v2”	No login required	Telnet	TSLv1

Please note that “SSLv3 rollback to v2” option in STS series means “Negotiate SSLv3/TSLv1” option in Serial/IP.

If encryption method of STS Series is set as “SSLv3”, then client (Serial/IP) cannot connect to STS

Series with “Negotiate SSLv3/TSLv1” option.

A 6.2. Connection example - Telnet and SSLv3 encryption

Step 1. Set host mode of serial port #1 of STS Series as follows,

Host mode = TCP,

TCP listening port = 7001,

Telnet protocol = Enabled

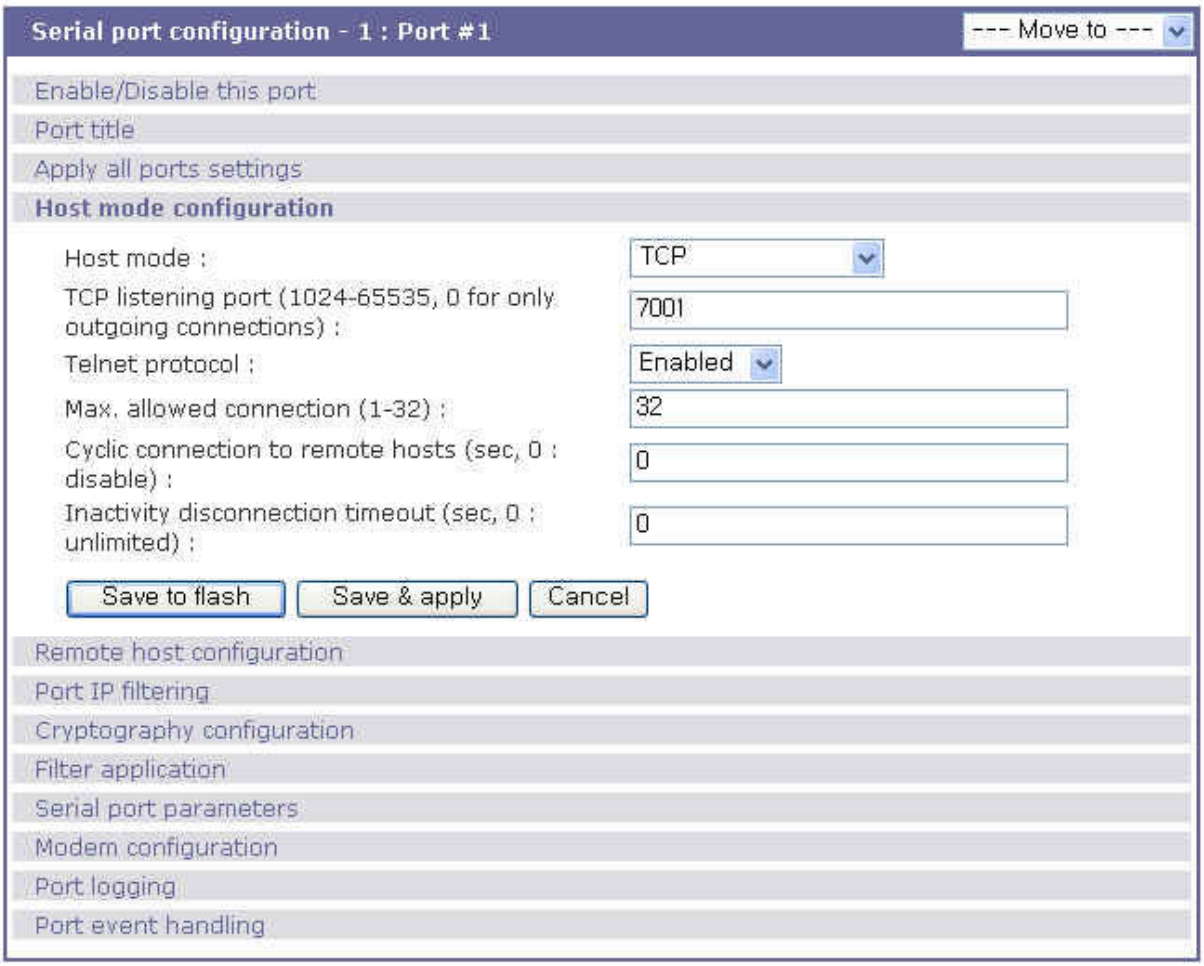


Figure A-11 Host mode configuration

Step 2. Set Cryptography configuration of serial port #1 of STS Series as follows,

Encryption method = SSLv3

Leave all other options as factory default.

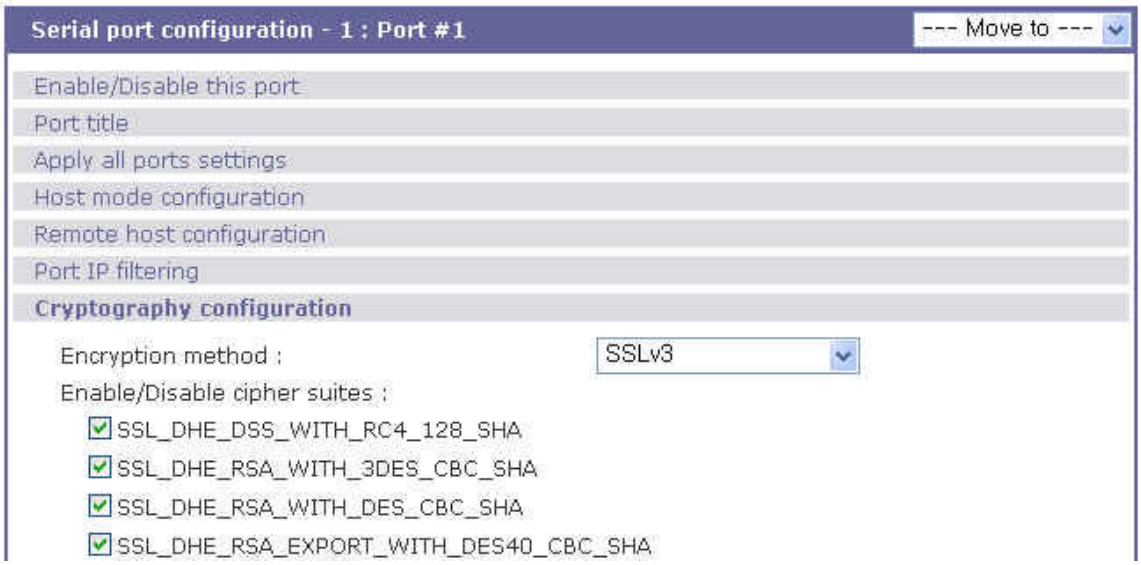


Figure A-12 Cryptography configuration

Step 3. Open Serial/IP Control Panel and check the COM port you want to use to communicate with serial port #1 of STS Series by pressing “Select Ports” button.

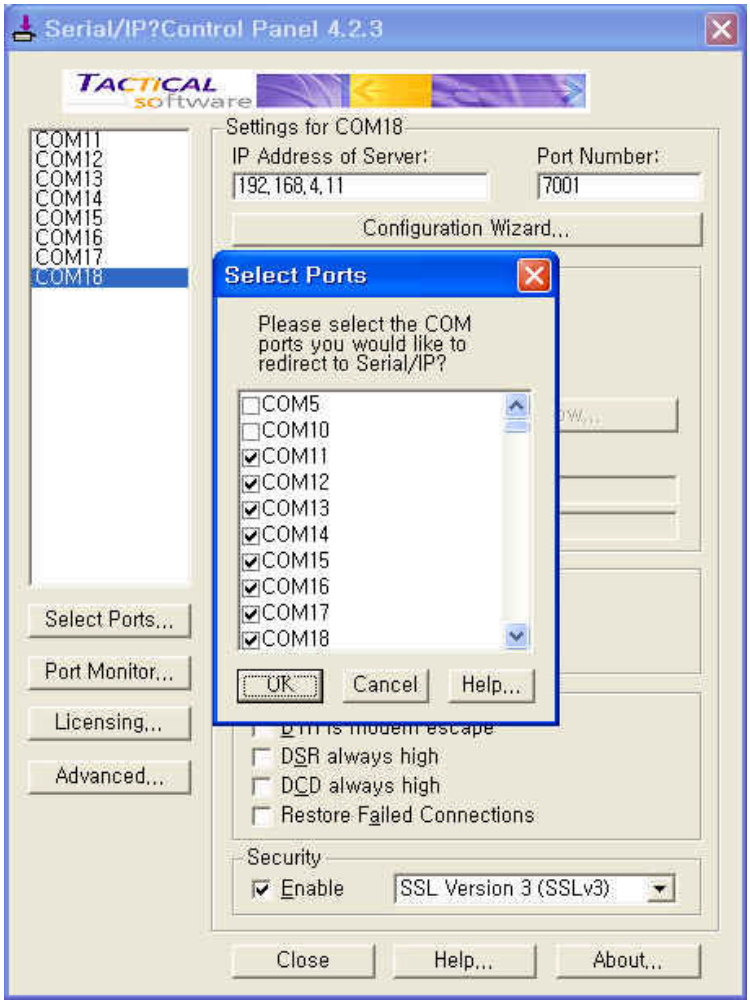


Figure A-13 Select Ports on Serial/IP Control Panel

Step 4. Enter IP address of Server(IP address of STS Series) and Port number (port number of serial port #1) correctly.

And then select other parameters as follows.

Credentials = No Login Required,

Connection Protocol = Telnet,

Security = SSL Version 3 (SSLv3)

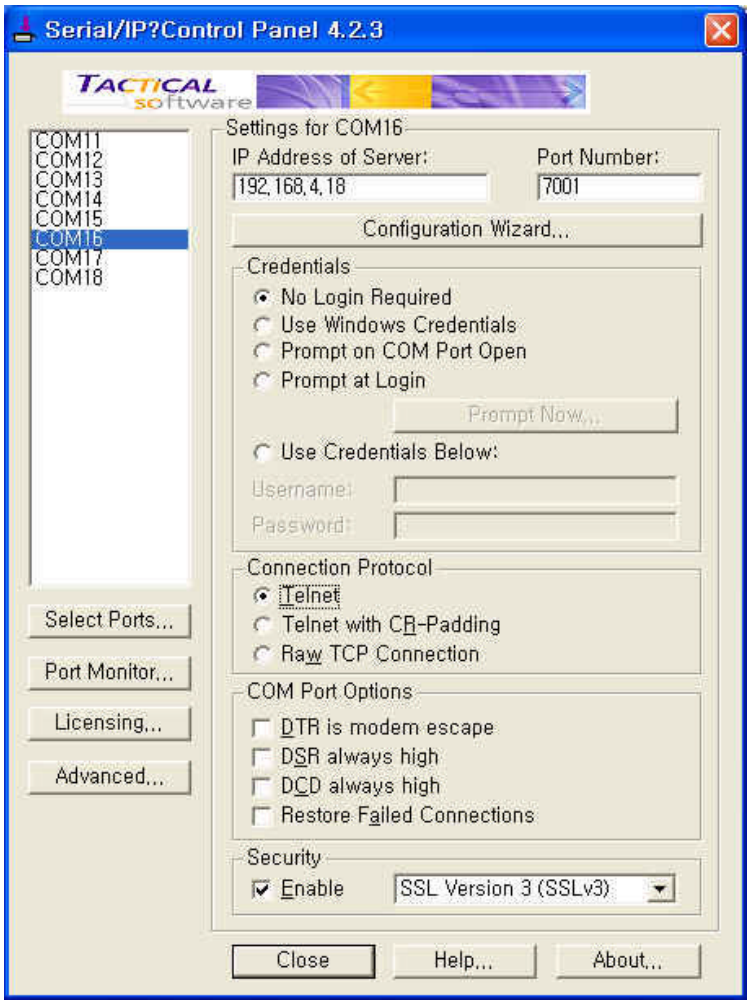


Figure A-14 Set parameters on Serial/IP Control Panel

Step 5. Open the terminal emulation program and select the corresponding COM port.

Then user can use the serial port of STS series using his local terminal emulation program as if it is one of COM ports on his PC.

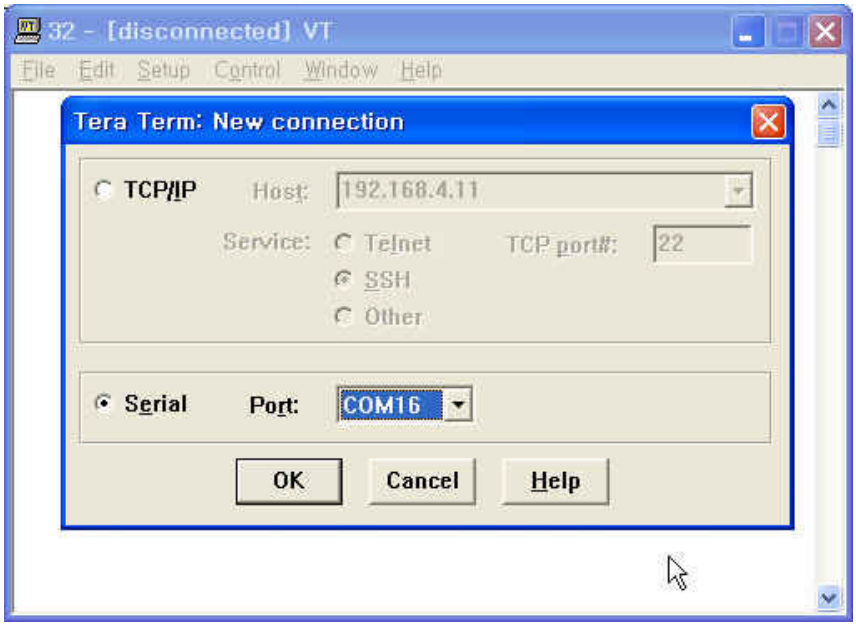


Figure A-15 Connect to serial port of STS series via Serial/IP

Step 6. User can monitor or trace the connection status using Serial/IP Port Monitor or Trace window.

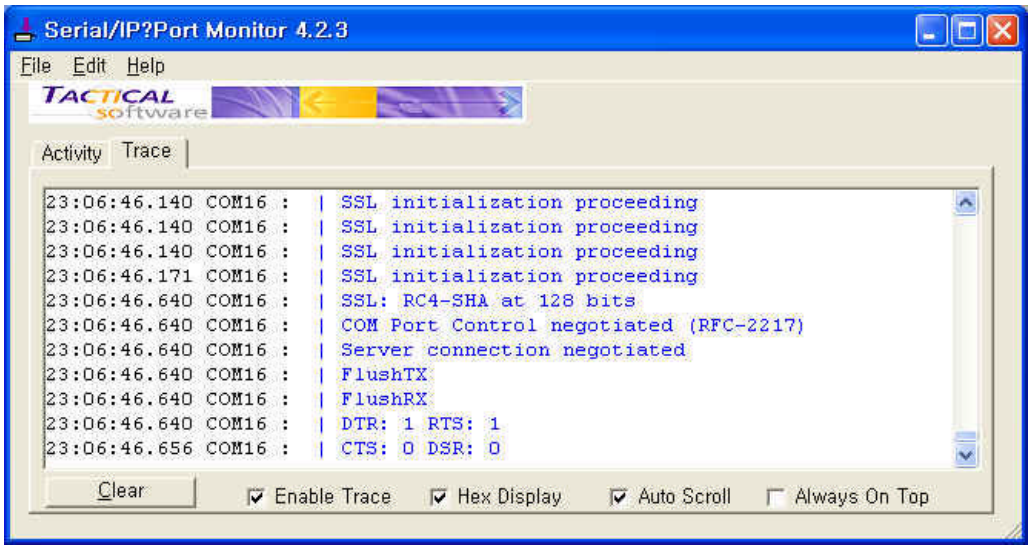


Figure A-16 Serial/IP Trace Window

Appendix 7. How to make a certificate for SSL encryption

A 7.1. Install the OpenSSL package

Step 1. Download the latest OpenSSL package. (http://www.openssl.org)

Step 2. Install the OpenSSL package.

<For Windows user>

Download OpenSSL for Windows binary file and run it.

(http://www.slproweb.com/products/Win32OpenSSL.html)

<For Linux user>

Download OpenSSL source code and compile it.

```
# cd /work/
# tar –xvzf openssl-0.9.7d.tar.gz
# cd openssl-0.9.7d
# ./config
# make
# make test
# make install
```

A 7.2. Make root CA (for Self-signed)

Step 1. Editing openssl configuration file.

Default configuration file location is as follows,

< Windows >

C:\Program Files\OpenSSL\bin

< Linux >

/usr/share/ssl/openssl.cnf

Modify [req_distinguished_name] section as follows,

```
countryName             = Country Name (2 letter code)
countryName_default     = KR
countryName_min         = 2
countryName_max         = 2

stateOrProvinceName     = State or Province Name (full name)
#stateOrProvinceName_default    = Some-State

localityName         = Locality Name(eg, city)
localityName_default        = Seoul

0.organizationName      = Organization Name (eg, company)
0.organizationName_default  = Sena Technologies Inc.
```

```
# we can do this but it is not needed normally :-)
#1.organizationName     = Second Organization Name (eg, company)
#1.organizationName_default = World Wide Web Pty Ltd

organizationalUnitName      = Organizational Unit Name (eg, section)
#organizationalUnitName_default =

commonName          = Common Name (eg, your name or your server\'s hostname)
commonName_default      = Sena Technologies
commonName_max          = 64

emailAddress        = Email Address
emailAddress_max    = 40
```

Modify [req_attributes] section as follows,

```
     challengePassword_min =0
     challengePassword_max =0
```

Step 2. Making self-signed Root CA(Certificate Authority)

< Windows >

cd /work/openssl-0.9.7d/

< Linux >

cd /work/openssl-0.9.7d/

mkdir CA

cd CA

sh /usr/local/ssl/misc/CA.sh –newca

```
CA certificate filename (or enter to create)
  ;(Press Enter to use default value)
Making CA certificate ...
; openssl is called here as follow from CA.sh
; openssl req -new -x509 -keyout ./demoCA/private/./cakey.pem \
;  -out ./demoCA/./cacert.pem -days 365
Using configuration from /usr/local/ssl/lib/ssleay.cnf
Generating a 1024 bit RSA private key
........++++++
.....................++++++
writing new private key to './demoCA/private/./cakey.pem'
Enter PEM pass phrase:  ; CA Password (Enter password and remember this)
Verifying password - Enter PEM pass phrase:  ; CA Password
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----  ; CA's Information
Country Name (2 letter code) [AU]: KR
State or Province Name (full name) [Some-State](Enter)
Locality Name (eg, city) []:Seoul
Organization Name (eg, company) [Internet Widgits Pty Ltd]: Sena Technologies
Organizational Unit Name (eg, section) [](Enter)
Common Name (eg, YOUR name) []:Sena Technologies
Email Address []:(Enter)
#
```

2-3. Check whether CA key file(demoCA/private/cakey.pem) and CA certificate

(demoCA/cacert.pem) is generated

ls demoCA/

```
cacert.pem certs   crl index.txt      newcerts
private    serial
```

ls demoCA/private

```
cakey.pem
```

A 7.3. Making a certificate request

To make new certificates, you should make a certificate request first.

cd /work/openssl-0.9.7c/CA

Run following commands,

openssl genrsa -out key.pem 1024

openssl req -new -key key.pem -out req.pem

(It is assumed that you are using sample configuration file

- “openssl.conf.sena”)

```
Using configuration from /usr/share/ssl/openssl.cnf
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [US]: (Enter)
State or Province Name (full name) [Minnesota]: (Enter)
Locality Name (eg, city) [Minneapolis]: (Enter)
Organization Name (eg, company) [Digi International]: (Enter)
Organizational Unit Name (eg, section) []:(Enter)
Common Name (eg, your name or your server's hostname) []:Sena VTS
Email Address []:(Enter)

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:(Press Enter – Do not enter any other characters)
An optional company name []:(Press Enter – Do not enter any other characters)
```

A 7.4. Signing a certificate request

4-1. Signing a certificate request

cd /work/openssl-0.9.7c/CA

cp req.pem newreq.pem

sh /usr/local/ssl/misc/CA.sh -sign

```
Using configuration from /usr/share/ssl/openssl.cnf
```

```
Enter PEM pass phrase: CA Password (Enter CA password in step 2-2)
Check that the request matches the signature
Signature ok
The Subjects Distinguished Name is as follows
countryName         :PRINTABLE:'US'
stateOrProvinceName   :PRINTABLE:'Minnesota'
localityName        :PRINTABLE:'Minneapolis'
organizationName     :PRINTABLE:'Digi International'
commonName          :PRINTABLE:'Digi PortServer CM'
Certificate is to be certified until Oct  6 09:39:59 2013 GMT (3653 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
Certificate:
   Data:
      Version: 3 (0x2)
      Serial Number: 1 (0x1)
      Signature Algorithm: md5WithRSAEncryption
      Issuer: C=US, ST=Minnesota, L=Minneapolis, O=Digi International, CN=Digi
International
      Validity
         Not Before: Oct  6 09:39:59 2003 GMT
         Not After : Oct  6 09:39:59 2013 GMT
      Subject: C=US, ST=Minnesota, L=Minneapolis, O=Digi International, CN=Digi
PortServer CM
      Subject Public Key Info:
         Public Key Algorithm: rsaEncryption
         RSA Public Key: (1024 bit)
....
-----BEGIN CERTIFICATE-----
....
-----END CERTIFICATE-----
Signed certificate is in newcert.pem
```

4-2. Check whether signed certificate(newcert.pem) is generated.

ls

```
  demoCA       key.pem     newcert.pem      newreq.pem     req.pem
```

A 7.5. Making certificate for STS

5-1. Removing headings in newcert.pem file

cd /work/openssl-0.9.7c/CA

cp newcert.pem server.pem

vi server.pem

```
Certificate:
   Data:
      Version: 3 (0x2)
      Serial Number: 1 (0x1)
      Signature Algorithm: md5WithRSAEncryption
      Issuer: C=KR, ST=, L=Seoul, O=Sena Technologies Inc., CN= Sena
Technologies
      Validity
         Not Before: Oct  6 09:39:59 2003 GMT
```

```
          Not After : Oct  6 09:39:59 2013 GMT
      Subject: C=US, ST=Minnesota, L=Minneapolis, O=Digi International, CN=Digi
PortServer CM
      Subject Public Key Info:
          Public Key Algorithm: rsaEncryption
          RSA Public Key: (1024 bit)
....
== Removing above lines ===
-----BEGIN CERTIFICATE-----
....
-----END CERTIFICATE-----
```

5-2. Concatenating key.pem file to server.pem

cat key.pem >> server.pem