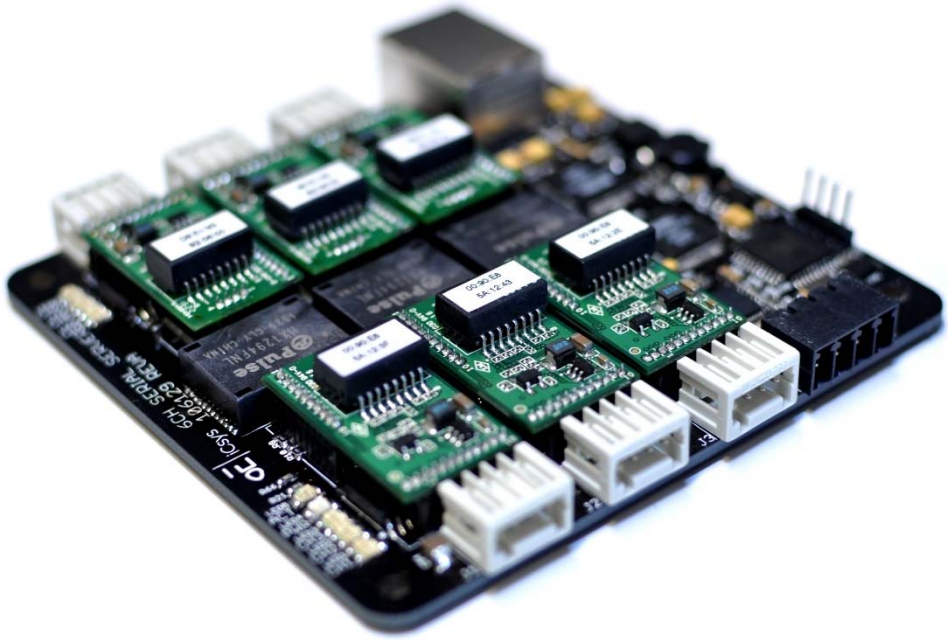
USER MANUAL



Equipment Description	6CH Serial Server PCB
iCsys Part Number:	106129

Document No.:	106129-ICS-PD-UMN-0001	**Document Name.:**	User Manual
Published	**Revision number**	**Revision reason**	**Revised by**
11.05.2017	A	Issued for release	VHA
11.05.2017	0	For Internal Review	VHA
Prepared		**Checked**	**Approved**
VHA		SHA	JOH

TABLE OF CONTENTS

1. INTRODUCTION

1.1. GENERAL NOTES

This document outlines and defines the installation, operation and maintenance procedures for the iCsys 6CH Serial Server PCB. The manual will contain all relevant data and methods to be able to use and maintain the device for its intended purpose.

It will be stated in the manual everything from technical specifications, installation and maintenance to troubleshooting.

1.2. PURPOSE AND SCOPE

The purpose of this manual is to give instructions to install, operate and maintain the 6CH Serial Server supplied by iCsys AS.

The manual is to be used by trained and competent personnel only.

1.3. ABBREVIATIONS

Abbreviation	Description
PCB	Printed Circuit Boards
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
ESD	Electrostatic Discharge
IP	Internet Protocol
EEPROM	Electric Erasable Read Only Memory

1.4. SUPPLIER CONTACT INFORMATION

iCsys AS
Hattelandsveien 92
N-4353 Klepp Stasjon
Norway

+47 51 42 22 22
post@icsys.no
www.icsys.no

2. HEALTH, SAFETY AND ENVIRONMENT

2.1. GENERAL

Safety Notes and General Precautions shall be presented to all personnel concerned prior to testing, operation, maintenance and repair. The operations shall be performed by the responsible engineer/supervisor. The personnel using this equipment must have knowledge of this type of equipment and have familiarized themselves with the applicable procedures and manuals for this product.

2.2. SAFETY MESSAGE LEVELS

Safety message level		Indication
	DANGER:	A hazardous situation which, if not avoided, will result in death or serious injury
	WARNING:	A hazardous situation which, if not avoided, could result in death or serious injury
	CAUTION:	A hazardous situation which, if not avoided, could result in minor or moderate injury or damage to equipment
	Electrical Hazard:	The possibility of electrical risks if instructions are not followed in a proper manner
NOTICE:		A potential situation which, if not avoided, could result in an undesirable result or state A practice not related to personal injury

3. SPECIFICATIONS

3.1. DESCRIPTION

The 6CH Serial Server is a printed circuit board with multiple serial communication ports for interfacing serial devices with Ethernet based network. Each port uses a Moxa Miineport module with its own IP address. A CPU on the board is used for setup and connection diagnose for the modules. Integrated network switches on the board combines all Ethernet ports to one common RJ45 port.

3.2. TECHNICAL DATA

General	
Manufacturer	iCsys AS
Part Number	106129
Description	6CH Serial Server PCB
Weight	~120g
Dimensions	96 x 90 x 16mm (PC104 format)

Electrical Data	
Supply Voltage	21.6 – 26.4 VDC
Power Consumption	< 5W

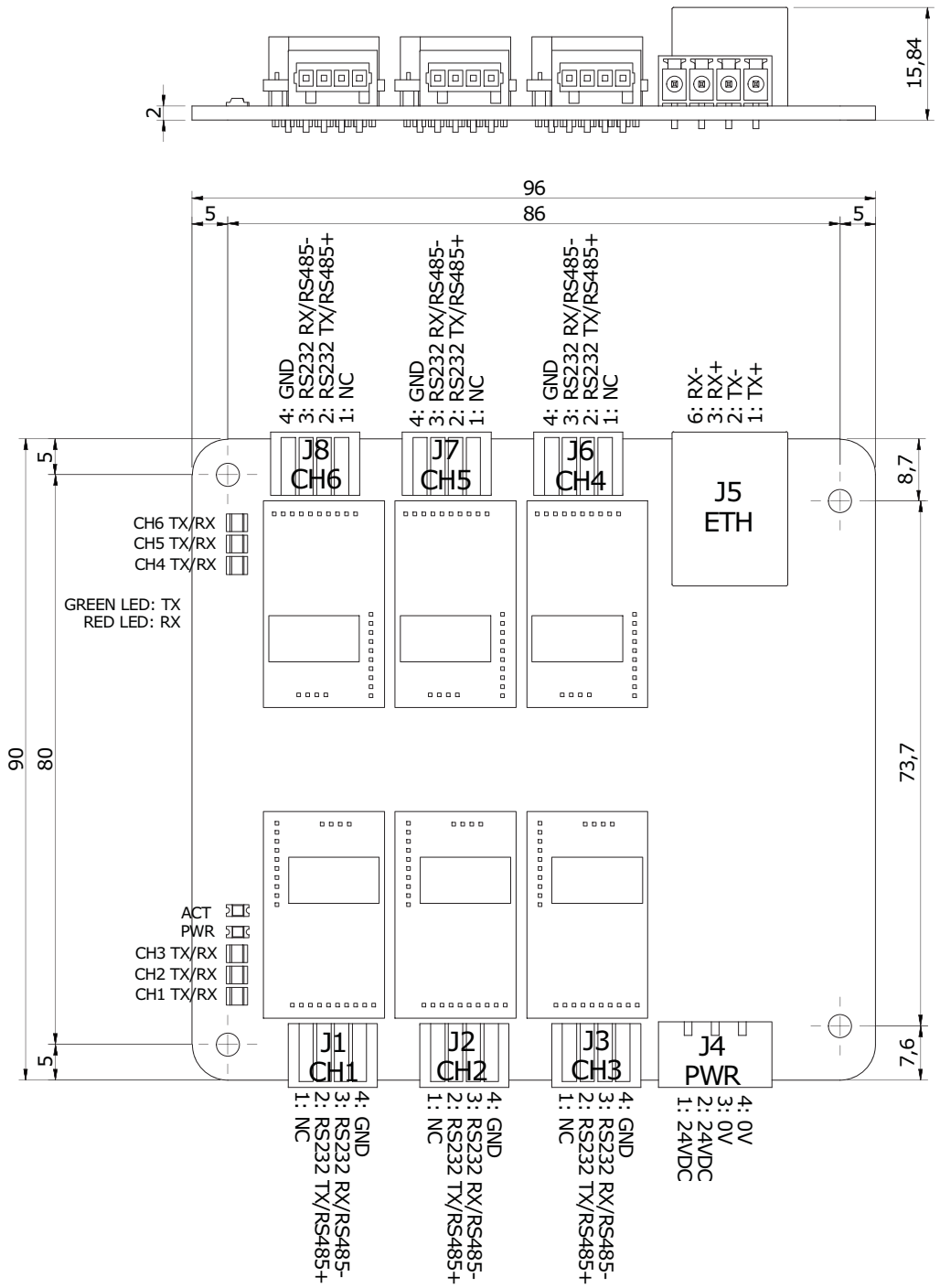
Cable Connectors	
J4 Power Supply	Weidmuller 1792790000
J1-3 & J6-8 Serial Ports	Wago 733-104
J5 Ethernet Port	RJ45

Other	
Speed	10/100 Mbps
Baudrate	50 bps – 921.6 Kbps
Port Module Type	Moxa Miineport E2-H-T
Default IP address of the CPU	10.0.37.240
Default IP address of the modules	10.0.37.241-246

3.3. WARRANTY CONDITIONS AND GUARANTEE

- Improper use of equipment where use is not reflected in what it was intended to.
- Where general maintenance is not performed leading to defective parts or other type of defect.
- Incorrect handling or use of equipment.
- Packing not carried out in an ESD protective way

4. DRAWING


15,84
2
96
86
5
5
4: GND
3: RS232 RX/RS485-
2: RS232 TX/RS485+
1: NC
4: GND
3: RS232 RX/RS485-
2: RS232 TX/RS485+
1: NC
4: GND
3: RS232 RX/RS485-
2: RS232 TX/RS485+
1: NC
6: RX-
3: RX+
2: TX-
1: TX+
J8
CH6
J7
CH5
J6
CH4
J5
ETH
8,7
CH6 TX/RX
CH5 TX/RX
CH4 TX/RX
GREEN LED: TX
RED LED: RX
90
80
73,7
ACT
PWR
CH3 TX/RX
CH2 TX/RX
CH1 TX/RX
J1
CH1
J2
CH2
J3
CH3
J4
PWR
7,6
4: GND
3: RS232 RX/RS485-
2: RS232 TX/RS485+
1: NC
4: GND
3: RS232 RX/RS485-
2: RS232 TX/RS485+
1: NC
4: GND
3: RS232 RX/RS485-
2: RS232 TX/RS485+
1: NC
4: 0V
3: 0V
2: 24VDC
1: 24VDC

5. OPERATION

5.1. NORMAL OPERATION

Operation of the Moxa Miineport modules are described in the last chapter of this manual which is partly taken from Moxas User Manual for the Moxa Miineport demo board.

5.2. SETUP

Use the last chapter in this manual as reference when setting the parameters in the Miineport Modules.

5.2.1. CONFIGURE SERIAL PORT TO RS485

- Set the Miineport module DIO-0 and DIO-2 to outputs and set DIO-2 to “high”
- Set DIO-0 to “high” to activate inbuilt termination resistor to the line if needed
- Select RS485 Mode and set the Baudrate, Databits, Parity and Stop bits according to your equipment setup.

5.2.2. CONFIGURE SERIAL PORT TO RS232

- Set the Miineport module DIO-0 and DIO-2 to outputs and set DIO-0 to “low” and DIO-2 to “low”
- Select RS232 Mode and set the Baudrate, Databits, Parity and Stop bits according to your equipment setup.

5.3. TROUBLESHOOTING / FAULTFINDING

Preliminary fault isolation Check

✓ The electrical connections are correct as described in drawing in chapter 4.

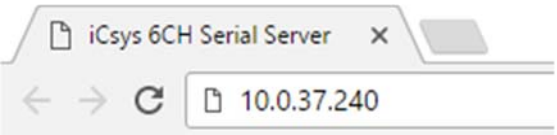
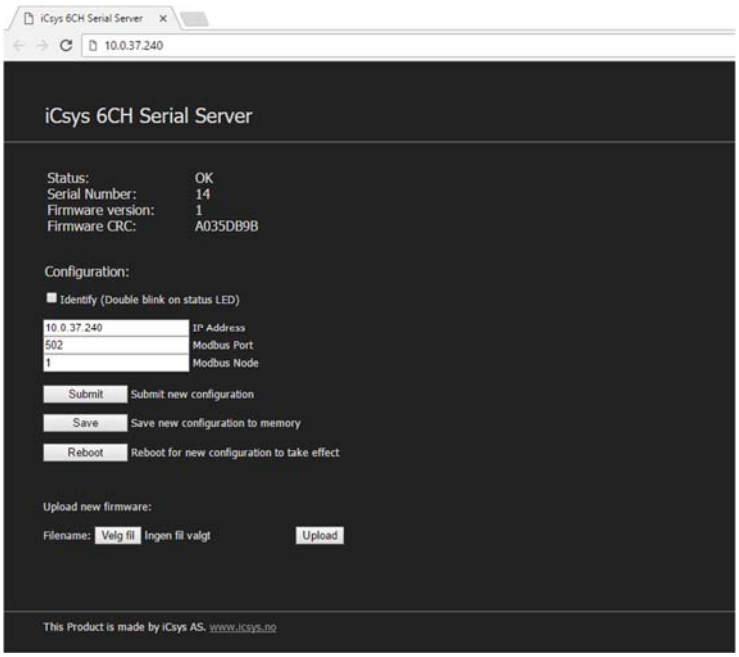
Trouble shooting		
Symptom	**Possible Causes**	**Remedy**
No communication with any Module	• Module is still booting. • No power to board. • Wrong ip address being used.	• Let the modules boot for one minute after it is turned on before communicating. • Be sure power in a range from 21.6 – 26.4 VDC is provided to the board. • Verify correct IP address being used
No communication with CPU on the board	• No power to board. • Wrong ip address being used.	• Be sure power in a range from 21.6 – 26.4 VDC is provided to the board. • Verify correct IP address being used • Verify orange activity LED (marked ACT) is blinking at a fixed rate. If the activity LED is not blinking at a fixed rate, the board must be replaced.
No communication on RS485	• RS485 is not configured in Miineport module • No termination resistor on the line	• Configure Miineport module to RS485 • Install or activate termination resistor.
No communication on RS232	• RS232 is not configured in Miineport module	• Configure Miineport module to RS232

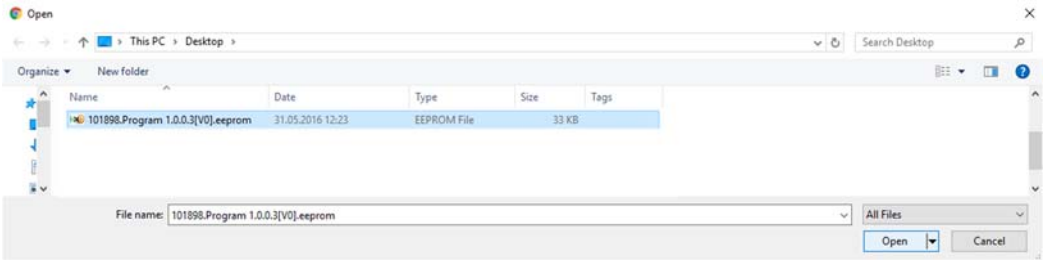
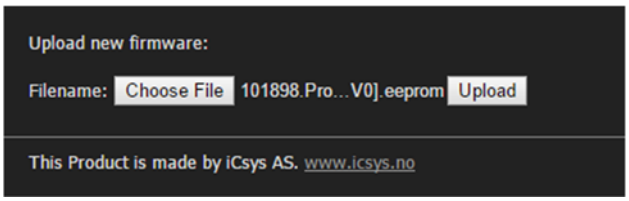
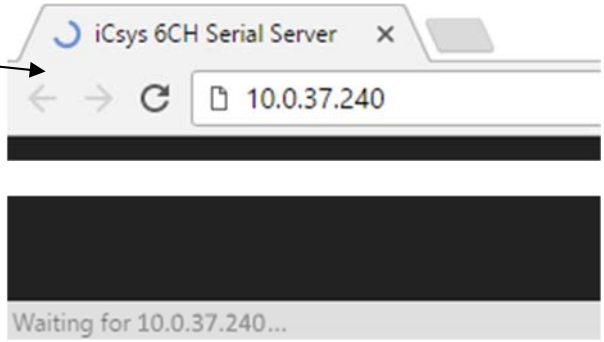
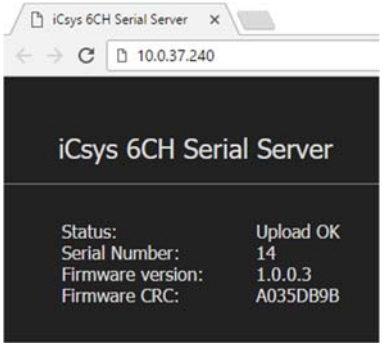
5.4. FIRMWARE UPDATE

Follow this guide to update the firmware in the 6CH Serial Server Main Controller.

CAUTION:

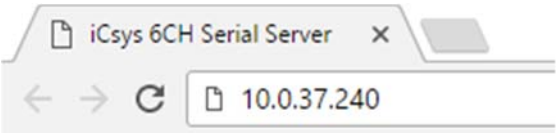
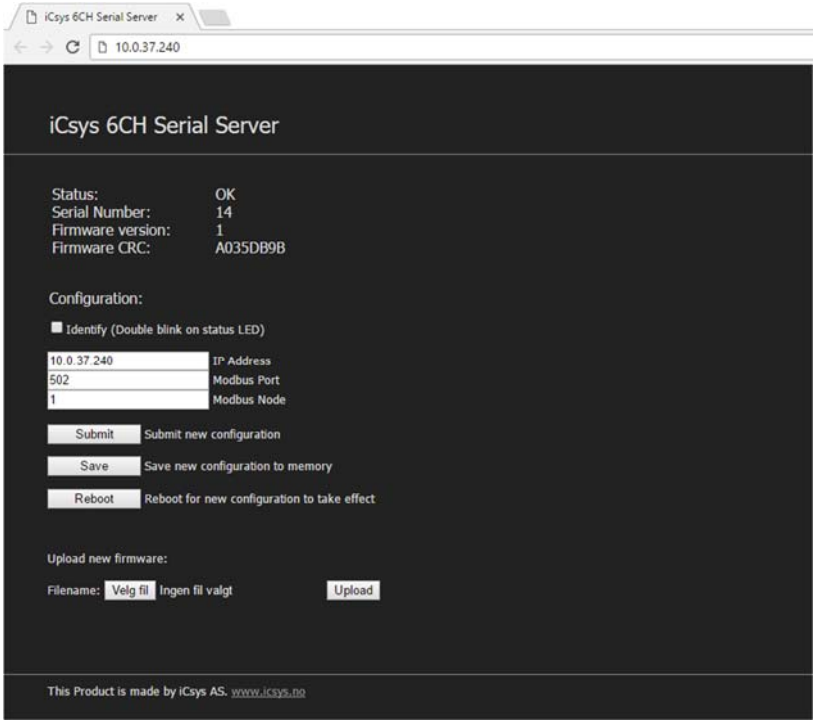
- Do not power down the board when updating new firmware, this will damage the component that is updating.

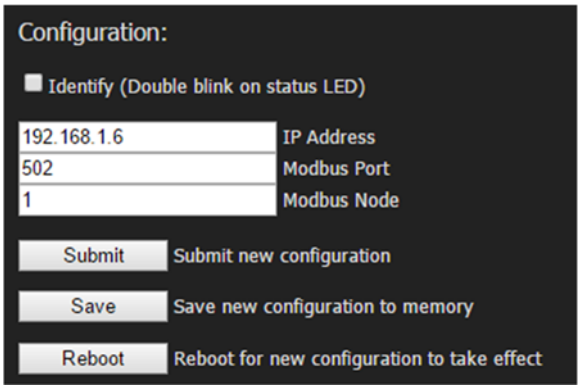
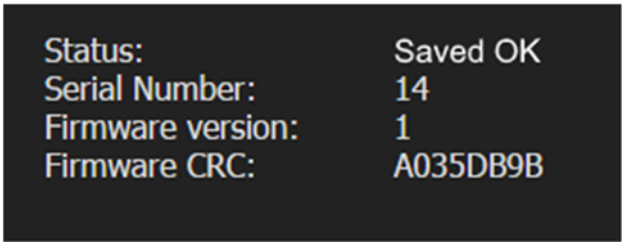
Step	Description	✓
1.	Open web browser, connect to desired 6CH Serial Server CPU by typing its IP address. Default IP address of the CPU is 10.0.37.240 	☐
2.	When connected, following page will show with info about status, serial number, firmware version, etc. More features will be added here by future development. 	☐
3.	To update firmware press “Choose File”	☐

4.	A file dialog will open, choose .eeprom file provided by iCsys AS. Press open.  	☐
5.	When correct .eeprom file is chosen, press upload.  	☐
6.	When firmware is uploaded, the browser will indicate that it is waiting for the page to respond. Top of page  Bottom of page	☐
7.	When uploading is finished, Status will indicate “Upload OK” or “Upload Failed”. If “Upload Failed” is shown or the web page times out, try one more time.  	☐
8.	Press reboot for the new firmware to take effect.	☐

5.5. CHANGING IP ADDRESS

Follow this procedure to change the IP address of the CPU.

Step	Description	✓
1.	Open web browser, connect to desired 6CH Serial Server CPU by typing its IP address. Default IP address of the CPU is 10.0.37.240  	☐
2.	When connected, following page will show with info about status, serial number, firmware version, etc. More features will be added here by future development.  	☐

3.	To change IP address, type in the new IP address in the IP address field, also fill in desired Modbus port and node id.  	☐
4.	Press “Submit” and then press “Save” to save the new settings to EEPROM.  	☐
5.	Press “Reboot” for the new settings to take effect.	☐

6. PARTS FROM MOXA MIINEPORT USER MANUAL

6.1. CHOOSING THE PROPER OPERATION MODE

The MiiNePort modules support operation modes for COM mapping and TCP/IP. After choosing the proper operation mode for your application, refer to subsequent chapters for configuration details.

The following topics are covered in this chapter:

- **Overview**
- **Real COM Mode**
- **TCP Server Mode**
- **TCP Client Mode**
- **RFC2217 Mode**
- **UDP Mode**
- **Ethernet Modem Mode**
- **MCSC Mode**

6.1.1. OVERVIEW

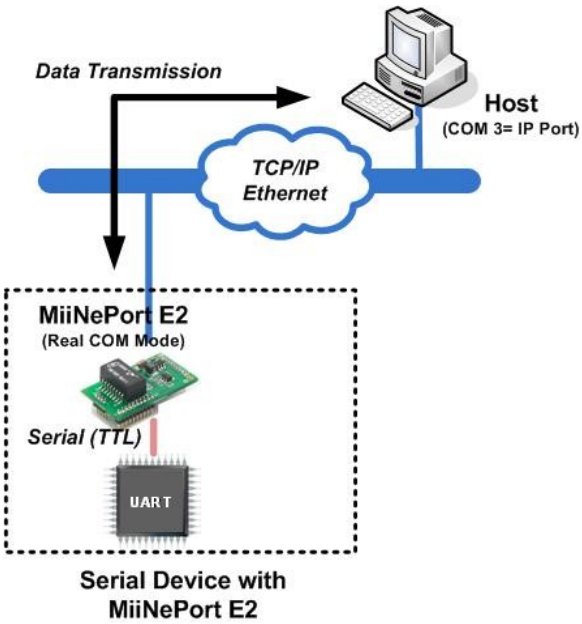
The MiiNePort acts as a bridge for connecting serial devices to Ethernet networks. After choosing the best operation mode for your application, you can use your computer to access, manage, and configure your serial devices from anywhere in the world over the Internet.

Traditional SCADA and data collection systems rely on serial ports (RS-232/422/485) to collect data from various kinds of instruments. Since MiiNePort modules convert between serial and Ethernet signals, you will be able to access your SCADA and data collection system from hosts connected to a standard TCP/IP network, regardless of whether the devices are used locally or at a remote site.

The MiiNePort modules support Real COM mode and six different socket modes—TCP Server, TCP Client, Ethernet Modem, RFC2217, UDP, and MCSC. The main difference between the TCP and UDP protocols is that TCP guarantees delivery of data by requiring the recipient to send an acknowledgement to the sender. UDP does not require this kind of verification, and consequently UDP is faster than TCP. UDP also allows multicasting of data to groups of IP addresses.

6.1.2. REAL COM MODE

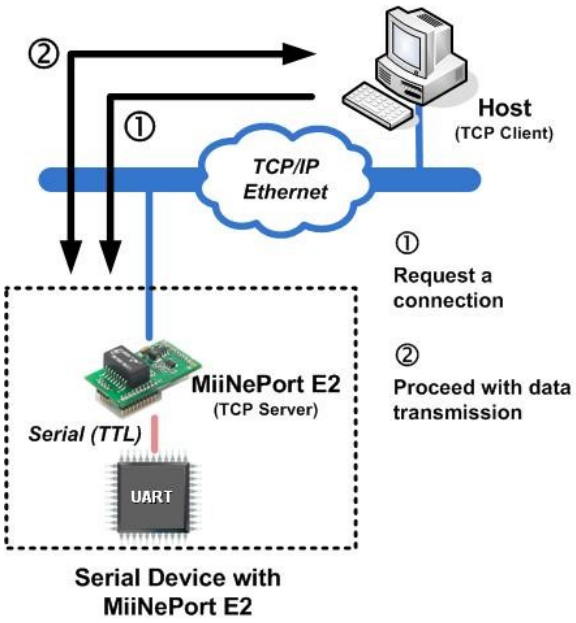
Real COM mode allows users to continue using software that was written for pure serial communications applications. Each module comes equipped with COM drivers for Windows systems (95 and above). The module's serial port is mapped by the driver to an IP address and port number. The driver intercepts data sent to the host's COM port, packs it into a TCP/IP packet, and then redirects it through the host's Ethernet card. At the other end of the connection, the module accepts the Ethernet frame, unpacks the TCP/IP packet, and then transparently sends the data to the attached serial device. In other words, a PC host can treat networked devices as though the devices were connected directly to the PC.



6.1.3. TCP SERVER MODE

In TCP Server mode, the module is assigned a unique IP address and port number on the TCP/IP network. The module waits passively to be contacted by the host computer, allowing the host computer to establish a connection with and obtain data from the serial device. TCP Server mode supports up to four simultaneous connections so that multiple hosts can collect data from the same serial device—at the same time. Data transmission proceeds as follows:

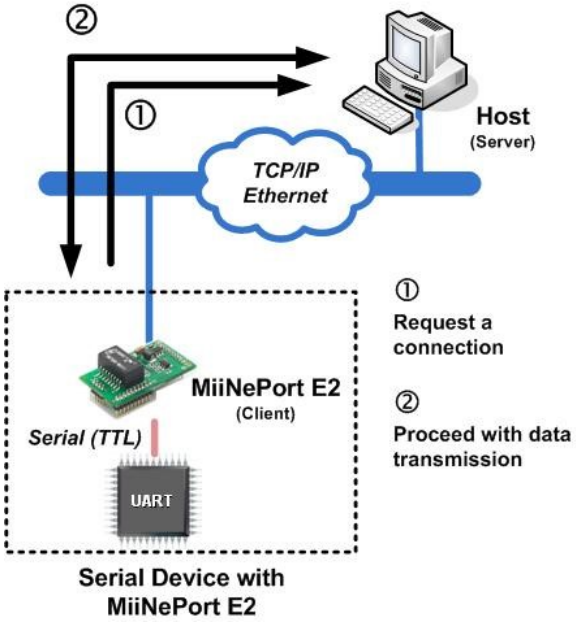
- The host connects to the module configured for TCP Server mode.
- Once the connection is established, data can be transmitted in both directions—from the host to the module, and from the module to the host.



6.1.4. TCP CLIENT MODE

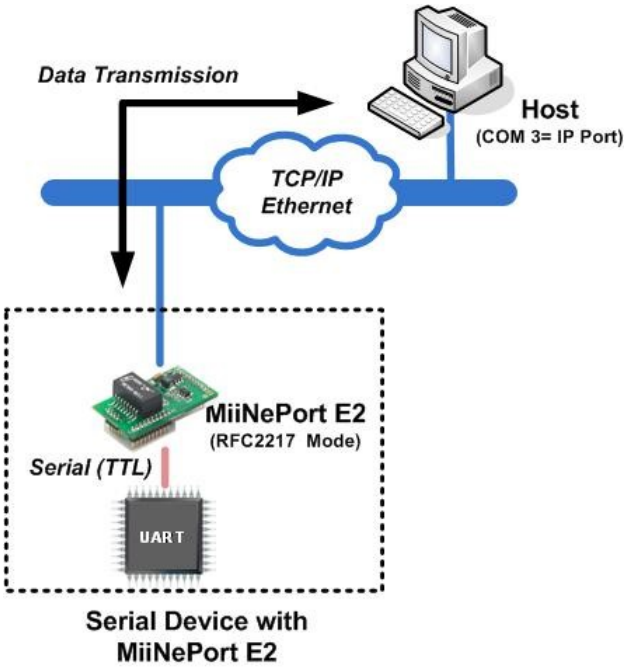
In TCP Client mode, the module can actively establish a TCP connection to a predefined host computer when serial data arrives. After the data has been transferred, the module can be automatically disconnected from the host computer by using the **TCP alive check time** or **Inactivity time** settings. Refer to subsequent chapters for details. Data transmission proceeds as follows:

The module actively establishes a connection based on the conditions set in the firmware. You may let the module connect to a remote host on startup, or connect later when data from the serial device arrives. Once the connection is established, data can be transmitted in both directions—from the host to the module, and from the module to the host.



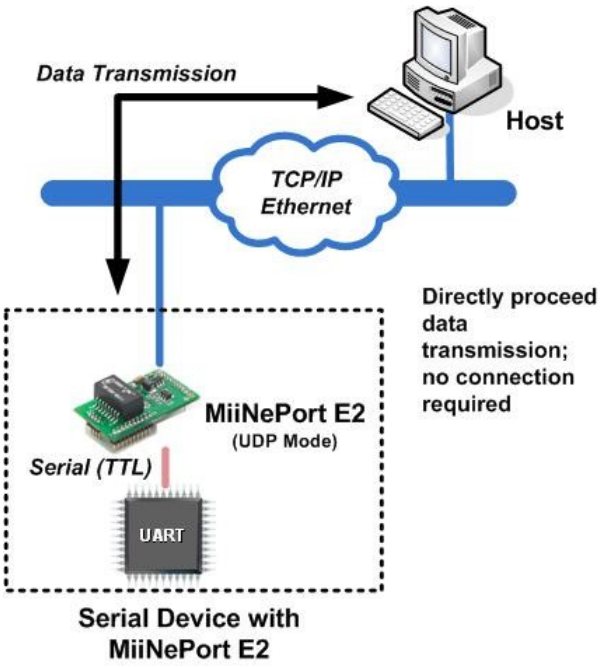
6.1.5. RFC2217 MODE

RFC2217 is an industrial public protocol for sharing serial devices over TCP/IP Ethernet networks. RFC2217 is similar to Moxa's proprietary Real COM mode in that it allows users to continue using software that was written for pure serial communications applications. Each module comes equipped with COM drivers for Windows systems (95 and above). The module's serial port is mapped by the driver to an IP address and port number. The driver intercepts data sent to the host's COM port, packs it into a TCP/IP packet, and then redirects it through the host's Ethernet card.



6.1.6. UDP MODE

UDP is similar to TCP but is faster and more efficient. Although data can be broadcast to or received from multiple network hosts, UDP does not support data verification, and consequently is not suitable for applications for which data integrity is critical. UDP would, however, be suitable for message display applications.



6.2. CHOOSING THE RIGHT CONFIGURATION TOOL

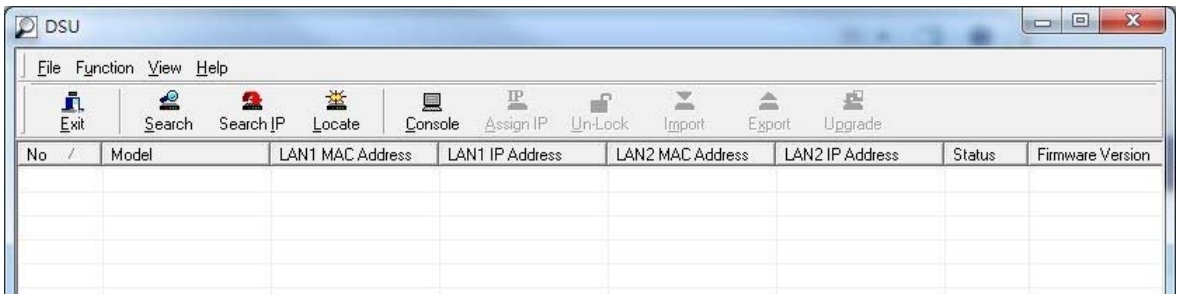
The MiiNePort supports several tools for configuring the module. In this chapter, we briefly describe the options available and appropriate situations for using those options.

The following topics are covered in this chapter:

- **Utility Console**
- **Web Console**
- **Telnet Console**
- **SCM (Serial Command Mode)**

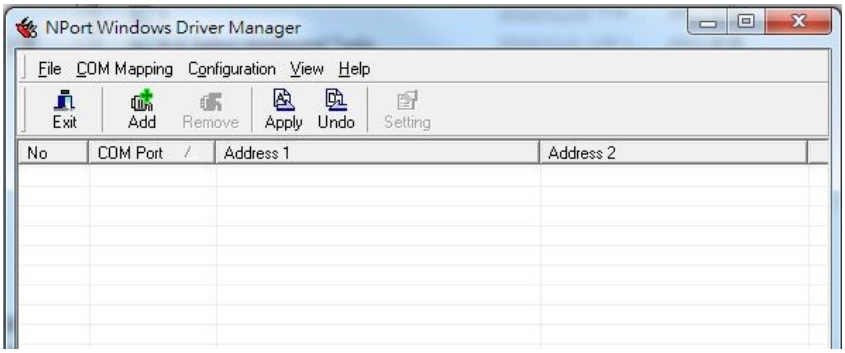
6.2.1. UTILITY CONSOLE

You can find the Device Search Utility on the CD-ROM that came with your product. NPort Search Utility is designed for Windows and is mainly used to search for the MiiNePort modules and for assigning IP addresses. Refer to the Web Console for additional configuration information.



6.2.2. NPORT WINDOWS DRIVER MANAGER

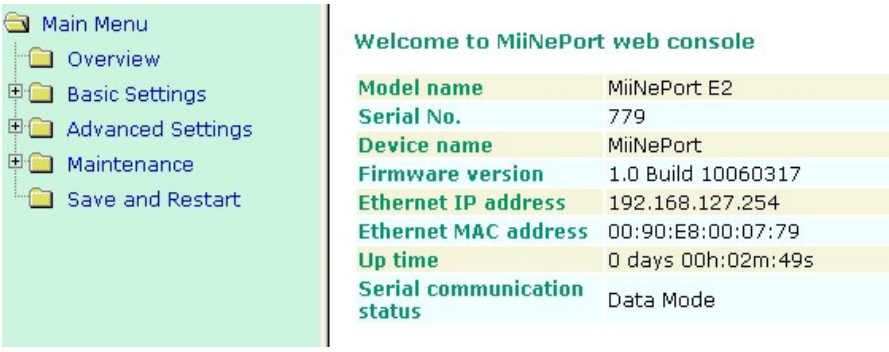
NPort Windows Driver Manager is intended for use with Real COM mode. The software manages the installation of drivers that allow you to map unused COM ports on your PC to serial ports on the MiiNePort.



Refer to **Chapter 6: Utility Console and Driver Installation** for details on how to use the Device Search Utility and NPort Windows Driver Manager.

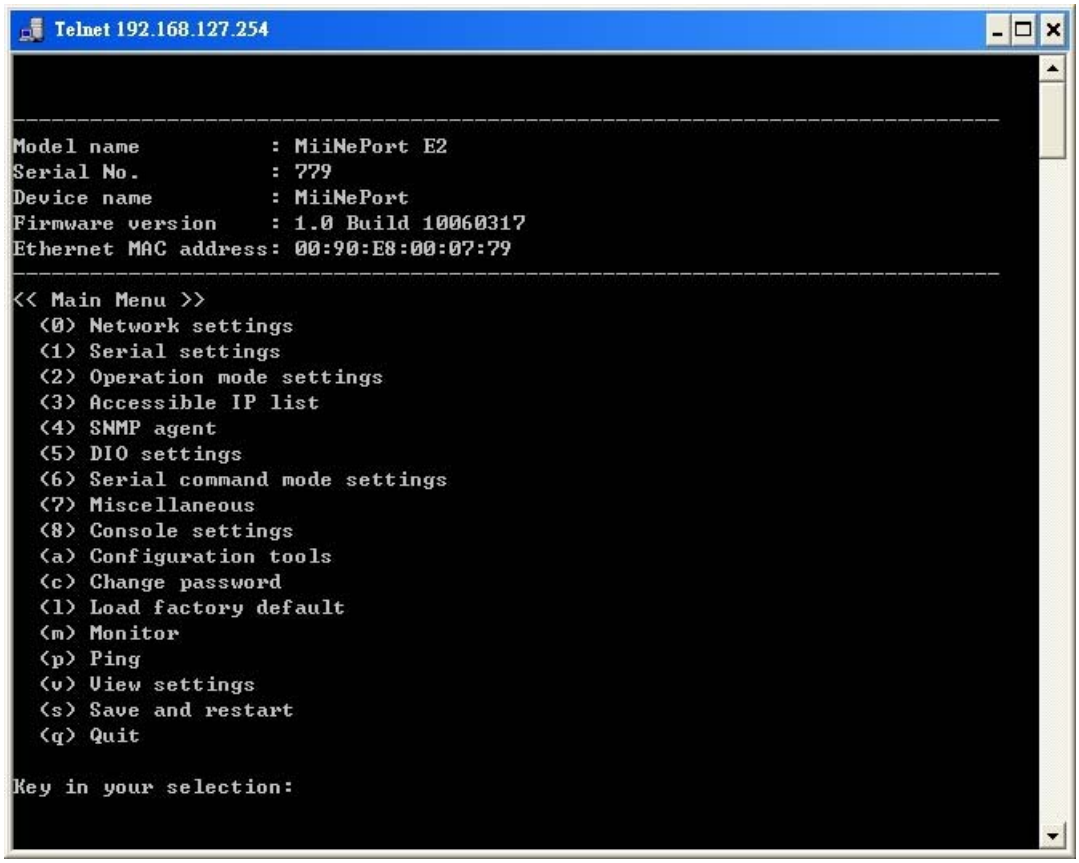
6.2.3. WEB CONSOLE

After locating a MiiNePort with the Device Search Utility, you may configure the MiiNePort using a standard web browser. Refer to **Chapter 7: Web Console Configuration** for details on how to access and use the MiiNePort web console.



6.2.4. TELNET CONSOLE

Your MiiNePort can be configured over the network with Telnet, which requires that the module has a network connection and an IP address. We briefly discuss Telnet console configuration in **Chapter 5: Initial IP Address Configuration**. All Telnet console commands are introduced in **Chapter 7: Web Console Configuration**.



6.3. INITIAL IP ADDRESS CONFIGURATION

When setting up your MiiNePort module for the first time, the first thing you should do is configure the IP address. This chapter introduces the methods that can be used to configure the module's IP address. For more details about network settings, refer to the Network Settings section in **Chapter 7: Web Console Configuration**.

The following topics are covered in this chapter:

- **Static vs. Dynamic IP Address**
- **Factory Default IP Address**
- **ARP**
- **Telnet Console**

6.3.1. STATIC VS. DYNAMIC IP ADDRESS

You should first determine whether the module will be assigned a Static IP or Dynamic IP (either DHCP or BOOTP application).

- If the module is used in a Static IP environment, you need to configure the IP address directly.
- If the module is used in a Dynamic IP environment, you need to configure the module to obtain an IP address dynamically with DHCP, DHCP/BOOTP, BOOTP, or AUTOIP.

ATTENTION

Consult your network administrator on how to reserve a fixed IP address for the module in the MAC-IP

mapping table when using a DHCP Server or BOOTP Server. For most applications, you should assign a fixed IP address to the module.

6.3.2. FACTORY DEFAULT IP ADDRESS

The MiiNePort module is configured with the following default private IP address:

192.168.127.254

IP addresses of the form 192.168.xxx.xxx are referred to as private IP addresses, since it is not possible to directly access a device configured with a private IP address from a public network. For example, you would not be able to ping such a device from an outside Internet connection. Applications that require sending data over a public network, such as the Internet, require setting up the server with a valid public IP address, which can be leased from a local ISP.

6.3.3. ARP

You can use the ARP (Address Resolution Protocol) command to set up the module's IP address. The ARP command tells your computer to associate the module's MAC address with the intended IP address. You must then use Telnet to access the module, at which point the module's IP address will be reconfigured.

ATTENTION

In order to use ARP, both your computer and the module must be connected to the same LAN. You may also use a crossover Ethernet cable to connect the module directly to your computer's Ethernet port. Your module must be configured with the factory default IP address before executing the ARP command.

To configure the IP address using ARP, follow these instructions:

✓ Obtain a valid IP address for the module from your network administrator.

✓ Obtain the module's MAC address from the label on the module.

✓ Execute the **arp -s** command from your computer's MS-DOS prompt by typing:

arp –s <new IP address> 00-90-E8-tt-tt-tt For example,

arp –s 192.168.200.100 00-90-E8-00-00-00

In this example, 192.168.200.100 is the new IP address and 00-90-E8-00-00-00 is the module's MAC address, as obtained in steps 1 and 2.

✓ Execute a special Telnet command by typing: **telnet <new IP address> 6000**

For example,

telnet 192.168.200.100 6000

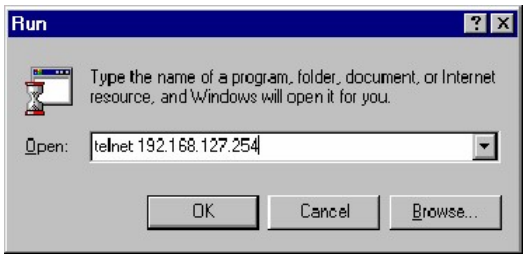
After issuing this command, a **Connect failed** message will appear. After the module reboots, its IP address will be updated to the new address, and you can reconnect the module using the utility, web, or Telnet console to verify that the update was successful.

6.3.4. TELNET CONSOLE

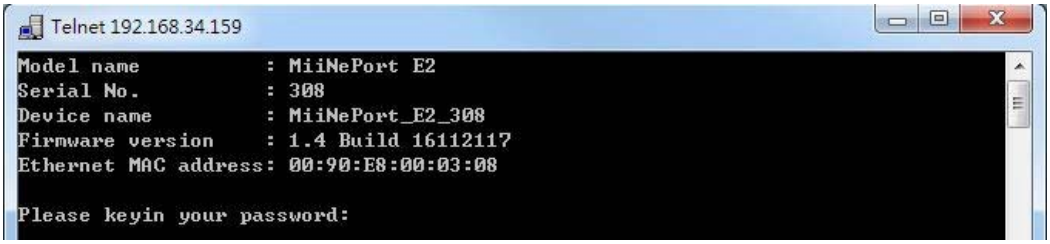
Depending on how your computer and network are configured, you may find it convenient to use network access to set up your module's IP address. This can be done using Telnet, which requires that the module has a network connection and an IP address.

☐ From the Windows desktop, click **Start** and then select **Run**.

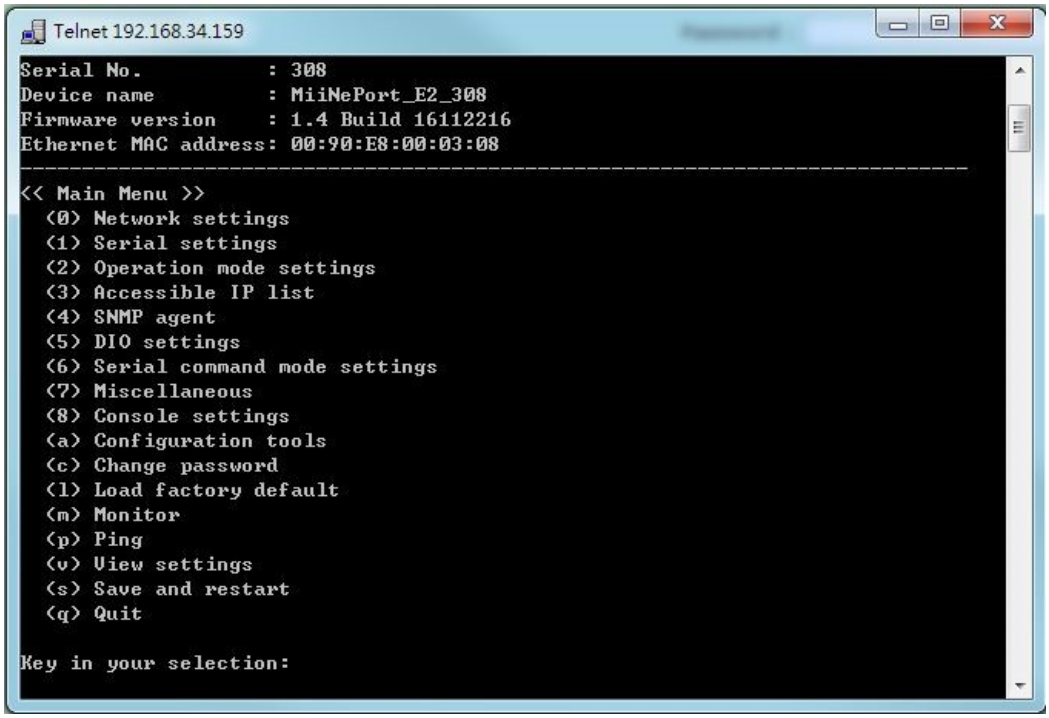
☐ Telnet to the module's current IP address. If this is the first time configuring the module, you will telnet to the default IP address by typing **telnet 192.168.127.254** in the **Open** text box. Click **OK** to proceed.



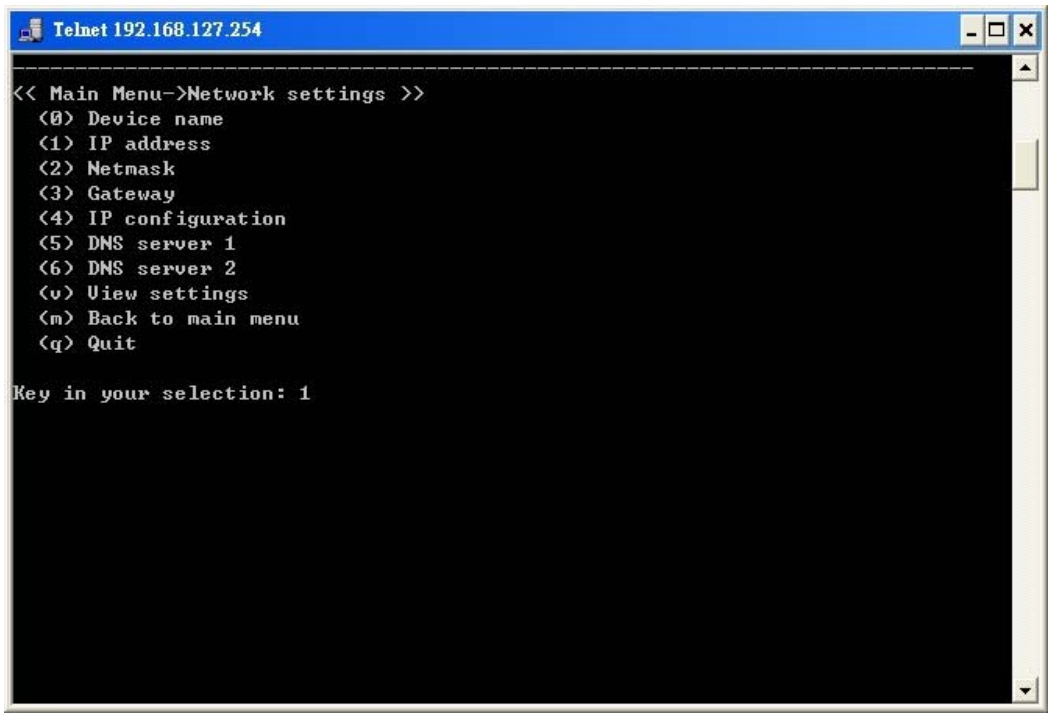
You will be asked to enter a password to access to the device. The default password for MiiNePort E2/E3 is **moxa**.



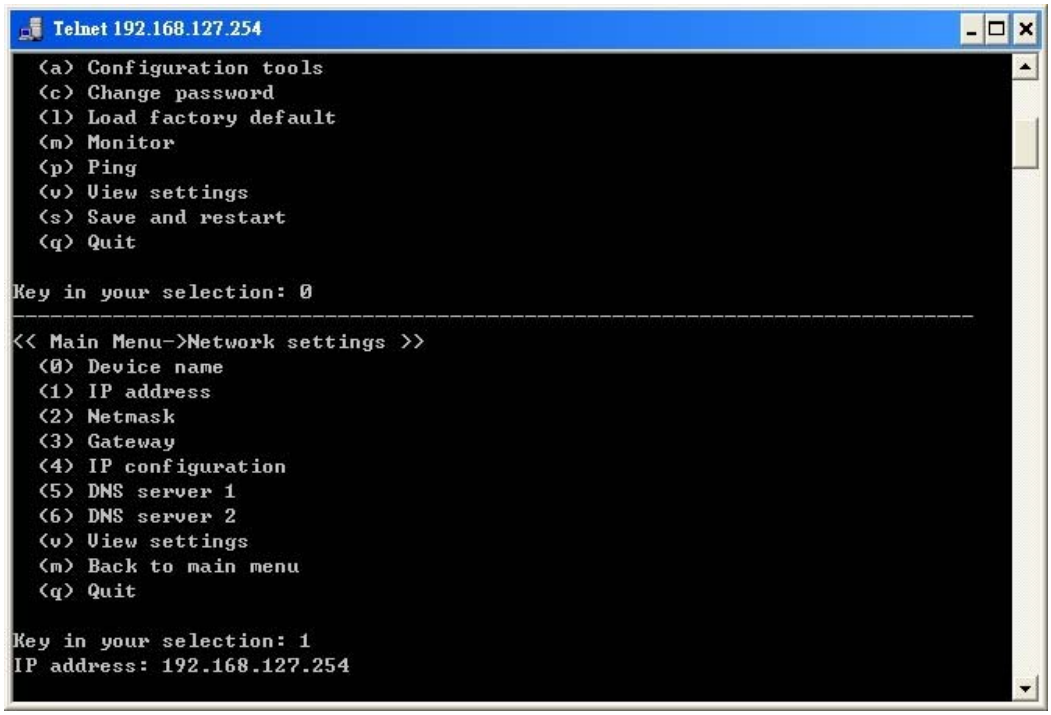
☐ Select Network settings by pressing **0** and then press **Enter**.



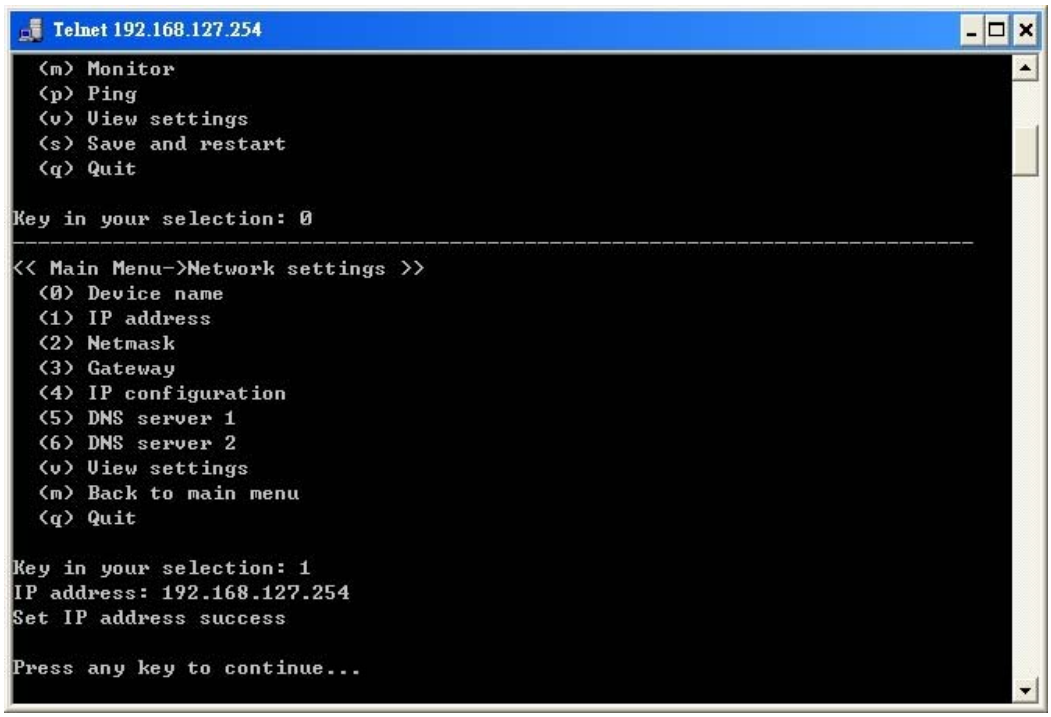
☐ Select IP address by pressing **1** and then press **Enter**.



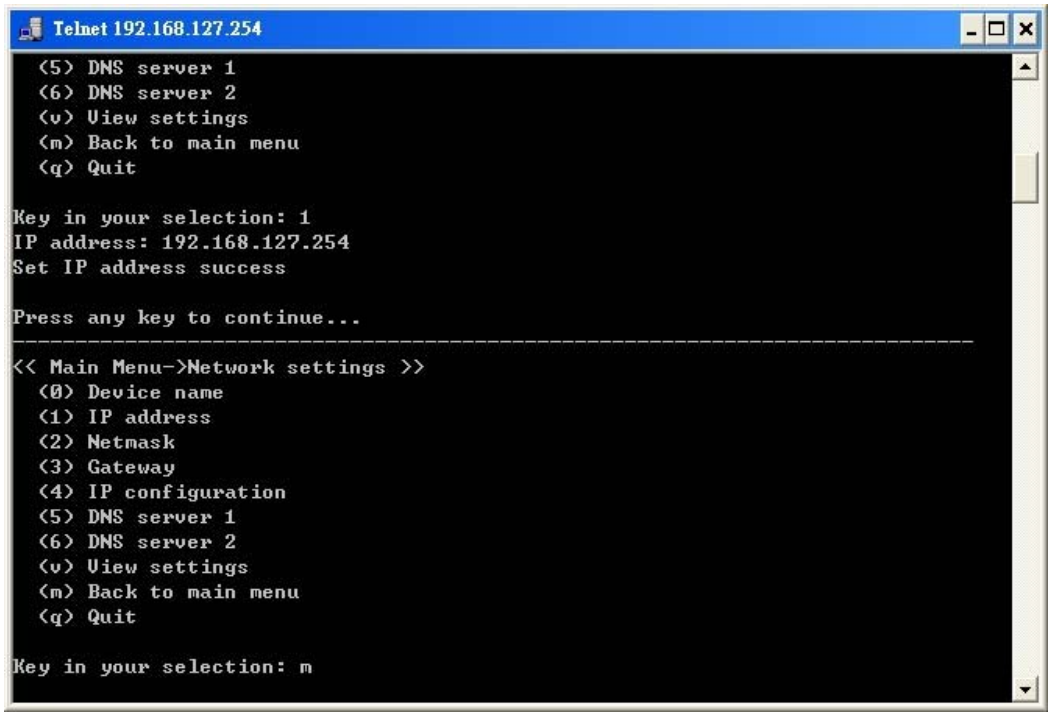
☐ Use the backspace key to erase the current IP address. Type in the new IP address and then press **Enter**.



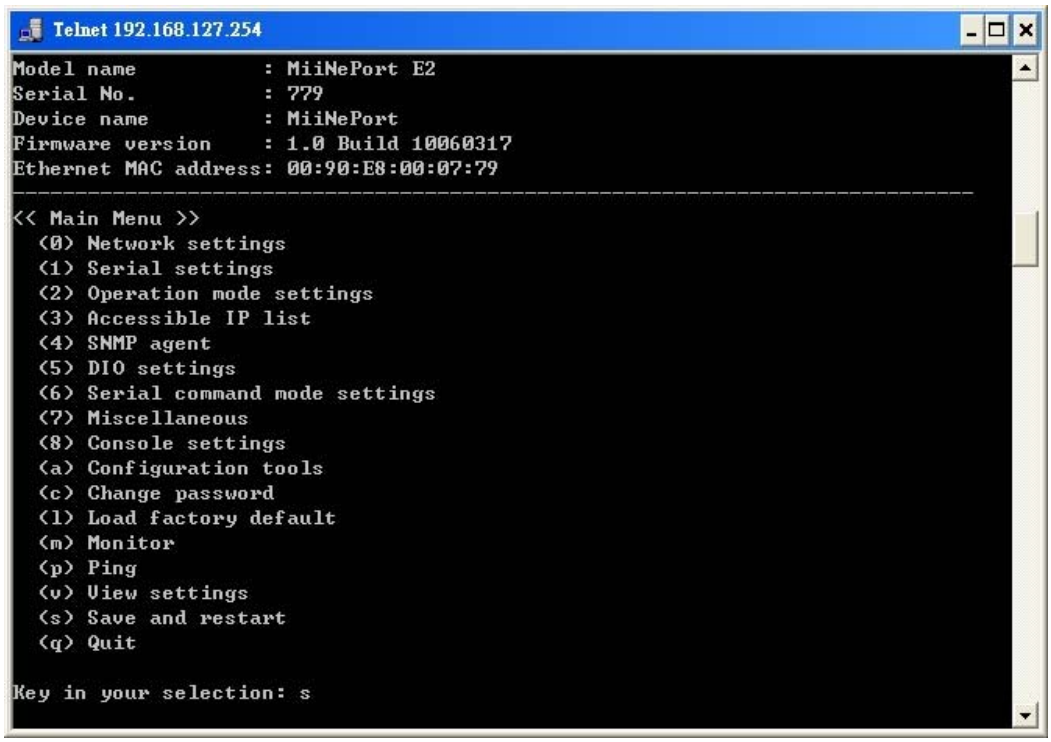
☐ Press any key to continue.



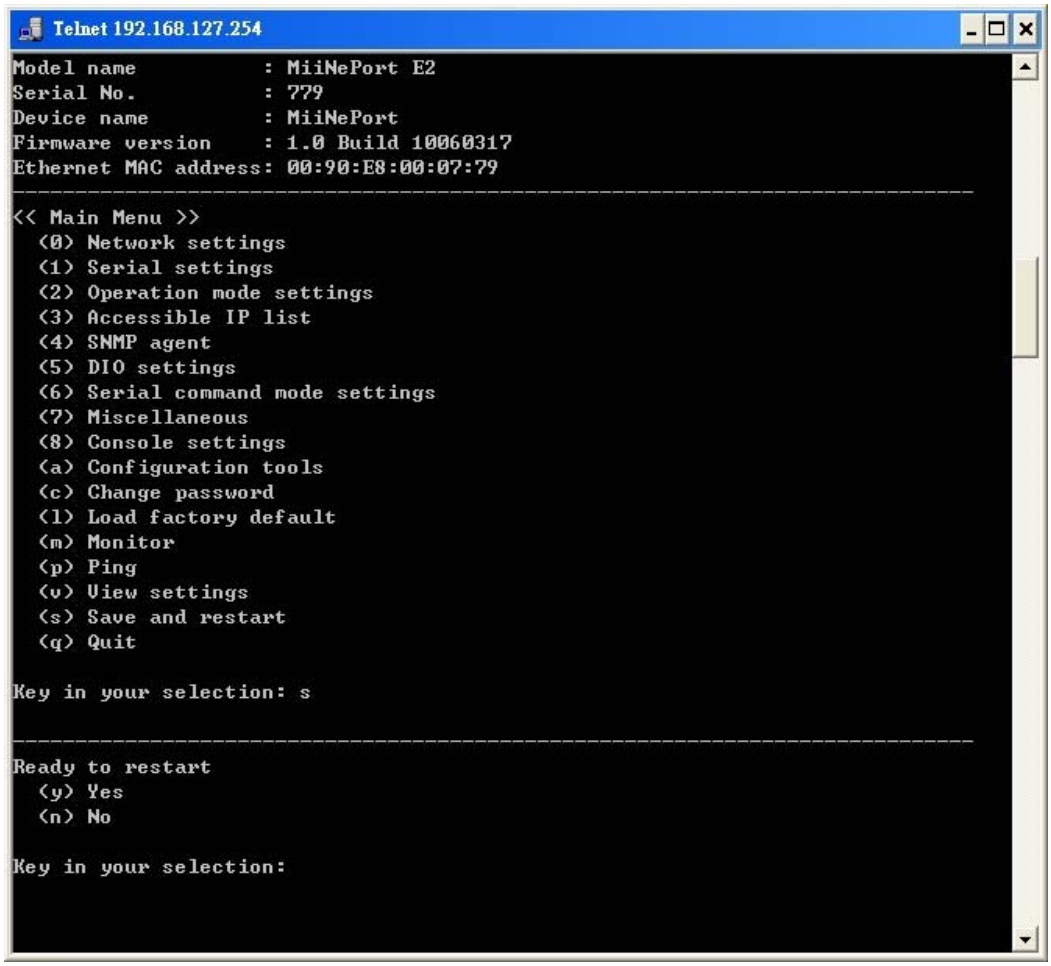
☐ Press **M** and then **Enter** to return to the main menu.



☐ Press **S** and then **Enter** to **Save/Restart** the system.



☐ Press **Y** and then **Enter** to save the new IP address and restart the module.



6.4. UTILITY CONSOLE AND DRIVER INSTALLATION

This chapter teaches you how to install the MiiNePort's utilities, use the utilities to perform simple configurations, and install the drivers.

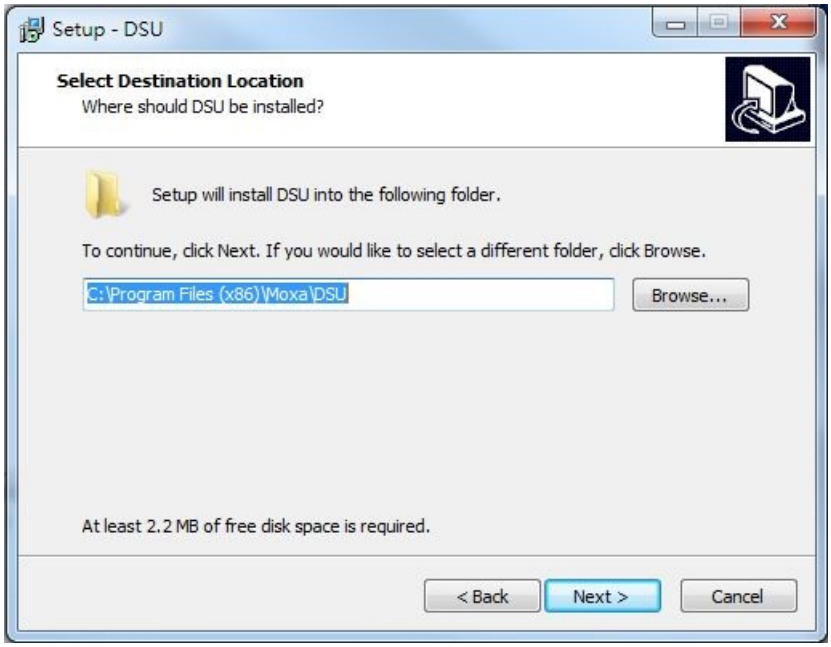
The following topics are covered in this chapter:

- ✓ **Device Search Utility (DSU)**
- o Installing the Device Search Utility
- o Device Search Utility Configuration **NPort Windows Driver Manager**
- o Installing NPort Windows Driver Manager
- o Using NPort Windows Driver Manager
- o Command Line Installation/Removal **The Linux Real TTY Driver**
- o Mapping TTY Ports
- o Removing Mapped TTY Ports
- o Removing Linux Driver Files
- ✓ **The UNIX Fixed TTY Driver**
- o Installing the UNIX Driver
- o Configuring the UNIX Driver

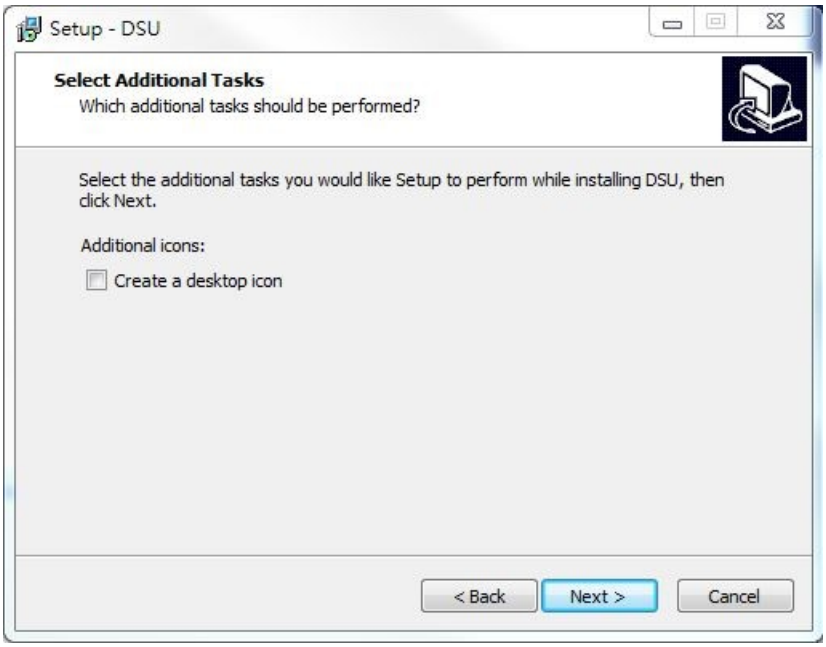
6.4.1. INSTALLING THE DEVICE SEARCH UTILITY

- Click the **INSTALL UTILITY** button in the MiiNePort Installation CD to install the Device Search Utility. Once the program starts running, click **Yes** to proceed.
- Click **Next** when the Welcome screen opens to proceed with the installation.

- Click **Browse** to select an alternate location and then click **Next** to install program files to the directory displayed in the input box.

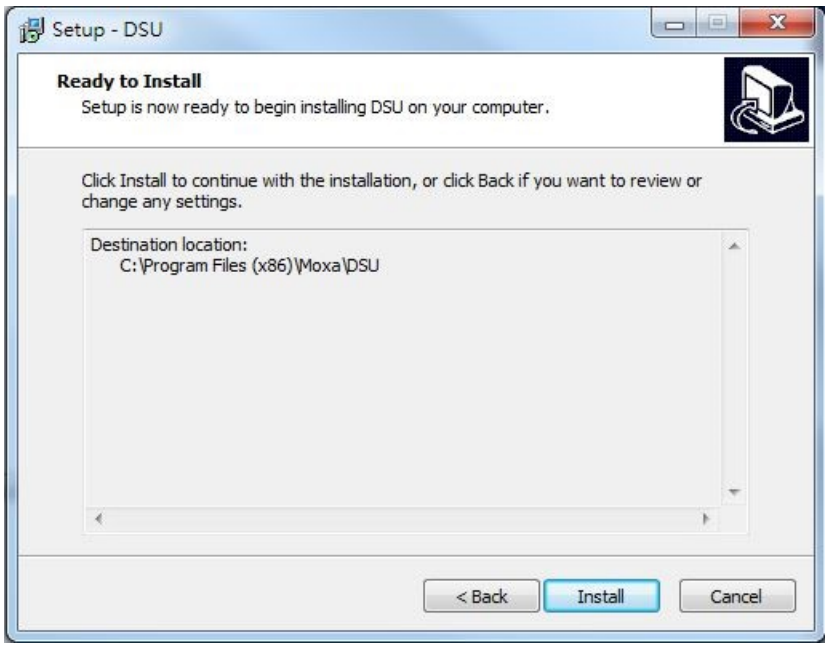


- Select the additional tasks you would like to set up to be performed while installing the DSU; then, click **Next**.

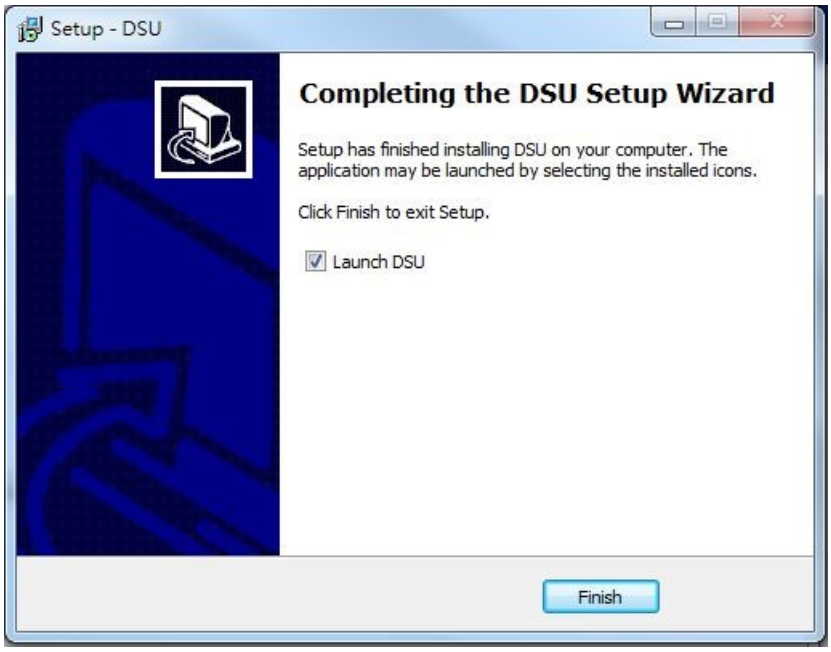


- The installer will display a summary of the installation options. Click **Install** to begin the installation.

The setup window will report the progress of the installation. To change the installation settings, click **Back** and navigate to the previous screen.



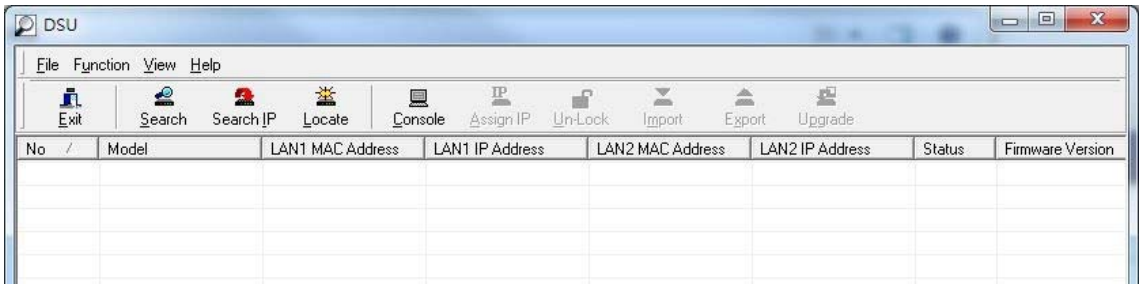
- **Finish** to complete the installation of the Device Search Utility.



6.4.2. DEVICE SEARCH UTILITY CONFIGURATION

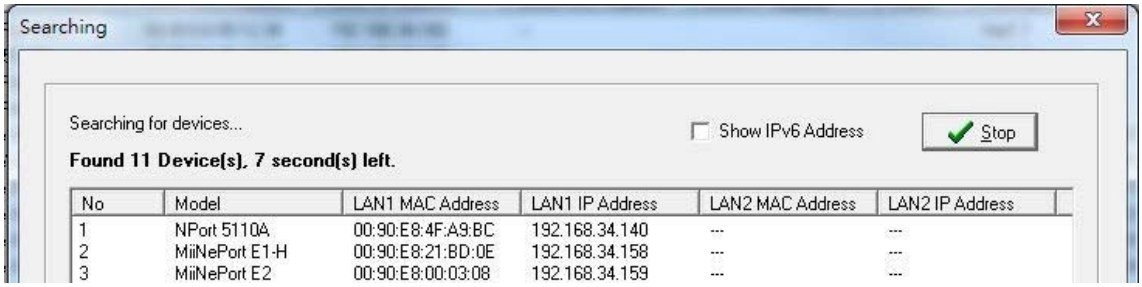
The Broadcast Search function is used to locate all MiiNePort modules that are connected to the same LAN as your computer. After locating a MiiNePort, you will be able to change its IP address. Since the Broadcast Search function searches by MAC address and not IP address, all MiiNePort modules connected to the LAN will be located, regardless of whether or not they are part of the same subnet as the host.

- Start the **Device Search Utility** and then click the **Search** icon.

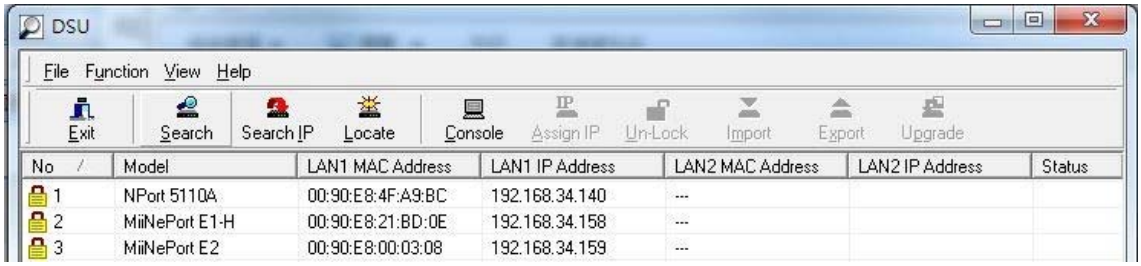


Note: Users running Windows Vista and Windows 7 will see a **User Account Control** pop-up and should allow the program.

- The **Searching** window indicates the progress of the search.



- When the search is complete, all MiiNePort modules that were located will be displayed in the Device Search Utility window.



- To modify the configuration of the highlighted MiiNePort, click the **Console** icon to open the web console.

This will take you to the web console, where you can make configuration changes. Refer to **Chapter 7: Web Console Configuration** for information on how to use the web console.

ATTENTION

If you are looking for information related to TCP Server, TCP Client, Ethernet Modem, RFC2217, or UDP

modes, you can ignore the following Driver sections, including NPort Windows Driver Manager and Linux Real TTY Driver, and instead jump directly to **Chapter 7: Web Console Configuration** for additional settings.

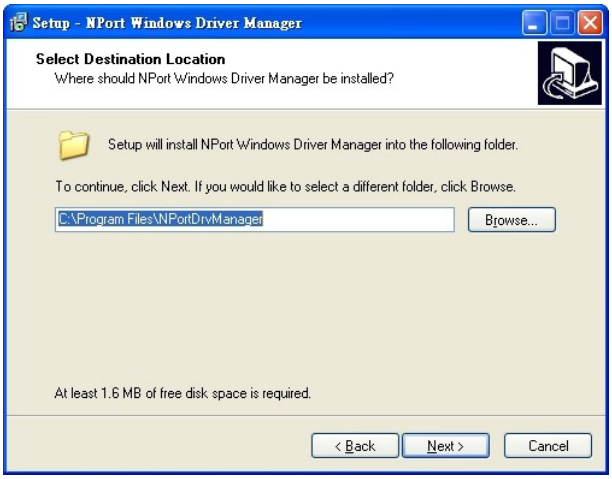
6.4.3. INSTALLING NPORT WINDOWS DRIVER MANAGER

The NPort Windows Driver Manager is intended for use with serial ports that are set to Real COM mode. The software manages the installation of drivers that allow you to map unused COM ports on your PC to your device through the MiiNePort's serial port. When the drivers are installed and configured, devices that are embedded with the MiiNePort will be treated as if they are attached to your PC's own COM ports.

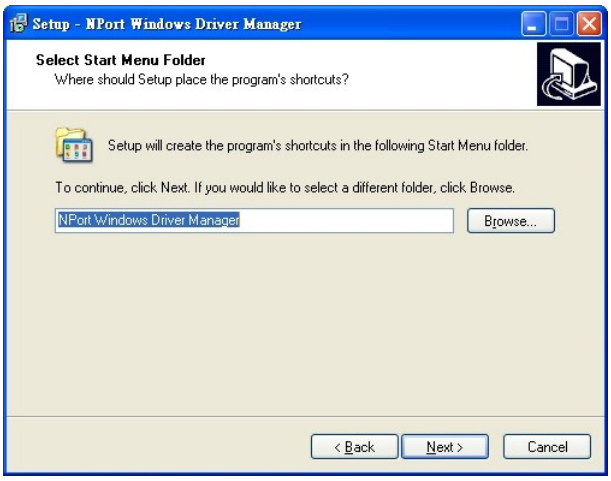
- Click the **INSTALL COM Driver** button in the MiiNePort Installation CD to install the NPort Windows Driver. Once the installation program starts running, click **Yes** to proceed.

- Click **Next** when the Welcome screen opens to proceed with the installation.

- **Browse** to select the destination directory and then click **Next** to install program files to the directory displayed in the input box.

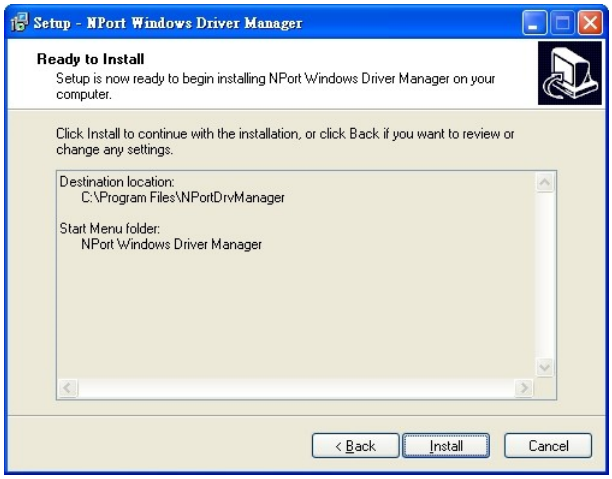


- Click **Next** to install the program's shortcuts in the appropriate **Start Menu** folder.



- The installer will display a summary of the installation options. Click **Install** to begin the installation.

The setup window will report the progress of the installation. To change the installation settings, click **Back** and navigate to the previous screen.



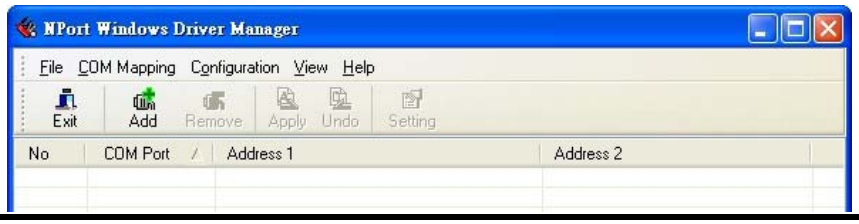
- **Finish** to complete the installation of NPort Windows Driver Manager.

6.4.4. USING NPORT WINDOWS DRIVER MANAGER

After you have installed the NPort Windows Driver Manager, you can set up the MiiNePort's serial port, which is connected to your device's main board, as remote COM ports for your PC host. Make sure that the serial port on your MiiNePort is already set to Real COM mode when mapping COM ports with the NPort Windows Driver Manager.

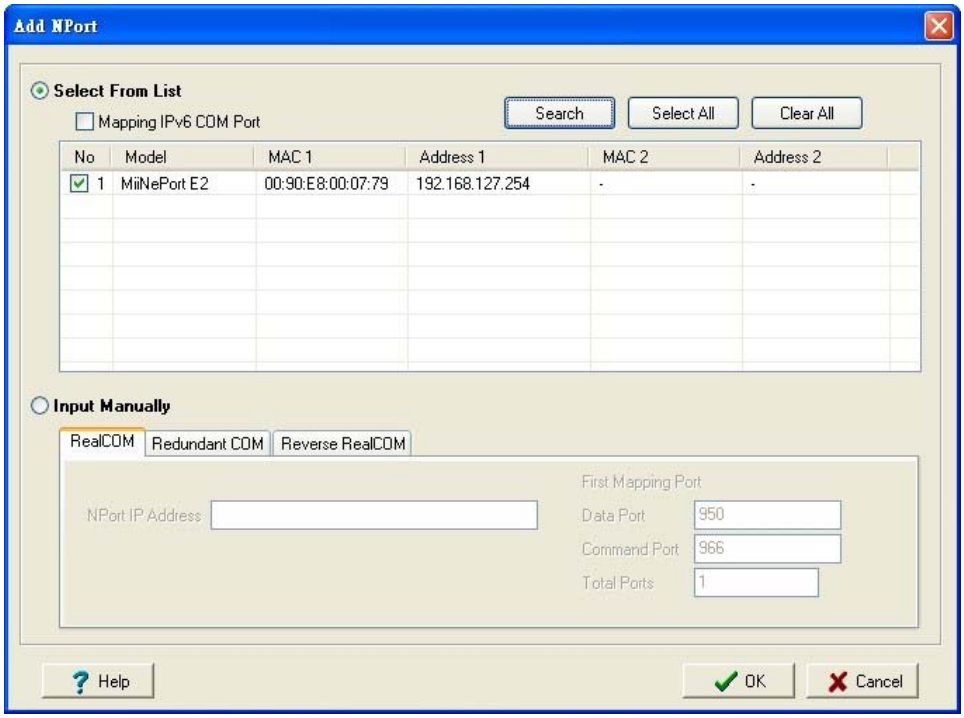
NOTE Refer to **Chapter 7: Web Console Configuration** to learn how to configure your MiiNePort to Real COM mode.

- Go to **Start** ⇨ **NPort Windows Driver Manager** ⇨ **NPort Windows Driver Manager** to start the COM mapping utility.
- Click the **Add** icon.

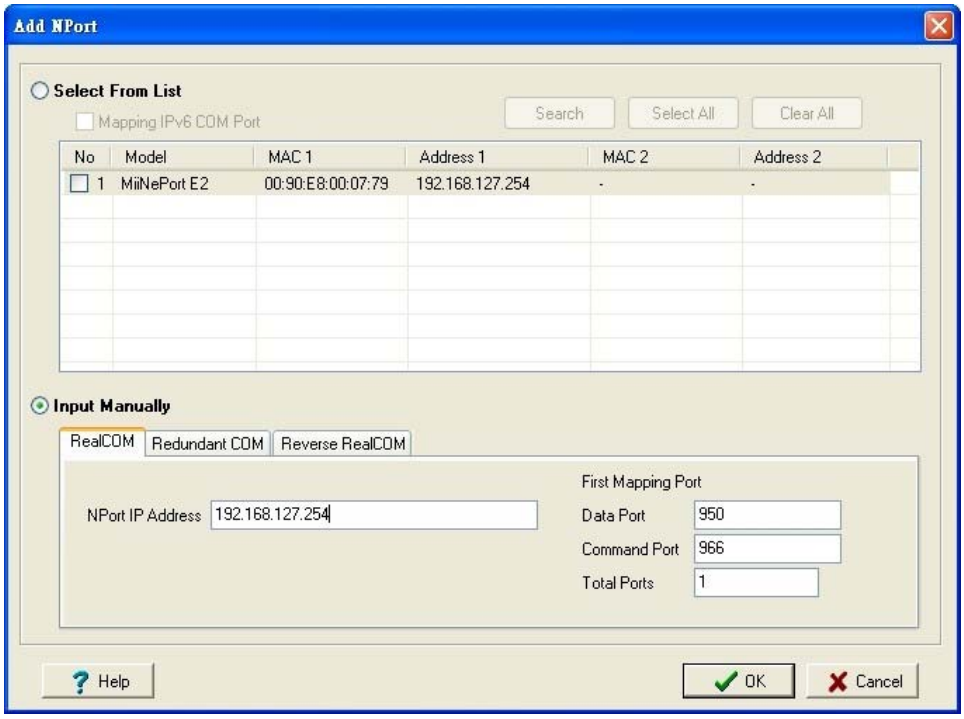


NOTE Users running Windows Vista and Windows 7 will see a **User Account Control** pop-up and should allow the program.

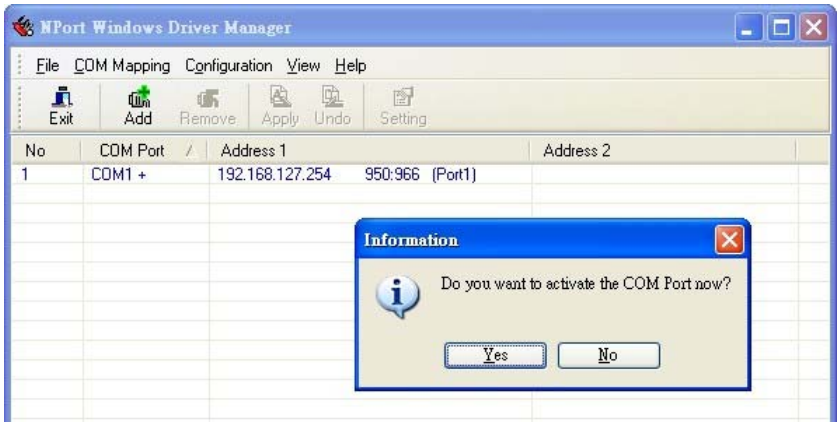
- **Search** to search for the MiiNePort modules. From the list that is generated, select the server to which you will map COM ports, and then click **OK**.



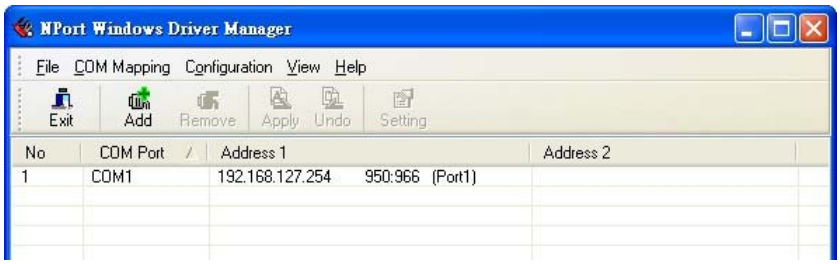
- Alternatively, you can select **Input Manually** and then manually enter the MiiNePort module's **IP Address**, **1st Data Port**, **1st Command Port**, and **Total Ports** to which COM ports will be mapped. Click **OK** to proceed to the next step. Note that the **Add NPort** page supports FQDN (Fully Qualified Domain Name), in which case the IP address will be filled in automatically.



- COM ports and their mappings will appear in blue until they are activated. Activating the COM ports saves the information in the host system registry and makes the COM port available for use. The host computer will not have the ability to use the COM port until the COM ports are activated. Click **Yes** to activate the COM ports at this time, or click **No** to activate the COM ports later.



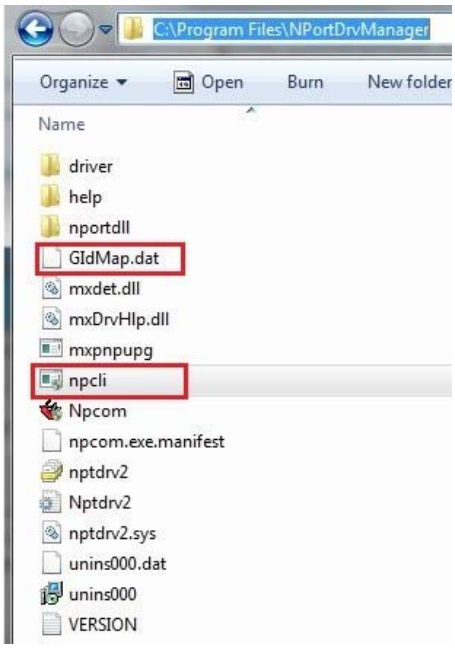
- Ports that have been activated will appear in black.



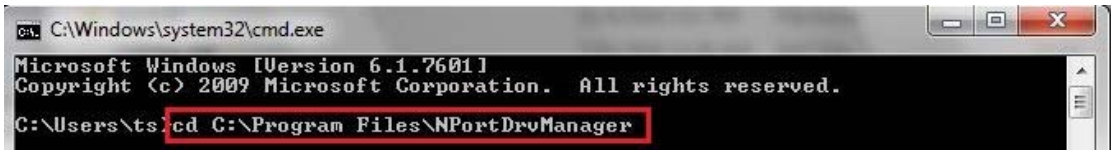
Command Line Installation/Removal

The NPort Windows Driver Manager v1.19 and above comes with a command-line script tool – ***npcli.exe*** for installation, removal of the driver, and configuring NPort driver functions.

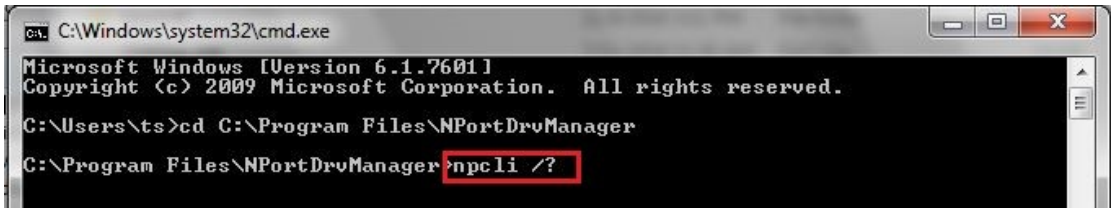
After successfully installing the NPort Windows Driver Manager v1.19 (or above), the default file path is ***C:\Program Files\NPortDrvManager*** as shown below. The main files that support the NPort command-line tool are ***npcli.exe*** and ***GIdMap.dat***. You may move these two files to your preferred location.



Once the NPort Windows Driver Manager v1.19 (or above) is installed, call up the ***cmd*** screen on your computer. Change the directory to the drive where you place the above two files.



Type ***npcli /?*** to get detailed information of what command lines are supported and the function descriptions.



The usage instructions will show up for user's reference.

6.4.5. THE LINUX REAL TTY DRIVER

9. Obtain the driver file from the included CD-ROM or the Moxa website, at http://www.moxa.com.

10. Log in to the console as a super user (root).

11. Execute **cd** / to go to the root directory.

12. Copy the driver file **npreal2xx.tg**z to the / directory.

13. Execute tar xvfz **npreal2xx.tgz** to extract all files into the system.

14. Execute **/tmp/moxa/mxinst**.

For RedHat AS/ES/WS and Fedora Core1, append an extra argument as follows: **#**
/tmp/moxa/mxinst SP1

The shell script will install the driver files automatically.

15. After installing the driver, you will be able to see several files in the **/usr/lib/npreal2/driver** folder:

> **mxaddsvr** (Add Server, mapping tty port)

> **mxdelsvr** (Delete Server, un-mapping tty port)

> **mxloadsvr** (Reload Server)

> **mxmknod** (Create device node/tty port)

> **mxrmnod** (Remove device node/tty port)

> **mxuninst** (Remove tty port and driver files)

At this point, you will be ready to map the MiiNePort serial port to the system tty port.

6.4.6. MAPPING TTY PORTS

Be sure to set the operation mode of the serial port of the MiiNePort to Real COM mode. After logging in as a super user, enter the directory **/usr/lib/npreal2/driver** and then execute **mxaddsvr** to map the target MiiNePort serial port to the host tty ports. The syntax of **mxaddsvr** is as follows: **mxaddsvr [MiiNePort IP Address] [Total Ports] ([Data port] [Cmd port])** The **mxaddsvr** command performs the following actions:

6. Modifies npreal2d.cf.

7. Creates tty ports in directory /dev with major and minor number configured in npreal2d.cf.

8. Restarts the driver.

6.4.7. MAPPING TTY PORTS AUTOMATICALLY

To map tty ports automatically, execute mxaddsvr with just the IP address and number of ports, as in the following example:

 # cd /usr/lib/npreal2/driver
 # ./mxaddsvr 192.168.3.4 16

In this example, 16 tty ports will be added, all with IP 192.168.3.4, with data ports from 950 to 965 and command ports from 966 to 981.

6.4.8. MAPPING TTY PORTS MANUALLY

To map tty ports manually, execute **mxaddsvr** and manually specify the data and command ports, as in the following example:

 # cd /usr/lib/npreal2/driver
 # ./mxaddsvr 192.168.3.4 16 4001 966

In this example, 16 tty ports will be added, all with IP 192.168.3.4, with data ports from 4001 to 4016 and command ports from 966 to 981.

6.4.9. REMOVING MAPPED TTY PORTS

After logging in as root, enter the directory **/usr/lib/npreal2/driver** and then execute **mxdelsvr** to delete a server. The syntax of mxdelsvr is:

mxdelsvr [IP Address] Example:

 # cd /usr/lib/npreal2/driver # ./mxdelsvr
192.168.3.4

The following actions are performed when executing mxdelsvr:

- **npreal2d.cf** is modified.
- Relevant tty ports in directory **/dev** are removed.
- The driver is restarted.

If the IP address is not provided in the command line, the program will list the installed servers and total ports on the screen. You will need to choose a server for deletion from the list.

6.4.10. REMOVING LINUX DRIVER FILES

A utility is included that will remove all driver files, mapped tty ports, and unload the driver. To do this, you only need to enter the directory **/usr/lib/npreal2/driver**, and then execute **mxuninst** to uninstall the driver. The following actions will be performed:

- The driver is unloaded.
- All files and directories in **/usr/lib/npreal2** are deleted.

- The directory **/usr/lib/npreal2** is deleted.
- The script file that initializes the system is modified.

6.4.11. INSTALLING THE UNIX DRIVER

1. Log in to UNIX and create a directory for the Moxa TTY. To create a directory named /usr/etc, execute the command:

 # mkdir –p /usr/etc

2. Copy **moxattyd.tar** to the directory you created. If you created the **/usr/etc** directory above, you would execute the following commands:

 # cp moxattyd.tar /usr/etc
 # cd /usr/etc

3. Extract the source files from the tar file by executing the command:

 # tar xvf moxattyd.tar

The following files will be extracted:

README.TXT **moxattyd.c** --- source code
moxattyd.cf --- an empty configuration file **Makefile** --- makefile
VERSION.TXT --- fixed tty driver version
FAQ.TXT

4. Compile and Link

For SCO UNIX:
 # make sco

For UnixWare 7:

 # make svr5

For UnixWare 2.1.x, SVR4.2:

 # make svr42

6.4.12. CONFIGURING THE UNIX DRIVER

Modify the configuration:

The configuration used by the **moxattyd program** is defined in the text file **moxattyd.cf**, which is in the same directory that contains the program **moxattyd**. You may use **vi**, or any text editor to modify the file, as follows: **ttyp1 192.168.1.1 950**

For more configuration information, view the file **moxattyd.cf**, which contains detailed descriptions of the various configuration parameters.

NOTE	The "Device Name" depends on the OS. See the Device Naming Rule section in README.TXT for more information.

Start the moxattyd daemon that you configured in moxattyd.cf. If you would like to set the connection timeout, you can add a "-t min" parameter to specify the connection timeout value in minutes. For example:

 # /usr/etc/moxattyd/moxattyd -t 1

NOTE	You will now be able to use tty, which is configured in moxattyd.cf.

To start the moxattyd daemon after system bootup, add an entry into **/etc/inittab**, with the tty name you configured in **moxattyd.cf**, as in the following example: **ts:2:respawn:/usr/etc/moxattyd/moxattyd –t 1**

6.4.13. DEVICE NAMING RULE

For UnixWare 7, UnixWare 2.1.x, and SVR4.2, use:

pts/**[*n*]**

For all other UNIX operating systems, use:

ttyp**[*n*]**

6.4.14. ADDING AN ADDITIONAL SERVER

Modify the text file **moxattyd.cf** to add an additional server. You may use **vi** or any text editor to modify the file. For more configuration information, look at the file **moxattyd.cf**, which contains detailed descriptions of the various configuration parameters.

Find the process ID (PID) of the program **moxattyd**. **# ps -ef |**
grep moxattyd

Update the configuration of the **moxattyd** program. **#** kill -
USR1 **[*PID*]**
(e.g., if moxattyd PID = 404, kill -USR1 404**)**

Execute the moxattyd program again to activate the new settings, as follows:

 # /usr/etc/moxattyd/moxattyd -t 1

This completes the process of adding an additional server.

6.5. WEB CONSOLE CONFIGURATION

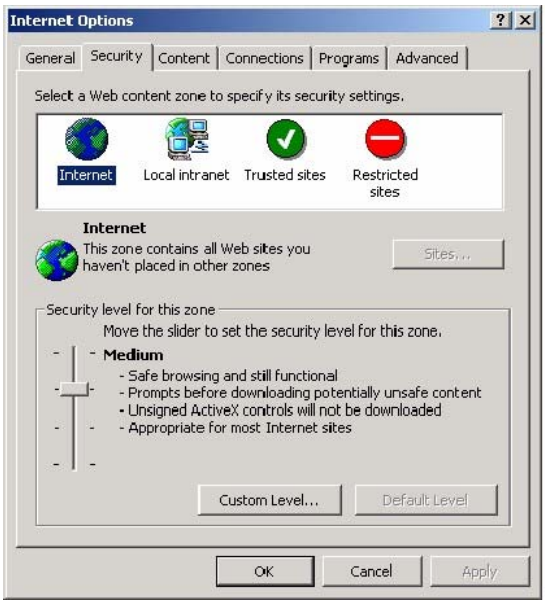
The web console is the most user-friendly way to configure your MiiNePort Series module. This chapter introduces the web console function groups and function definitions.

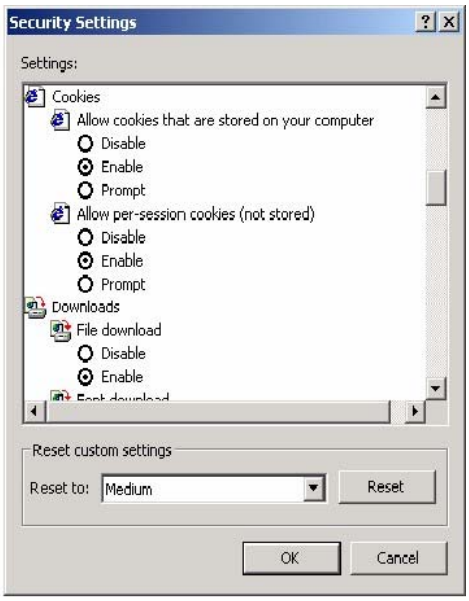
The following topics are covered in this chapter:

1. **Opening Your Brower**
2. **Web Console Fundamentals Basic Settings**
    1. Network Settings
    2. Serial Port Settings
    3. Operation Modes
3. **Advanced Settings**
    1. Accessible IP List
    2. SNMP Agent
    3. DIO Settings
    4. Serial Command Mode (SCM)
    5. Miscellaneous
4. **Maintenance**
    1. Console Settings
    2. Firmware Upgrade
    3. Configuration Tools
    4. Change Password

6.5.1. OPENING YOUR BROWER

1. Open your browser with the cookie function enabled. (To enable your Internet Explorer for cookies, right click on your desktop Internet Explorer icon, select **Properties**, click on the **Security** tab, and then select the three Enable options as shown in the figure below.)





2. Type 192.168.127.254 in the **Address** box (use the correct IP address if different from the default), and then press **Enter**. You will be asked to enter a password to access the device. The default password for MiiNePort E1 is **moxa**.

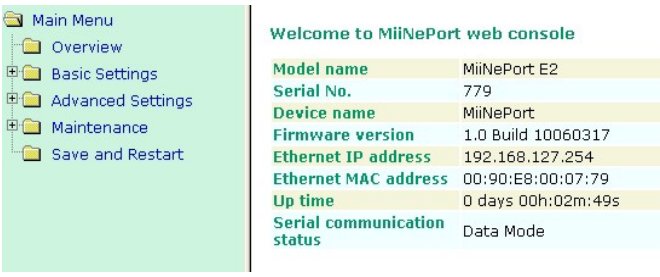
ATTENTION

If you use other web browsers, remember to enable the functions to **allow cookies that are stored on your computer** or **allow per-session cookies**. MiiNePort modules only use cookies for password transmission.

ATTENTION

Refer to **Chapter 5: Initial IP Address Configuration** for instructions on IP configuration.

3. The web console will open. On this page, you can see a brief description of the web console's function groups in the left part of the page. You can also see a configuration overview of your MiiNePort module.



6.5.2. WEB CONSOLE FUNDAMENTALS

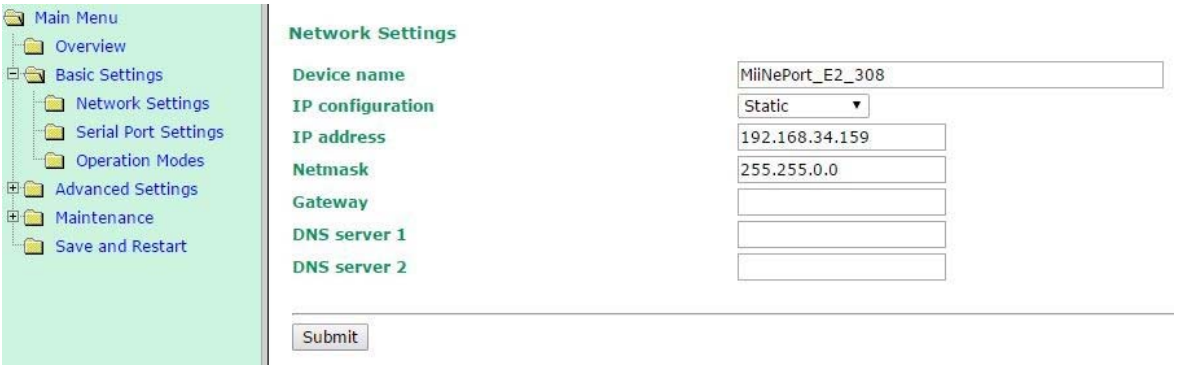
In the web console, the left panel is the navigation panel. It contains an expandable menu tree for navigating among the various settings and categories. When you click on a menu item in the navigation panel, the main window will display the corresponding options for that item.

Configuration changes can then be made in the main window. For example, if you click on **Basic Settings** ⍰ **Network Settings** in the navigation panel, the main window will show a page of network settings that you can configure.

You must click on the **Submit** button to keep your configuration changes. The Submit button will be located at the bottom of every page that has configurable settings. If you navigate to another page without clicking the Submit button, your settings will not be retained.

Changes will not take effect until they are saved and the module is restarted! You may complete this in one step by clicking on **Save/Restart** after you submit a change. If you restart the module without saving your configuration, the module will discard all submitted changes.

6.6. NETWORK SETTINGS



You must assign a valid IP address to the module before it will work in your network environment. Your network system administrator should provide you with an IP address and related settings for your network. The IP address must be unique within the network; otherwise the module will not have a valid connection to the network. First-time users can refer to **Chapter 5: Initial IP Address Configuration** for more information.

6.6.1. DEVICE NAME

Setting	Factory Default	Necessity
1 to 39 characters	[model name]_[Serial No.]	Optional

This option can be used to specify the location or application of the module, which may be useful when managing more than one module on the network.

6.6.2. IP CONFIGURATION

Method	Function Definition
Static	User defined IP address, Netmask, Gateway.
DHCP	DHCP Server assigned IP address, Netmask, Gateway, DNS
DHCP/BOOTP	DHCP Server assigned IP address, Netmask, Gateway, DNS, or BOOTP Server assigned IP address
BOOTP	BOOTP Server assigned IP address
AUTOIP	AUTOIP protocols automatically negotiate and assign IP in 169.254/16 network

IP configuration is a required field. The default setting is Static.

ATTENTION

In Dynamic IP environments, the module will attempt to obtain an IP address from the DHCP or BOOTP server

3 times at 30-second intervals. The timeout for the first try will be 1 second, the second try will be 3 seconds, and the last try will be 5 seconds.

If the DHCP/BOOTP Server is unavailable, the module will use the default IP address (192.168.127.254), netmask, and gateway settings.

6.6.3. IP ADDRESS

Setting	Description	Factory Default
E.g., 192.168.1.1 (IP addresses of the form x.x.x.0 and x.x.x.255 are invalid.)	192.168.127.254	Required

An IP address is a number assigned to a network device, such as a computer, as a permanent address on the network. Computers use the IP address to identify and talk to each other over the network. Choose a proper IP address that is unique and valid in your network environment. ***Netmask***

Setting	Factory Default	Necessity
E.g., 255.255.255.0	255.255.255.0	Required

A subnet mask represents all the network hosts at one geographic location, in one building, or on the same local area network. When a packet is sent out over the network, the module will use the subnet mask to check whether the host specified in the packet is on a local network segment. If the address is on the same network segment as the module, a connection is established directly from the module. Otherwise, the connection is established through the default gateway. ***Gateway***

Setting	Factory Default	Necessity
E.g., 192.168.1.1	None	Optional

A gateway acts as an entrance to another network. Usually, the computers that control traffic within the network or at the local Internet service provider are gateway nodes. The module needs to know the IP address of the default gateway computer in order to communicate with the hosts outside the local network environment. For correct gateway IP address information, consult your network administrator.

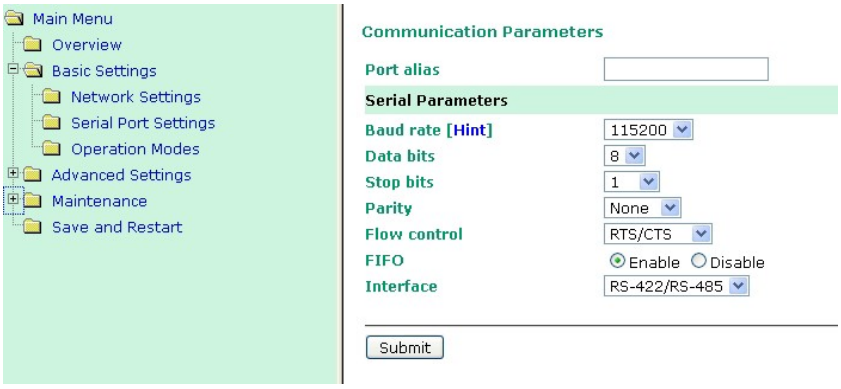
6.6.4. DNS SERVER 1 / DNS SERVER 2

Setting	Factory Default	Necessity
E.g., 192.168.1.1 (IP addresses of the form x.x.x.0 and x.x.x.255 are invalid)	None	Optional

The Domain Name System (DNS) is used to identify and translate Internet domain names into IP addresses. A domain name is an alphanumeric name, such as moxa.com, that is usually easier to remember. A DNS server is a host that translates the text-based domain name into the corresponding numeric IP address, which is used to establish a TCP/IP connection. When the user enters a website address, the computer asks a DNS server for the website's IP address to connect to the web server.

When a DNS server is specified, the module acts as a DNS client and will allow domain names instead of IP addresses to be used on the web console. The following web console fields support the use of domain names: TCP Client-Destination IP Address, and IP Address Report Server. Two DNS servers can be specified, DNS

6.7. SERIAL PORT SETTINGS



6.7.1. PORT ALIAS

Setting	Factory Default	Necessity
1 to 15 characters (E.g., PLC-No.1)	None	Optional

This function is designed for future use. You may enter a string to help in the module's serial port from other serial ports.

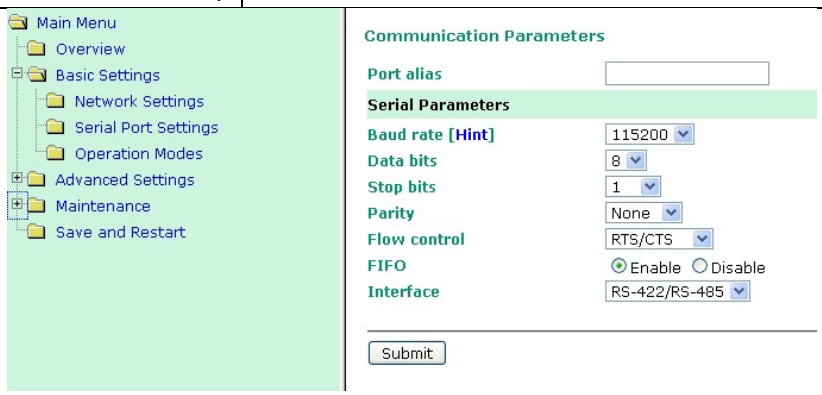
ATTENTION

Refer to the serial communication parameters in your serial device's user's manual. The module's serial

parameters should be the same as the parameters used by your serial device.

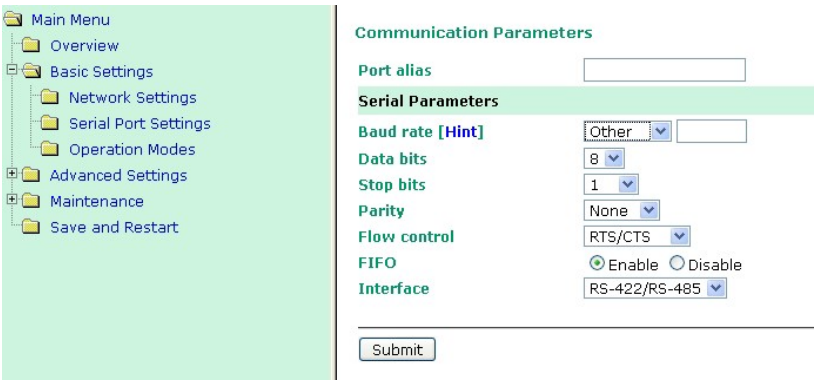
6.7.2. BAUDRATE

Setting	Factory Default	Necessity
50 bps to 921.6 Kbps (supports non-standard baudrates)	115.2 Kbps	Required



The MiiNePort supports the **Any Baudrate (non-standard baudrate)** feature. If your baudrate is not listed, select **Other** from the drop-down list and type the baudrate in the input box.

The MiiNePort will use the closest baudrate that is supported.



6.7.3. DATA BITS

Setting	Factory Default	Necessity
5, 6, 7, 8	8	Required

Stop Bits

Setting	Factory Default	Necessity
1, 1.5, 2	1	Required

Stop Bits will be set to 1.5 when Data Bits is set to 5 bits.

Parity

Setting	Factory Default	Necessity
None, Even, Odd, Space, Mark	None	Required

Flow control

Setting	Factory Default	Necessity
None, RTS/CTS, DTR/DSR, XON/XOFF	RTS/CTS	Required

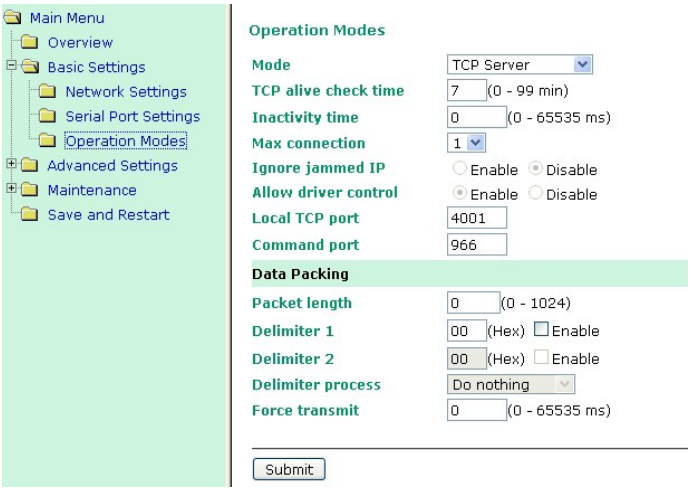
FIFO

Setting	Factory Default	Necessity
Enable, Disable	Enable	Required

Each module's serial port provides a 128-byte FIFO both in the Tx and Rx directions. Disable the FIFO setting when your serial device does not have a FIFO to prevent data loss during communication. ***Interface***

Setting	Description	Necessity
RS-232/422/485	RS-232, RS-422/485	Required

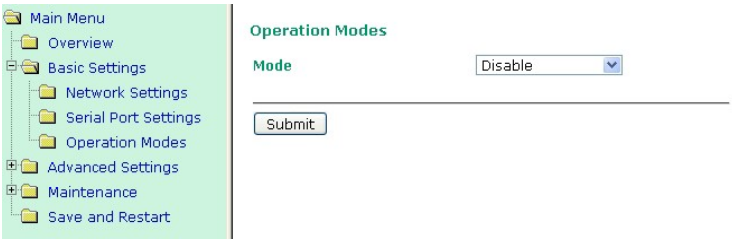
6.8. OPERATION MODES



Before reading this section, refer to **Chapter 3: Choosing the Proper Operation Mode** to select the operation mode that best fits your device application.

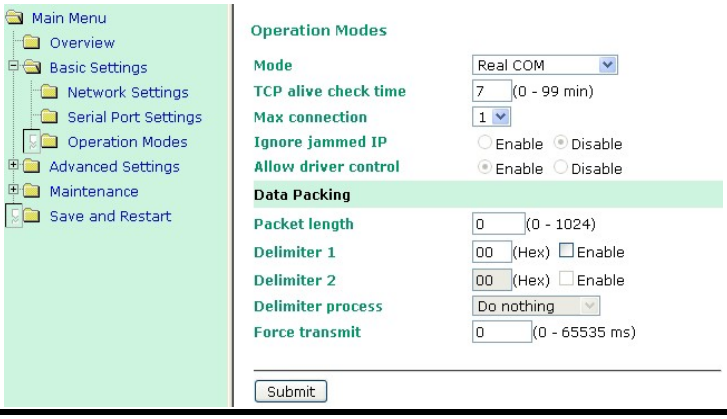
Click **Operation Modes**, located under the Main Menu, to display the operating settings for the MiiNePort's two serial ports.

6.8.1. DISABLE MODE



When **Mode** is set to **Disable**, that particular port will be disabled.

6.8.2. REAL COM MODE



ATTENTION

To use Real COM mode, refer to **Chapter 6: Utility Console and Driver Installation** for instructions on

how to install the Real COM driver on Windows or Linux machines.

TCP alive check time

Setting	Factory Default	Necessity
0 to 99 min	7 min	Optional

1. **min:** The TCP connection is not closed due to an idle TCP connection.

2. **to 99 min:** The module automatically closes the TCP connection if there is no TCP activity for the given time. After the connection is closed, the module starts listening for another host's TCP connection.

Max connection

Setting	Factory Default	Necessity
1, 2, 3, 4	1	Required

Max connection is used when the device needs to receive data from different hosts simultaneously.

The factory default only allows one connection at a time. When Max Connection is set to 1, the Real COM driver on the specific host has full control.

Max connection 1: The module will only allow one host's Real COM driver to open a connection to the module's serial port.

Max connections 2 to 4: When set to 2 or higher, Real COM drivers for up to the specified number of hosts may open this port at the same time. When Real COM drivers for multiple hosts open the port at the same time, the COM driver only provides a pure data tunnel with no control ability. The serial port parameters will use firmware settings instead of your application program (AP) settings.

Application software that is based on the COM driver will receive a driver response of "success" when the software uses any of the Win32 API functions. The firmware will only send data back to the driver on the host. Data will be sent first-in,-first-out when data is received by the MiiNePort from the Ethernet interface.

ATTENTION

When Max connection is greater than one, the MiiNePort module will use a multiple connection application

(i.e., two to four hosts are allowed access to the port at the same time). When using a multi-connection application, the module will use the serial communication parameters as defined here in the web console, and all hosts connected to the port must use identical serial settings. If one of the hosts opens the COM port with different serial settings, data will not be transmitted properly.

Ignore jammed IP

Setting	Factory Default	Necessity
Enable, Disable	Disable	Required when Max connection is greater than 1

This option determines how the port will proceed if more of the hosts stops responding as the port is transmitting data. If you select **Disable**, the port will wait until the data has been transmitted successfully to all hosts before transmitting the next group of data. If you select **Enable**, the port will ignore the host that stopped responding and continue data transmission to the other hosts.

NOTE Ignore Jammed IP is only active when Max connection is greater than 1.

Allow driver control

Setting	Factory Default	Necessity
Enable, Disable	Enable	Required when Max connection is greater than 1

This option determines how the port will proceed if multiple hosts that are connected to the port. If **Disable** is selected, driver control commands will be ignored. If **Enable** is selected, control commands will be accepted, with the most recent command received taking precedence.

NOTE Allow driver control is only active when Max connection is greater than one.

Packet length

Setting	Factory Default	Necessity
0 to 1024 bytes	0 byte	Required

The **Packet length** setting refers to the maximum amount of data that is allowed to accumulate in the serial port buffer before sending. When packet length is set to 0 (the default), a maximum amount is not specified and data in the buffer will be sent as specified by the delimiter settings or when the buffer is full. When a packet length between 1 and 1024 bytes is specified, data in the buffer will be sent as soon as it reaches the specified length.

Delimiter 1

Setting	Factory Default	Necessity
00 to FF	“0” for None	Optional

Delimiter 2

Setting	Factory Default	Necessity
00 to FF	“0” for None	Optional

The delimiter fields are used to specify a one-character or two-character sequence that acts as a marker to control packing of serial data. By default, delimiter characters are not defined, so the module transmits data as soon as it is received. When a delimiter character or characters are defined, the module will hold data in its buffer until it receives the delimiter character or two-character sequence. When the delimiter is received, the module will pack the data into its buffer and send it out through the Ethernet port.

Use Delimiter 1 to define the first delimiter character in hex. If only one delimiter character is used, Delimiter 2 should be set to “0”. If the delimiter is a two-character sequence, use Delimiter 2 to define the second character. To disable the use of delimiters, set both Delimiter 1 and Delimiter 2 to “0”.

Note that data packing is controlled not only by the delimiter but is also influenced by the module’s buffer size and the Force transmit field. If the delimiter has not been received by the time the 1K buffer is full, the module will pack the data for network transmission and clear the buffer. In addition, the module will also pack data for network transmission if the next byte of data is not received within the Force transmit time.

ATTENTION

Delimiter 2 is optional. If left blank, then Delimiter 1 alone trips clearing of the buffer. If the size of the serial

data received is greater than 1 KB, the MiiNePort will automatically pack the data and send it to the Ethernet. However, to use the delimiter function, you must at least enable Delimiter 1. If Delimiter 1 is left blank and Delimiter 2 is enabled, the delimiter function will not work properly.

Delimiter process

Setting	Factory Default	Necessity
Do Nothing, Delimiter+1, Delimiter+2, Strip Delimiter	Do Nothing (only effective when Delimiter is enabled)	Optional

The **Delimiter process** field determines how the data is handled when a delimiter is received. Delimiter 1 must be enabled for this field to have effect. If Delimiters 1 and 2 are both enabled, both characters must be received for the delimiter process to take place.

[Do Nothing]: Data in the buffer will be transmitted when the delimiter is received.

[Delimiter + 1] or [Delimiter + 2]: The data will be transmitted when an additional byte (for Delimiter +1), or an additional two bytes (for Delimiter +2) of data is received after receiving the delimiter.

[Strip Delimiter]: When the delimiter is received, the delimiter is deleted (i.e., stripped), and the remaining data is transmitted.

Force transmit

Setting	Factory Default	Necessity
0 to 65535 ms	0 ms	Optional

0: The force transmit timeout is disabled.

1 to 65535: If the module does not receive the next byte of data within the time specified, it will pack the data in its buffer into the same data frame for network transmission.

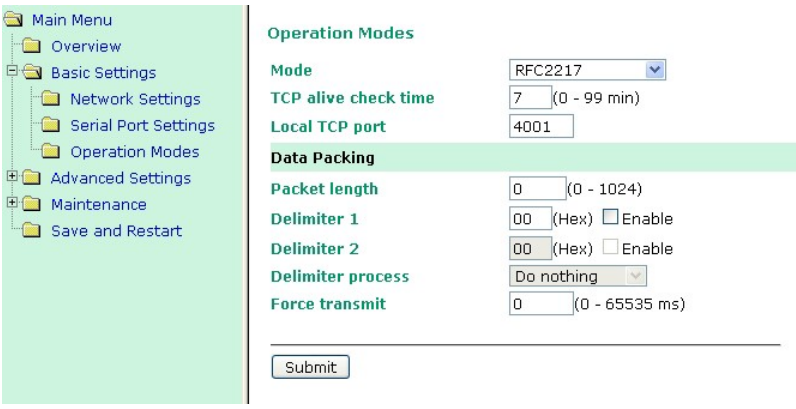
The **Force transmit** field is typically used in conjunction with the delimiter fields to specify how data in the module's buffer is packed for network transmission. When delimiters are used, the module accumulates data in its buffer as it waits to receive a delimiter. If there is a break in communication, data will be held in the buffer as the module continues to wait for a delimiter. The Force transmit field allows you to specify the maximum amount of time that the module will wait for data. With Force transmit enabled, the module will automatically pack the data in the buffer for network transmission if no data is received for the specified time.

When set to 0, Force transmit is disabled, which means there is no time limit for how long the module will wait to receive data. When set between 1 and 65535, the module will pack data if serial data is not received in the specified time.

The optimal Force transmit time depends on your application, but it should be larger than one character interval within the specified baudrate to have any effect. 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 needed to send one character is 10 bits, and the time required to transfer one character is **(10 bits / 1200 bits/s) × 1000 ms/s = 8.3 ms.**

Since it requires about 9 ms to send one character, the Force transmit should be 10 ms or more to have any effect. At 9 ms or less, the module will simply pack every character as it is received, which would be the same as if delimiter characters or a Force transmit time were not specified.

6.8.3. RFC 2217 MODE



TCP alive check time

Setting	Factory Default	Necessity
0 to 99 min	7 min	Optional

1. **min:** TCP connection is not closed due to an idle TCP connection.
2. **to 99 min:** The MiiNePort automatically closes the TCP connection if there is no TCP activity for the given time. After the connection is closed, the MiiNePort starts listening for another host's TCP connection.

Local TCP port

Setting	Factory Default	Necessity
1 to 65535	4001	Required

The **Local TCP port** is the TCP port that the MiiNePort uses to listen to connections, and that other devices must use to contact the MiiNePort. To avoid conflicts with well-known TCP ports, the default is set to 4001.

Packet length

Setting	Factory Default	Necessity
0 to 1024	0	Optional

The Packet length setting refers to the maximum amount of data that is allowed to accumulate in the serial port buffer before sending. When packet length is set to 0 (the default), a maximum amount is not specified and data in the buffer will be sent as specified by the delimiter settings or when the buffer is full. When a packet length between 1 and 1024 bytes is specified, data in the buffer will be sent as soon as it reaches the specified length.

Delimiter 1

Setting	Factory Default	Necessity
00 to FF	None	Optional

Delimiter 2

Setting	Factory Default	Necessity
00 to FF	None	Optional

The delimiter fields are used to specify a one-character or two-character sequence that acts as a marker to control packing of serial data. By default, delimiter characters are not defined, so the module transmits data as soon as it is received. When a delimiter character or characters are defined, the module will hold data in its buffer until it receives the delimiter character or two-character sequence. When the delimiter is received, the module will pack the data into its buffer and send it out through the Ethernet port.

Use Delimiter 1 to define the first delimiter character in hex. If only one delimiter character is used, Delimiter 2 should be set to “0”. If the delimiter is a two-character sequence, use Delimiter 2 to define the second character. To disable the use of delimiters, set both Delimiter 1 and Delimiter 2 to “0”.

Note that data packing is controlled not only by the delimiter but is also influenced by the module’s buffer size and the Force transmit field. If the delimiter has not been received by the time the 1K buffer is full, the module will pack the data for network transmission and clear the buffer. In addition, the module will also pack data for network transmission if the next byte of data is not received within the Force transmit time. ***Delimiter process***

Setting	Factory Default	Necessity

Do Nothing, Delimiter +1, Delimiter +2, Strip Delimiter	Do Nothing	Optional

Thedelimiter process field determines how the data is handled when a delimiter is received. Delimiter 1 must be enabled for this field to have effect. If Delimiters 1 and 2 are both enabled, both characters must be received for the delimiter process to take place.

[Do Nothing]: Data in the buffer will be transmitted when the delimiter is received.

[Delimiter + 1] or [Delimiter + 2]: The data will be transmitted when an additional byte (for Delimiter +1), or an additional two bytes (for Delimiter +2) of data is received after receiving the delimiter.

[Strip Delimiter]: When the delimiter is received, the delimiter is deleted (i.e., stripped), and the remaining data is transmitted. ***Force transmit***

Setting	Factory Default	Necessity
0 to 65535 ms	0 ms	Optional

0: The force transmit timeout is disabled.

1 to 65535: If the module does not receive the next byte of data within the time specified, it will pack the data in its buffer into the same data frame for network transmission.

The **Force transmit** field is typically used in conjunction with the delimiter fields to specify how data in the module's buffer is packed for network transmission. When delimiters are used, the module accumulates data in its buffer as it waits to receive a delimiter. If there is a break in communication, data will be held in the buffer as the module continues to wait for a delimiter. The Force transmit field allows you to specify the maximum amount of time that the module will wait for data. With Force transmit enabled, the module will automatically pack the data in the buffer for network transmission if no data is received for the specified time.

When set to 0, Force transmit is disabled, which means there is no time limit for how long the module will wait to receive data. When set between 1 and 65535, the module will pack data if serial data is not received in the specified time.

The optimal force transmit time depends on your application, but it should be larger than one character interval within the specified baudrate to have any effect. 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 needed to send one character is 10 bits, and the time required to transfer one character is **(10 bits / 1200 bits/s) × 1000 ms/s = 8.3 ms.**

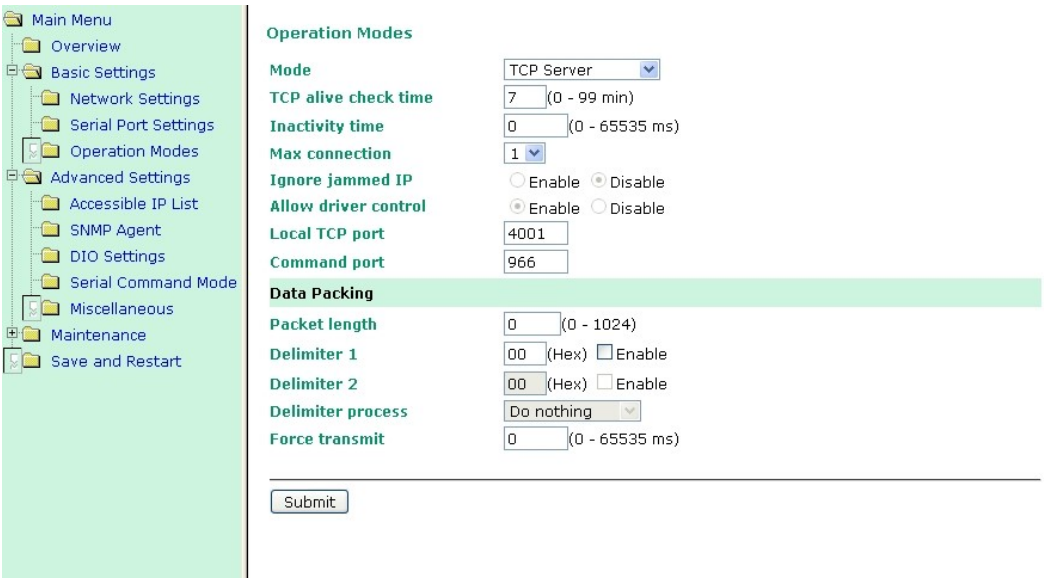
Since it requires about 9 ms to send one character, the Force transmit should be 10 ms or more to have any effect. At 9 ms or less, the module will simply pack every character as it is received, which would be the same as if a delimiter character or Force transmit time were not specified.

ATTENTION

If you want to send a series of characters in the same packet, the serial device attached to the MiiNePort

should send that series of characters during a time interval less than the Force transmit timeout for the MiiNePort, and the total length of data must be less than or equal to the MiiNePort's internal buffer size. The serial communication buffer size for the MiiNePort is 1 KB per port.

6.8.4. TCP SERVER MODE



TCP alive check time

Setting	Factory Default	Necessity
0 to 99 min	7 min	Optional

1. **min:** TCP connection is not closed due to an idle TCP connection.

2. **to 99 min:** The MiiNePort automatically closes the TCP connection if there is no TCP activity for the given time. After the connection is closed, the MiiNePort starts listening for another host's TCP connection.

Inactivity time

Setting	Factory Default	Necessity
0 to 65535 ms	0 ms	Optional

1. **ms:** TCP connection is not closed due to an idle serial line.

0-65535 ms: The MiiNePort automatically closes the TCP connection if there is no serial data activity for the given time. After the connection is closed, the MiiNePort starts listening for another host's TCP connection.

This parameter defines the maintenance status as Closed or Listen for the TCP connection. The connection is closed if there is no incoming or outgoing data through the serial port during the specific Inactivity time.

If the Inactivity time is set to 0, the current TCP connection is kept active until a connection close request is received. Although Inactivity time is disabled, the MiiNePort will check the connection status between the MiiNePort and the remote host by sending “keep alive” packets periodically. If the remote host does not respond to the packet, the MiiNePort assumes that the connection was closed unintentionally. The MiiNePort will then force the existing TCP connection to close.

ATTENTION

The Inactivity time should be greater than the Force transmit timeout. To prevent the unintended loss of data

due to the session being disconnected, it is highly recommended that this value is set large enough so that the intended data transfer is completed.

ATTENTION

Inactivity time is ONLY active when **TCP connect on** is set to **Any character**.

Max connection

Setting	Factory Default	Necessity
1, 2, 3, 4	1	Required

Max connection is used when the device needs to receive data from different hosts simultaneously.

The factory default only allows 1 connection at a time. When Max Connection is set to 1, the Real COM driver on the specific host has full control.

Max connection 1: The module will only allow one host’s Real COM driver to open a connection to the module’s serial port.

Max connections 2 to 4: When set to two or higher, Real COM drivers for up to the specified number of hosts may open this port at the same time. When Real COM drivers for multiple hosts open the port at the same time, the COM driver only provides a pure data tunnel with no control ability. The serial port parameters will use firmware settings instead of your application program (AP) settings.

Application software that is based on the COM driver will receive a driver response of “success” when the software uses any of the Win32 API functions. The firmware will only send data back to the driver on the host. Data will be sent first-in,first-out when data is received by the MiiNePort from the Ethernet interface. ***Ignore jammed IP***

Setting	Factory Default	Necessity
Enable, Disable	Disable	Optional

This option determines how the port will proceed if multiple hosts are connected, and one or more of the hosts stops responding as the port is transmitting data. If you select **Disable**, the port will wait until the data has been transmitted successfully to all hosts before transmitting the next group of data. If you select **Enable**, the port will ignore the host that stopped responding and continue data transmission to the other hosts.

NOTE Ignore Jammed IP is only active when Max connection is greater than 1.

Allow driver control

Setting	Factory Default	Necessity
Enable, Disable	Disable	Optional

This option determines how the port will proceed if driver control commands are received from multiple hosts that are connected to the port. If **Disable** is selected, driver control commands will be ignored. If **Enable** is selected, control commands will be accepted, with the most recent command received taking precedence.

NOTE Allow driver control is only active when Max connection is greater than one.

Local TCP port

Setting	Factory Default	Necessity
1 to 65535	4001	Required

The **Local TCP port** is the TCP port that the MiiNePort uses to listen to connections and that other devices must use to contact the MiiNePort. To avoid conflicts with well-known TCP ports, the default is set to 4001.

Command port

Setting	Factory Default	Necessity
1 to 65535	966	Optional

The **Command port** is a "listen TCP port" for IP-Serial Lib commands from the host. In order to prevent a TCP port conflict with other applications, the user can set the Command port to another port if needed. IP-Serial Lib will automatically check the Command Port on the MiiNePort so that the user does not need to configure the program.

Packet length

Setting	Factory Default	Necessity
0 to 1024	0	Optional

The **Packet length** setting refers to the maximum amount of data that is al accumulate in the serial port buffer before sending. When packet length is set to maximum amount is not specified and data in the buffer specified by the delimiter settings or when the buffer is f packet length between 1 and 1024 bytes is specified, data in t sent as soon as it reaches the specified length.

Delimiter 1

Setting	Factory Default	Necessity
00 to FF	None	Optional

Delimiter 2

Setting	Factory Default	Necessity
00 to FF	None	Optional

The delimiter fields are used to specify a one-character or two-character sequence that acts as a marker to control packing of serial data. By default, delimiter characters are not defined, so the module transmits data as soon as it is received. When a delimiter character or characters are defined, the module will hold data in its buffer until it receives the delimiter character or two-character sequence. When the delimiter is received, the module will pack the data into its buffer and send it out through the Ethernet port.

Use Delimiter 1 to define the first delimiter character in hex. If only one delimiter character is used, Delimiter 2 should be set to "0". If the delimiter is a two-character sequence, use Delimiter 2 to define the second character. To disable the use of delimiters, set both Delimiter 1 and Delimiter 2 to "0".

Note that data packing is controlled not only by the delimiter but is also influenced by the module's buffer size and the Force transmit field. If the delimiter has not been received by the time the 1K buffer is full, the module will pack the data for network transmission and clear the buffer. In addition, the module will also pack data for network transmission if the next byte of data is not received within the Force transmit time.

ATTENTION

Delimiter 2 is optional. If left blank, then Delimiter 1 alone trips clearing of the buffer. If the size of the serial

data received is greater than 1 KB, the MiiNePort will automatically pack the data and send it to the Ethernet. However, to use the delimiter function, you must at least enable Delimiter 1. If Delimiter 1 is left blank and Delimiter 2 is enabled, the delimiter function will not work properly.

Delimiter process

Setting	Factory Default	Necessity
Do Nothing, Delimiter + 1, Delimiter + 2, Strip Delimiter	Do Nothing	Optional

The **Delimiter process** field determines how the data is handled when a delimiter is received. Delimiter 1 must be enabled for this field to have effect. If Delimiters 1 and 2 are both enabled, both characters must be received for the delimiter process to take place.

[Do Nothing]: Data in the buffer will be transmitted when the delimiter is received.

[Delimiter + 1] or [Delimiter + 2]: The data will be transmitted when an additional byte (for Delimiter +1), or an additional 2 bytes (for Delimiter +2) of data is received after receiving the delimiter.

[Strip Delimiter]: When the delimiter is received, the delimiter is deleted (i.e., stripped), and the remaining data is transmitted. ***Force transmit***

Setting	Factory Default	Necessity

0 to 65535 ms	0 ms	Optional

0: The force transmit timeout is disabled.

1 to 65535: If the module does not receive the next byte of data within the time specified, it will pack the data in its buffer into the same data frame for network transmission.

The **Force transmit** field is typically used in conjunction with the delimiter fields to specify how data in the module's buffer is packed for network transmission. When delimiters are used, the module accumulates data in its buffer as it waits to receive a delimiter. If there is a break in communication, data will be held in the buffer as the module continues to wait for a delimiter. The Force transmit field allows you to specify the maximum amount of time that the module will wait for data. With Force transmit enabled, the module will automatically pack the data in the buffer for network transmission if no data is received for the specified time.

When set to 0, Force transmit is disabled, which means there is no time limit for how long the module will wait to receive data. When set between 1 and 65535, the module will pack data if serial data is not received in the specified time.

The optimal force transmit time depends on your application, but it should be larger than one character interval within the specified baudrate to have any effect. 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 needed to send one character is 10 bits, and the time required to transfer one character is **(10 bits / 1200 bits/s) × 1000 ms/s = 8.3 ms.**

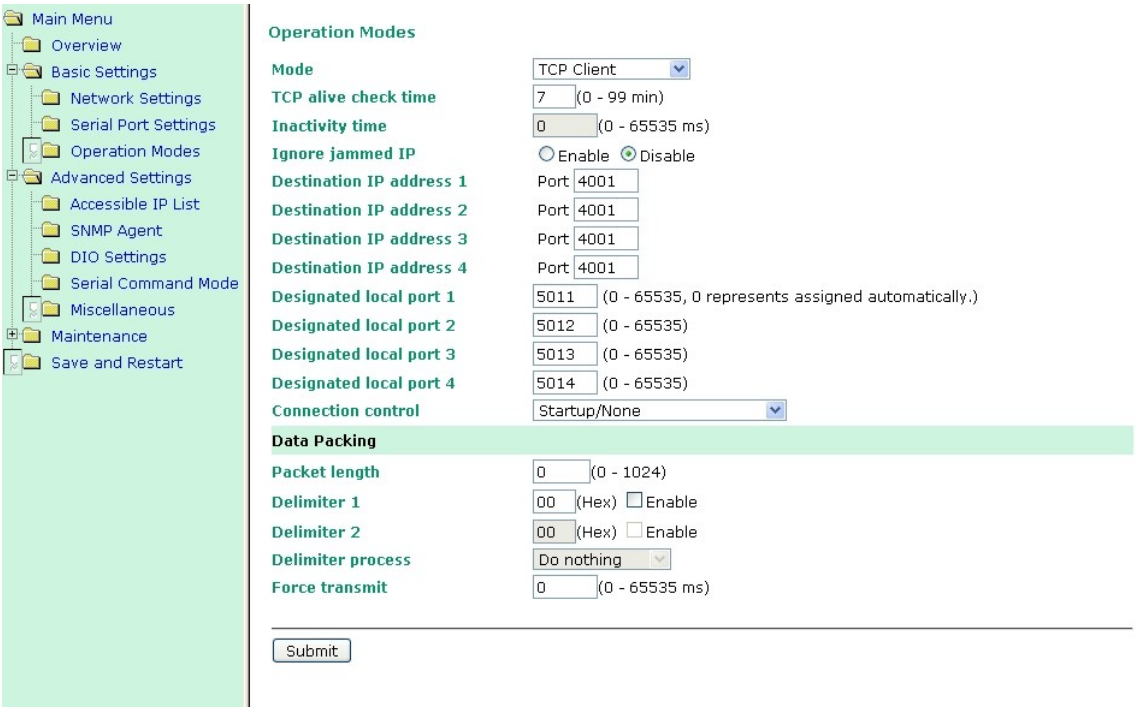
Since it requires about 9 ms to send one character, the Force transmit should be 10 ms or more to have any effect. At 9 ms or less, the module will simply pack every character as it is received, which would be the same as if delimiter characters or a Force transmit time were not specified.

ATTENTION

If you want to send a series of characters in the same packet, the serial device attached to the MiiNePort

should send that series of characters during a time interval less than the Force transmit timeout for the MiiNePort, and the total length of data must be less than or equal to the MiiNePort's internal buffer size. The serial communication buffer size for the MiiNePort is 1 KB per port.

6.8.5. TCP CLIENT MODE



TCP alive check time

Setting	Factory Default	Necessity
0 to 99 min	7 min	Optional

1. **min:** The TCP connection is not closed due to an idle TCP connection.
2. **to 99 min:** The module automatically closes the TCP connection if there is no TCP activity for the given time. After the connection is closed, the module starts listening for another host's TCP connection.

Inactivity time

Setting	Factory Default	Necessity
0 to 65535 ms	0 ms	Optional

1. **ms:** TCP connection is not closed due to an idle serial line.

0-65535 ms: The MiiNePort automatically closes the TCP connection if there is no serial data activity for the given time. After the connection is closed, the MiiNePort starts listening for another host's TCP connection.

This parameter defines the maintenance status as Closed or Listen for the TCP connection. The connection is closed if there is no incoming or outgoing data through the serial port during the specific Inactivity time.

If the Inactivity time is set to 0, the current TCP connection is kept active until a connection close request is received. Although Inactivity time is disabled, the MiiNePort will check the connection status between the MiiNePort and the remote host by sending "keep alive" packets periodically. If the remote host does not respond to the packet, the MiiNePort assumes that the connection was closed unintentionally. The MiiNePort will then force the existing TCP connection to close.

ATTENTION

The Inactivity time should be greater than the Force transmit timeout. To prevent the unintended loss of data due to the session being disconnected, it is highly recommended that this value is set large enough so that the intended data transfer is completed.

ATTENTION

Inactivity time is ONLY active when **TCP connect on** is set to **Any character**.

Ignore jammed IP

Setting	Factory Default	Necessity
Enable, Disable	Disable	Optional

This option determines how the port will proceed if multiple hosts are connect more of the hosts

stops responding as the port is transmitting data. If you select **Disable**, the port will wait until the data has been transmitted successfully to all hosts before transmitting the next group of data. If you select **Enable**, the port will ignore the host that stopped responding and continue data transmission to the other hosts.

NOTE Ignore Jammed IP is only active when Max connection is greater than one.

Packet length

Setting	Factory Default	Necessity
0 to 1024	0	Optional

The **Packet length** setting refers to the maximum amount of data that is allowe

in the serial port buffer before sending. When packet length is set to 0 (the def

a

and data in the buffer will be sent as specified by the delimiter settings or whe mount is not full. When a packet specified

length between 1 and 1024 bytes is specified, data in t

length. sent as soon as it reaches the specified

Delimiter 1

Setting	Factory Default	Necessity
00 to FF	“00” for None	Optional

Delimiter 2

Setting	Factory Default	Necessity
00 to FF	“00” for None	Optional

The delimiter fields are used to specify a one-character or two-character sequence that acts as a marker to control packing of serial data. By default, delimiter characters are not defined, so the module transmits data

as soon as it is received. When a delimiter character or characters are defined, the module will hold data in its buffer until it receives the delimiter character or two-character sequence. When the delimiter is received, the module will pack the data into its buffer and send it out through the Ethernet port.

Use Delimiter 1 to define the first delimiter character in hex. If only one delimiter character is used, Delimiter 2 should be set to “0”. If the delimiter is a two-character sequence, use Delimiter 2 to define the second character. To disable the use of delimiters, set both Delimiter 1 and Delimiter 2 to “0”.

Note that data packing is controlled not only by the delimiter but is also influenced by the module’s buffer size and the Force transmit field. If the delimiter has not been received by the time the 1K buffer is full, the module will pack the data for network transmission and clear the buffer. In addition, the module will also pack data for network transmission if the next byte of data is not received within the Force transmit time.

ATTENTION

Delimiter 2 is optional. If left blank, then Delimiter 1 alone trips clearing of the buffer. If the size of the serial

data received is greater than 1 KB, the MiiNePort will automatically pack the data and send it to the Ethernet. However, to use the delimiter function, you must at least enable Delimiter 1. If Delimiter 1 is left blank and Delimiter 2 is enabled, the delimiter function will not work properly.

Delimiter process

Setting	Factory Default	Necessity
Do Nothing, Delimiter + 1, Delimiter + 2, Strip Delimiter	Do Nothing	Optional

The **Delimiter process** field determines how the data is handled when a delimiter is received. Delimiter 1 must be enabled for this field to have effect. If Delimiters 1 and 2 are both enabled, both characters must be received for the delimiter process to take place.

[Do Nothing]: Data in the buffer will be transmitted when the delimiter is received.

[Delimiter + 1] or [Delimiter + 2]: The data will be transmitted when an additional byte (for Delimiter +1), or an additional 2 bytes (for Delimiter +2) of data is received after receiving the delimiter.

[Strip Delimiter]: When the delimiter is received, the delimiter is deleted (i.e., stripped), and the remaining data is transmitted. ***Force transmit***

Setting	Factory Default	Necessity
0 to 65535 ms	0 ms	Optional

0: The Force transmit timeout is disabled.

1 to 65535: If the module does not receive the next byte of data within the time specified, it will pack the data in its buffer into the same data frame for network transmission.

The **Force transmit** field is typically used in conjunction with the delimiter fields to specify how data in the module's buffer is packed for network transmission. When delimiters are used, the module accumulates data in its buffer as it waits to receive a delimiter. If there is a break in communication, data will be held in the buffer as the module continues to wait for a delimiter. The Force transmit field allows you to specify the maximum amount of time that the module will wait for data. With Force transmit enabled, the module will automatically pack the data in the buffer for network transmission if no data is received for the specified time.

When set to 0, Force transmit is disabled, which means there is no time limit for how long the module will wait to receive data. When set between 1 and 65535, the module will pack data if serial data is not received in the specified time.

The optimal force transmit time depends on your application, but it should be larger than one character interval within the specified baudrate to have any effect. 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 needed to send one character is 10 bits, and the time required to transfer one character is **(10 bits / 1200 bits/s) × 1000 ms/s = 8.3 ms.**

Since it requires about 9 ms to send one character, the Force transmit should be 10 ms or more to have any effect. At 9 ms or less, the module will simply pack every character as it is received, which would be the same as if delimiter characters or a Force transmit time were not specified.

Destination IP address 1/2/3/4

Setting	Factory Default	Necessity
IP address or Domain Name (E.g., 192.168.1.1)	None	Required

Up to four **Destination IP addresses** (or domain names) can be specified. The MiiNePort will be able to actively connect to each of these remote addresses.

ATTENTION

The connection speed or throughput may be slow if one of the four connections is slow, since the one slow connection will slow down the other three connections.

ATTENTION

Both IP address and Domain Name can be used in the **Destination IP address** field.

Designated Local Port 1/2/3/4

Setting	Factory Default	Necessity

TCP Port No.	5011 (Port 1) 5012 (Port 2) 1. (Port 3) 2. (Port 4)	Required

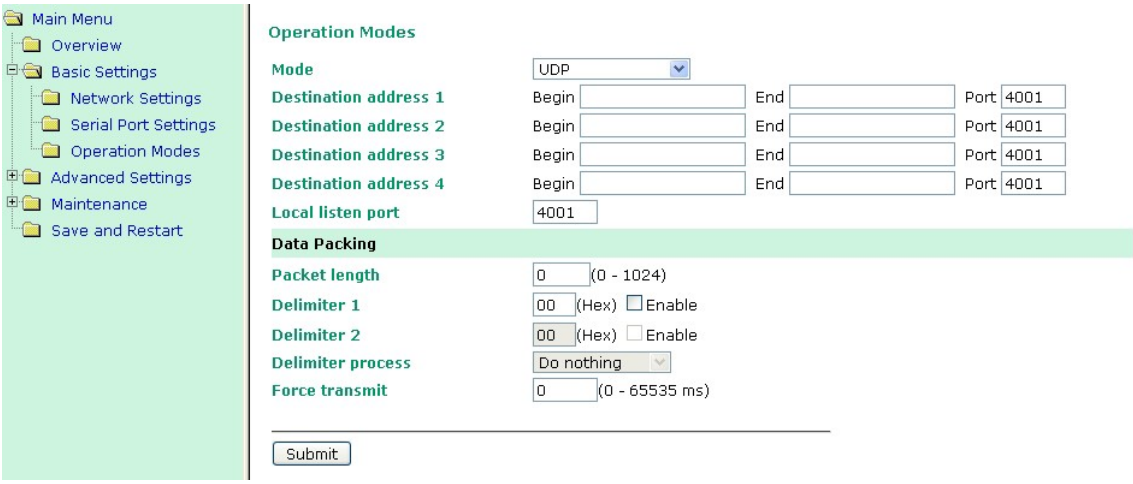
Connection control

Setting	Factory Default	Necessity
Startup/None, Any Character/None, Any Character/Inactivity Time, DSR ON/DSR OFF, DSR ON/None, DCD ON/DCD OFF, DCD ON/None	Startup/None	Required

The meaning of each of the above settings is given in the following table. In general, both the Connect condition and Disconnect condition are given.

Connect/Disconnect	Description
Startup / None (default)	A TCP connection will be established on startup and will remain active indefinitely.
Any Character / None	A TCP connection will be established when any character is received from the serial interface and will remain active indefinitely.
Any Character / Inactivity Time	A TCP connection will be established when any character is received from the serial interface and will be disconnected when the Inactivity time out is reached.
DSR On / DSR Off	A TCP connection will be established when a DSR “On” signal is received and will be disconnected when a DSR “Off” signal is received.
DSR On / None	A TCP connection will be established when a DSR “On” signal is received and will remain active indefinitely.
DCD On / DCD Off	A TCP connection will be established when a DCD “On” signal is received and will be disconnected when a DCD “Off” signal is received.
DCD On / None	A TCP connection will be established when a DCD “On” signal is received and will remain active indefinitely.

6.8.6. UDP MODE



Destination IP address 1

Setting	Factory Default	Necessity
IP address range	Begin: Empty	Required
E.g., Begin: 192.168.1.1	End: Empty	
End: 192.168.1.10	Port: 4001	

Destination IP address 2/3/4

Setting	Factory Default	Necessity
IP address range	Begin: Empty	Optional
E.g., Begin: 192.168.1.11	End: Empty	
End: 192.168.1.20	Port: 4001	

Local listen port

Setting	Factory Default	Necessity
1 to 65535	4001	Required

The **Local listen port** is the UDP port that the MiiNePort listens to, and that other devices must use to contact the MiiNePort. To avoid conflicts with well-known UDP ports, the default is set to 4001. ***Packet length***

Setting	Factory Default	Necessity
0 to 1024	0	Optional

The **Packet length** setting refers to the maximum amount of data that is allowed to accumulate in the serial port buffer before sending. When packet length is set to 0 (the default), a maximum amount is not specified and data in the buffer will be sent as specified by the delimiter settings or when the buffer is full. When a packet length between 1 and 1024 bytes is specified, data in the buffer will be sent as soon as it reaches the specified length.

Delimiter 1

Setting	Factory Default	Necessity
00 to FF	“00” for None	Optional

Delimiter 2

Setting	Factory Default	Necessity
00 to FF	"00" for None	Optional

The delimiter fields are used to specify a one-character or two-character sequence that acts as a marker to control packing of serial data. By default, delimiter characters are not defined, so the module transmits data as soon as it is received. When a delimiter character or characters are defined, the module will hold data in its buffer until it receives the delimiter character or two-character sequence. When the delimiter is received, the module will pack the data into its buffer and send it out through the Ethernet port.

Use Delimiter 1 to define the first delimiter character in hex. If only one delimiter character is used, Delimiter 2 should be set to "0". If the delimiter is a two-character sequence, use Delimiter 2 to define the second character. To disable the use of delimiters, set both Delimiter 1 and Delimiter 2 to "0".

Note that data packing is controlled not only by the delimiter but is also influenced by the module's buffer size and the Force transmit field. If the delimiter has not been received by the time the 1K buffer is full, the module will pack the data for network transmission and clear the buffer. In addition, the module will also pack data for network transmission if the next byte of data is not received within the Force transmit time.

ATTENTION

Delimiter 2 is optional. If left blank, then Delimiter 1 alone trips clearing of the buffer. If the size of the serial

data received is greater than 1 KB, the MiiNePort will automatically pack the data and send it to the Ethernet. However, to use the delimiter function, you must at least enable Delimiter 1. If Delimiter 1 is left blank and Delimiter 2 is enabled, the delimiter function will not work properly.

Delimiter process

Setting	Factory Default	Necessity
Do Nothing, Delimiter + 1, Delimiter + 2, Strip Delimiter	Do Nothing	Optional

The **Delimiter process** field determines how the data is handled when a delimiter is received. Delimiter 1 must be enabled for this field to have effect. If Delimiters 1 and 2 are both enabled, both characters must be received for the delimiter process to take place.

[Do Nothing]: Data in the buffer will be transmitted when the delimiter is received.

[Delimiter + 1] or [Delimiter + 2]: The data will be transmitted when an additional byte (for Delimiter +1), or an additional two bytes (for Delimiter +2) of data is received after receiving the delimiter.

[Strip Delimiter]: When the delimiter is received, the delimiter is deleted (i.e., stripped), and the remaining data is transmitted. ***Force transmit***

Setting	Factory Default	Necessity

0 to 65535 ms	0 ms	Optional

0: The force transmit timeout is disabled.

1 to 65535: If the module does not receive the next byte of data within the time specified, it will pack the data in its buffer into the same data frame for network transmission.

The **Force transmit** field is typically used in conjunction with the Delimiter fields to specify how data in the module's buffer is packed for network transmission. When delimiters are used, the module accumulates data in its buffer as it waits to receive a delimiter. If there is a break in communication, data will be held in the buffer as the module continues to wait for a delimiter. The Force transmit field allows you to specify the maximum amount of time that the module will wait for data. With Force transmit enabled, the module will automatically pack the data in the buffer for network transmission if no data is received for the specified time.

When set to 0, Force transmit is disabled, which means there is no time limit for how long the module will wait to receive data. When set between 1 and 65535, the module will pack data if serial data is not received in the specified time.

The optimal force transmit time depends on your application, but it should be larger than one character interval within the specified baudrate to have any effect. 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 needed to send one character is 10 bits, and the time required to transfer one character is **(10 bits / 1200 bits/s) × 1000 ms/s = 8.3 ms.**

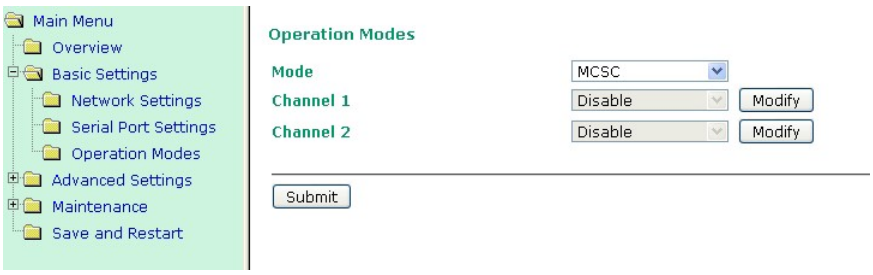
Since it requires about 9 ms to send one character, the Force transmit should be 10 ms or more to have any effect. At 9 ms or less, the module will simply pack every character as it is received, which would be the same as if delimiter characters or a Force transmit time were not specified.

ATTENTION

If you want to send a series of characters in the same packet, the serial device attached to the MiiNePort

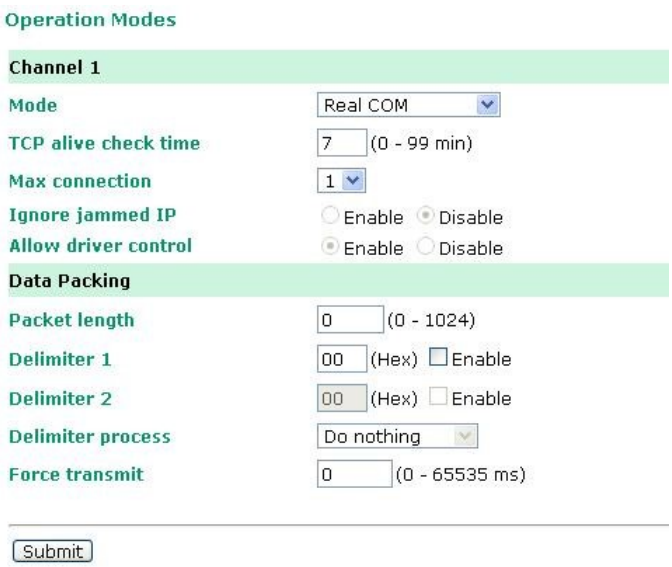
should send that series of characters during a time interval less than the Force transmit timeout for the MiiNePort, and the total length of data must be less than or equal to the MiiNePort's internal buffer size. The serial communication buffer size for the MiiNePort is 1 KB per port.

6.8.7. MCSC MODE



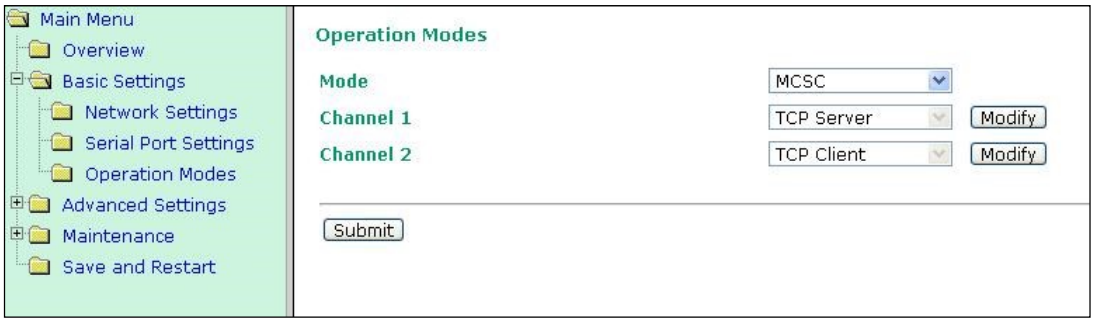
After selecting **MCSC mode**, you will be able to configure Channel 1 and Channel 2. MCSC mode is designed so that the two channels work independently. Click the **Modify** button of a channel to update the channel's configuration.

We use Channel 1 to illustrate.



All of the configuration items shown in the above screenshot have been discussed earlier in this chapter. Use the **Mode** drop-down list to select the operation mode for the channel, and then update the configuration items. When you click **Submit**, the following confirmation window will appear.

Click the **Close** button to return to the initial **Operation Modes** configuration screen (shown below). At this point you can click the Modify button for Channel 2 to configure the settings for that channel. The configuration settings for Channel 1 will be shown, but you are free to update each configuration item. When you click Submit, the configuration will be saved to the MiiNePort's memory and will not be erased until the next time you change the configuration.



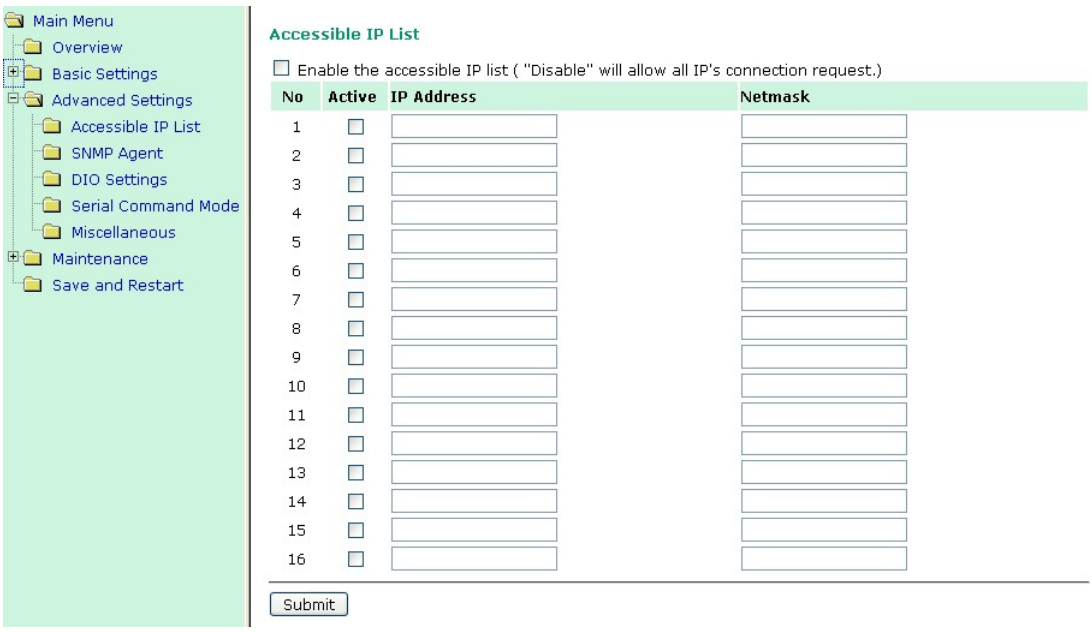
Click **Modify** to configure Channel 2. For example, if you want to set Channel 2 to Real COM mode, you need to finish your Real COM settings and then click **Submit**.

When the confirmation screen appears, choose either **Save/Restart** to activate the changes you've made or **Close** to do other configurations.

For details on MCSC's functionality and configuration, refer to **Chapter 8: NetEZ Technologies' MCSC**. MCSC is only supported on the MiiNePort E2.

6.9. ADVANCED SETTINGS

6.9.1. ACCESSIBLE IP LIST



The Accessible IP List allows you to configure the following types of accessibility:

1. **To allow access to a specific IP address**

   Enter the IP address in the corresponding field; enter 255.255.255.255 for the netmask.

2. **To allow access to hosts on a specific subnet**

For both the IP address and netmask, use 0 for the last digit (e.g., 192.168.1.0 and 255.255.255.0).

3. **To allow unrestricted access**

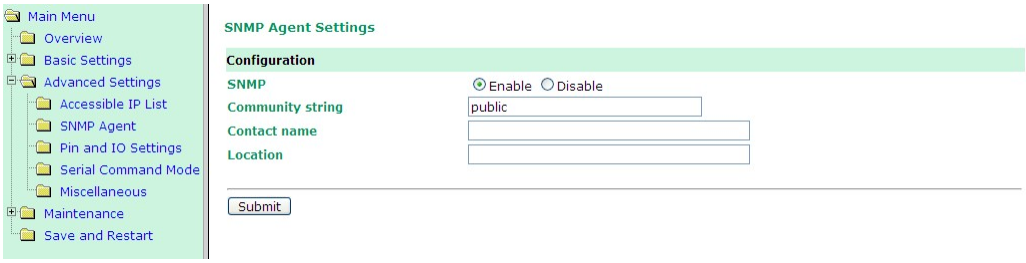
Deselect the **Enable the accessible IP list** option.

Refer to the following table for more configuration examples.

Allowed Hosts	Entered IP address/Netmask
Any host	Disable
192.168.1.120	192.168.1.120 / 255.255.255.255
192.168.1.1 to 192.168.1.254	192.168.1.0 / 255.255.255.0
192.168.0.1 to 192.168.255.254	192.168.0.0 / 255.255.0.0
192.168.1.1 to 192.168.1.126	192.168.1.0 / 255.255.255.128
192.168.1.129 to 192.168.1.254	192.168.1.128 / 255.255.255.128

6.9.2. SNMP AGENT

To enable the SNMP agent function, click the **Enable** option located to the right of **SNMP** under **Configuration**.



6.9.3. COMMUNITY STRING

Setting	Factory Default	Necessity
1 to 39 characters (E.g., support, 886-89191230 #300)	Public	Optional

A community name is a plain-text password mechanism that is used to authenticate queries to agents of managed network devices. ***Contact Name***

Setting	Factory Default	Necessity
1 to 39 characters (E.g., support, 886-89191230 #300)	None	Optional

The SNMP contact information usually includes an emergency contact name and telephone or pager number.

6.9.4. LOCATION

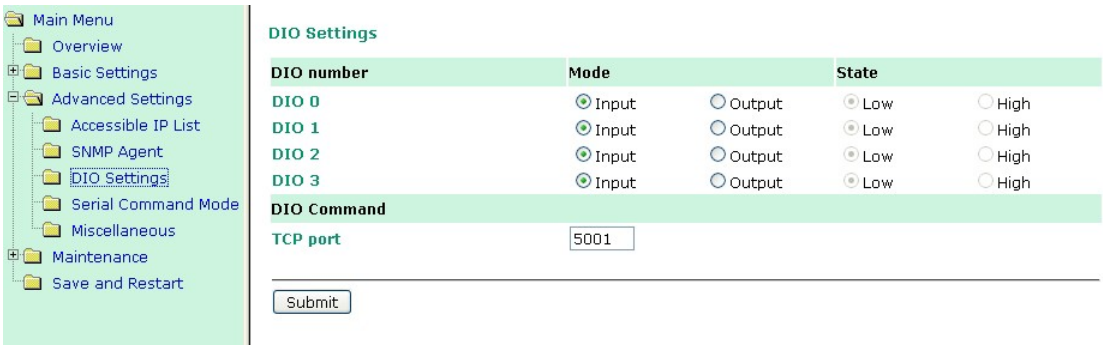
Setting	Factory Default	Necessity

1 to 39 characters (E.g., floor 1, office 2)	None	Optional

Enter a location string for SNMP agents. This string is usually set to the street address where the module is physically located.

6.9.5. DIO SETTINGS

The DIO channel is configured on the DIO Settings page. DIO commands can be used to control and obtain data from the module's DIO channels. When digital input operation is selected, the channel's status is controlled by an external digital switch. When digital output operation is selected, the channel transmits a high or low signal. Refer to **Appendix D: DIO Commands** for additional information.



DIO number Mode

Select **Input** or **Output** under **Mode** to specify digital input or digital output operation.

6.9.6. DIO NUMBER STATE

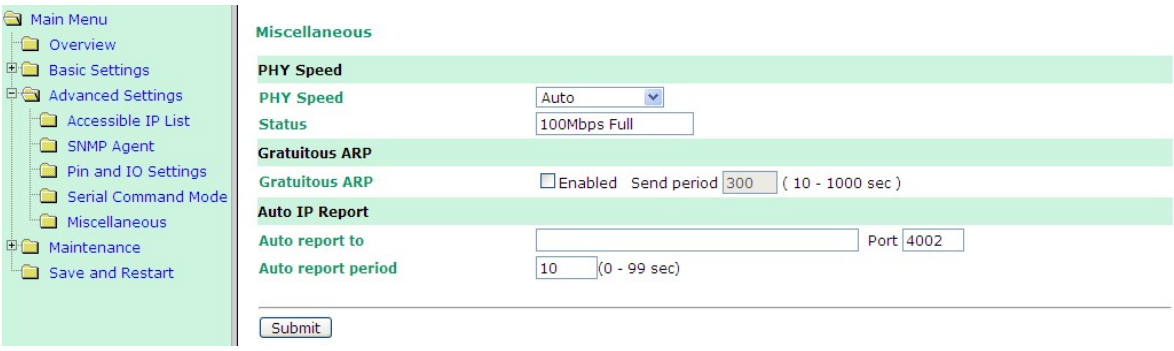
Select **Low** or **High** under **State** to specify a high or low state for digital output channels. Note that when Mode is set to Input (as shown in the above screenshot), the State option is deactivated.

6.9.7. DIO COMMAND TCP PORT

Select an appropriate **TCP port** under **DIO Command** to indicate the port number that will be reserved for DIO commands.

6.9.8. MISCELLANEOUS

PHY Speed, **Gratuitous ARP**, and **Auto IP Report** are configured in the **Miscellaneous** section under **Advanced Settings**.



6.9.9. PHY SPEED

PHY Speed

Setting	Factory Default	Necessity
Auto	Auto	Required

Status

Setting	Factory Default	Necessity
100 Mbps Full Duplex, 100 Mbps Half Duplex, 10 Mbps. Full Duplex, 10 Mbps Half Duplex		Required

You can set PHY speed to Auto, 100 Mbps Full Duplex, 100 Mbps Half Duplex, 10 Mbps Full Duplex, or 10 Mbps Half Duplex.

6.9.10. GRATUITOUS ARP

For some applications, you may want the MiiNePort to send broadcast packets to update the ARP table on the server. If you enable this function, the MiiNePort will send broadcast packets periodically based on the specified time interval.

6.9.11. AUTO IP REPORT

When the MiiNePort is used in a dynamic IP environment, additional time must be spent on IP management tasks. For example, when a module is operating as a server (TCP Server or UDP operation modes), a PC operating as a client will need to know the module's IP address. If the DHCP server assigns a new IP address to the module, the PC must have some way of obtaining the module's new IP address. The IP Address report fields are used to set up periodic reporting of the module's IP address when the module's IP address is assigned by a server. The IP address report is sent automatically at regular intervals to the IP address and TCP port number that you specify. Auto IP Report allows you to view the module's IP address report on a PC:

To develop software that parses the IP address report data, see **Appendix C: Auto IP Report Protocol** for details.

Auto report to

Setting	Factory Default	Necessity
E.g., 192.168.1.1 or URL (IP addresses of the form x.x.x.0 and x.x.x.255 are invalid.)	None	Optional

Auto report to TCP port

Setting	Factory Default	Necessity
E.g., 4001	4002	Optional

If left blank, auto IP reporting is disabled. If an IP address is entered along with a TCP port number, the IP address reports will be sent to the specified address and port number. ***Auto report period***

Setting	Factory Default	Necessity
Time interval (in seconds)	10	Optional

The Auto report period field specifies how often the module will report its IP address. For example, if the auto report period is set to 10 seconds, an IP address report will be sent every 10 seconds.

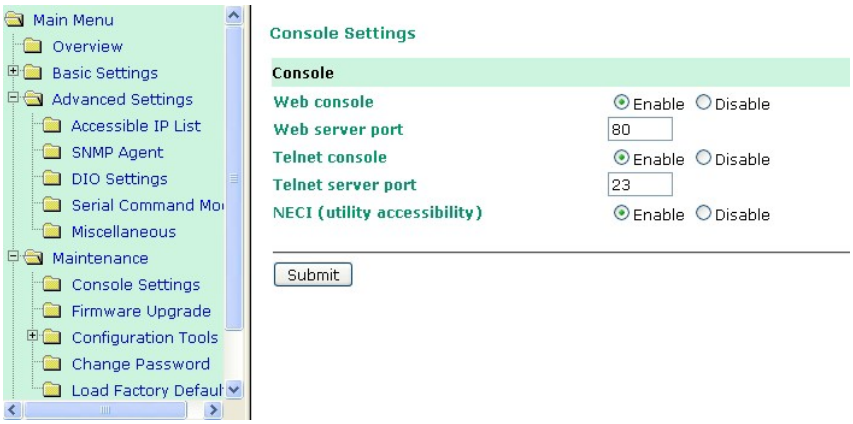
ATTENTION

The module will send IP address reports only when assigned an IP address from a DHCP or BOOTP server. If a connection to a DHCP or BOOTP server is not available, IP address reports will not be sent.

6.10. MAINTENANCE

6.10.1. CONSOLE SETTINGS

You can enable or disable the Web, Telnet, and Utility console functions on the **Console Settings** page. In addition, you can configure the port numbers for the web Telnet servers.



6.10.2. FIRMWARE UPGRADE

The MiiNePort's firmware can be upgraded though the web console, or through the Device Search utility. If you have made any changes to your configuration, remember to save the configuration first before upgrading the firmware. Refer to the **Save Configuration** section in this chapter for more information. Any unsaved changes will be discarded when the firmware is upgraded.

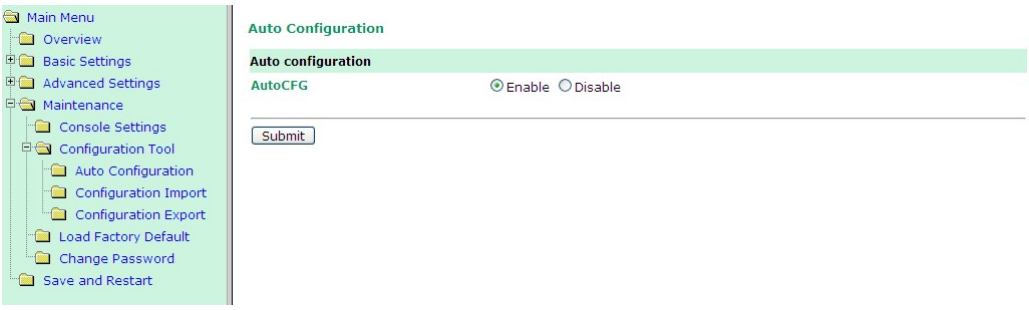
To upgrade the firmware, simply enter the file name and click **Submit**. The latest firmware can be downloaded from Moxa's website at www.moxa.com.

6.10.3. CONFIGURATION TOOLS

The MiiNePort's **Auto Configuration**, **Configuration Import**, and **Configuration Export** functions from the **Maintenance** ⍰ **Configuration Tool** sub-menu.

6.10.4. AUTO CONFIGURATION

You can enable or disable **AutoCFG** from the **Auto Configuration** page. AutoCFG is designed for enabling automatic network configuration during the production of your device. AutoCFG allows you to realize true device mass production without needing to set up the network modules one by one.



For more information about AutoCFG, refer to **Chapter 8: NetEZ Technologies**.

6.10.5. CONFIGURATION IMPORT

The MiiNePort E2/E3 Series can share or back up its configuration by exporting all settings to a file, which can then be imported into another MiiNePort E2/3. The passwords in exported file will be encrypted by a cipher

key assigned by the user, which will be asked again when importing back to the MiiNePort E2/E3 module. The Configuration Import utility allows you to choose whether to import the IP configuration at the same time.

6.10.6. CONFIGURATION EXPORT

Configuration Export allows you to export the current configuration to a text file. You will be asked to input an user defined cipher key for encrypting the passwords in the configuration file prior to clicking the download button. The exported configuration file, named **MiiNePortE2.txt** (or **MiiNePortE2-H.txt, MiiNePortE3.txt, MiiNePortE3-H.txt**) by default, can be imported into another MiiNePort module later with the Configuration Import utility. It can also be used with AutoCFG for mass configuration deployment during your device's mass production stage.

6.10.7. CHANGE PASSWORD

Click on **Change Password** in the navigation panel to display the Change Password window.

Enter the old password and new password (twice) and then click **Submit**. The default password for MiiNePort device is **moxa**.

6.10.8. LOAD FACTORY DEFAULT

To load the factory default settings, click on **Load Factory Default** in the navigation panel and then click **Submit**. All previous modifications will be lost, but you can choose to keep the IP settings by checking the **Keep IP settings** checkbox.