MiiNePort E1 User's Manual

Version 8.0, May 2021

www.moxa.com/product

MiiNePort E1 User’s Manual

The software described in this manual is furnished under a license agreement and may be used only in accordance with the terms of that agreement.

Copyright Notice

Trademarks

Disclaimer

Information in this document is subject to change without notice and does not represent a commitment on the part of Moxa.

Moxa provides this document as is, without warranty of any kind, either expressed or implied, including, but not limited to, its particular purpose. Moxa reserves the right to make improvements and/or changes to this manual, or to the products and/or the programs described in this manual, at any time.

Information provided in this manual is intended to be accurate and reliable. However, Moxa assumes no responsibility for its use, or for any infringements on the rights of third parties that may result from its use.

This product might include unintentional technical or typographical errors. Changes are periodically made to the information herein to correct such errors, and these changes are incorporated into new editions of the publication.

Technical Support Contact Information

www.moxa.com/support

Moxa Americas

Toll-free:	1-888-669-2872
Tel:	+1-714-528-6777
Fax:	+1-714-528-6778

Moxa Europe

Tel:	+49-89-3 70 03 99-0
Fax:	+49-89-3 70 03 99-99

Moxa India

Tel:	+91-80-4172-9088
Fax:	+91-80-4132-1045

Moxa China (Shanghai office)

Toll-free:	800-820-5036
Tel:	+86-21-5258-9955
Fax:	+86-21-5258-5505

Moxa Asia-Pacific

Tel:	+886-2-8919-1230
Fax:	+886-2-8919-1231

Table of Contents

1

Introduction

The MiiNePort E1 Series embedded device servers are compact drop-in modules that can be integrated with your serial devices to enable connectivity to an Ethernet network. All MiiNePort E1 Series modules come equipped with built-in TCP/IP protocols and other easy-to-use network enabling tools for fast integration, allowing you to provide network access to any electronic device with a serial port.

The following topics are covered in this chapter:

- ❒ **Overview**
- ❒ **Package Checklist**
- ❒ **Product Features**
- ❒ **NetEZ**
  - ➢ MiiNePort E1 Module Dimensions
  - ➢ Recommended Device PCB Layout
- ❒ **Panel Layout and Pin Assignments**
  - ➢ MiiNePort E1-ST Evaluation Board Panel Layout
  - ➢ Pin Assignments
- ❒ **Block Diagram**
- ❒ **LED Indicators**

Overview

Moxa's MiiNePort E1 embedded device servers are designed for manufacturers who want to add sophisticated network connectivity to their serial devices with minimal integration effort. The MiiNePort E1 is powered by the MiiNe, Moxa's second generation SoC, which supports 10/100 Mbps Ethernet and up to 921.6 Kbps serial baudrate. The MiiNePort E1 comes with a versatile selection of ready-to-use operation modes, and requires only a small amount of power. By using Moxa's innovative NetEZ technology, the MiiNePort E1 can be used to convert any device with a standard serial interface to an Ethernet-enabled device in no time. In addition, the MiiNePort E1 is the size of an RJ45 connector, making it easy to fit into virtually any existing serial device.

Package Checklist

Module Package (one of the following)

- MiiNePort E1 (0 to 55°C operating temp., 50 bps to 230.4 Kbps baudrate)
- MiiNePort E1-H (0 to 55°C operating temp., 50 bps to 921.6 Kbps baudrate)
- MiiNePort E1-T (-40 to 85°C operating temp., 50 bps to 230.4 Kbps baudrate)
- MiiNePort E1-H-T (-40 to 85°C operating temp., 50 bps to 921.6 Kbps baudrate)

Starter Kit Package

- MiiNePort E1 module
- MiiNePort E1 evaluation board
- Universal power adapter
- 2 power cords
- Null modem serial cable
- Cross-over Ethernet cable
- Quick installation guide
- Warranty card

NOTE Please notify your sales representative if any of the above items is missing or damaged.

Product Features

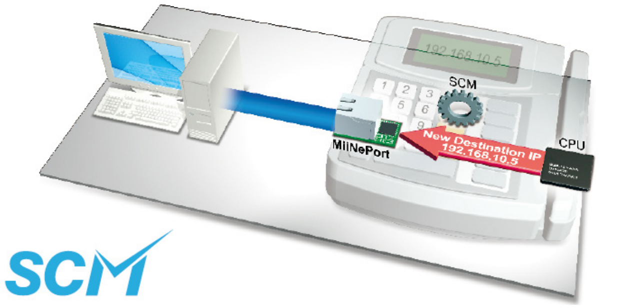
All MiiNePort E1 Series modules have the following general features:

- Same size as an RJ45 connector—only 33.9 x 16.25 x 13.5 mm.
- Extremely low power consumption
- Uses the MiiNe, Moxa's second generation SoC.
- NetEZ technology makes integration incredibly easy.
- A versatile choice of operation modes: Real COM, TCP, UDP, and MCSC.

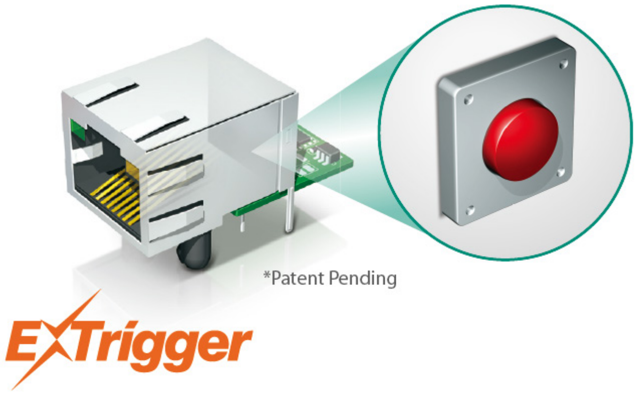
NetEZ

Moxa's NetEZ technology is designed to give serial device manufacturers a range of powerful tools for integrating Ethernet capability into serial devices. NetEZ technology includes:

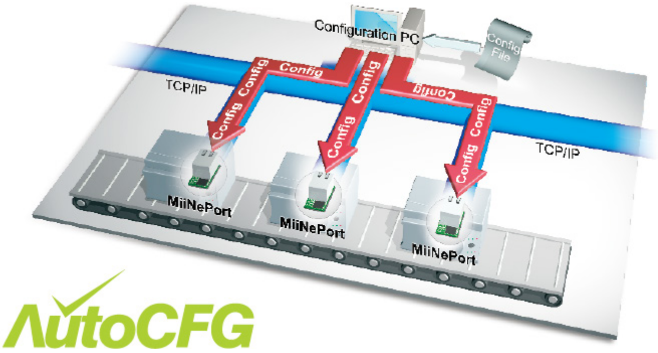
SCM (Serial Command Mode) can be used to easily configure the MiiNePort E1 through serial communication inside the device.



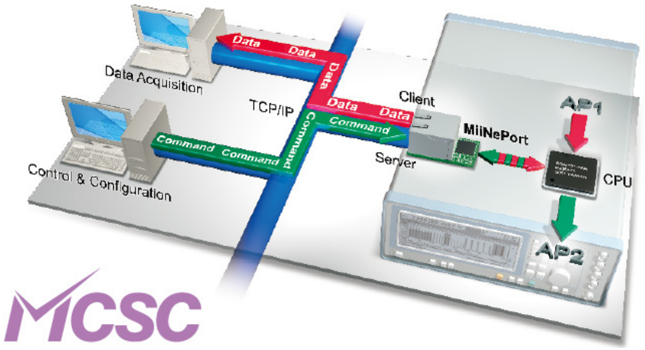
EXTrigger (External Trigger) restarts the MiiNePort E1 modules or resets the modules to factory defaults with just one click for easy troubleshooting.



AutoCFG (Auto Configuration) saves time and effort when setting up the MiiNePort E1 one by one during the device production process.



MCSC (Multiple Channel Serial Communication) provides dual connections and dual channels so your device can act as a server and client at the same time.



ATTENTION

Users must set jumpers JP15, JP16, and JP17 to correspond with the functions of pins 6, 7, and 8, respectively, to ensure that the evaluation board works properly.

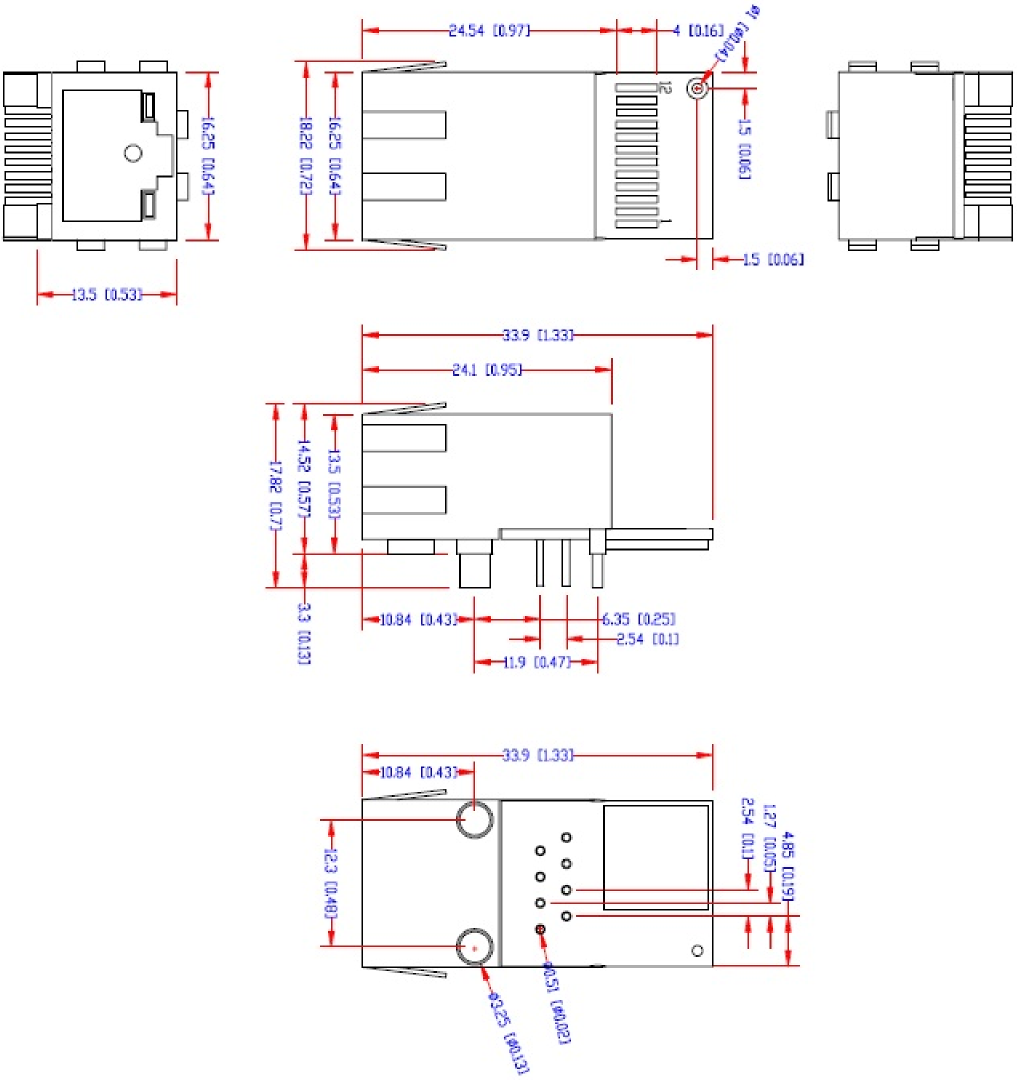
ATTENTION

When you are in RS-485 mode, 485EN must be configured by either JP15 or JP16, and the 6-pin jumper must be moved from JP19 to JP20.

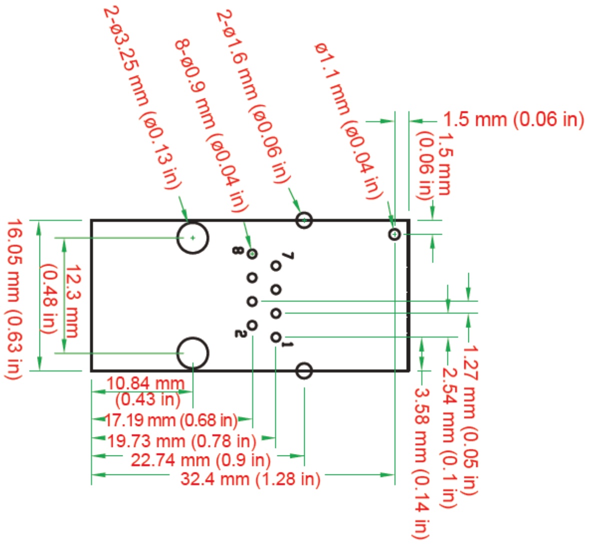
ATTENTION

Before you manipulate the jumpers, be sure to disconnect the power first.

MiiNePort E1 Module Dimensions

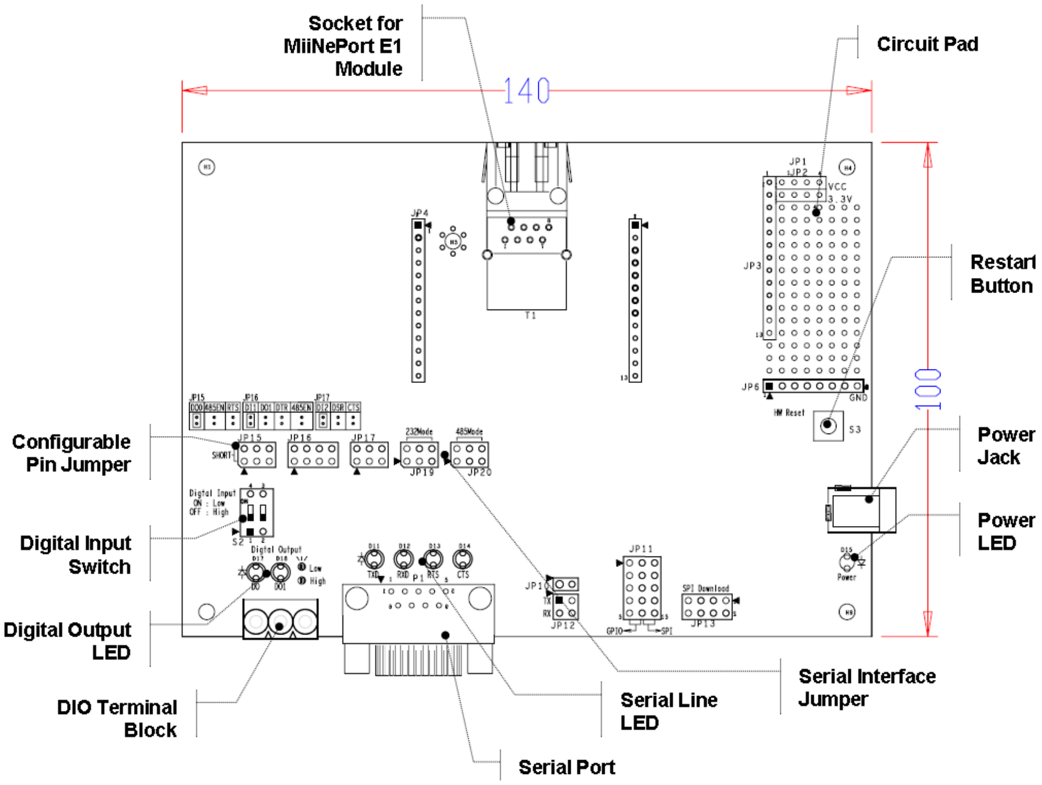


Recommended Device PCB Layout



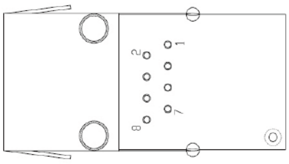
Panel Layout and Pin Assignments

MiiNePort E1-ST Evaluation Board Panel Layout



Pin Assignments

Serial Signal Pins for MiiNePort E1 Modules



Pin	Function
1	GND
2	VCC
3	Reset
4	Data Out
5	Data In
6	Ready/RTS[a]
7	Reset to Default[b]
8	CTS[c]

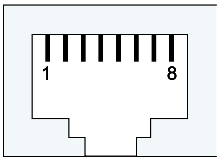
a. Pin 6 can be configured as Ready/RTS (Request to Send), Ready/DO, or RS-485 Tx Enabled (the default it Ready/RTS).

b. Pin 7 can be configured as Reset to Default, DIO, DTR, or RS-485 Tx Enabled (the default is Reset to Default).

c. Pin 8 can be configured as CTS (Clear to Send), DI, or DSR (the default is CTS)

Ethernet Port Pins for MiiNePort E1 Modules

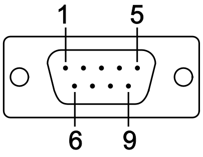
RJ45



Pin	Signal
1	Tx+
2	Tx-
3	Rx+
6	Rx-

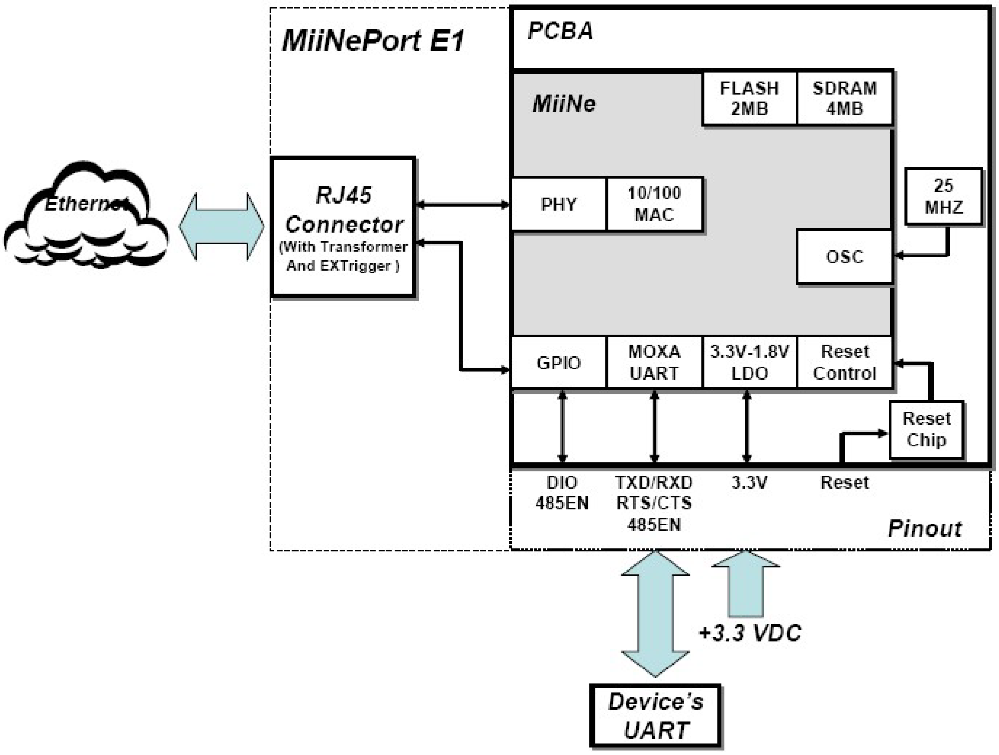
Serial Pin Signals for the MiiNePort E1-ST Evaluation Board

DB9 Male



Pin	RS-232	2-wire RS-485
1	DCD	–
2	RxD	–
3	TxD	Data+
4	DTR	Data-
5	GND	GND
6	DSR	–
7	RTS	–
8	CTS	–

Block Diagram



LED Indicators

MiiNePort E1 Series Modules

LED	Color	Meaning
Left	Green	100BASE-TX Link Activity (constant on when link exists, blinks when data transmitting)
	Amber	10BASE-T Link Activity (constant on when link exists, blinks when data transmitting)
Right	Green	In Use (constant on when connection established)
	Amber	Fault (blinks when IP fault)

MiiNePort E1-ST Evaluation Board

Power LED

- Power LED (D15) shows the power input status.

Serial Signal LED

- LED D11 shows the TxD status
- LED D12 shows the RxD status
- LED D13 shows the RTS status
- LED D14 shows the CTS status

Digital Output LED

- LED D17 shows the DO0 (Pin No. 6) status
- LED D18 show the DO1 (Pin No. 7) status.

2

Getting Started

This chapter includes information about how to install MiiNePort E1 Series modules for development and testing.

The following topics are covered in this chapter:

- ❒ **Wiring Precautions**
- ❒ **Installing the MiiNePort E1 onto the MiiNePort E1-ST**
- ❒ **Selecting the Serial Interface**
- ❒ **Circuit Pad for External Connection**
- ❒ **Connecting the Power**
- ❒ **Connecting to the Network**
- ❒ **Connecting to a Serial Device**
- ❒ **Digital I/O Channel Settings**
- ❒ **Schematic Design Guide**

Wiring Precautions

This section describes some important safety precautions that you should pay attention to before proceeding with any installation.

ATTENTION

Be sure to disconnect the power cord before installing or wiring the evaluation board.

ATTENTION

Determine the maximum possible current in each power wire and common wire. Observe all electrical codes dictating the maximum current allowable for each wire size. If the current goes above the maximum ratings, the wiring could overheat, causing serious damage to your equipment.

ATTENTION

Take care when handling the evaluation boards. When plugged in, the evaluation boards' internal components generate heat, and consequently the board may feel hot to the touch.

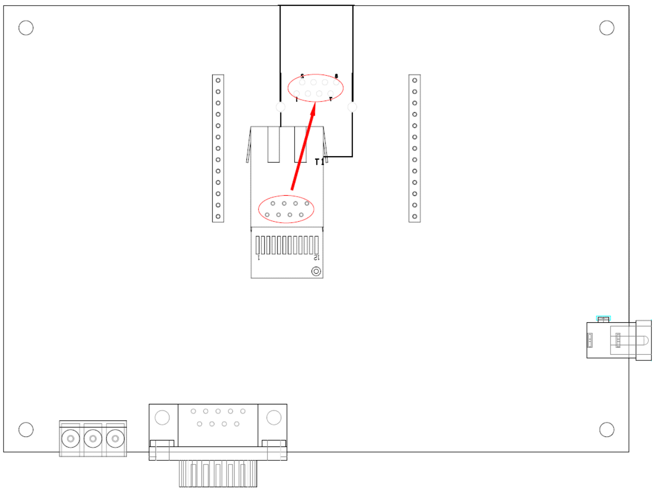
You should also pay attention to the following:

- Do not run signal or communication wiring and power wiring in the same wire conduit. To avoid interference, wires with different signal characteristics should be routed separately. Separate paths should be used to route wiring for power and devices. You can use the type of signal transmitted through a wire to determine which wires should be kept separate. The rule of thumb is that wires sharing similar electrical characteristics may be bundled together.
- Keep input wiring and output wiring separate.
- If power wiring and device wiring paths must cross paths, make sure the wires are perpendicular at the intersection point.
- All wiring should be clearly labeled.

Installing the MiiNePort E1 onto the MiiNePort E1-ST

Before using the MiiNePort E1-ST evaluation board with the module, be sure to disconnect the power supply, network, and serial device. In the top center of the evaluation board, there is a MiiNePort E1 profile that indicates where you should install the module on the evaluation board (shown in the below figure). When attaching the module to the evaluation board, make sure the 8 pins on the module are securely plugged into the 8 pin headers on the evaluation board.

After the module is installed, connect the power supply, network, and serial device to the evaluation board.



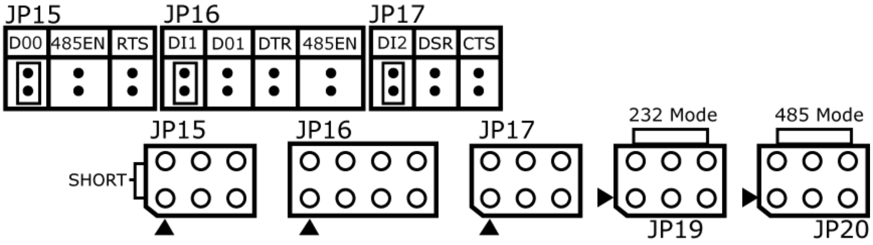
Selecting the Serial Interface

RS-232

To use an RS-232 serial interface, place the 6-pin jumper on JP19.

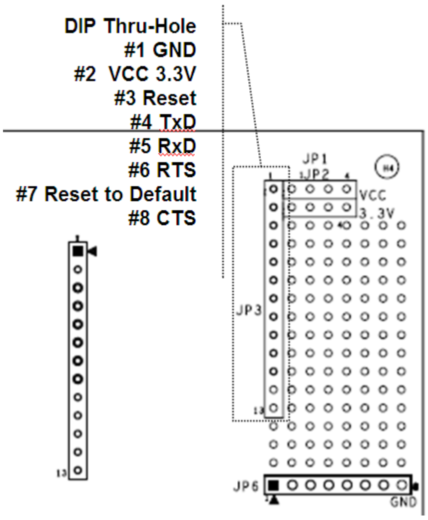
RS-485

To use an RS-485 serial interface, place the 2-pin jumper on the middle two pins of JP15 or the right-most two pins of JP16 (labeled as 485EN), and place the 6-pin jumper on JP20.



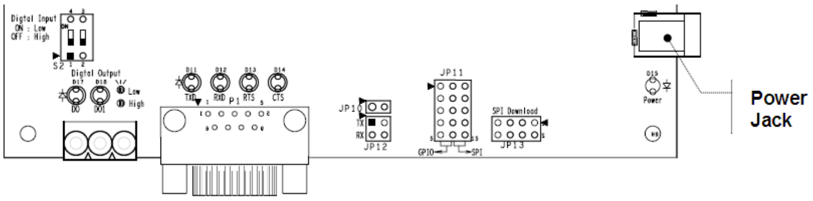
Circuit Pad for External Connection

A circuit pad is provided on the right side of each evaluation board for the development of additional application circuits.



Connecting the Power

Connect the 12-48 VDC power line with the evaluation board's power jack. If the power is properly supplied, the power LED (D15, as shown in the following figure) on the evaluation board will show a solid red color until the system is ready, at which time the ready LED on the module will show a solid green color.



Connecting to the Network

To connect to the network for testing and development purposes, install the module on the evaluation board and then plug the Ethernet cable into the RJ45 jack on the module. If the cable is properly connected, the LED on the module will indicate a valid connection to the Ethernet as follows:

LED	Color	Meaning
Left	Green	100BASE-TX Link Activity (constant on when link existed, blinks when data transmitting)
	Amber	10BASE-T Link Activity (constant on when link existed, blinks when data transmitting)
Right	Green	In Use (always on when connection established)
	Amber	Fault (blinks when IP fault)

When using a private IP address for the module, which is the factory default, make sure the netmask and IP settings are configured properly to access the module from a host on the network.

Connecting to a Serial Device

To connect to a serial device for testing and development purposes, the module should be installed on the evaluation board. Make sure the serial interface are configured correctly before connecting the evaluation board to the serial device. (Refer to **Selecting the Serial Interface** section above when you are using jumper blocks to select the serial interface on the evaluation board.) The module's serial signals are routed to and from the RS-232 or RS-485 COM port on the evaluation board. Use a serial data cable to connect the serial device to the COM port on the evaluation board.

Digital I/O Channel Settings

Each module has 3 digital I/O (DIO) channels. (Refer to the **Pin Assignments** section above for the module's configurable DIO pin description. Refer to **Configurable Pin Jumpers** to select the corresponding setting on the evaluation board.) All 3 DIO channels may be configured by software. A DI channel is a channel that is operating in digital input mode; a DO channel is a channel that is operating in digital output mode. You may use the evaluation board's Digital Output LEDs and Digital Input DIP switches as the digital input and output devices, or you may connect digital input/output devices to the DI/O Terminal Block.

ATTENTION

When using a digital input device connected to the DI/O Terminal Block, the corresponding Digital Input DIP switch must be set to **OFF** or **High**. Setting the DIP switch to **ON** or **Low** will interfere with the signal from your digital input device.

For channels in digital output mode, **Low** and **High** status is controlled from within the web console. When using a Digital Output LED as your output device, **Low** status will be expressed by the LED lighting up, and **High** status will be expressed by the LED turning off.

Schematic Design Guide

For guidance and suggestions on integrating your device's hardware with the MiiNePort E1, refer to the MiiNePort E1 Schematic Design Guide in our Document and Software CD.

3

Choosing the Proper Operation Mode

In this chapter, we describe the operation modes supported by MiiNePort E1 modules. Modes are available for COM port mapping from the host computer, as well as operation modes for TCP/IP protocols. After choosing the an operation mode in this chapter, refer to subsequent chapters for configuration details.

The following topics are covered in this chapter:

Overview

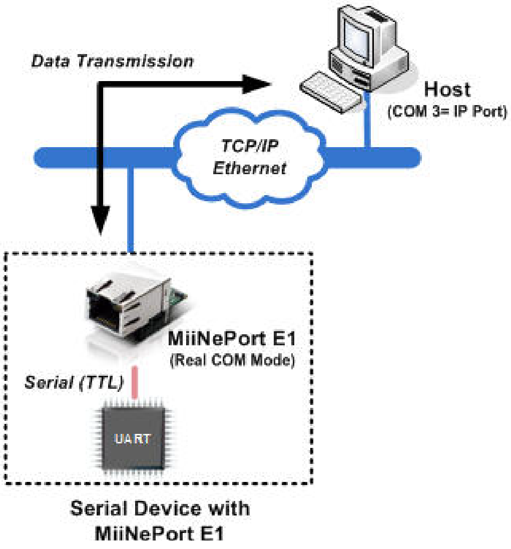
MiiNePort E1 modules act as a bridge to connect your serial devices to the Ethernet. The built-in TCP/IP stack frees you from the tedious task of programming networking protocols. With one step you can choose the proper operation mode, and then use your computer to access, manage, and configure your 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 a variety of instruments. Since MiiNePort E1 modules convert between serial and Ethernet signals, your SCADA and data collection system can be made accessible from any device connected to a standard TCP/IP network, regardless of whether the devices are used locally or at a remote site.

The MiiNePort E1 supports Real COM mode and four different socket modes—TCP Server, TCP Client, TCP Mixed, and UDP Server/Client. 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 offers speedier delivery by not requiring this type of verification. In addition, UDP also allows multicasting of data to groups of IP addresses.

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. The driver thus establishes a transparent connection between the host and serial device, allowing the host to treat the networked device as if it were directly attached.



ATTENTION

The Real COM driver comes with NPort Windows Driver Manager, which can be downloaded from the product resource page at moxa.com

ATTENTION

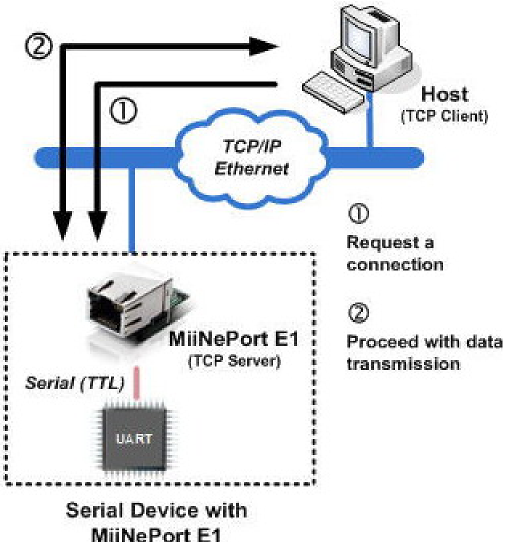
Real COM mode allows several hosts to simultaneously access the module. The driver controls host access to attached serial devices by checking the host's IP address against the Accessible IP list. Use the Accessible IP table to restrict access to the module when a public IP address is required for your application.

TCP Modes

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. This operation mode also supports up to 4 simultaneous connections, so that multiple hosts can collect data from the same serial device—at the same time. As illustrated in the figure, data transmission proceeds as follows:

1. The host connects to the module configured for TCP Server mode.
2. 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.

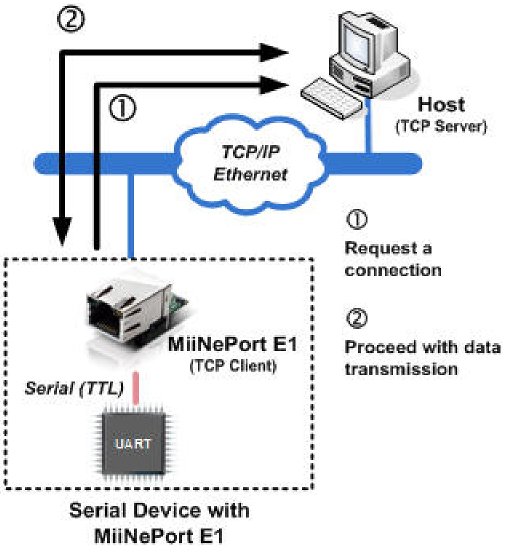


TCP Client Mode

In TCP Client mode, the module can actively establish a TCP connection to a pre-defined 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 the following chapters for more details.

As illustrated in the figure, data transmission proceeds as follows:

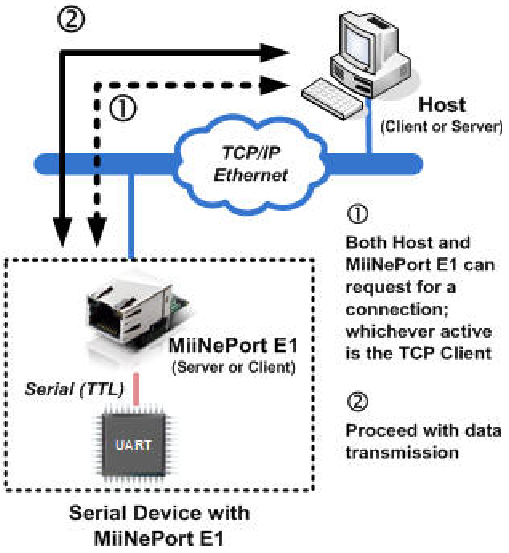
1. The module actively establishes a connection based on the conditions set in the firmware. You may let the module connect to a remote host on startup, or connect later when data from the serial device arrives.
2. Once the connection is established, data can be transmitted in both directions—from the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



TCP Mixed Mode

In TCP Mixed mode, the module has both TCP Server and Client settings. The MiiNePort module can act as a TCP Server or Client depending on the application. This mode is suitable for applications where the serial device actively sends data to the remote host as well as for applications where the remote host sends commands or other data to the serial device at the same time. As illustrated in the figure below, data transmission proceeds as follows:

1. Both the remote host and MiiNePort E1 can request a connection; if one acts as the TCP Client, the other acts as the TCP Server.
2. After the connection is established, data transmission will proceed.


②
①
Host
(Client or Server)
TCP/IP
Ethernet
①
Both Host and MiiNePort E1 can request for a connection; whichever active is the TCP Client
②
Proceed with data transmission
MiiNePort E1
(Server or Client)
Serial (TTL)
UART
Serial Device with
MiiNePort E1

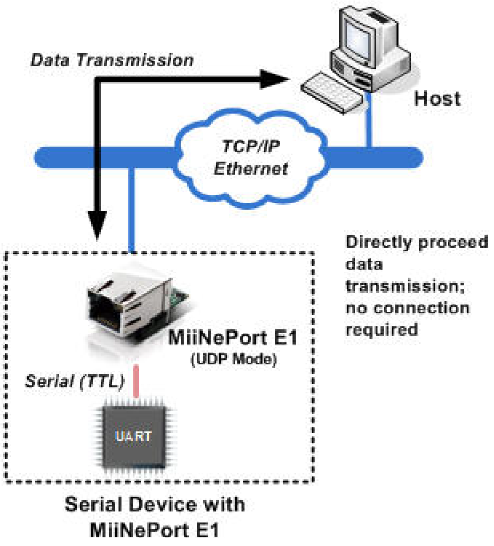
RFC2217 Mode

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.

NOTE To select RFC 2217 mode, first select TCP Server mode and then set Communication protocol to RFC 2217. For more details, refer to **Chapter 7: Web Console Configuration**.

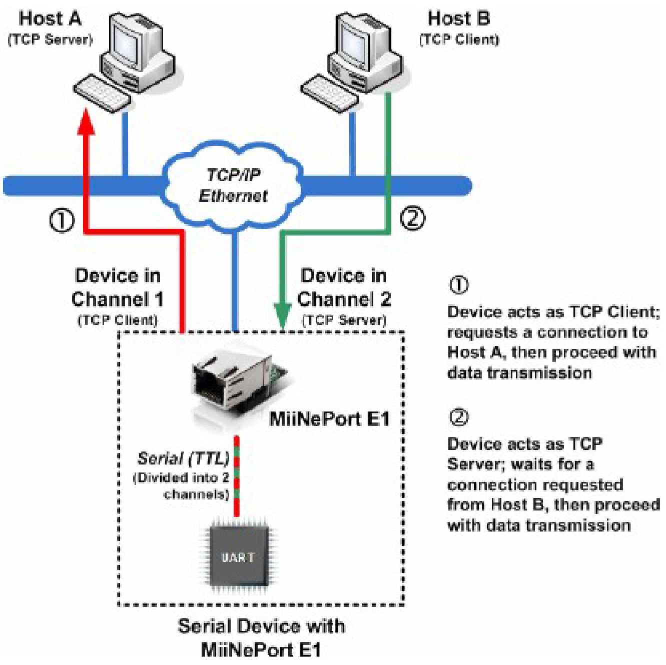
UDP Mode

UDP mode is faster and more efficient than TCP modes. In UDP mode, you can multicast data from the serial device to multiple host computers, and the serial device can also receive data from multiple host computers, making this mode ideal for message display applications.



MCSC Mode

MCSC (Multiple Channel Serial Communication) was developed for multiple serial-to-Ethernet applications that use only one serial port. For example, if you need your device to act as a TCP Server and TCP Client at the same time (as illustrated below), you can use MCSC.


Host A
(TCP Server)
Host B
(TCP Client)
TCP/IP
Ethernet
①
②
Device in
Channel 1
(TCP Client)
Device in
Channel 2
(TCP Server)
MiiNePort E1
Serial (TTL)
(Divided into 2
channels)
UART
Serial Device with
MiiNePort E1
①
Device acts as TCP Client; requests a connection to Host A, then proceed with data transmission
②
Device acts as TCP Server; waits for a connection requested from Host B, then proceed with data transmission

For details on MCSC's functionality and configuration, refer to "Chapter 8: NetEZ Technologies' MSCS."

4

Choosing the Configuration Tool

The MiiNePort E1 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**

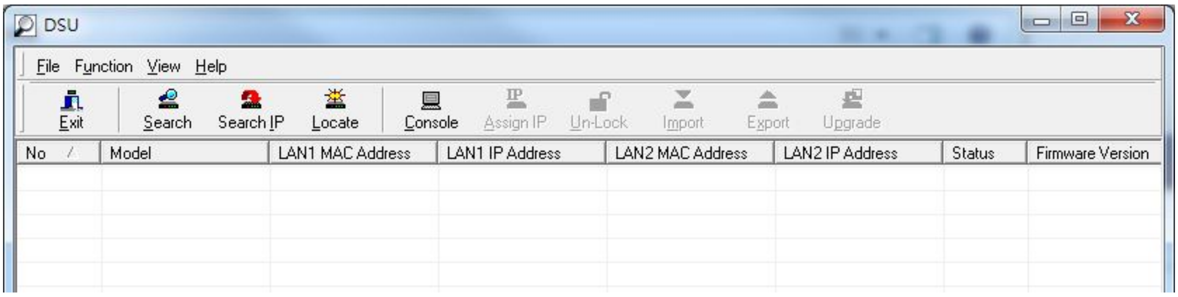
❐ **Telnet Console**

❐ **SCM (Serial Command Mode)**

Utility Console

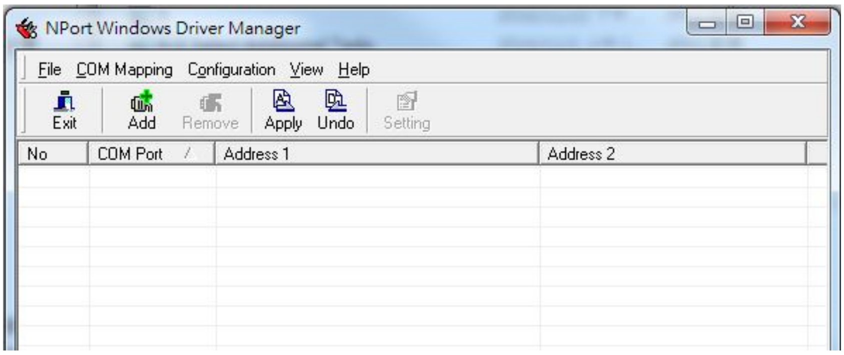
Device Search Utility

You can find the Device Search Utility (DSU) on the CD-ROM. The Device Search Utility is designed for Windows and is mainly used to search for the MiiNePort E1 modules and for assigning IP addresses. Refer to the Web Console for additional configuration information.



NPort Windows Driver Manager

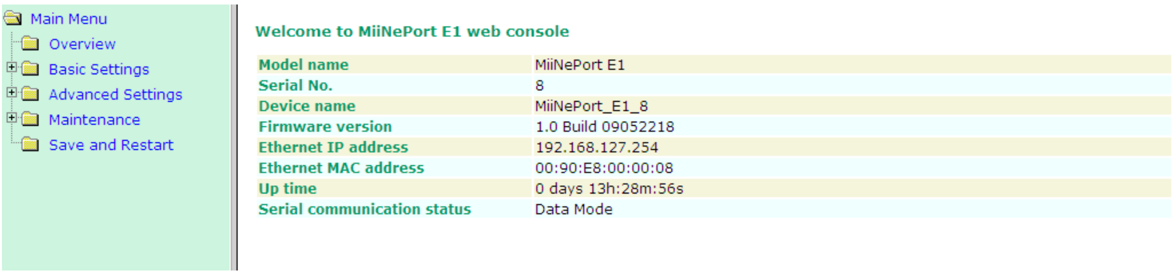
NPort Windows Driver Manager is intended for use with Real COM mode. The software manages the installation of drivers that allow you to map unused COM ports on your PC to serial ports on the MiiNePort E1.


NPort Windows Driver Manager
File COM Mapping Configuration View Help
Exit
Add
Remove
Apply
Undo
Setting
No
COM Port
Address 1
Address 2

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

Web Console

After locating a MiiNePort E1 with the Device Search Utility, you may configure the MiiNePort E1 using a standard web browser. Refer to **Chapter 7: Web Console Configuration** for details on how to access and use the MiiNePort E1 web console.

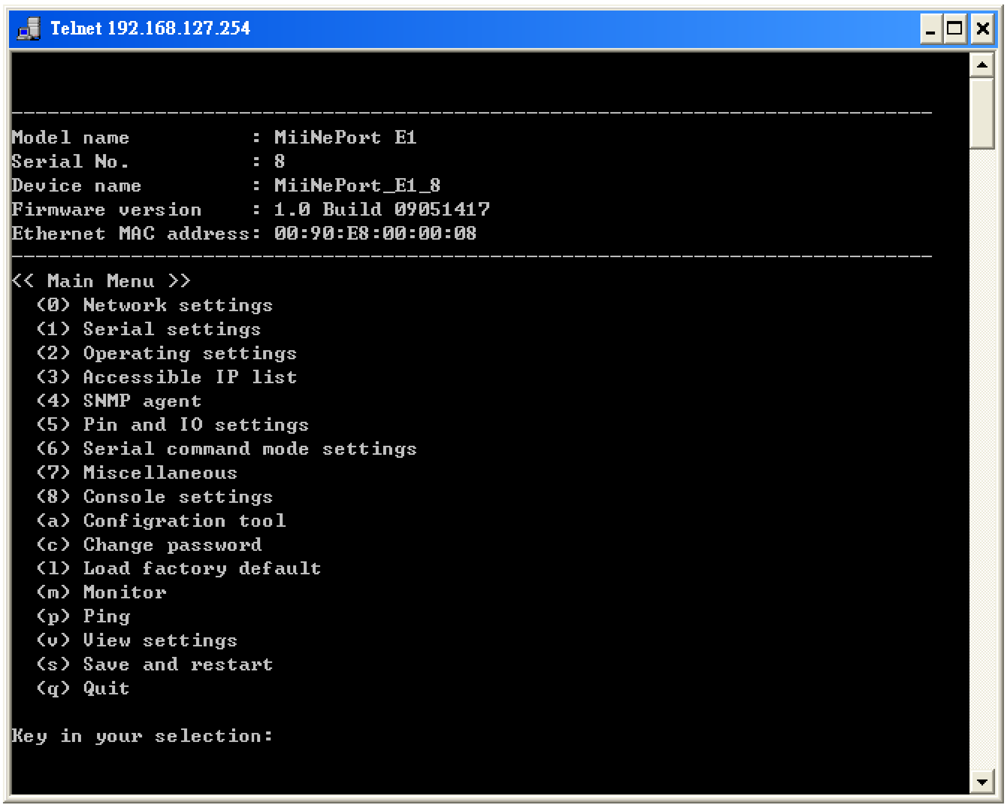


Welcome to MiiNePort E1 web console

Model name	MiiNePort E1
Serial No.	8
Device name	MiiNePort_E1_8
Firmware version	1.0 Build 09052218
Ethernet IP address	192.168.127.254
Ethernet MAC address	00:90:E8:00:00:08
Up time	0 days 13h:28m:56s
Serial communication status	Data Mode

Telnet Console

Your MiiNePort E1 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**.



```
----------------------------------------------------------------------------
Model name          : MiiNePort E1
Serial No.          : 8
Device name         : MiiNePort_E1_8
Firmware version    : 1.0 Build 09051417
Ethernet MAC address: 00:90:E8:00:00:08
----------------------------------------------------------------------------
<< Main Menu >>
  (0) Network settings
  (1) Serial settings
  (2) Operating settings
  (3) Accessible IP list
  (4) SNMP agent
  (5) Pin and IO settings
  (6) Serial command mode settings
  (7) Miscellaneous
  (8) Console settings
  (a) Configration tool
  (c) Change password
  (l) Load factory default
  (m) Monitor
  (p) Ping
  (v) View settings
  (s) Save and restart
  (q) Quit

Key in your selection:
```

SCM (Serial Command Mode)

The MiiNePort E1's SCM (Serial Command Mode) allows the module's parameters to be retrieved and configured through the serial port. This is accomplished using specially parsed commands sent to the module through the serial port.

SCM is often used when your device has already been used in a real application and a configuration change, such as changing the device's IP address with the device's key pad, is required.

Refer to **Chapter 7: Web Console Configuration** for details on how to access and use the MiiNePort E1's SCM. Refer to Appendix A: Introduction to SCM (Serial Command Mode) for the SCM command set instructions.

5

Initial IP Address Configuration

When setting up your MiiNePort E1 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**

Static vs. Dynamic IP Address

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

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.

Factory Default IP Address

The MiiNePort E1 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.

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 cross-over 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:

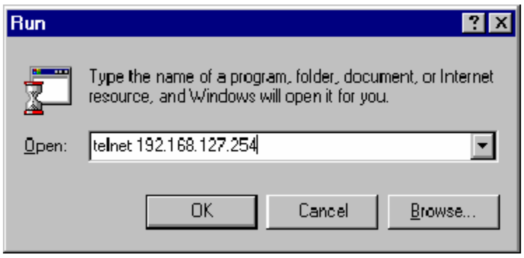
1. Obtain a valid IP address for the module from your network administrator.
2. Obtain the module's MAC address from the label on the module.
3. Execute the **arp -s** command from your computer's MS-DOS prompt by typing:
   ```
   arp –s <new IP address> 00-90-E8-tt-tt-tt
   ```
   For example,
   ```
   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.
4. 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.

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.

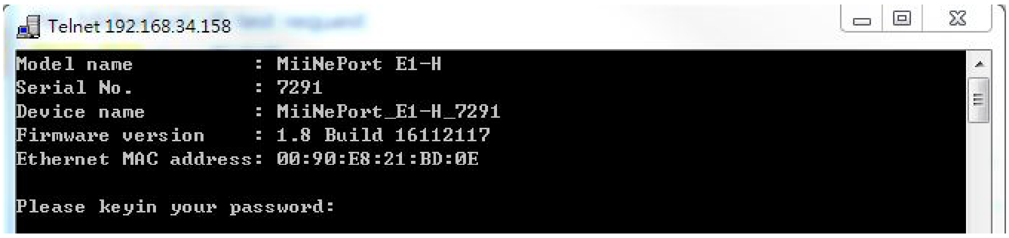
1. From the Windows desktop, click **Start** and then select **Run**.
2. Telnet to the module's current IP address. If this is the first time configuring the module, you will telnet to the default IP address by typing **telnet 192.168.127.254** in the **Open** text box. Click **OK** to proceed.



Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.

Open: telnet 192.168.127.254

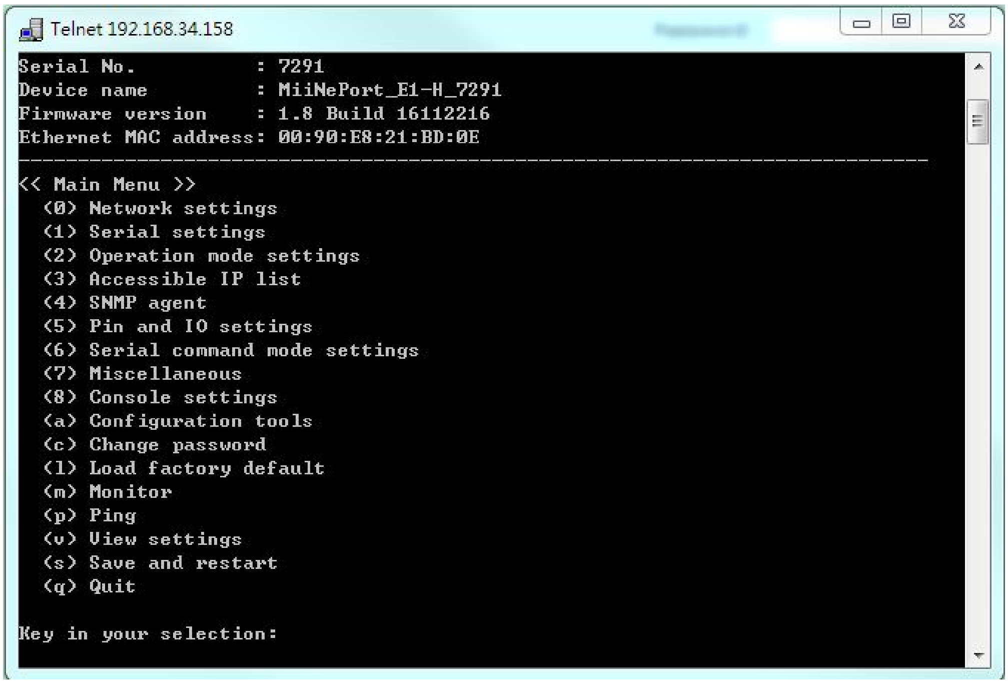
You will be asked to enter a password to access to the device. The default password for MiiNePort E1 is **moxa**.



```
Model name           : MiiNePort E1-H
Serial No.           : 7291
Device name          : MiiNePort_E1-H_7291
Firmware version     : 1.8 Build 16112117
Ethernet MAC address: 00:90:E8:21:BD:0E

Please keyin your password:
```

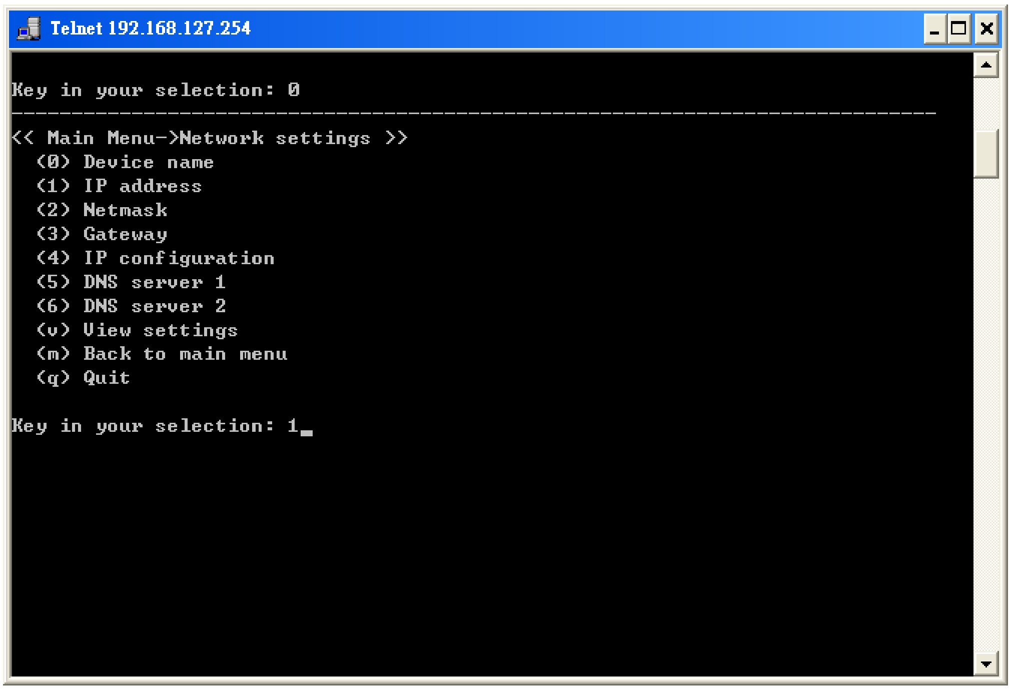
3. Select Network settings by pressing **0** and then press **Enter**.



```
Serial No.           : 7291
Device name          : MiiNePort_E1-H_7291
Firmware version     : 1.8 Build 16112216
Ethernet MAC address: 00:90:E8:21:BD:0E
------------------------------------------------------------------------------
<< Main Menu >>
  (0) Network settings
  (1) Serial settings
  (2) Operation mode settings
  (3) Accessible IP list
  (4) SNMP agent
  (5) Pin and IO settings
  (6) Serial command mode settings
  (7) Miscellaneous
  (8) Console settings
  (a) Configuration tools
  (c) Change password
  (l) Load factory default
  (m) Monitor
  (p) Ping
  (v) View settings
  (s) Save and restart
  (q) Quit

Key in your selection:
```

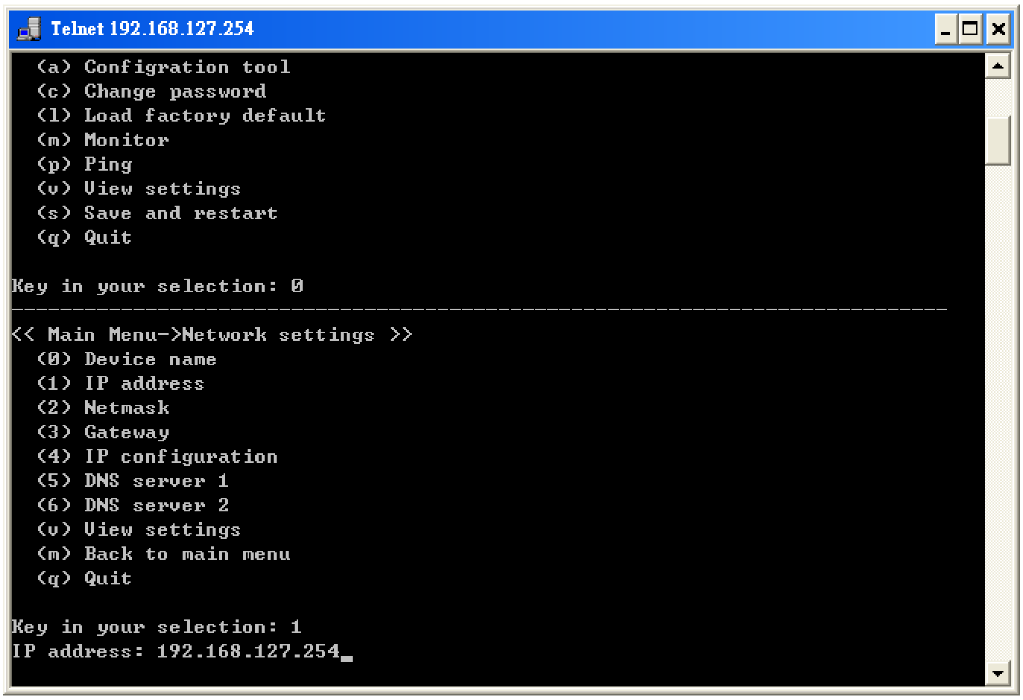
4. Select IP address by pressing **1** and then press **Enter**.



```
Key in your selection: 0
------------------------------------------------------------------------------
<< Main Menu->Network settings >>
  (0) Device name
  (1) IP address
  (2) Netmask
  (3) Gateway
  (4) IP configuration
  (5) DNS server 1
  (6) DNS server 2
  (v) View settings
  (m) Back to main menu
  (q) Quit

Key in your selection: 1
```

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

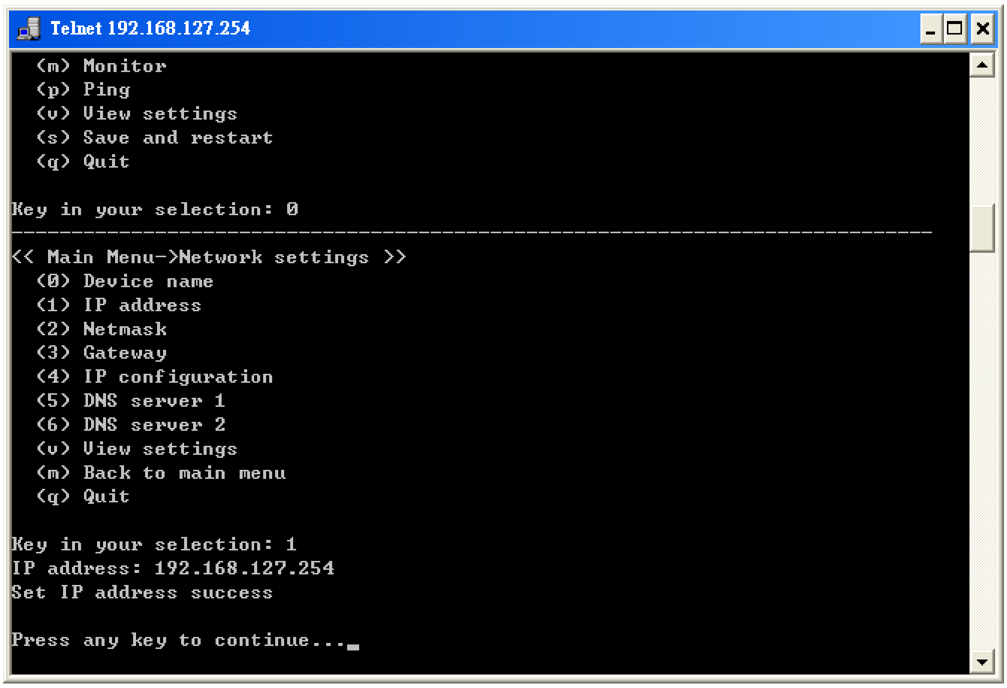


```
  (a) Configration tool
  (c) Change password
  (l) Load factory default
  (m) Monitor
  (p) Ping
  (v) View settings
  (s) Save and restart
  (q) Quit

Key in your selection: 0
------------------------------------------------------------------------------
<< Main Menu->Network settings >>
  (0) Device name
  (1) IP address
  (2) Netmask
  (3) Gateway
  (4) IP configuration
  (5) DNS server 1
  (6) DNS server 2
  (v) View settings
  (m) Back to main menu
  (q) Quit

Key in your selection: 1
IP address: 192.168.127.254
```

6. Press any key to continue.



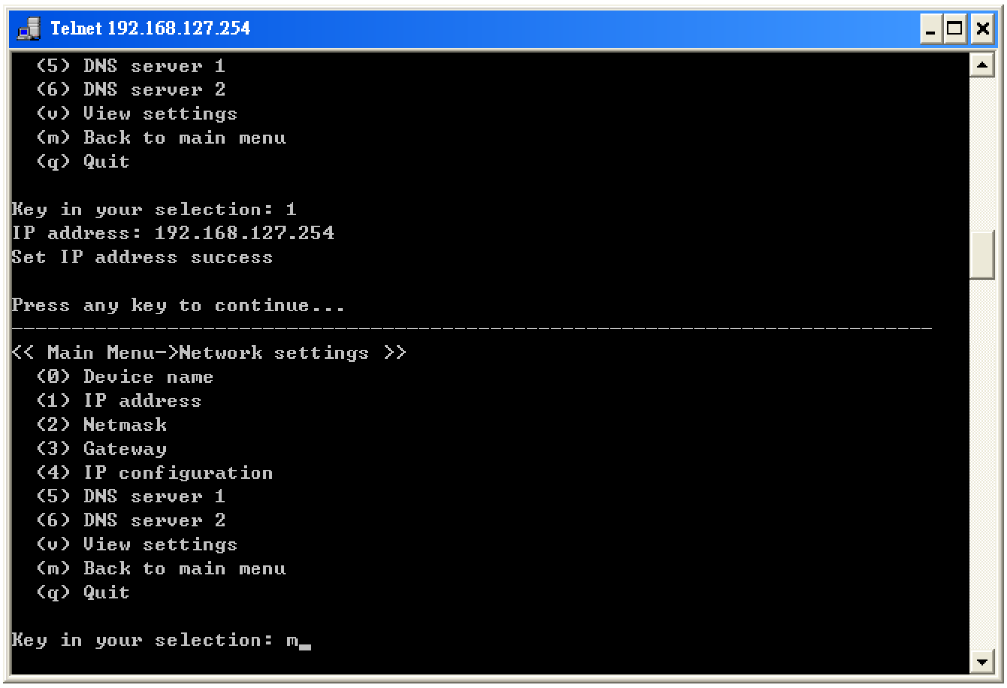
```
  <m> Monitor
  <p> Ping
  <v> View settings
  <s> Save and restart
  <q> Quit

Key in your selection: 0
-----------------------------------------------------------------------------
<< Main Menu->Network settings >>
  <0> Device name
  <1> IP address
  <2> Netmask
  <3> Gateway
  <4> IP configuration
  <5> DNS server 1
  <6> DNS server 2
  <v> View settings
  <m> Back to main menu
  <q> Quit

Key in your selection: 1
IP address: 192.168.127.254
Set IP address success

Press any key to continue..._
```

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



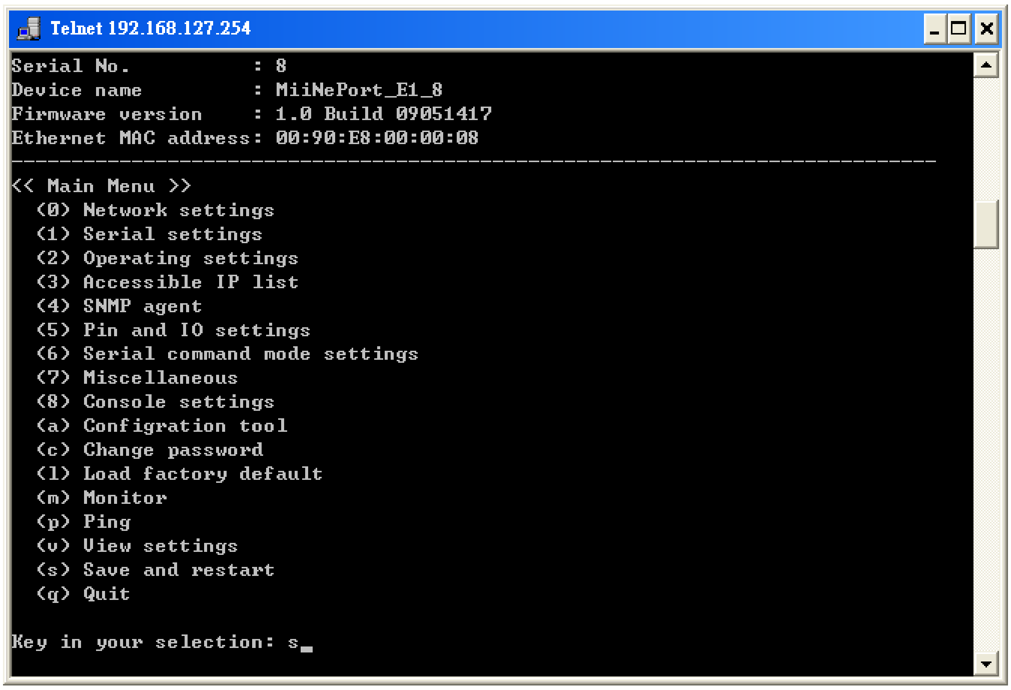
```
  <5> DNS server 1
  <6> DNS server 2
  <v> View settings
  <m> Back to main menu
  <q> Quit

Key in your selection: 1
IP address: 192.168.127.254
Set IP address success

Press any key to continue...
-----------------------------------------------------------------------------
<< Main Menu->Network settings >>
  <0> Device name
  <1> IP address
  <2> Netmask
  <3> Gateway
  <4> IP configuration
  <5> DNS server 1
  <6> DNS server 2
  <v> View settings
  <m> Back to main menu
  <q> Quit

Key in your selection: m_
```

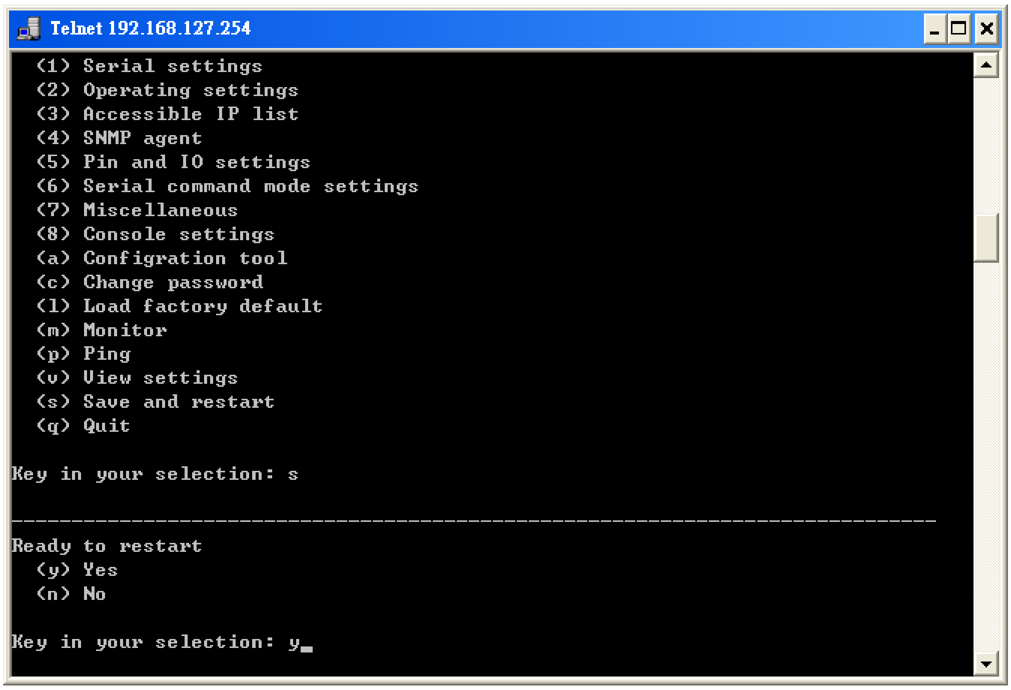
8. Press **S** and then **Enter** to **Save/Restart** the system.



```
Serial No.            : 8
Device name           : MiiNePort_E1_8
Firmware version      : 1.0 Build 09051417
Ethernet MAC address: 00:90:E8:00:00:08
-----------------------------------------------------------------------------
<< Main Menu >>
  (0) Network settings
  (1) Serial settings
  (2) Operating settings
  (3) Accessible IP list
  (4) SNMP agent
  (5) Pin and IO settings
  (6) Serial command mode settings
  (7) Miscellaneous
  (8) Console settings
  (a) Configration tool
  (c) Change password
  (l) Load factory default
  (m) Monitor
  (p) Ping
  (v) View settings
  (s) Save and restart
  (q) Quit

Key in your selection: s_
```

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



```
  (1) Serial settings
  (2) Operating settings
  (3) Accessible IP list
  (4) SNMP agent
  (5) Pin and IO settings
  (6) Serial command mode settings
  (7) Miscellaneous
  (8) Console settings
  (a) Configration tool
  (c) Change password
  (l) Load factory default
  (m) Monitor
  (p) Ping
  (v) View settings
  (s) Save and restart
  (q) Quit

Key in your selection: s

-----------------------------------------------------------------------------
Ready to restart
  (y) Yes
  (n) No

Key in your selection: y_
```

6

Utility Console and Driver Installation

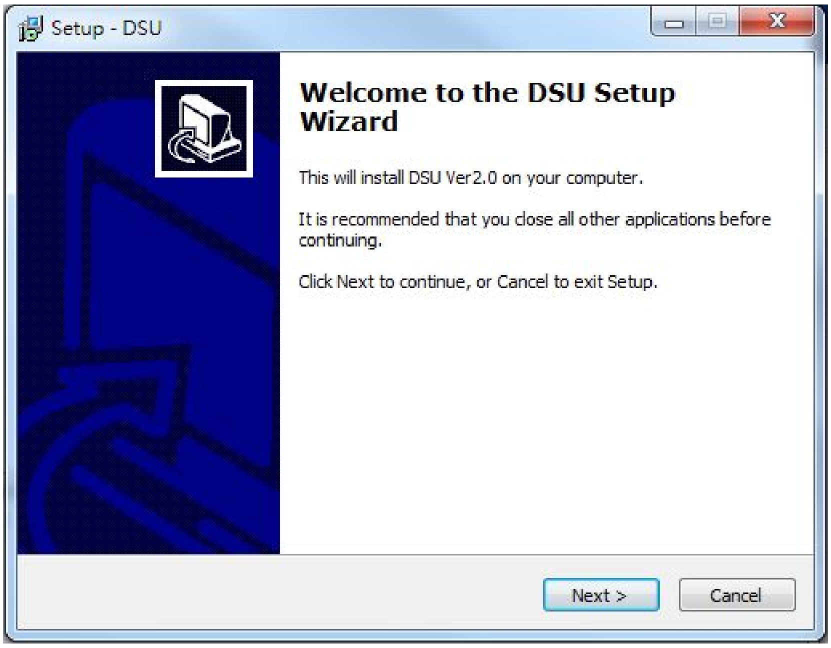
This chapter teaches you how to install the MiiNePort E1's utilities, use the utilities to perform simple configurations, and install the drivers.

The following topics are covered in this chapter:

Device Search Utility (DSU)

Installing the Device Search Utility

1. Click the **INSTALL UTILITY** button in the MiiNePort E1 Installation CD auto-run window to install the Device Search Utility. Once the program starts running, click **Yes** to proceed.
2. Click **Next** when the Welcome screen opens to proceed with the installation.



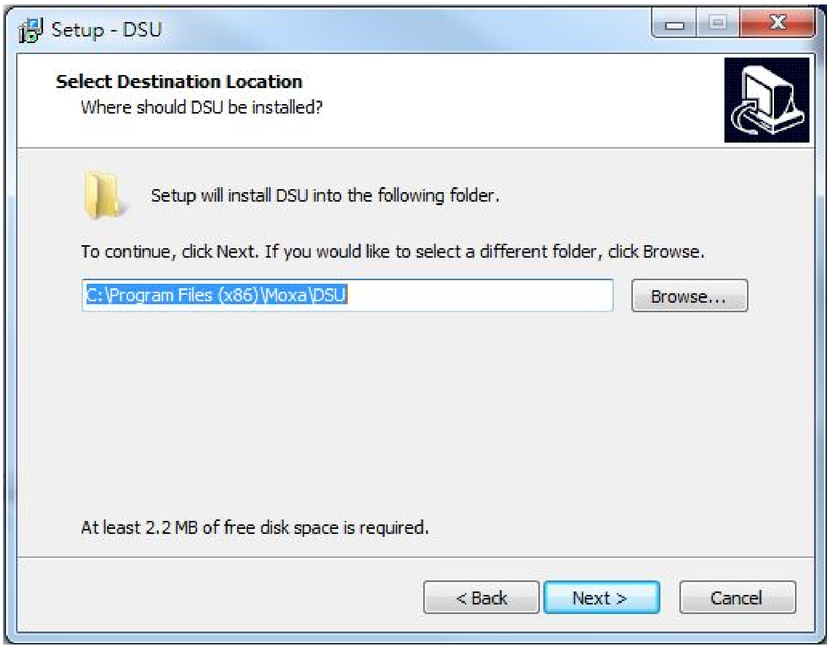
Welcome to the DSU Setup Wizard

This will install DSU Ver2.0 on your computer.

It is recommended that you close all other applications before continuing.

Click Next to continue, or Cancel to exit Setup.

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



Select Destination Location

Where should DSU be installed?

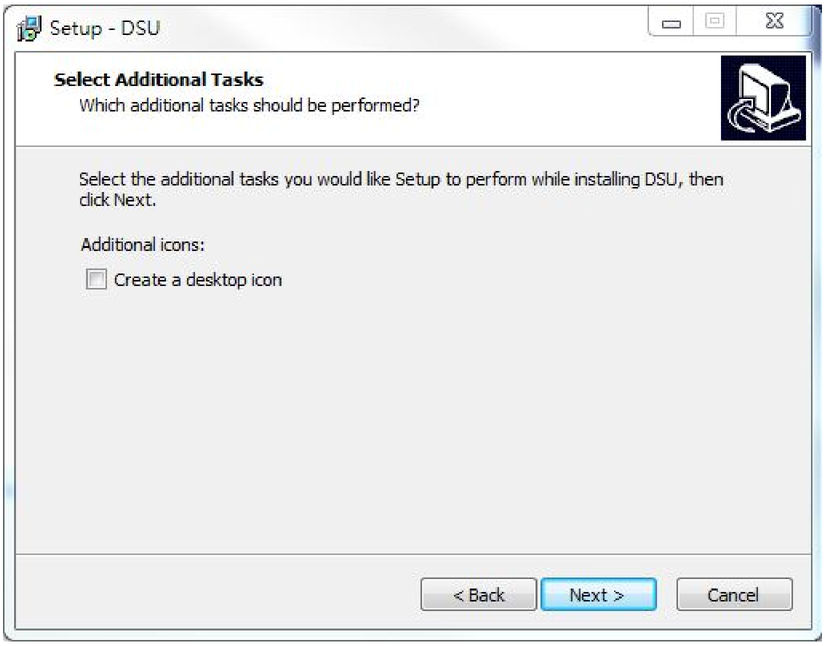
Setup will install DSU into the following folder.

To continue, click Next. If you would like to select a different folder, click Browse.

C:\Program Files (x86)\Moxa\DSU

At least 2.2 MB of free disk space is required.

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



Setup - DSU

Select Additional Tasks

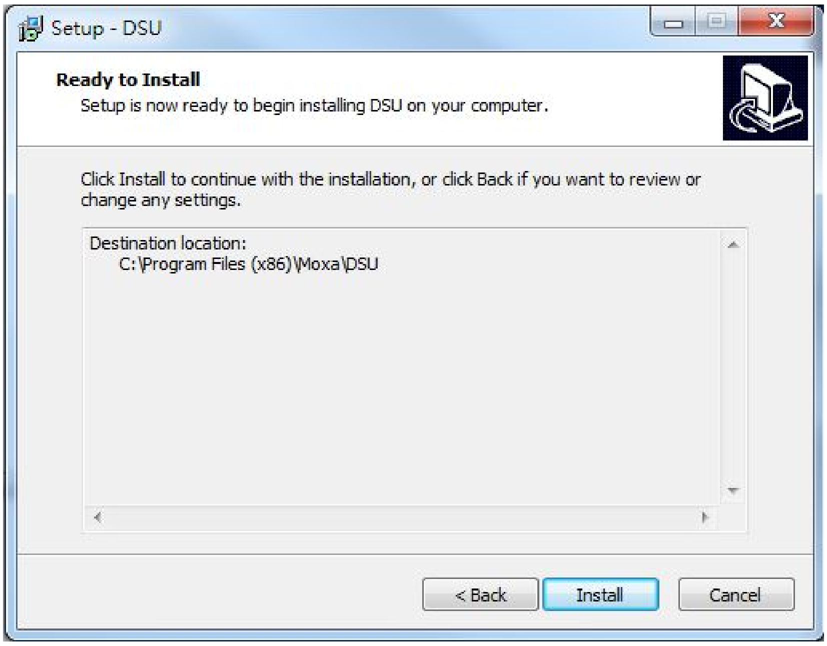
Which additional tasks should be performed?

Select the additional tasks you would like Setup to perform while installing DSU, then click Next.

Additional icons:

□ Create a desktop icon

5. 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.



Setup - DSU

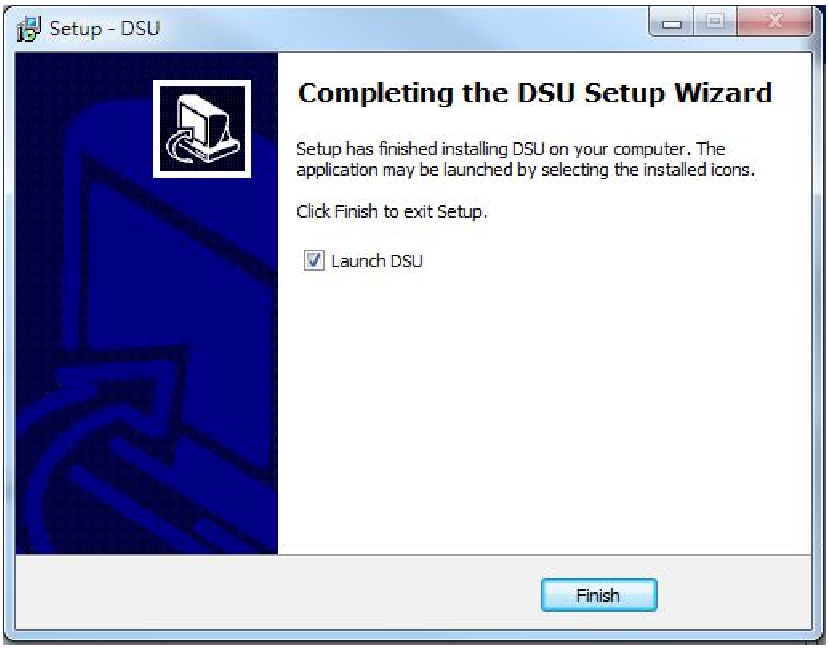
Ready to Install

Setup is now ready to begin installing DSU on your computer.

Click Install to continue with the installation, or click Back if you want to review or change any settings.

Destination location:
C:\Program Files (x86)\Moxa\DSU

6. Click **Finish** to complete the installation of the Device Search Utility.



Completing the DSU Setup Wizard

Setup has finished installing DSU on your computer. The application may be launched by selecting the installed icons.

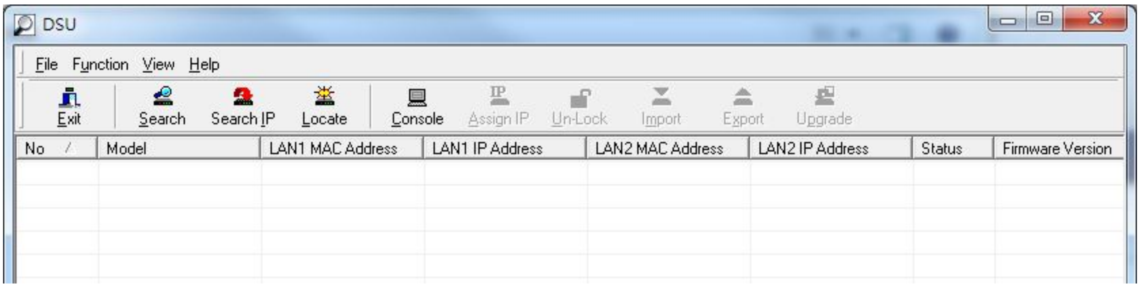
Click Finish to exit Setup.

☑ Launch DSU

Device Search Utility Configuration

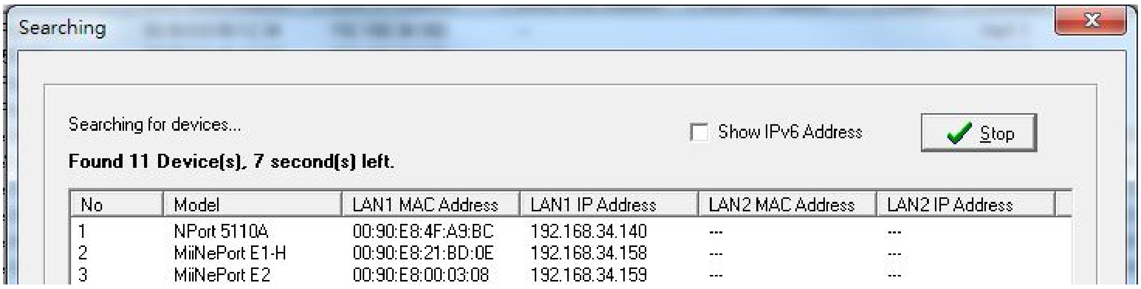
The Broadcast Search function is used to locate all MiiNePort E1 modules that are connected to the same LAN as your computer. After locating a MiiNePort E1, you will be able to change its IP address. Since the Broadcast Search function searches by MAC address and not IP address, all MiiNePort E1 modules connected to the LAN will be located, regardless of whether or not they are part of the same subnet as the host.

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



No	Model	LAN1 MAC Address	LAN1 IP Address	LAN2 MAC Address	LAN2 IP Address	Status	Firmware Version

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



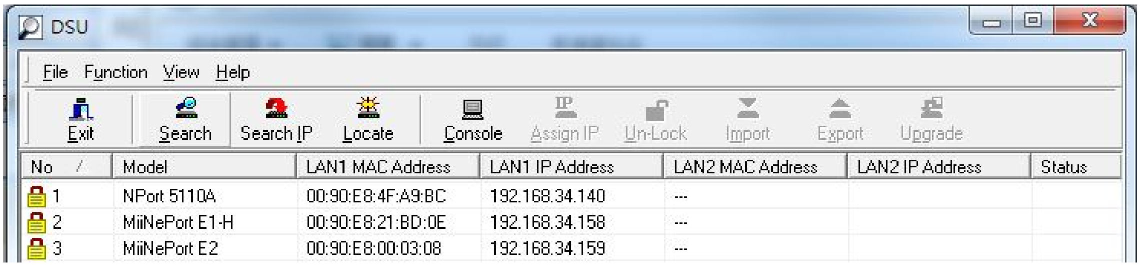
Searching for devices...

Found 11 Device(s), 7 second(s) left.

□ Show IPv6 Address

No	Model	LAN1 MAC Address	LAN1 IP Address	LAN2 MAC Address	LAN2 IP Address
1	NPort 5110A	00:90:E8:4F:A9:BC	192.168.34.140	---	---
2	MiiNePort E1-H	00:90:E8:21:BD:0E	192.168.34.158	---	---
3	MiiNePort E2	00:90:E8:00:03:08	192.168.34.159	---	---

3. When the search is completed, all the located MiiNePort E1 modules will be displayed in the Device Search Utility window. Select the device you wish to access and press the **UnLock** button to input the password for the device. The default password is **moxa**.



No	Model	LAN1 MAC Address	LAN1 IP Address	LAN2 MAC Address	LAN2 IP Address	Status
1	NPort 5110A	00:90:E8:4F:A9:BC	192.168.34.140	---		
2	MiiNePort E1-H	00:90:E8:21:BD:0E	192.168.34.158	---		
3	MiiNePort E2	00:90:E8:00:03:08	192.168.34.159	---		

4. To modify the configuration of the highlighted MiiNePort E1, 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

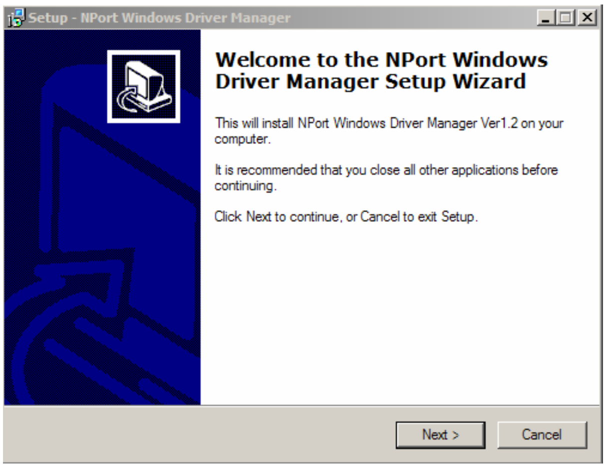
If you are looking for information related to TCP, RFC2217, or UDP modes, you can ignore the following Driver sections, including NPort Windows Driver Manager and Linux Real TTY Driver, and instead jump directly to Chapter 7: Web Console Configuration for additional settings.

NPort Windows Driver Manager

Installing the 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 presented by MiiNePort E1's serial port. When the drivers are installed and configured, devices that are embedded with the MiiNePort E1 will be treated as if they are attached to your PC's own COM ports.

1. Click the **INSTALL COM Driver** button in the MiiNePort E1 Installation CD auto-run window to install the NPort Windows Driver. Once the installation program starts running, click **Yes** to proceed.
2. Click **Next** when the Welcome screen opens to proceed with the installation.



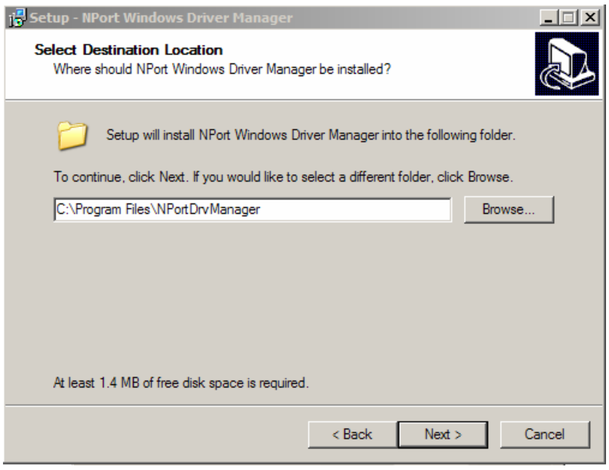
Welcome to the NPort Windows Driver Manager Setup Wizard

This will install NPort Windows Driver Manager Ver1.2 on your computer.

It is recommended that you close all other applications before continuing.

Click Next to continue, or Cancel to exit Setup.

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


Setup - NPort Windows Driver Manager
Select Destination Location
Where should NPort Windows Driver Manager be installed?

Setup will install NPort Windows Driver Manager into the following folder.

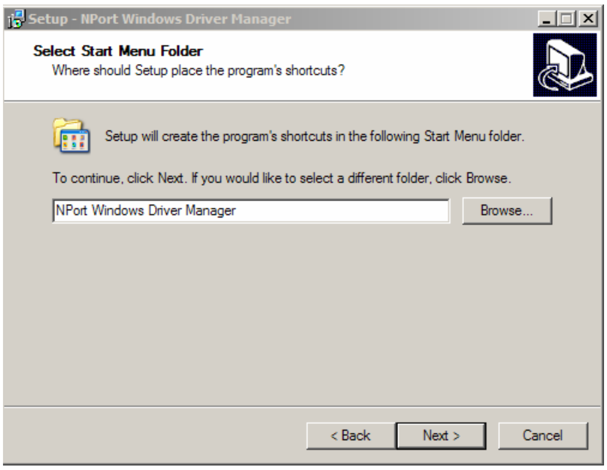
To continue, click Next. If you would like to select a different folder, click Browse.

C:\Program Files\NPortDrvManager

At least 1.4 MB of free disk space is required.

Browse...
< Back
Next >
Cancel

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


Setup - NPort Windows Driver Manager
Select Start Menu Folder
Where should Setup place the program's shortcuts?

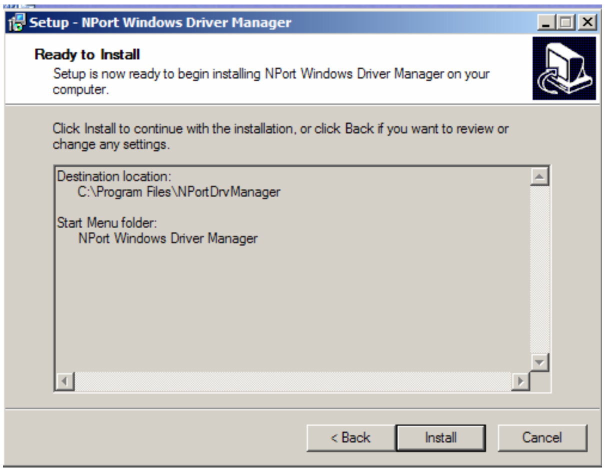
Setup will create the program's shortcuts in the following Start Menu folder.

To continue, click Next. If you would like to select a different folder, click Browse.

NPort Windows Driver Manager

Browse...
< Back
Next >
Cancel

5. 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.



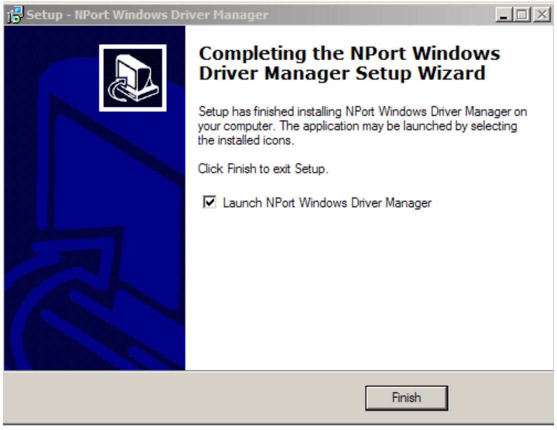
Setup is now ready to begin installing NPort Windows Driver Manager on your computer.

Click Install to continue with the installation, or click Back if you want to review or change any settings.

Destination location:
C:\Program Files\NPortDrvManager

Start Menu folder:
NPort Windows Driver Manager

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



Completing the NPort Windows Driver Manager Setup Wizard

Setup has finished installing NPort Windows Driver Manager on your computer. The application may be launched by selecting the installed icons.

Click Finish to exit Setup.

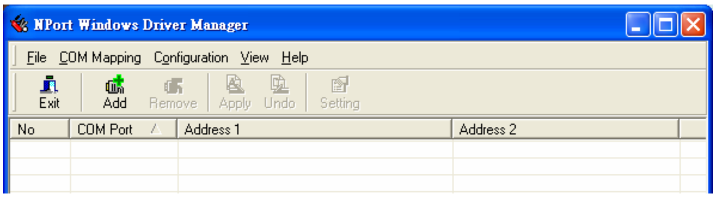
☑ Launch NPort Windows Driver Manager

Using the NPort Windows Driver Manager

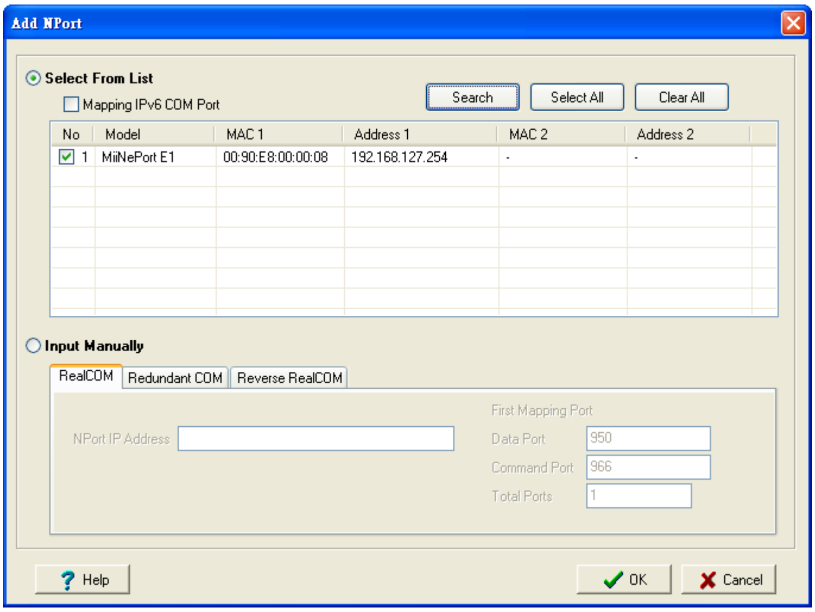
After you install the NPort Windows Driver Manager, you can set up the MiiNePort E1's serial port, which is connected to your device's mainboard, as remote COM ports for your PC host. Make sure that the serial port on your MiiNePort E1 is already set to Real COM mode when mapping COM ports with the NPort Windows Driver Manager.

NOTE Refer to Chapter 7: Web Console Configuration to learn how to configure your MiiNePort E1 to Real COM mode.

1. Go to **Start → NPort Windows Driver Manager → NPort Windows Driver Manager** to start the COM mapping utility.
2. Click the **Add** icon.



3. Click **Search** to search for the MiiNePort E1 modules. From the list that is generated, select the server to which you will map COM ports, and then click **OK**.



● Select From List

□ Mapping IPv6 COM Port

No	Model	MAC 1	Address 1	MAC 2	Address 2
☑ 1	MiiNePort E1	00:90:E8:00:00:08	192.168.127.254	-	-

○ Input Manually

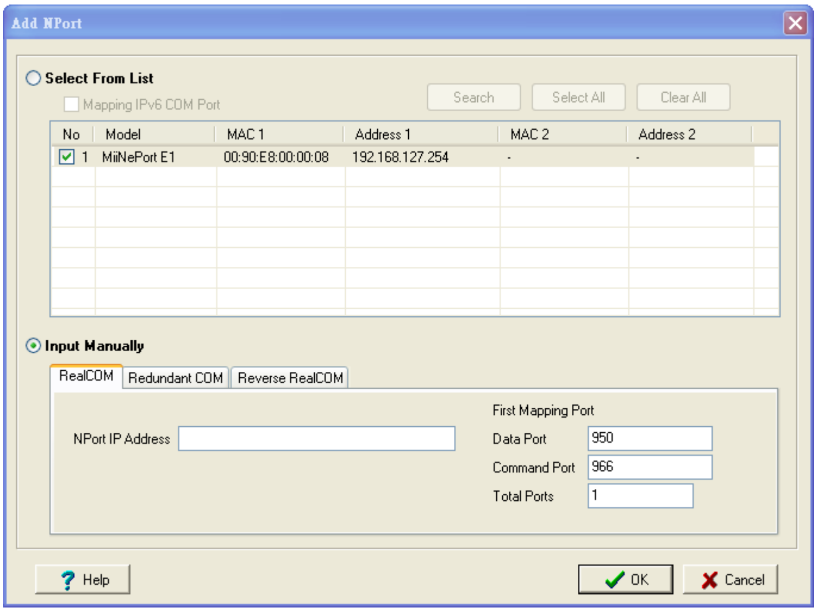
NPort IP Address

Data Port 950

Command Port 966

Total Ports 1

4. Alternatively, you can select **Input Manually** and then manually enter the MiiNePort E1 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.



Add NPort

○ Select From List

□ Mapping IPv6 COM Port

No	Model	MAC 1	Address 1	MAC 2	Address 2
☑ 1	MiiNePort E1	00:90:E8:00:00:08	192.168.127.254	-	-

● Input Manually

NPort IP Address:

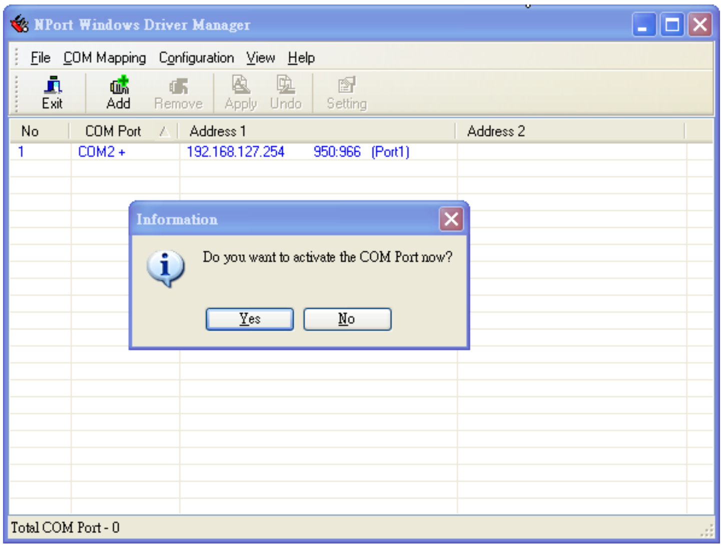
First Mapping Port

Data Port: 950

Command Port: 966

Total Ports: 1

5. 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.



NPort Windows Driver Manager

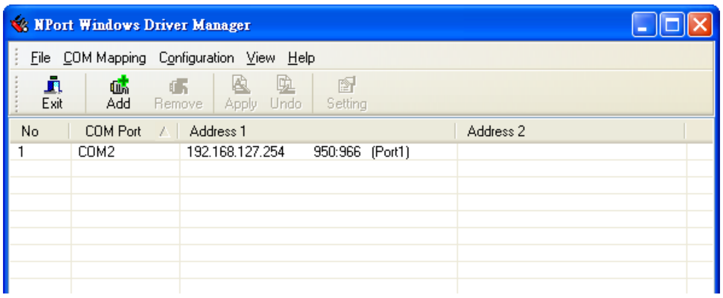
No	COM Port	Address 1	Address 2
1	COM2 +	192.168.127.254 950:966 (Port1)	

Information

Do you want to activate the COM Port now?

Total COM Port - 0

6. Ports that have been activated will appear in black.



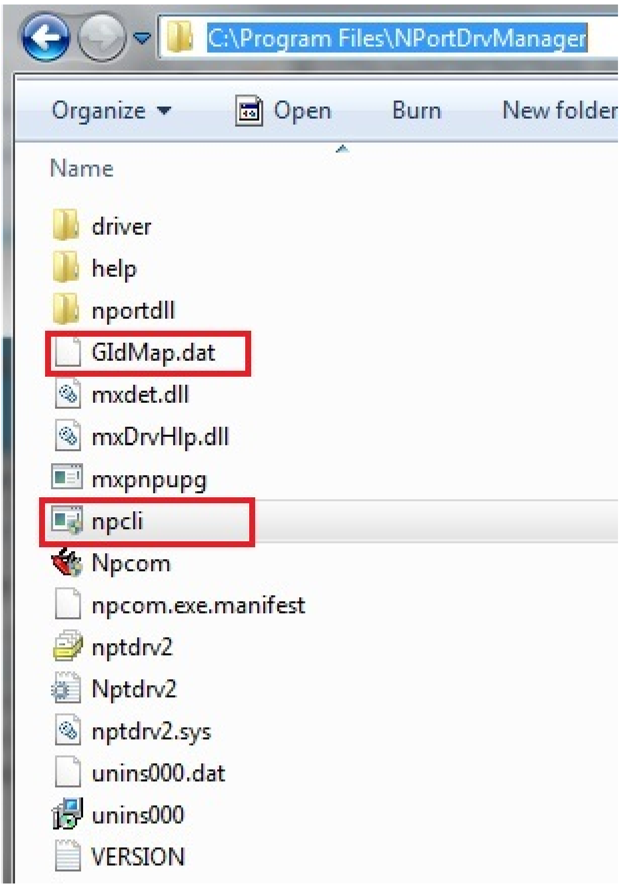
NPort Windows Driver Manager

No	COM Port	Address 1	Address 2
1	COM2	192.168.127.254 950:966 (Port1)	

Command Line Installation/Removal

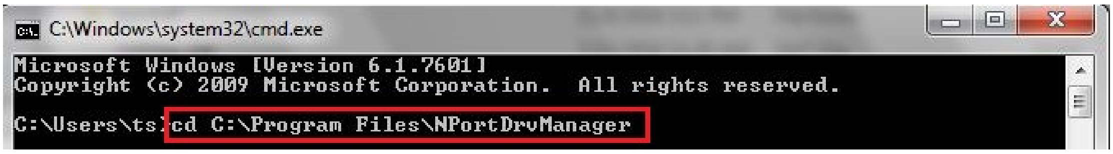
The NPort Windows Driver Manager v1.19 and above comes with a command-line script tool – ***npcli.exe*** for installation, removal of the driver and configuring NPort driver functions.

After successfully installing 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.



C:\Program Files\NPortDrvManager

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 placed the above two files.

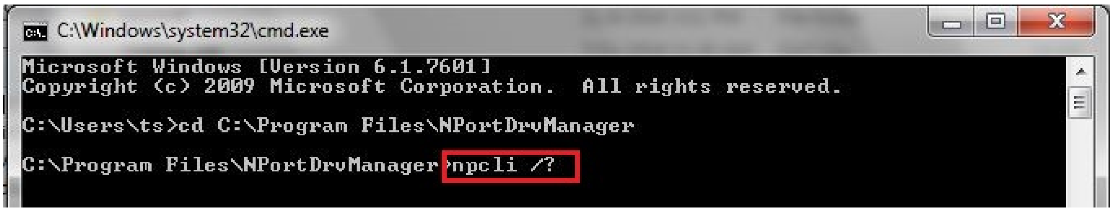


C:\Windows\system32\cmd.exe

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\ts>cd C:\Program Files\NPortDrvManager
```

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



C:\Windows\system32\cmd.exe

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\ts>cd C:\Program Files\NPortDrvManager

C:\Program Files\NPortDrvManager>npcli /?
```

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

```
------------------------------------------------------------------------------
NPort Command-Line Interface Ver2.0 Build 16052400
------------------------------------------------------------------------------
Usage:

  1. NPort Driver operation:
    npcli /driver [/install | /uninstall | /upgrade] [PATH_NAME]

  /install     Install specified driver to host.
  /uninstall   Uninstall current installed driver from host.
  /upgrade     Upgrade specified driver without modifying the mapped ports.
  PATH_NAME    Specify the installer file of NPort Driver Manager to install
              or upgrade.

  2. RealCOM port operation:
    npcli /driver /add IP_ADDR /port PORT_NO /com COM_NO [/txmode [hiperf |
         classical]] [/fifo [enable | disable]] [/flush [fast | normal]]
    npcli /driver /remove /com [COM_NO | all]

  /add         Add a RealCOM with a valid IP address (IP_ADDR).
  /port        Specify the NPort port number (PORT_NO) to add.
  /com         Specify the COM number to add or remove (COM_NO).
  /txmode      Set the TX mode as hi-performance (hiperf) or classical. The
              default is hiperf.
  /fifo        Set the FIFO as enable or disable. The default is enable.
  /flush       Set to enable fast flush(fast) or disable fast flush(normal).
              The default is fast.
  /remove      Remove specified COM number (COM_NO) or all RealCOM ports.

  3. NPort devices operation:
    npcli /devicd /search
    npcli /device /set ID /network [/ip IP_ADDR] [/mask SUBNET]
         [/gateway IP_ADDR] [/password CIPHER]
    npcli /device /apply ID [/password CIPHER]

  /search      Search the NPort and store the list to the memory.
  /set         Specify the ID to set. Users must specify one of the searched
              NPorts for further operations. The default is 1.
  /port        Specify the NPort port number (PORT_NO) to set.
  /password    Specify the password (CIPHER) if the NPort has one.
  /network     Set to change the network settings.
  /ip          Change the IP address (IP_ADDR) of NPort.
  /mask        Change the subnet mask (SUBNET) of NPort.
  /gateway     Change the IP address (IP_ADDR) of gateway.
  /apply       Specify the ID to save changes and restart the NPort.

  4. Examples
    npcli /driver /install D:\Users\drvmgr_setup_Ver1.19.0_Build_15122492
    npcli /driver /uninstall
    npcli /driver /add 192.168.127.254 /port 1 /com 3
    npcli /driver /add 192.168.127.254 /port 2 /com 4 /flush normal
    npcli /device /search
    npcli /device /set 1 /network /ip 192.168.10.7 /mask 255.255.255.0
```

```
          /password moxa
    npcli /device /apply 1

Note:
  Npcli.exe requires an administrator privilege to change device settings.
  It support only IPv4 and it must be run under Windows XP and later versions.
```

Installing Linux Real TTY Driver Files

1. Obtain the driver file from the included CD-ROM or the Moxa website, at http://www.moxa.com.
2. Log in to the console as a super user (root).
3. Execute **cd** / to go to the root directory.
4. Copy the driver file **npreal2xx.tg**z to the / directory.
5. Execute tar xvfz **npreal2xx.tgz** to extract all files into the system.
6. 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.
7. 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 E1 serial port to the system tty port.

Mapping TTY Ports

Make sure that you set the operation mode of the serial port of the MiiNePort E1 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 E1 serial port to the host tty ports. The syntax of **mxaddsvr** is as follows:

mxaddsvr [MiiNePort E1 IP Address] [Total Ports] ([Data port] [Cmd port])

The **mxaddsvr** command performs the following actions:

1. Modifies npreal2d.cf.
2. Creates tty ports in directory /dev with major and minor number configured in npreal2d.cf.
3. Restarts the driver.

Mapping tty ports automatically

To map tty ports automatically, you may 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.

Mapping tty ports manually

To map tty ports manually, you may 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.

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:

1. Modifies **npreal2d.cf**.
2. Removes the relevant tty ports in directory **/dev**.
3. Restart the driver.

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 from the list for deletion.

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. This program will perform the following actions:

1. Unload the driver.
2. Delete all files and directories in **/usr/lib/npreal2**.
3. Delete directory **/usr/lib/npreal2**.
4. Modify the system initializing script file.

Installing macOS Pseudo-TTY Drivers—NPortConnect

Basic Procedures

To map an NPort 5000 serial port to a macOS host’s tty port, follow these instructions:

1. Set up the NPort 5000. After verifying that the IP configuration works and that you can access the NPort 5000 (by using ping, Telnet, etc.), configure the desired serial port on the NPort 5000 to Real COM mode.

2. Install the macOS NPortConnect pseudo-tty driver files on the host

3. Map the NPort serial port to the host’s tty port

Hardware Setup

Before proceeding with the software installation, make sure you have completed the hardware installation. The default IP address for the NPort 5000 is 192.168.127.254.

NOTE After installing the hardware, you must configure the operating mode of the serial port on your NPort 5000 to Real COM or TCP server mode only.

Installing macOS Pseudo-TTY Drivers

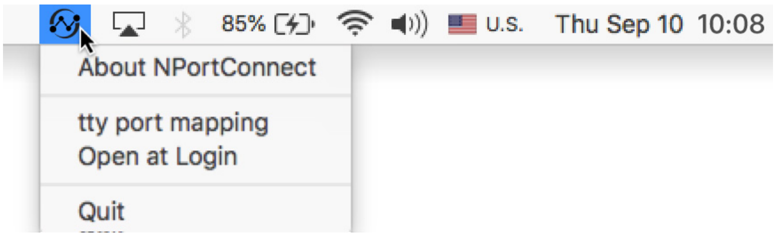
NOTE For the newest information, please refer to readme.txt on macOS NPortConnect pseudo-tty driver

Obtain the driver file from Moxa's website, at http://www.moxa.com. You may find it in the Resource section under the product page.

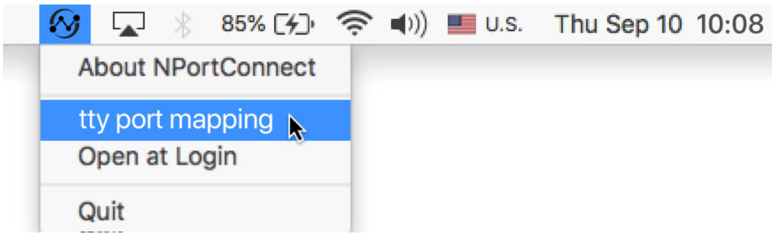
Follow the installation steps under "Install NPortConnect".

Mapping TTY Ports

Make sure that you set the operation mode of the desired NPort 5000 serial port to Real COM or TCP server mode. You may launch tty port mapping utility by clicking the NPortConnect icon on your macOS title bar

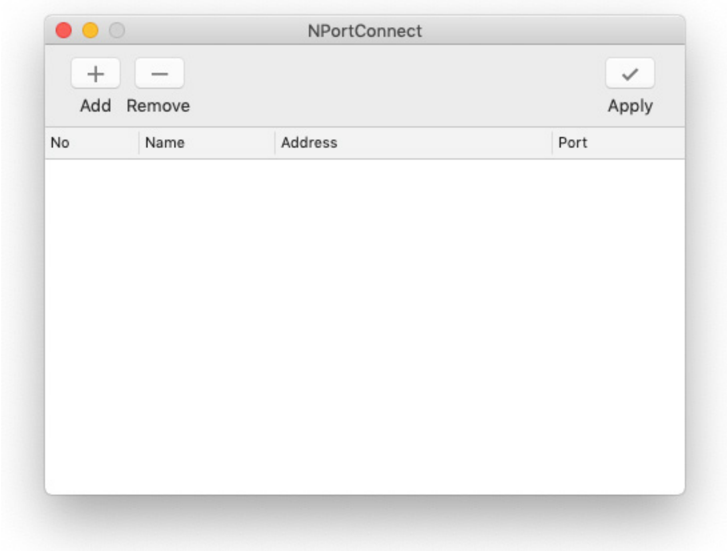


And, by clicking **"tty port mapping"** to bring up the utility.

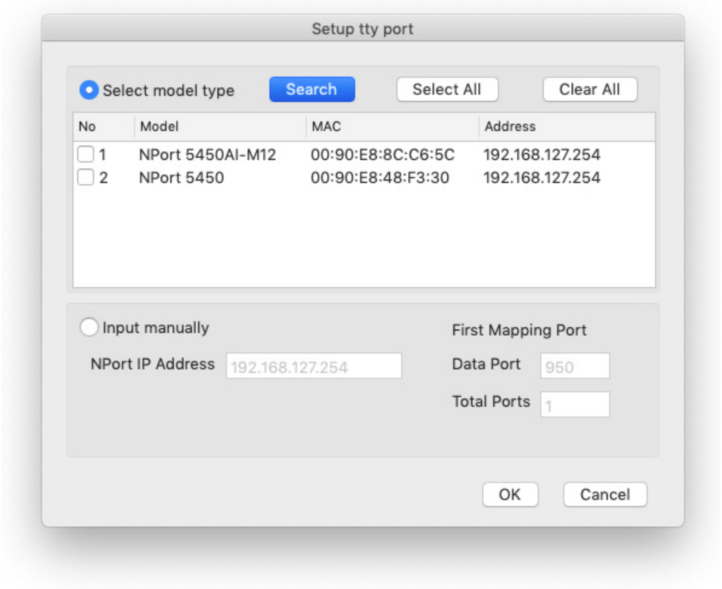


Mapping TTY Ports Automatically

You may add or remove the pseudo-tty port on the tool bar.



Click **Add** to add a new pseudo-tty port.



● Select model type

No	Model	MAC	Address
□ 1	NPort 5450AI-M12	00:90:E8:8C:C6:5C	192.168.127.254
□ 2	NPort 5450	00:90:E8:48:F3:30	192.168.127.254

○ Input manually

NPort IP Address: 192.168.127.254

First Mapping Port

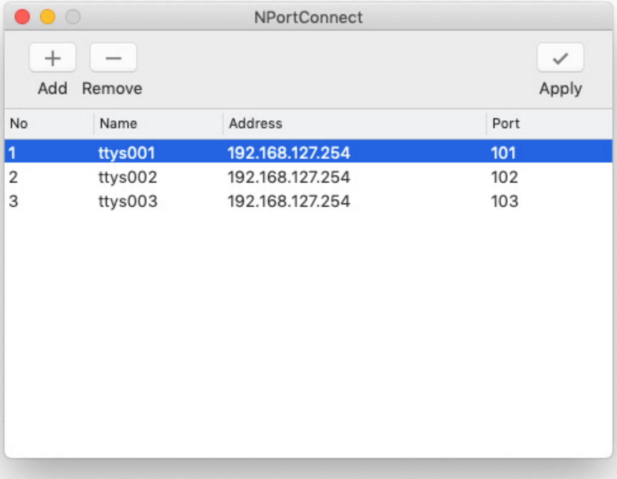
Data Port: 950

Total Ports: 1

If your NPort is up and running, the NPort(s) should appear on your search result after clicking the **Search** button. Check the NPort that you wish to map. The tty port for all the physical serial ports will be set up automatically.

Mapping TTY Ports Manually

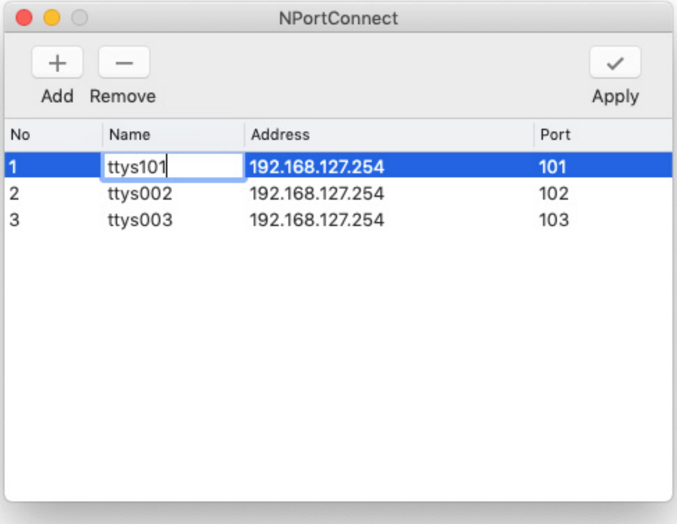
If your NPort has already an assigned IP, or you would like to assign a data port number and the total number of ports manually, select **Input manually** and then input the IP address, data port number, and the total number of ports. The data port number will be created sequentially, starting from the port number you input.



No	Name	Address	Port
1	ttys001	192.168.127.254	101
2	ttys002	192.168.127.254	102
3	ttys003	192.168.127.254	103

Editing Mapped TTY Ports

You may edit the name and port number after the setup by clicking the field to modify.

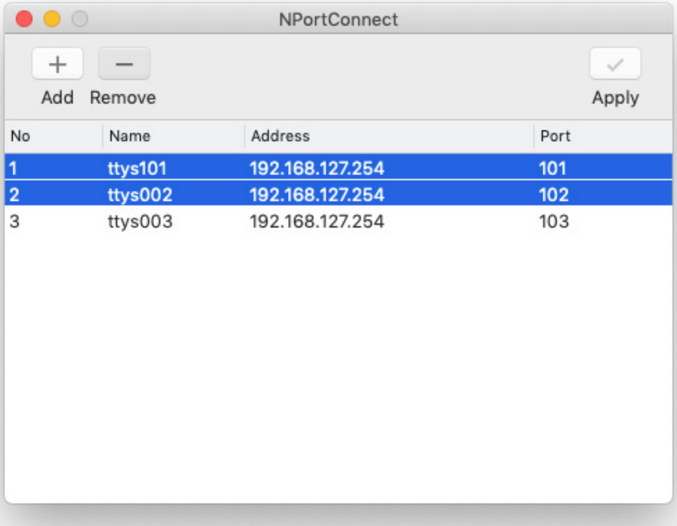


No	Name	Address	Port
1	ttys101	192.168.127.254	101
2	ttys002	192.168.127.254	102
3	ttys003	192.168.127.254	103

When all the details are confirmed, click **Apply** to complete the setup.

Removing Mapped TTY Ports

When you need to remove the tty ports that have already been set up, select the ports and then click **Remove**.



No	Name	Address	Port
1	ttys101	192.168.127.254	101
2	ttys002	192.168.127.254	102
3	ttys003	192.168.127.254	103

Uninstalling NPortConnect

Execute sudo bash /Library/NPortConnect/uninstall.sh in the terminal to uninstall the driver. If you need to install the driver again, it is suggested to reboot your Mac before installation.

7

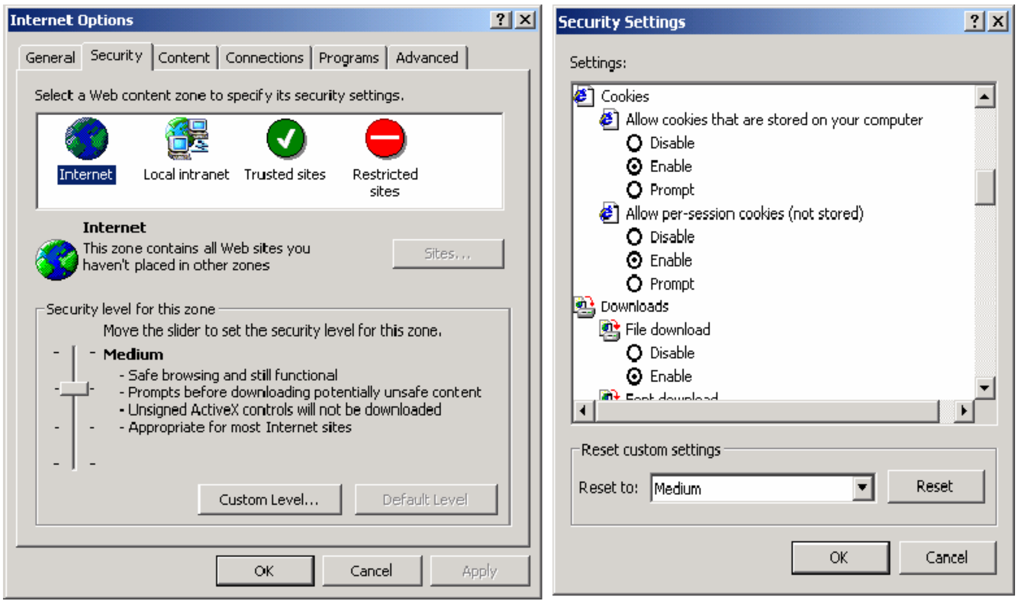
Web Console Configuration

The web console is the most user-friendly way to configure your MiiNePort E1 Series module. This chapter introduces the web console function groups and function definitions.

The following topics are covered in this chapter:

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.)



Internet Options
General
Security
Content
Connections
Programs
Advanced
Select a Web content zone to specify its security settings.
Internet
Local intranet
Trusted sites
Restricted sites
Internet
This zone contains all Web sites you haven't placed in other zones
Sites...
Security level for this zone
Move the slider to set the security level for this zone.
Medium
- Safe browsing and still functional
- Prompts before downloading potentially unsafe content
- Unsigned ActiveX controls will not be downloaded
- Appropriate for most Internet sites
Custom Level...
Default Level
OK
Cancel
Apply
Security Settings
Settings:
Reset custom settings
Reset to:
Medium
Reset
OK
Cancel

Cookies
- Allow cookies that are stored on your computer
  - ○ Disable
  - ● Enable
  - ○ Prompt
- Allow per-session cookies (not stored)
  - ○ Disable
  - ● Enable
  - ○ Prompt

Downloads
- File download
  - ○ Disable
  - ● Enable
- Font download

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**.

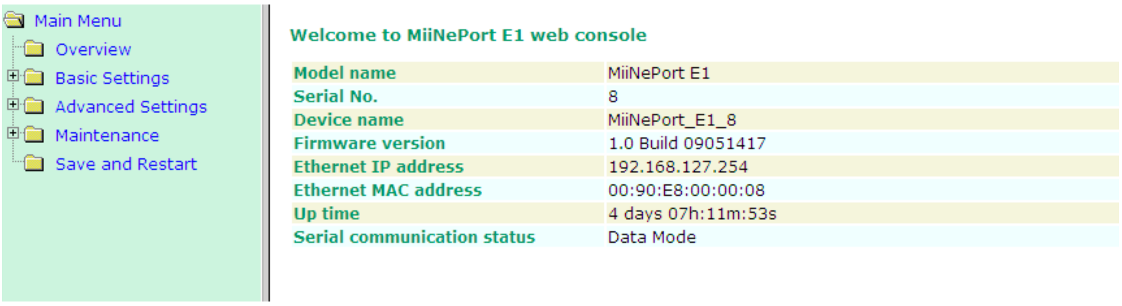
ATTENTION

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 E1 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 E1 module.



Welcome to MiiNePort E1 web console

Model name	MiiNePort E1
Serial No.	8
Device name	MiiNePort_E1_8
Firmware version	1.0 Build 09051417
Ethernet IP address	192.168.127.254
Ethernet MAC address	00:90:E8:00:00:08
Up time	4 days 07h:11m:53s
Serial communication status	Data Mode

Web Console Fundamentals

On the web console, the left panel is the navigation panel and 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.

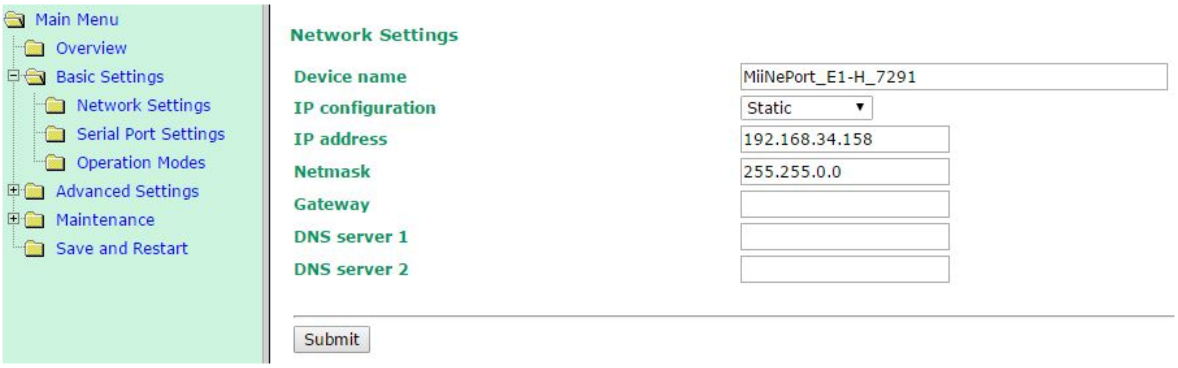
Configuration changes can then be made in the main window. For example, if you click on **Basic Settings** → **Network Settings** in the navigation panel, the main window will show a page of network settings that you can configure.

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.

Basic Settings

Network Settings



Network Settings

Device name: MiiNePort_E1-H_7291
IP configuration: Static
IP address: 192.168.34.158
Netmask: 255.255.0.0
Gateway:
DNS server 1:
DNS server 2:

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.

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.

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.

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.

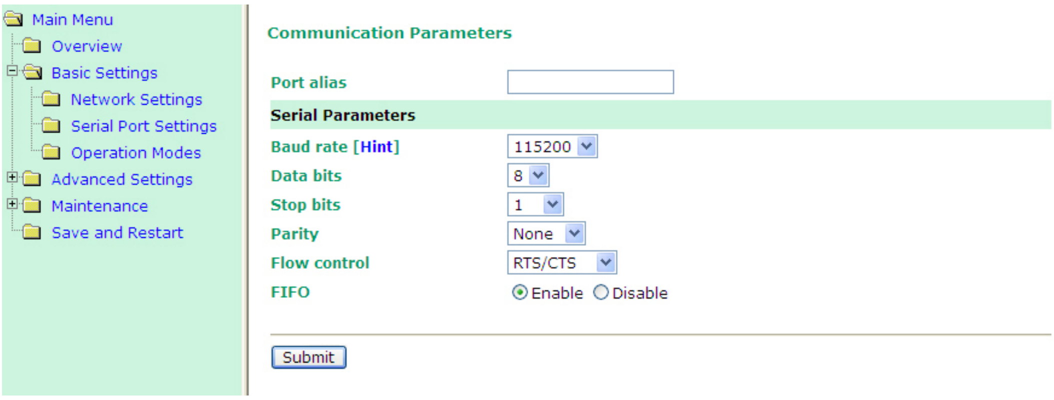
DNS server 1 / DNS server 2

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

Domain Name System (DNS) is how Internet domain names are identified and translated into IP addresses. A domain name is an alphanumeric name, such as moxa.com, that it 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 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 may be specified, DNS server 1 and DNS server 2. DNS server 2 is included for use when DNS sever 1 is unavailable.

Serial Port Settings



Communication Parameters

Port alias

Serial Parameters

Baud rate [Hint] 115200

Data bits 8

Stop bits 1

Parity None

Flow control RTS/CTS

FIFO ● Enable ○ Disable

Port Alias

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

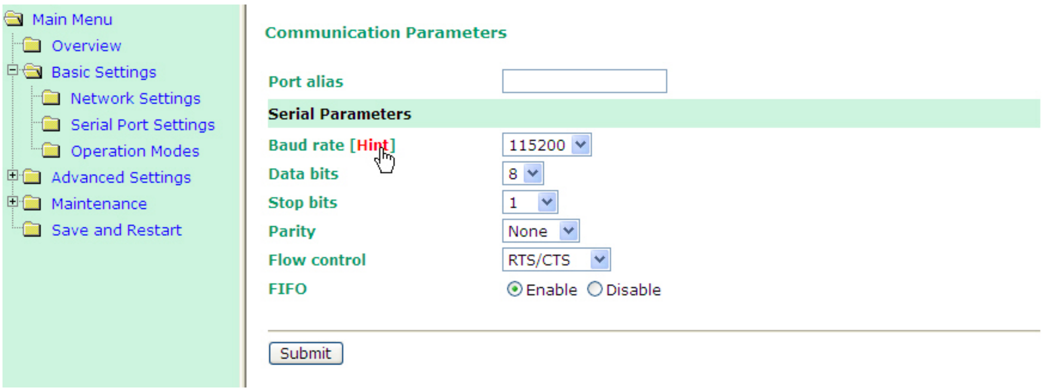
This function is designed for future use. You may enter a string to help in the module's serial port from other serial ports.

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.

Baud Rate

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



Communication Parameters

Port alias

Serial Parameters

Baud rate [Hint] 115200

Data bits 8

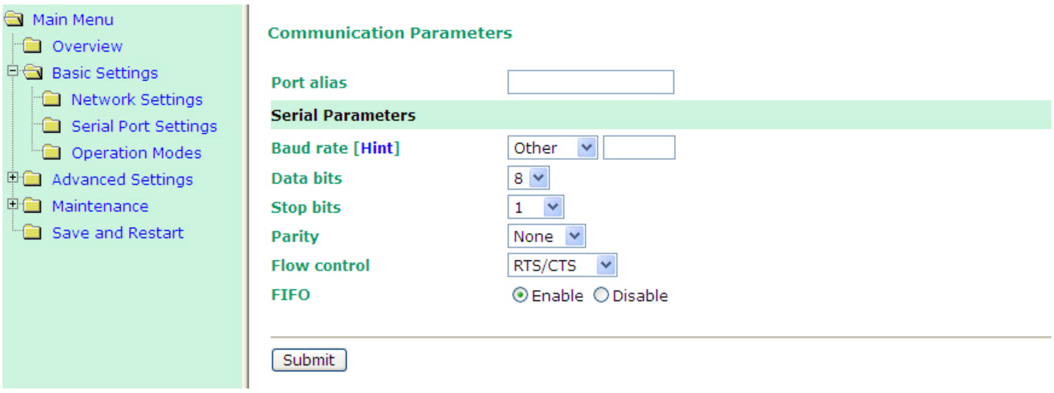
Stop bits 1

Parity None

Flow control RTS/CTS

FIFO ● Enable ○ Disable

MiiNePort E1 supports the **Any Baud Rate (non-standard baud rate)** feature. If your baud rate is not listed, select **Other** from the drop-down list and type the baud rate in the input box. MiiNePort E1 will use the closest baud rate we can support.



Communication Parameters

Port alias

Serial Parameters

Baud rate [Hint]: Other

Data bits: 8

Stop bits: 1

Parity: None

Flow control: RTS/CTS

FIFO: ● Enable ○ Disable

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

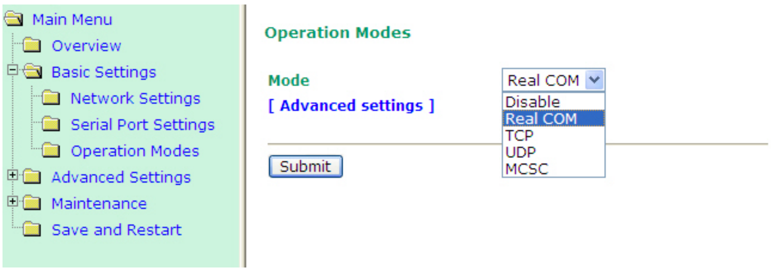
Setting	Factory Default	Necessity
None, RTS/CTS, DTR/DSR, XON/XOFF	RTS/CTS	Required

FIFO

Setting	Factory Default	Necessity
Enable, Disable	Enable	Required

Each module's serial port provides a 16-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.

Operation Modes



Operation Modes

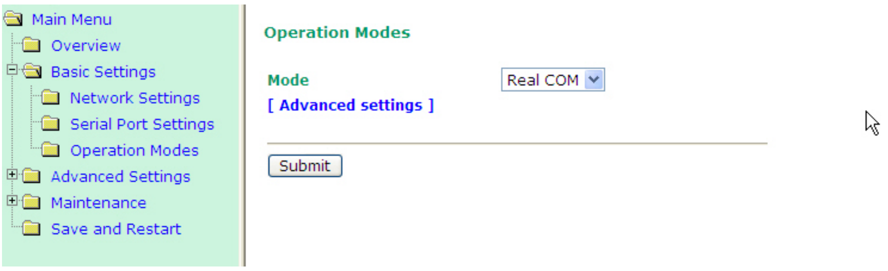
Mode: Real COM

[Advanced settings]

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

To save time, in each Operation Mode configuration page we only show the settings that are used most often. You can check and configure other functions by clicking **Advanced Settings** in each operation mode configuration page. However, your application should work properly without making any changes in Advanced Settings.

Real COM Mode



Operation Modes

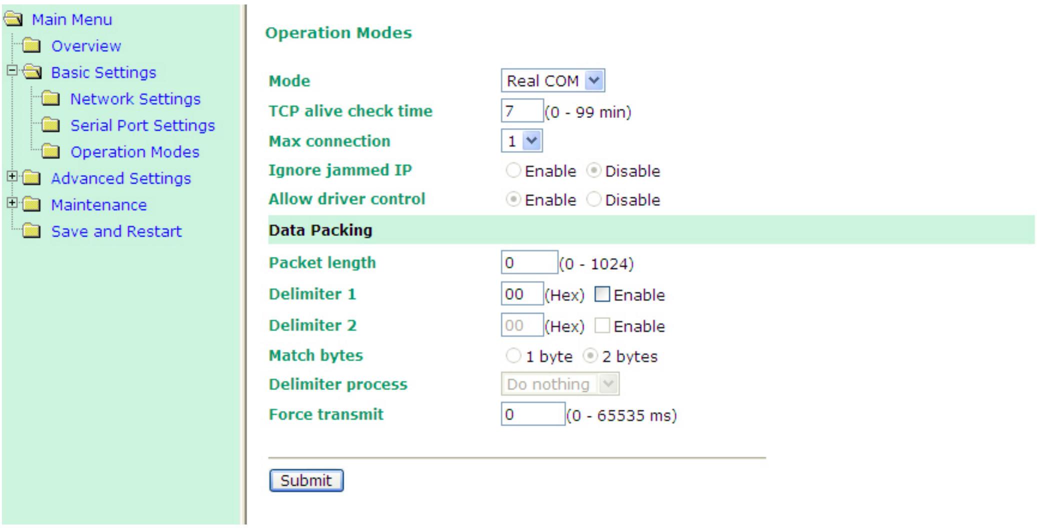
Mode: Real COM

[Advanced settings]

ATTENTION

To use Real COM mode, refer to Chapter 6: Utility Console and Driver Installation to install the Real COM driver on Windows or Linux.

Advanced Settings for Real COM Mode



Operation Modes

Mode: Real COM

TCP alive check time: 7 (0 - 99 min)

Max connection: 1

Ignore jammed IP: ○ Enable ● Disable

Allow driver control: ● Enable ○ Disable

Data Packing

Packet length: 0 (0 - 1024)

Delimiter 1: 00 (Hex) □ Enable

Delimiter 2: 00 (Hex) □ Enable

Match bytes: ○ 1 byte ● 2 bytes

Delimiter process: Do nothing

Force transmit: 0 (0 - 65535 ms)

TCP alive check time

Setting	Factory Default	Necessity
0 to 99 min	7 min	Optional

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

1 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 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 1 host's Real COM driver to open a connection to the module's serial port.

Max connection 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 E1 from the Ethernet interface.

ATTENTION

When Max connection is greater than 1, the MiiNePort E1 module will use a multiple connection application (i.e., 2 to 4 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 greater than 1

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 valid when the Max connection is greater than 1.

Allow driver control

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

NOTE Allow drive control is only valid when Max connection is greater than 1.

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.

Data Packing

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. At the default of 0 for packet length, no maximum amount is 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 character or 2-character sequence which will act as a marker to control packing of serial data. By default, no delimiter characters are 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 2-character sequence. When the delimiter is received, the module will pack up the data in its buffer and send it through the Ethernet port.

Use Delimiter 1 to define the first delimiter character in hex. If only one delimiter character will be used, Delimiter 2 should be set to “0”. If the delimiter will be 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 not only controlled by the delimiter; it 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. Also, if the module will also pack data for network transmission if the next byte of data is not received within the Force transmit time.

Match bytes

Setting	Factory Default	Necessity
1 byte to 2 bytes	2 bytes (only effective when Delimiter is enabled)	Optional

1 byte: MiiNePort E1 packs the serial data and sends out TCP/IP packets once it receives either one of Delimiter 1 and Delimiter 2.

2 bytes: MiiNePort E1 only packs the serial data and sends out TCP/IP packets once it receives both of Delimiter 1 and Delimiter 2. And the receiving order should be first Delimiter 1 then Delimiter 2.

Delimiter process

Setting	Factory Default	Necessity
Do Nothing, Delimiter+1, Delimiter+2	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: Data in the buffer will be transmitted after 1 additional byte is received following the delimiter.

Delimiter + 2: Data in the buffer will be transmitted after 2 additional bytes are received following the delimiter.

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 as soon as there is no serial communication for the specified time.

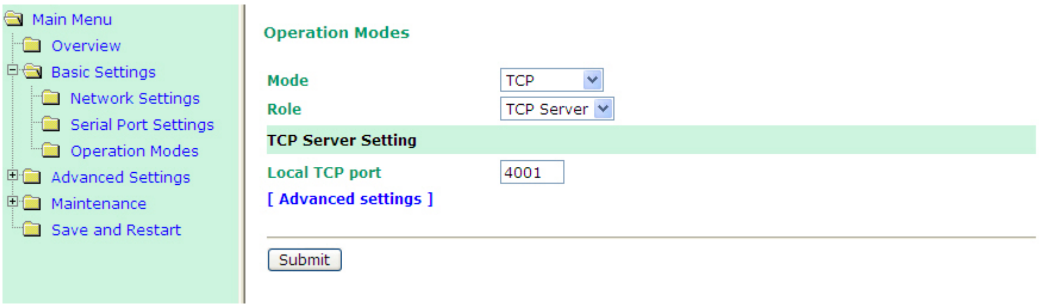
The optimal force transmit time depends on your application, but it should be larger than one character interval within the specified baud rate to have any effect. For example, assume that the serial port is set to 1200 bps, 8 data bits, 1 stop bit, and no parity. In this case, the total number of bits 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 no delimiter characters or Force transmit time were specified at all.

TCP Server Mode

To select TCP Server Mode, first select TCP as Mode and then select TCP Server as Role.



Operation Modes

Mode: TCP

Role: TCP Server

TCP Server Setting

Local TCP port: 4001

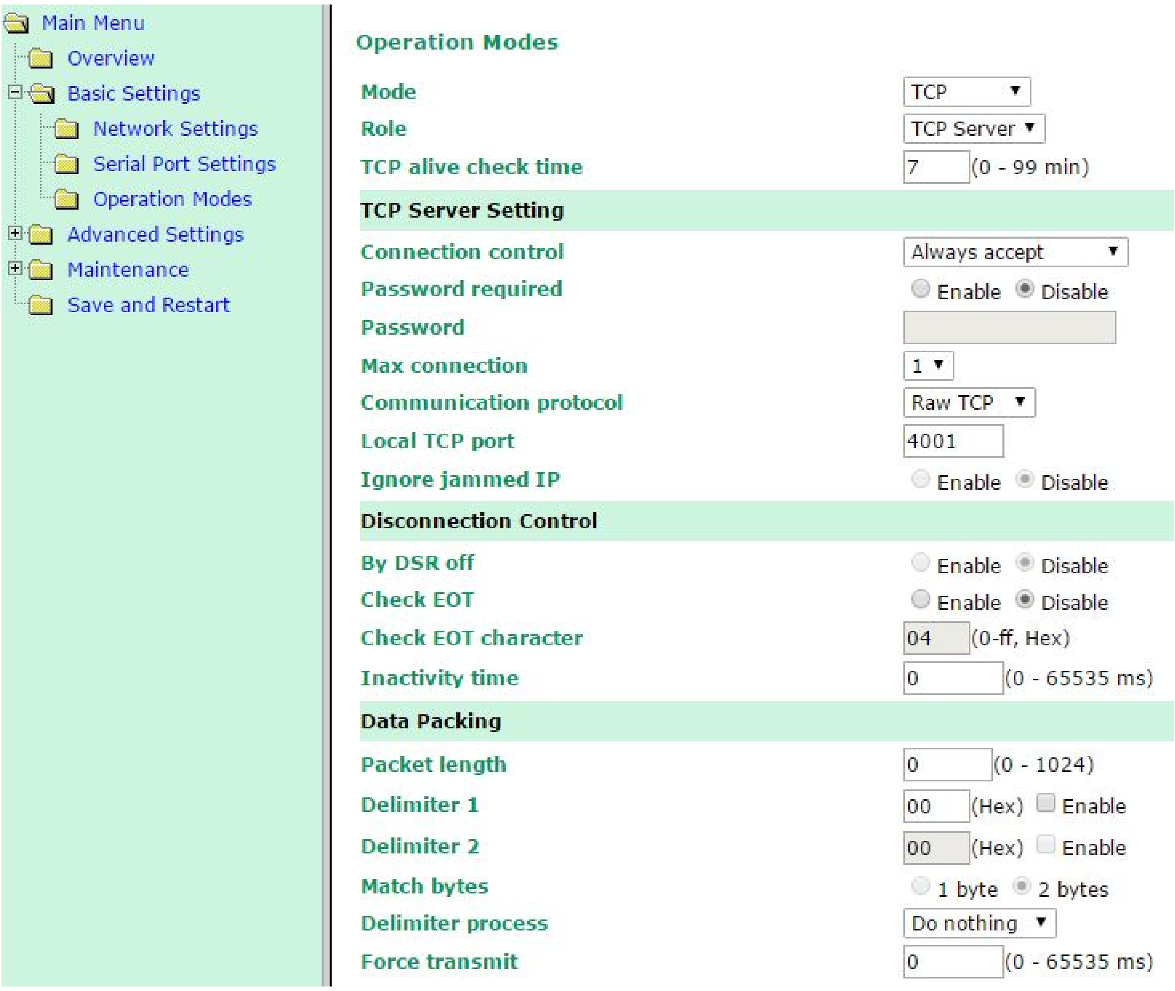
[Advanced settings]

Local TCP port

Setting	Factory Default	Necessity
1 to 65535	4001	Required

Use this field to indicate the TCP port that the module will use to listen to connections, and that other devices must use to contact the module. To avoid conflicts with well known TCP ports, the default is set to 4001.

Advanced Settings for TCP Server Mode



Operation Modes

Mode	TCP ▼
Role	TCP Server ▼
TCP alive check time	7 (0 - 99 min)
TCP Server Setting	
Connection control	Always accept ▼
Password required	○ Enable ● Disable
Password	
Max connection	1 ▼
Communication protocol	Raw TCP ▼
Local TCP port	4001
Ignore jammed IP	○ Enable ● Disable
Disconnection Control	
By DSR off	○ Enable ● Disable
Check EOT	○ Enable ● Disable
Check EOT character	04 (0-ff, Hex)
Inactivity time	0 (0 - 65535 ms)
Data Packing	
Packet length	0 (0 - 1024)
Delimiter 1	00 (Hex) ☐ Enable
Delimiter 2	00 (Hex) ☐ Enable
Match bytes	○ 1 byte ● 2 bytes
Delimiter process	Do nothing ▼
Force transmit	0 (0 - 65535 ms)

TCP alive check time

Setting	Factory Default	Necessity
0 to 99 min	7 min	Optional

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

1 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.

TCP Server Setting

Connection Control

Setting	Factory Default	Necessity
Always accept, Accept when DSR on	Always accept	Optional

Always accept: Always accept TCP connection from host.

Accept with DSR on: Accept TCP connection from host when the DSR is pulled On.

Default is **Always accept**.

ATTENTION

When setting the Connection Control or Disconnection Control by DSR signal in TCP Server mode, you must configure pin 8 to transmit the DSR signal.

Password required

Setting	Factory Default	Necessity

Enable, Disable	Disable	Optional

Enable: When host is requesting a TCP connection to the MiiNePort E1 in TCP Server mode, the host will be required to enter a password.

Disable: The host can establish a TCP connection to the MiiNePort E1 without requiring a password.

Default is **Disable**.

Password

Setting	Factory Default	Necessity
1-15 English or number characters without space in between	None	Optional

After you enable the Password required function, you must set a set of passwords composed of 1 to 15 English or number characters, without spaces.

Max connection

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

Maximum 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 1 host’s Real COM driver to open a connection to the module’s serial port.

Max connection 2 to 4: The module will allow requests from between 2 and 4 different hosts to open the module’s serial port at the same time. When multiple hosts’ Real COM drivers open the serial port at the same time, the COM driver only provides a pure data tunnel without control ability.

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 the data back to the driver on the host. Data will be sent first-in-first-out when data comes into the module from the Ethernet interface.

Communication Protocol

Setting	Factory Default	Necessity
Raw TCP, RFC 2217	Raw TCP	Required

RAW TCP: Standard TCP/IP protocol.

RFC 2217 11: Refer **Chapter 3: Choosing the Proper Operation Mode** for more details of RFC 2217 mode.

NOTE Be sure you install the RFC 2217 driver before choosing RFC 2217.

Default is **RAW TCP**.

Local TCP port

Setting	Factory Default	Necessity
1 to 65535	4001	Required

Use this field to indicate the TCP port that the module will use to listen to connections, and that other devices must use to contact the module. To avoid conflicts with well known TCP ports, the default is set to 4001.

Ignore jammed IP

Setting	Factory Default	Necessity

Enable, Disable	Disable	Required when Max connection is greater than 1

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

This option determines how the port will proceed if multiple hosts are connected and one or more of the hosts stop 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.

MiiNePort E1 can actively disconnect the connection established by the host.

Disconnection Control

By DSR off

Setting	Factory Default	Necessity
Enable, Disable	Disable	Required

Enable: MiiNePort E1 will actively cut off the connection when DSR is pulled Off.

Disable: Disable foresaid function

Default is **Disable**.

ATTENTION

When set the Connection Control or Disconnection Control by DSR signal, configure Pin 8 as DSR function is required.

Check EOT

Setting	Factory Default	Necessity
Enable, Disable	Disable	Required

Enable: Your device main system can send an serial EOT character to MiiNePort E1 to stop current TCP connection.

Disable: Disable this function.

Default is **Disable**.

Check EOT character

Setting	Factory Default	Necessity
0 to FF, Hex	04	Optional

Set up EOT character.

Inactivity time

Setting	Factory Default	Necessity
0 to 65535 ms	0 ms	Optional

0 ms: The TCP connection is not closed due to an idle serial line.

1-65535 ms: The module automatically closes the TCP connection if there is no serial data activity for the given time. After the connection is closed, the module starts listening for another host's TCP connection.

This parameter defines the maintenance status as Closed or Listen on 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 Inactivity time is set to 0, the current TCP connection is maintained until there is a connection close request. Even with Inactivity time is disabled, the module will still check the connection status between itself and the remote host by sending "keep alive" packets periodically. If the remote host does not respond to the packet, it assumes that the connection was closed down unintentionally. The module will then force the existing TCP connection to close.

ATTENTION

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

Data Packing

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. At the default of 0 for packet length, no maximum amount is 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 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 character or 2-character sequence which will act as a marker to control packing of serial data. By default, no delimiter characters are 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 2-character sequence. When the delimiter is received, the module will pack up the data in its buffer and send it through the Ethernet port.

Use Delimiter 1 to define the first delimiter character in hex. If only one delimiter character will be used, Delimiter 2 should be set to "0". If the delimiter will be 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 not only controlled by the delimiter; it 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. The module will also pack data for network transmissions if the next byte of data is not received within the Force transmit time.

Match bytes

Setting	Factory Default	Necessity
1 byte to 2 bytes	2 bytes (only effective when Delimiter is enabled)	Optional

1 byte: MiiNePort E1 packs the serial data and sends out TCP/IP packets once it receives either one of Delimiter 1 and Delimiter 2.

2 bytes: MiiNePort E1 only packs the serial data and sends out TCP/IP packets once it receives both of Delimiter 1 and Delimiter 2. And the receiving order should be first Delimiter 1 then Delimiter 2.

Delimiter process

Setting	Factory Default	Necessity
Do Nothing, Delimiter+1, Delimiter+2	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: Data in the buffer will be transmitted after 1 additional byte is received following the delimiter.

Delimiter + 2: Data in the buffer will be transmitted after 2 additional bytes are received following the delimiter.

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 as soon as there has been no serial communication for the specified time.

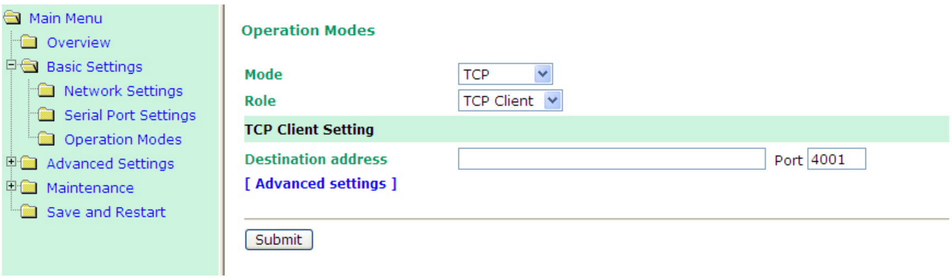
The optimal force transmit time depends on your application, but it should be larger than one character interval within the specified baud rate to have any effect. For example, assume that the serial port is set to 1200 bps, 8 data bits, 1 stop bit, and no parity. In this case, the total number of bits 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 no delimiter characters or Force transmit time were specified at all.

TCP Client Mode

To select TCP Client Mode, first select TCP Mode and then set Role to TCP Client.



Operation Modes

Mode: TCP

Role: TCP Client

TCP Client Setting

Destination address: Port 4001

[Advanced settings]

Destination address

Setting	Factory Default	Necessity
IP address or domain name (e.g., 192.168.1.1)	None	Required

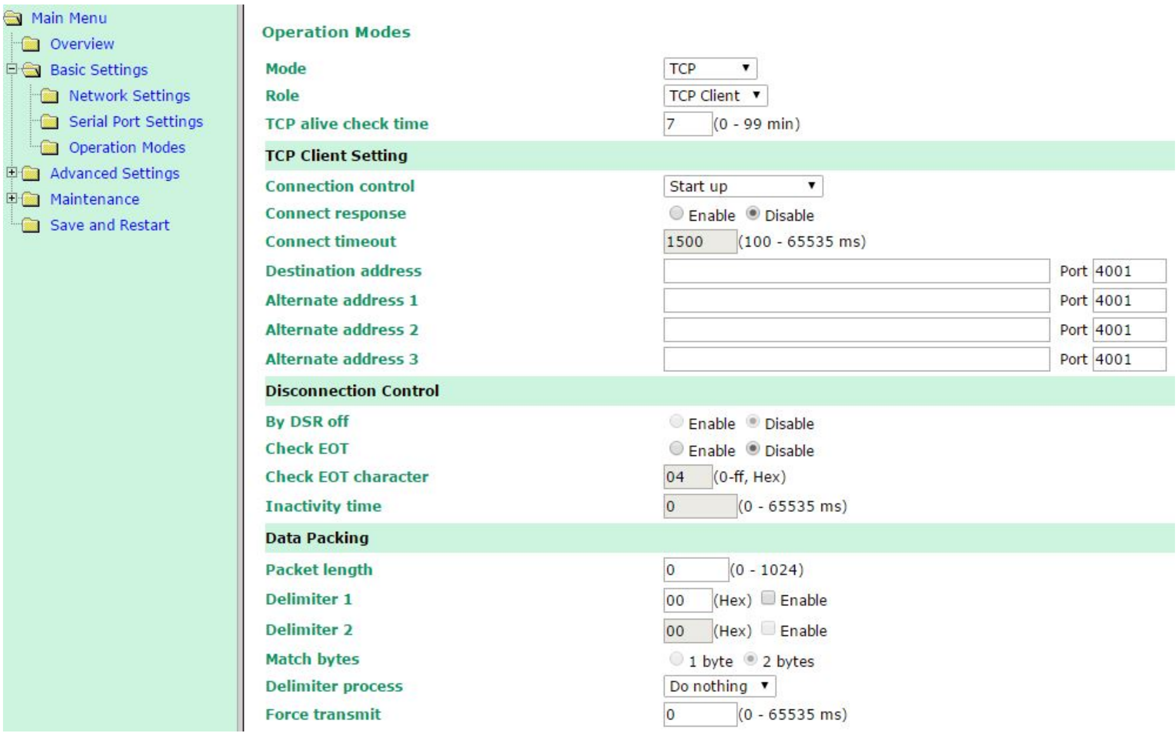
Use this field to specify the remote hosts that the module will connect to.

TCP port

Setting	Factory Default	Necessity
1 to 65535	4001	Required

Use this field to indicate the TCP port of the destination host that will be connected to by MiiNePort E1. To avoid conflicts with well known TCP ports, the default is set to 4001.

Advanced Settings for TCP Client Mode



Operation Modes

Mode	TCP ▾
Role	TCP Client ▾
TCP alive check time	7 (0 - 99 min)
TCP Client Setting	
Connection control	Start up ▾
Connect response	○ Enable ● Disable
Connect timeout	1500 (100 - 65535 ms)
Destination address	Port 4001
Alternate address 1	Port 4001
Alternate address 2	Port 4001
Alternate address 3	Port 4001
Disconnection Control	
By DSR off	○ Enable ● Disable
Check EOT	○ Enable ● Disable
Check EOT character	04 (0-ff, Hex)
Inactivity time	0 (0 - 65535 ms)
Data Packing	
Packet length	0 (0 - 1024)
Delimiter 1	00 (Hex) □ Enable
Delimiter 2	00 (Hex) □ Enable
Match bytes	○ 1 byte ● 2 bytes
Delimiter process	Do nothing ▾
Force transmit	0 (0 - 65535 ms)

TCP alive check time

Setting	Factory Default	Necessity
0 to 99 min	7 min	Optional

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

1 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.

TCP Client Setting

Connection control

Setting	Factory Default	Necessity
Any character, Manual connection, Start up, With DSR on	Start up	Optional

Any character: The module will attempt to establish a TCP connection as soon as the module starts receiving serial data from your device’s main system.

Manual connection: Your device main system can request MiiNePort E1 establish a connection by command. The command should begin with “C” and end with “LF” or “CR-LF”, and IP address/port in the middle. For example: C192.168.32.221/4001CR-LF.

Start up: The module will attempt to establish a TCP connection as soon as it is powered on.

With DSR on: The module will attempt to establish a TCP connection with the DSR being pulled On.

ATTENTION

When setting the Connection Control or Disconnection Control by DSR signal of TCP Client mode, you must first configure pin 8 to transmit the DSR signal.

Connect response

Setting	Factory Default	Necessity
Enable, Disable	Disable	Optional

Enable: MiiNePort E1 will send a one-byte message back to device main system reporting the connection status; "N" means connection failed to be established. "C" means connection successfully established. "D" means the connection lost.

Disable: Disable above function.

Default is **Disable**.

Connect Timeout

Setting	Factory Default	Necessity
100 to 65535 ms	1500 ms	Required

NOTE Connect timeout is only valid with manual connections.

Connect Timeout is used to limit the time the MiiNePort E1 tries to establish a connection to the Destination address, when configured in TCP Client mode,. When the time is reached, the MiiNePort E1 will abort the attempt.

The MiiNePort E1 can actively disconnect the connection established by the host.

Destination address

Setting	Factory Default	Necessity
IP address or domain name (e.g., 192.168.1.1)	None	Required

Use this field to specify which remote hosts the module will connect to.

ATTENTION

The Destination address parameter can accept domain name addresses in place of IP addresses if a DNS server has been configured. For some applications, you may need to send the data actively to a remote destination's domain name address.

Disconnection Control

By DSR off

Setting	Factory Default	Necessity
Enable, Disable	Disable	Required

Enable: MiiNePort E1 will actively cut off the connection when DSR is pulled Off.

Disable: Disable foresaid function.

Default is **Disable**.

Check EOT

Setting	Factory Default	Necessity
Enable, Disable	Disable	Required

Enable: Your device's main system can send an EOT character to the MiiNePort E1 to stop the current TCP connection.

Disable: Disable this function.

Default is **Disable**.

Check EOT character

Setting	Factory Default	Necessity
0 to FF, Hex	04	Optional

Set up EOT character.

Inactivity time

Setting	Factory Default	Necessity
0 to 65535 ms	0 ms	Optional

0 ms: The TCP connection is not closed due to an idle serial line.

1-65535 ms: The module automatically closes the TCP connection if there is no serial data activity for the given time. After the connection is closed, the module starts listening for another host's TCP connection.

This parameter defines the maintenance status as Closed or Listen on 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 Inactivity time is set to 0, the current TCP connection is maintained until there is a connection close request. Even with Inactivity time is disabled, the module will still check the connection status between itself and the remote host by sending "keep alive" packets periodically. If the remote host does not respond to the packet, it assumes that the connection was closed down unintentionally. The module will then force the existing TCP connection to close.

ATTENTION

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

Data Packing

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. At the default of 0 for packet length, no maximum amount is 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 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 character or 2-character sequence which will act as a marker to control packing of serial data. By default, no delimiter characters are 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 2-character sequence. When the delimiter is received, the module will pack up the data in its buffer and send it through the Ethernet port.

Use Delimiter 1 to define the first delimiter character in hex. If only one delimiter character will be used, Delimiter 2 should be set to "0". If the delimiter will be 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 not only controlled by the delimiter; it 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. Also, if the module will also pack data for network transmission if the next byte of data is not received within the Force transmit time.

Match bytes

Setting	Factory Default	Necessity
1 byte to 2 bytes	2 bytes (only effective when Delimiter is enabled)	Optional

1 byte: MiiNePort E1 packs the serial data and sends out TCP/IP packets once it receive either one of Delimiter 1 and Delimiter 2.

2 bytes: MiiNePort E1 only packs the serial data and sends out TCP/IP packets once it receive both of Delimiter 1 and Delimiter 2. And the receiving order should be first Delimiter 1 then Delimiter 2.

Delimiter process

Setting	Factory Default	Necessity
Do Nothing, Delimiter+1, Delimiter+2	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: Data in the buffer will be transmitted after 1 additional byte is received following the delimiter.

Delimiter + 2: Data in the buffer will be transmitted after 2 additional bytes are received following the delimiter.

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 packed 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 as soon as there has been no serial communication for the specified time.

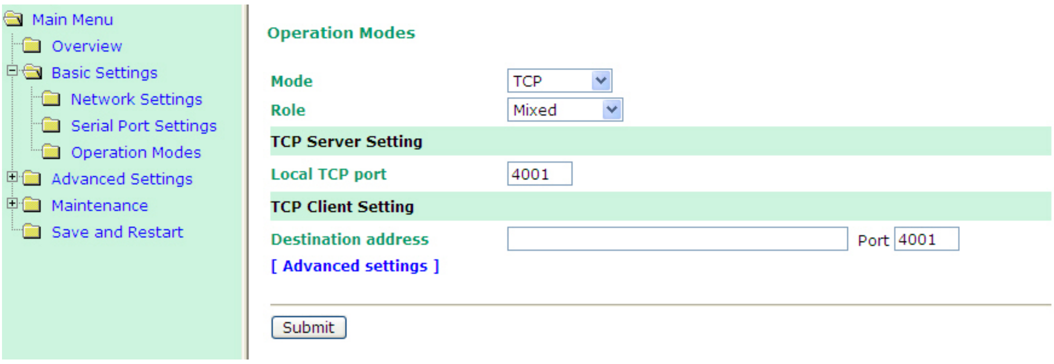
The optimal force transmit time depends on your application, but it should be larger than one character interval within the specified baud rate to have any effect. For example, assume that the serial port is set to 1200 bps, 8 data bits, 1 stop bit, and no parity. In this case, the total number of bits 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 no delimiter characters or Force transmit time were specified at all.

TCP Mixed Mode

To select TCP Mixed Mode, select TCP as Mode first then set Role to TCP Mixed.

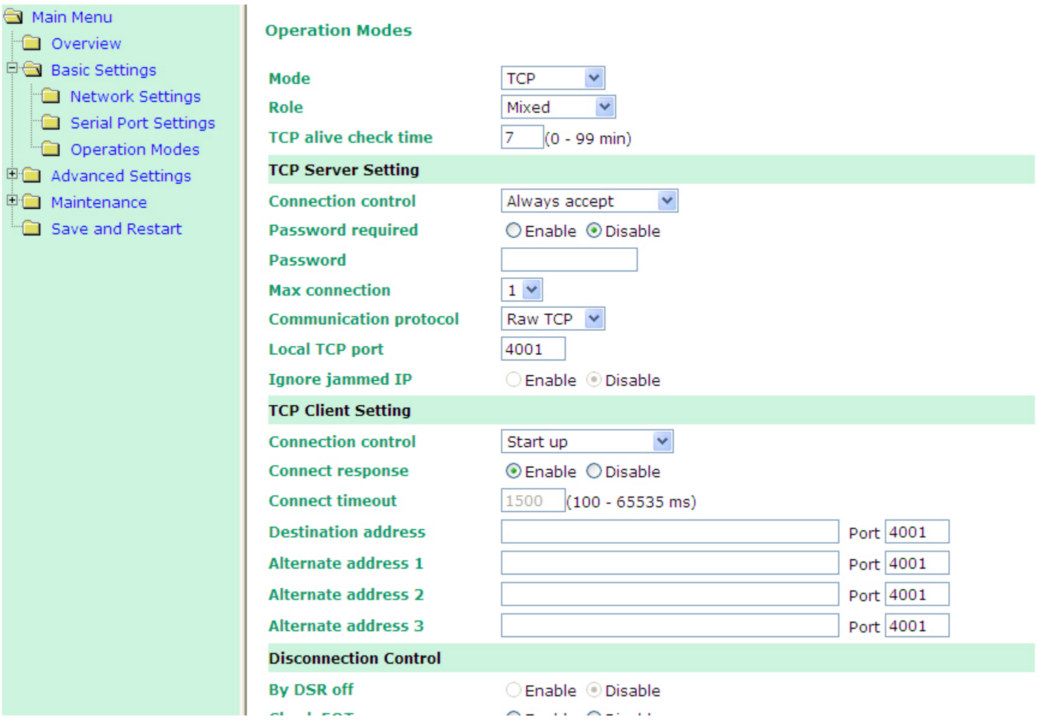


Operation Modes

Mode	TCP
Role	Mixed
TCP Server Setting	
Local TCP port	4001
TCP Client Setting	
Destination address	Port 4001

For information related to configuration settings in **TCP Mixed Mode**, refer to the descriptions for **TCP Server Mode** and **TCP Client Mode**.

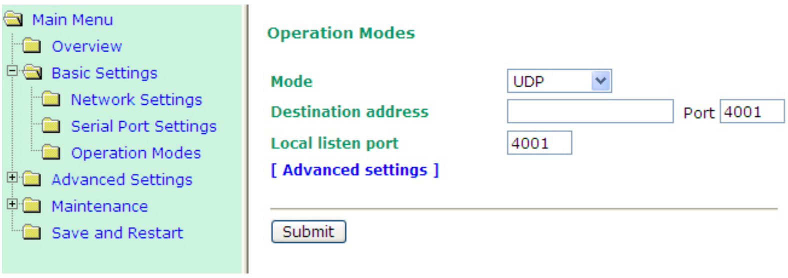
Advanced Settings for TCP Mixed Mode



Operation Modes

Mode	TCP
Role	Mixed
TCP alive check time	7 (0 - 99 min)
TCP Server Setting	
Connection control	Always accept
Password required	○ Enable ● Disable
Password	
Max connection	1
Communication protocol	Raw TCP
Local TCP port	4001
Ignore jammed IP	○ Enable ● Disable
TCP Client Setting	
Connection control	Start up
Connect response	● Enable ○ Disable
Connect timeout	1500 (100 - 65535 ms)
Destination address	Port 4001
Alternate address 1	Port 4001
Alternate address 2	Port 4001
Alternate address 3	Port 4001
Disconnection Control	
By DSR off	○ Enable ● Disable

UDP Mode



Operation Modes

Mode: UDP

Destination address: Port 4001

Local listen port: 4001

[Advanced settings]

Destination address

Setting	Factory Default	Necessity
IP address (e.g., 192.168.1.1)	None	Required

Use this field to specify the remote hosts that the module will connect to.

Port

Setting	Factory Default	Necessity
1 to 65535	4001	Required

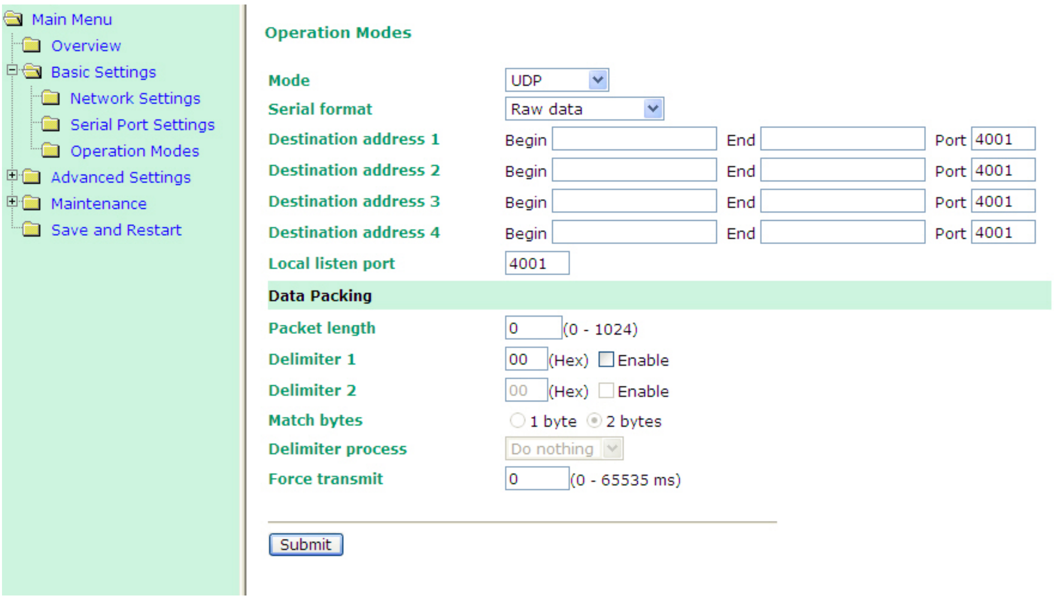
Use this field to indicate the UDP port of the destination host that will be connected to by MiiNePort E1. To avoid conflicts with well known UDP ports, the default is set to 4001.

Local listen port

Setting	Factory Default	Necessity
1 to 65535	4001	Required

Use this field to indicate the local listen UDP port of the MiiNePort E1. To avoid conflicts with well known UDP ports, the default is set to 4001.

Advanced Settings for UDP Mode



Operation Modes

Mode: UDP

Serial format: Raw data

Destination address 1: Begin End Port 4001

Destination address 2: Begin End Port 4001

Destination address 3: Begin End Port 4001

Destination address 4: Begin End Port 4001

Local listen port: 4001

Data Packing

Packet length: 0 (0 - 1024)

Delimiter 1: 00 (Hex) □ Enable

Delimiter 2: 00 (Hex) □ Enable

Match bytes: ○ 1 byte ● 2 bytes

Delimiter process: Do nothing

Force transmit: 0 (0 - 65535 ms)

Serial format

Setting	Factory Default	Necessity
Raw data, Packet command	Raw data	Optional

When the MiiNePort E1 receives serial data and then sends out a UDP packet, it is possible to dynamically change the destination address during this operation with the Serial Format function. If you set the Serial format as Raw data, you can then set the up to four desired destinations. But if you set it as Packet command, you will then be able to use the following command to change the destination IP address on the fly.

The command format is:

Start byte + Destination IP + Length + Data

For example, if you want to change the destination IP address to 192.168.35.100 with data "0123", you will need to send the serial data as 02C0A82364000430313233.

The ASCII to Hex code transformation of this example is shown below:

	Start byte	Destination IP				Length (2 bytes)	Data
ASCII	SOT	192	168	35	100	04	0123
Hex	02	C0	A8	23	64	00 04	30 31 32 33

Note that the destination port should be the same as the local listen port.

Destination address 1/2/3/4 16

Setting	Factory Default	Necessity
IP address range (e.g., Begin: 192.168.1.1 End: 192.168.1.10) or single IP address (e.g., Begin 192.168.1.1)	None	Required

Use this field to specify the remote hosts that the module will connect to.

NOTE If you already enter one set of Destination address and Port, you will see the data is kept when you enter into Advanced settings.

ATTENTION

The Destination address parameter can accept domain name addresses in place of IP addresses if a DNS server has been configured. For some applications, you may need to send the data actively to a remote destination's domain name address.

ATTENTION

The maximum selectable IP address range is 64 addresses. However, when using multi-unicast, you may enter IP addresses of the form xxx.xxx.xxx.255 in the Begin field. For example, enter 192.127.168.255 to allow the MiiNePort E1 to broadcast UDP packets to all hosts with IP addresses between 192.127.168.1 and 192.127.168.254.

Data Packing

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. At the default of 0 for packet length, no maximum amount is 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 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 character or 2-character sequence which will act as a marker to control packing of serial data. By default, no delimiter characters are 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 2-character sequence. When the delimiter is received, the module will pack up the data in its buffer and send it 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 not only controlled by the delimiter; it 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. Also, if the module will also pack data for network transmission if the next byte of data is not received within the Force transmit time.

Match bytes

Setting	Factory Default	Necessity
1 byte to 2 bytes	2 bytes (only effective when Delimiter is enabled)	Optional

1 byte: MiiNePort E1 packs the serial data and sends out TCP/IP packets once it receives either Delimiter 1 or Delimiter 2.

2 bytes: MiiNePort E1 only packs the serial data and sends out TCP/IP packets once it receives both Delimiter 1 and Delimiter 2. And the receiving order should be first Delimiter 1 then Delimiter 2.

Delimiter process

Setting	Factory Default	Necessity
Do Nothing, Delimiter+1, Delimiter+2	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: Data in the buffer will be transmitted after 1 additional byte is received following the delimiter.

Delimiter + 2: Data in the buffer will be transmitted after 2 additional bytes are received following the delimiter.

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.

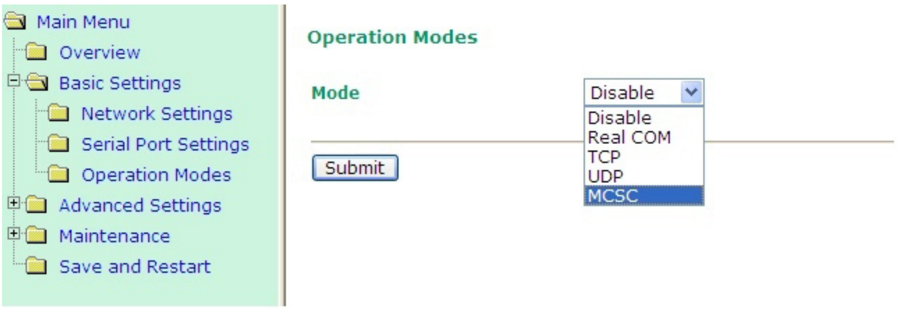
When set to 0, Force transmit is disabled, which means there is no time limit for how long the module will wait to receive data. When set between 1 and 65535, the module will pack data as soon as there has been no serial communication for the specified time.

The optimal force transmit time depends on your application, but it should be larger than one character interval within the specified baud rate 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

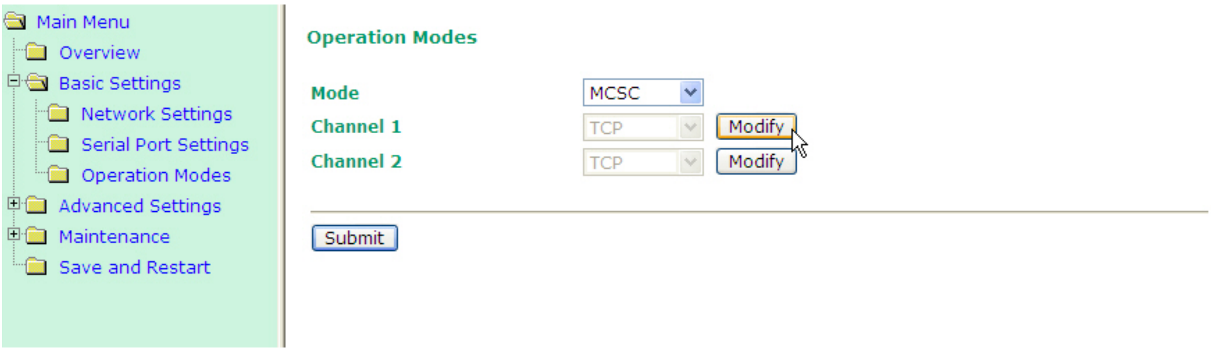
(10 bits / 1200 bits/s) × 1000 ms/s = 8.3 ms.

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 no delimiter characters or Force transmit time were specified at all.

MCSC Mode

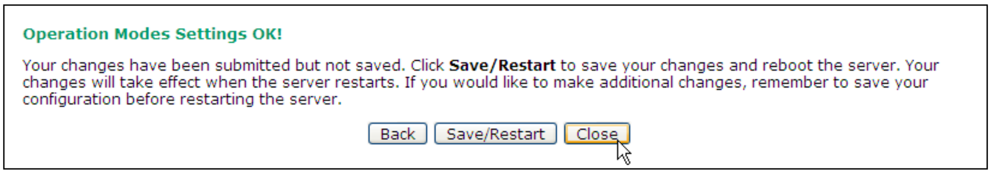


After **MCSC mode** is selected, you will see 2 channels that are ready to configured. Under the MCSC structure, each channel works independently so you need to configure each channel separately according to your application. Click **Modify** to configure the Channel 1.



Mode MCSC
Channel 1 TCP Modify
Channel 2 TCP Modify

After you click Modify, the channel configuration window will pop up and you will see exactly the same configuration options as stated earlier in this chapter. For example, if you want to set Channel 1 to TCP Client mode, just finish your configuration, click **Submit**, and a confirmation window will appear.

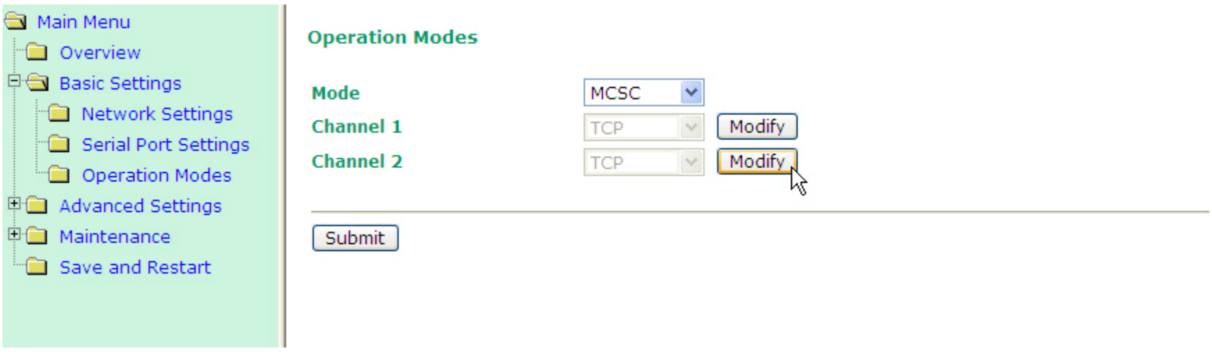


Operation Modes Settings OK!

Your changes have been submitted but not saved. Click **Save/Restart** to save your changes and reboot the server. Your changes will take effect when the server restarts. If you would like to make additional changes, remember to save your configuration before restarting the server.

We suggest that you to click **Close** to go back to the **Operation Mode** configuration screen to finish configuring your Channel 2. After you get back to the screen, you will see the TCP mode setting for Channel

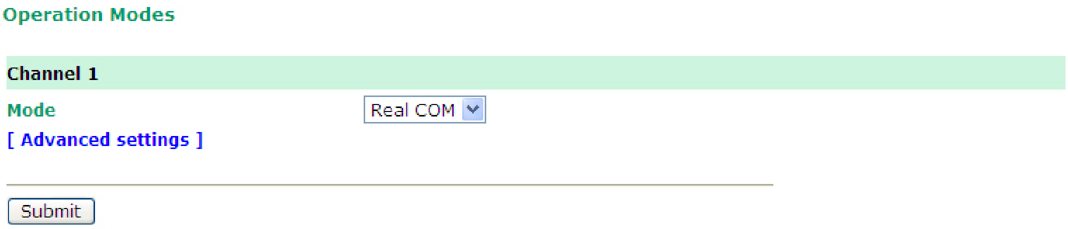
1 is already shown on the screen. Note that after you click Submit, your configuration is already stored in MiiNePort E1’s memory and won’t be erase unless your change the configuration again.



Operation Modes

Mode: MCSC
Channel 1: TCP
Channel 2: TCP

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**.



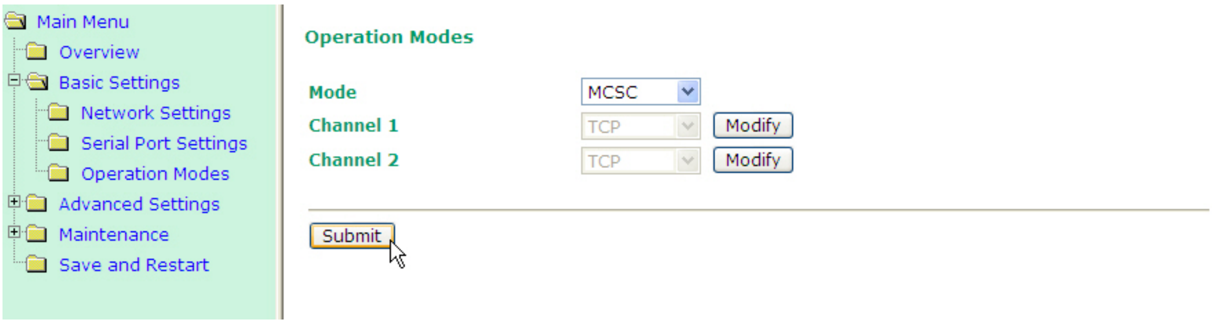
Operation Modes

Channel 1

Mode: Real COM

A confirmation window will then appear; **click** Close to return to the Operation Modes setting screen.

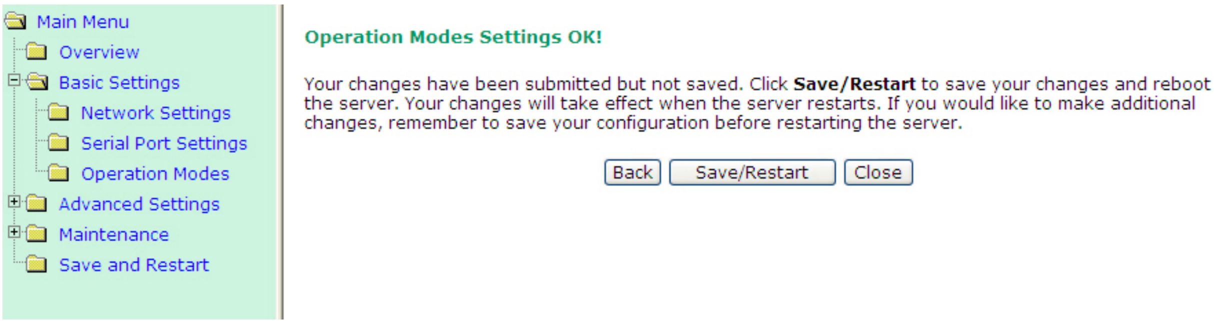
Click **Submit** to finish MCSC configuration on your MiiNePort E1.



Operation Modes

Mode: MCSC
Channel 1: TCP
Channel 2: TCP

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



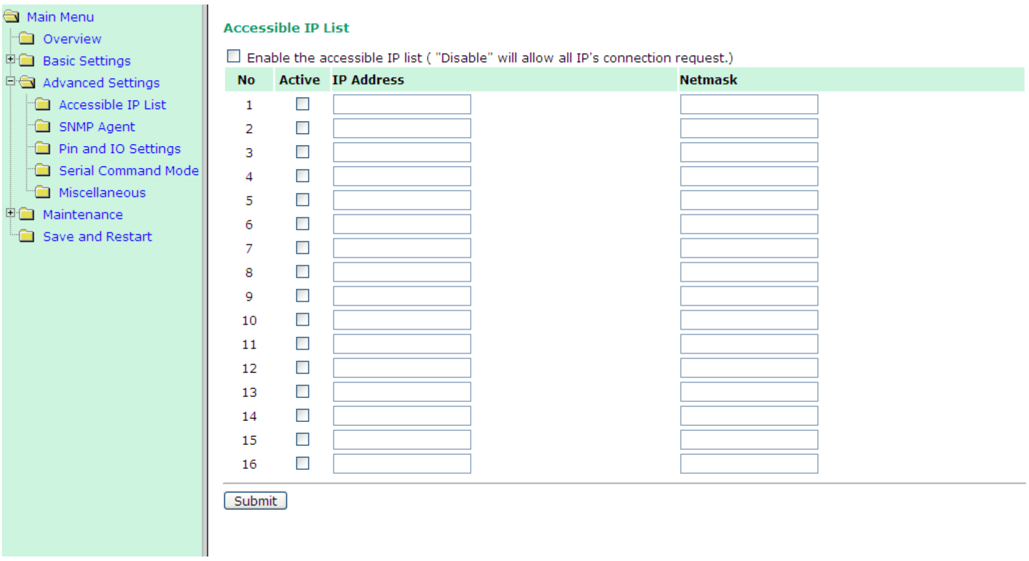
Operation Modes Settings OK!

Your changes have been submitted but not saved. Click **Save/Restart** to save your changes and reboot the server. Your changes will take effect when the server restarts. If you would like to make additional changes, remember to save your configuration before restarting the server.

Besides the configuration settings, refer to **Chapter 8: NetEZ Technologies** for the MCSC command format and device system design guidance.

Advanced Settings

Accessible IP List



Accessible IP List

☐ Enable the accessible IP list ("Disable" will allow all IP's connection request.)

No	Active	IP Address	Netmask
1	☐		
2	☐		
3	☐		
4	☐		
5	☐		
6	☐		
7	☐		
8	☐		
9	☐		
10	☐		
11	☐		
12	☐		
13	☐		
14	☐		
15	☐		
16	☐		

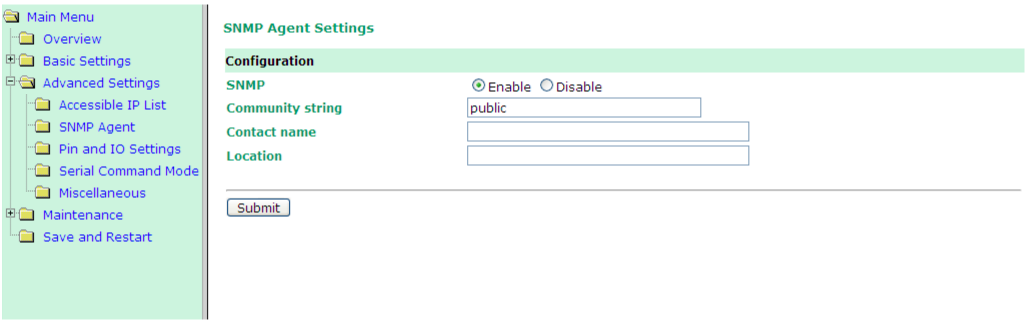
- **To allow access to a specific IP address**
  Enter the IP address in the corresponding field; enter 255.255.255.255 for the netmask.
- **To allow access to hosts on a specific subnet**
  For both the IP address and netmask, use 0 for the last digit (e.g., 192.168.1.0 and 255.255.255.0).
- **To allow 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

SNMP Agent

To enable the SNMP agent function, select the Enable option for SNMP under Configuration.



SNMP Agent Settings

Configuration

SNMP ● Enable ○ Disable
Community string public
Contact name
Location

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

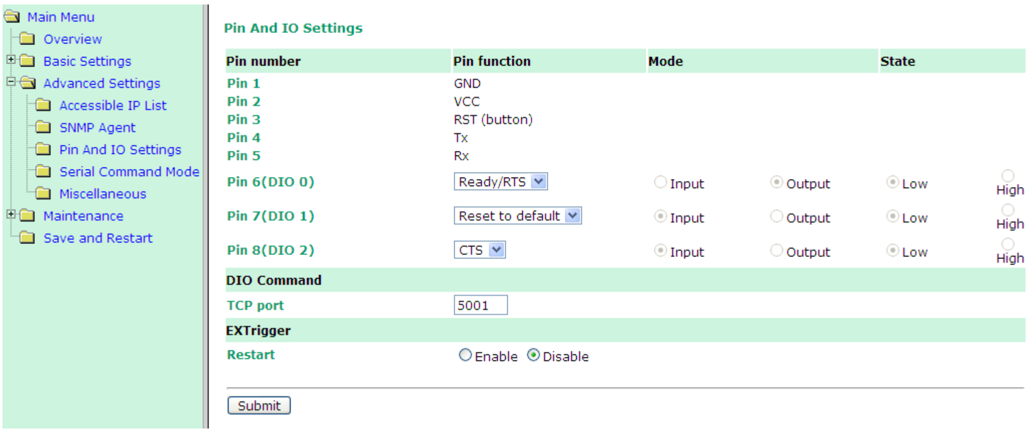
The SNMP contact information usually includes an emergency contact name and telephone or pager number.

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.

Pin and IO Settings



Pin And IO Settings

Pin number	Pin function	Mode		State	
Pin 1	GND				
Pin 2	VCC				
Pin 3	RST (button)				
Pin 4	Tx				
Pin 5	Rx				
Pin 6(DIO 0)	Ready/RTS	○ Input	● Output	● Low	○ High
Pin 7(DIO 1)	Reset to default	● Input	○ Output	● Low	○ High
Pin 8(DIO 2)	CTS	● Input	○ Output	● Low	○ High
DIO Command					
TCP port	5001				
EXTrigger					
Restart	○ Enable ● Disable				

Pin Function

For the 3 configurable pins (Pin 6, 7, and 8), refer to **Chapter 1: Pin Assignments** for their default settings and change to the appropriate function for your application.

NOTE Default settings: Pin 6 as Ready/RTS, Pin 7 as Reset to Default, Pin 8 as CTS.

DI is for digital input operation, where the channel's status is controlled by an external digital switch. DO is for digital output operation, where the channel transmits a high or low signal. Use Status to control high or low status for digital output channels; status setting will be ignored for digital input channels.

Reset to Default (Use Pin 7, DIO 1) is used to clear the password or reset the MiiNePort E1 to Moxa's factory default settings. When this function is enabled, Pin 7, DIO 1, will be forced to digital input mode and will act as an internal reset mechanism. Pulling Pin 7, DIO 1, "low" for 100 ms will load the Moxa factory default settings. This function is enabled by default.

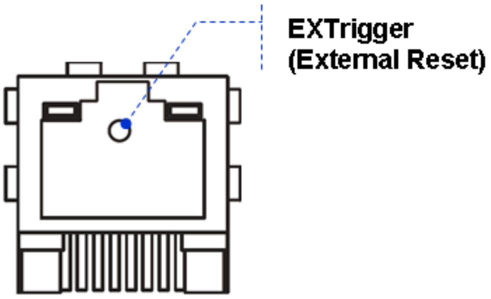
DIO Command

DIO Command TCP Port is the port number that will be reserved for DIO commands. DIO commands may be used to control and obtain data from the module's DIO channels. Refer to **Appendix D: DIO Commands** for additional information on DIO commands.

EXTrigger

MiiNePort E1's EXTrigger has 2 functions; reset the module to Moxa's default configuration and restarting the Module. EXTrigger is designed to provide an easy tool for network troubleshooting without interrupting device operation.

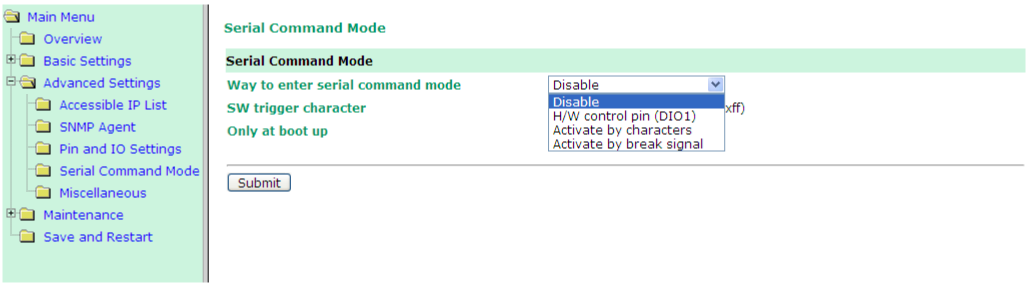
The Reset to Default function is enabled by default and is not subject to be disabled. You can choose to enable or disable the Restart function in the Pin and IO Settings.



For more details on how to use EXTrigger, refer to **Chapter 8: NetEZ Technologies**.

Serial Command Mode (SCM)

SCM (Serial Command Mode) uses serial communication between the MiiNePort E1 and your device's main system to configure the MiiNePort E1, usually during device operation. For more details about SCM commands, refer to **Chapter 8: NetEZ Technologies.**



Serial Command Mode

Serial Command Mode	
Way to enter serial command mode	Disable Disable H/W control pin (DIO1) Activate by characters Activate by break signal
SW trigger character	xff)
Only at boot up	

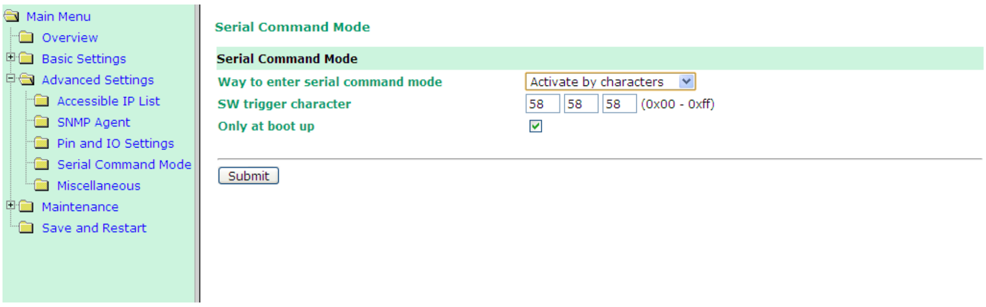
Way to enter serial command mode

Setting	Factory Default	Necessity
Disable, H/W control pin (DIO1), Activate by characters, Activate by break signal	Activate by characters (Only at boot up)	Required

There are three ways to access SCM:

H/W control pin (DIO 1): Access SCM by pulling Pin 7, DIO1, to "low".

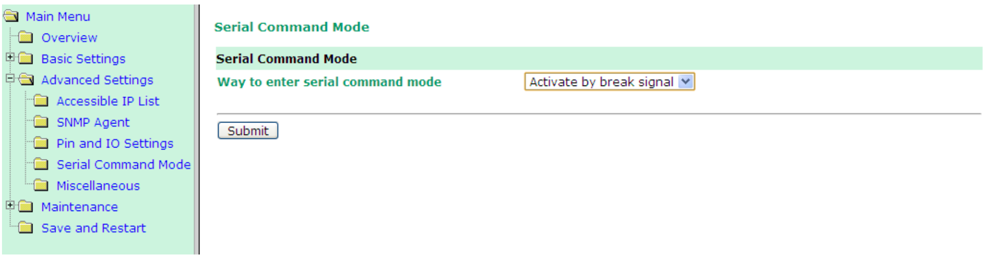
Activate by characters: Access SCM by sending the MiiNePort E1's configurable three characters from your device's main system. Check Only at boot up if you want to make this effective only when booting up the MiiNePort E1.



Serial Command Mode

Serial Command Mode	
Way to enter serial command mode	Activate by characters
SW trigger character	58 58 58 (0x00 - 0xff)
Only at boot up	☑

Activate by break signal: Break signals are caused by sending continuous Spacing values (no Start or Stop bits). When there is no electricity present on the data circuit, the line is considered to be sending a Break. The Break signal must be of duration longer than the time it takes to send a complete byte plus Start, Stop, and Parity bits.

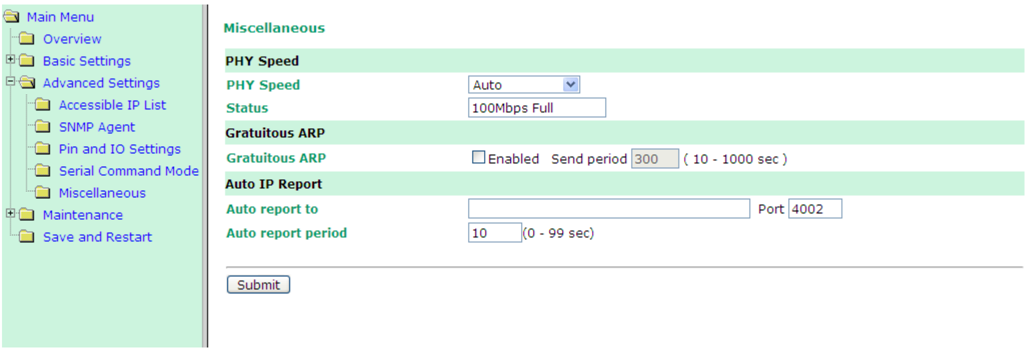


Serial Command Mode

Serial Command Mode

Way to enter serial command mode: Activate by break signal

Miscellaneous



Miscellaneous

PHY Speed

PHY Speed: Auto

Status: 100Mbps Full

Gratuitous ARP

Gratuitous ARP: □ Enabled Send period 300 (10 - 1000 sec)

Auto IP Report

Auto report to: Port 4002

Auto report period: 10 (0 - 99 sec)

PHY Speed

PHY Speed

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

Auto IP Report

When the module 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 or UDP 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 an IP address and TCP port number of your choice. There are two ways to view the module's IP address report on a PC:

Develop software that parses the IP address report data. Refer to Appendix C: Auto IP Report Protocol for details on how to parse the module's IP address report data.

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. An auto report period of 10 seconds means that an IP address report will be sent every 10 seconds.

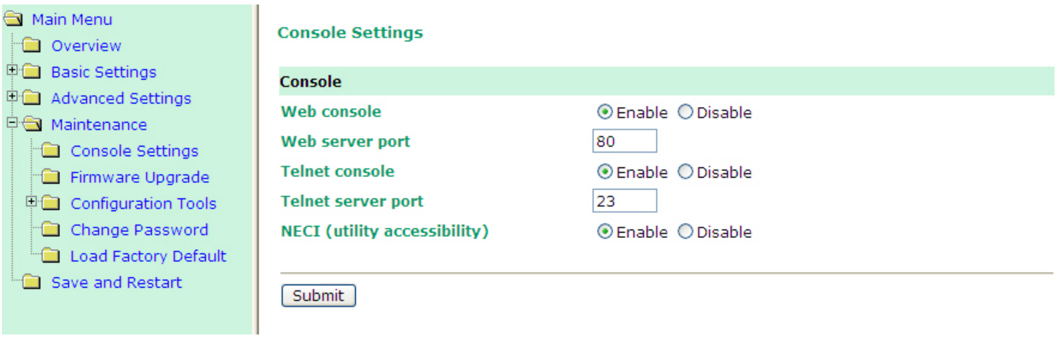
ATTENTION

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, no IP address report will be sent.

Maintenance

Console Settings

You can enable or disable the Web, Telnet, and Utility console functions on this page. You can also set up the Web server port number and Telnet server port number here.



Console Settings

Console	
Web console	● Enable ○ Disable
Web server port	80
Telnet console	● Enable ○ Disable
Telnet server port	23
NECI (utility accessibility)	● Enable ○ Disable

Firmware Upgrade

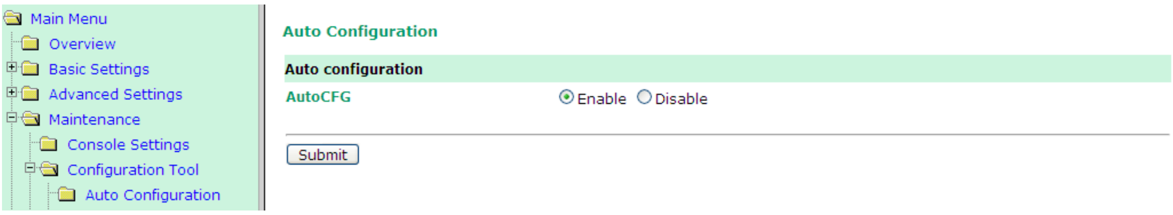
The firmware can be upgraded though the web console or through the Device Search Utility. If you have made any changes to your configuration, remember to save the configuration first before upgrading the firmware. Any unsaved changes will be discarded when the firmware is upgraded. To upgrade the firmware, simply enter the file name and click **Submit**. The latest firmware can be downloaded from www.moxa.com.

Configuration Tools

The MiiNePort E1 has Auto Configuration, Configuration Import, and Configuration Export functions built in to make the configuration process more efficient.

Auto Configuration

You can enable or disable the AutoCFG (Auto Configuration) function here. AutoCFG is designed for enabling automatic network configuration during your device production. By using the AutoCFG, you can realize true device mass production without needing to set up the network modules one by one.



Auto Configuration

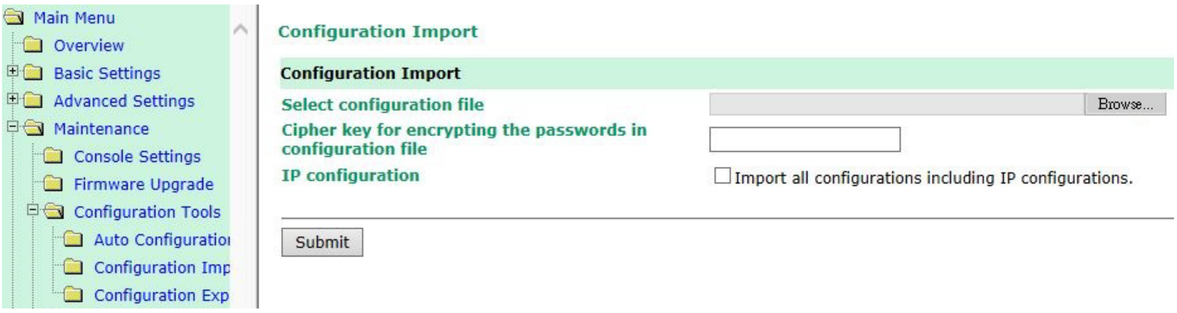
Auto configuration

AutoCFG ● Enable ○ Disable

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

Configuration Import

The MiiNePort E1 Series can share or back up its configuration by exporting all settings to a file, which can then be imported into another MiiNePort E1. The passwords in the exported file will be encrypted by a cipher key assigned by the user, which will be asked again when importing back to the MiiNePort E1 module. The Configuration Import utility allows you to choose whether to import the IP configuration at the same time.



Configuration Import

Configuration Import

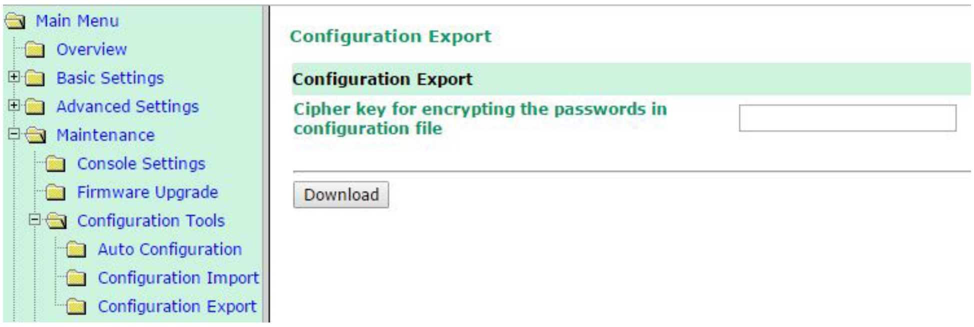
Select configuration file

Cipher key for encrypting the passwords in configuration file

IP configuration □ Import all configurations including IP configurations.

Configuration Export

You can download the current configuration as a MiiNePortE1.txt (MiiNePortE1-H.txt) file by clicking on Configuration Export. You will be asked to input an user-defined cipher key for encrypting the passwords in the configuration file before clicking the download button. The configuration MiiNePortE1.txt (MiiNePortE1-H.txt) file can be used later in Configuration Import for effortless configuration replication or AutoCFG for mass configuration deployment during your device's mass production stage.



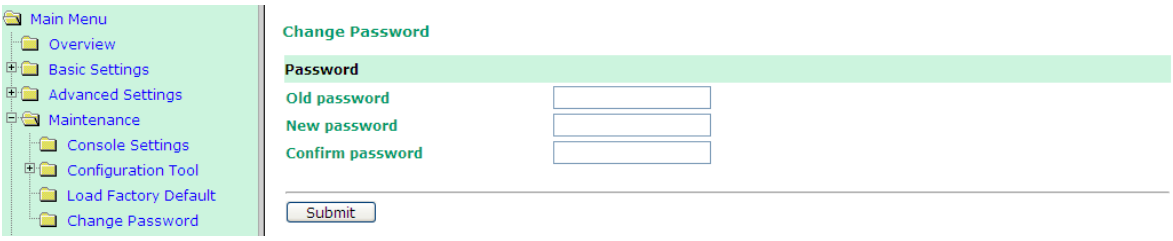
Configuration Export

Configuration Export

Cipher key for encrypting the passwords in configuration file

Change Password

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



Change Password

Password	
Old password	
New password	
Confirm password	

Submit

Enter the old password and new passwords and click on **Submit**. The default password for MiiNePort device is **moxa**.

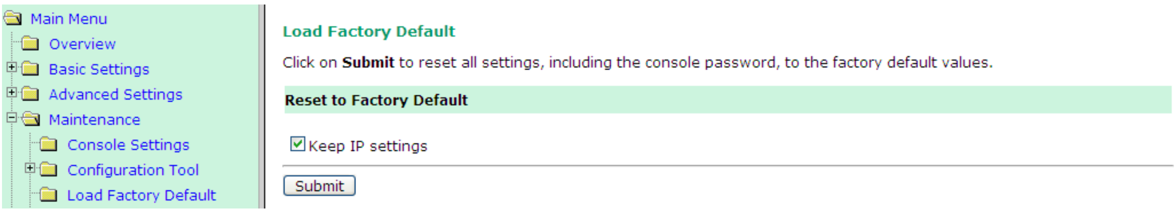
ATTENTION

If you forget your MiiNePort E1's password you may want to reset the module to factory defaults. Since you may not be able to use the function through any of the usual software interfaces, we designed two hardware tools for you to handle this problem;

1. Pin 7 on the Module: To use the "Load Factory Defaults" function, refer to Chapter 1: Pin Assignments. Check the Pin 7's definition. If you pull low Pin 7 for longer than 100 ms, the pull high and the MiiNePort E1 will automatically load factory default settings.
2. EXTrigger: By pressing the EXTrigger button when the device is powering on, the MiiNePort E1 module will reset itself to factory default settings. For details, refer to Chapter 8: NetEZ Technologies.

Load Factory Default

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



Load Factory Default

Click on **Submit** to reset all settings, including the console password, to the factory default values.

Reset to Factory Default

☑ Keep IP settings

Submit

8

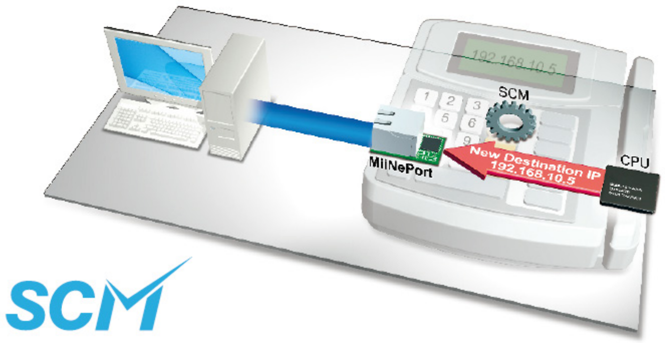
NetEZ Technologies

This chapter introduces the NetEZ technology family and its four innovative functions.

The following topics are covered in this chapter:

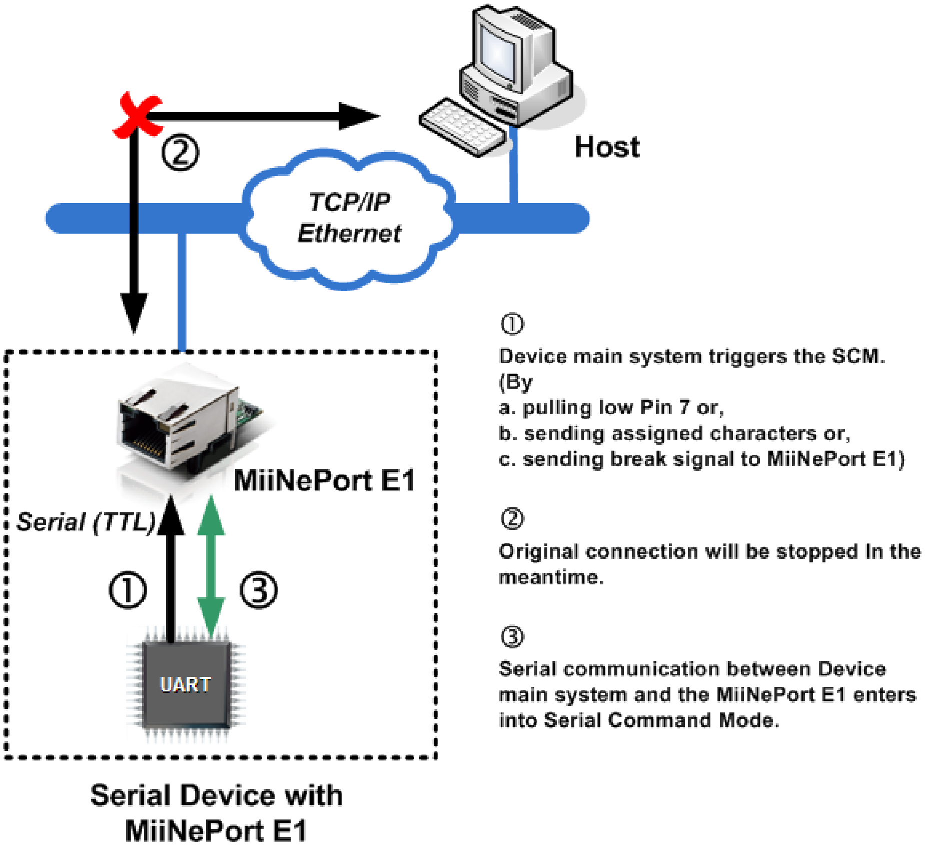
- ❒ **SCM (Serial Command Mode)**
- ❒ **EXTrigger (External Trigger)**
- ❒ **AutoCFG (Auto Configuration)**
- ❒ **MCSC (Multiple Channel Serial Communication)**
  - ➢ Command Packets
  - ➢ SCM (Serial Command Mode) under MCSC

SCM (Serial Command Mode)



MiiNePort E1's SCM (Serial Command Mode) allows the module's parameters to be retrieved or configured through the serial port, rather than over the network. This is done through the use of specially parsed commands sent to the module through the serial port.

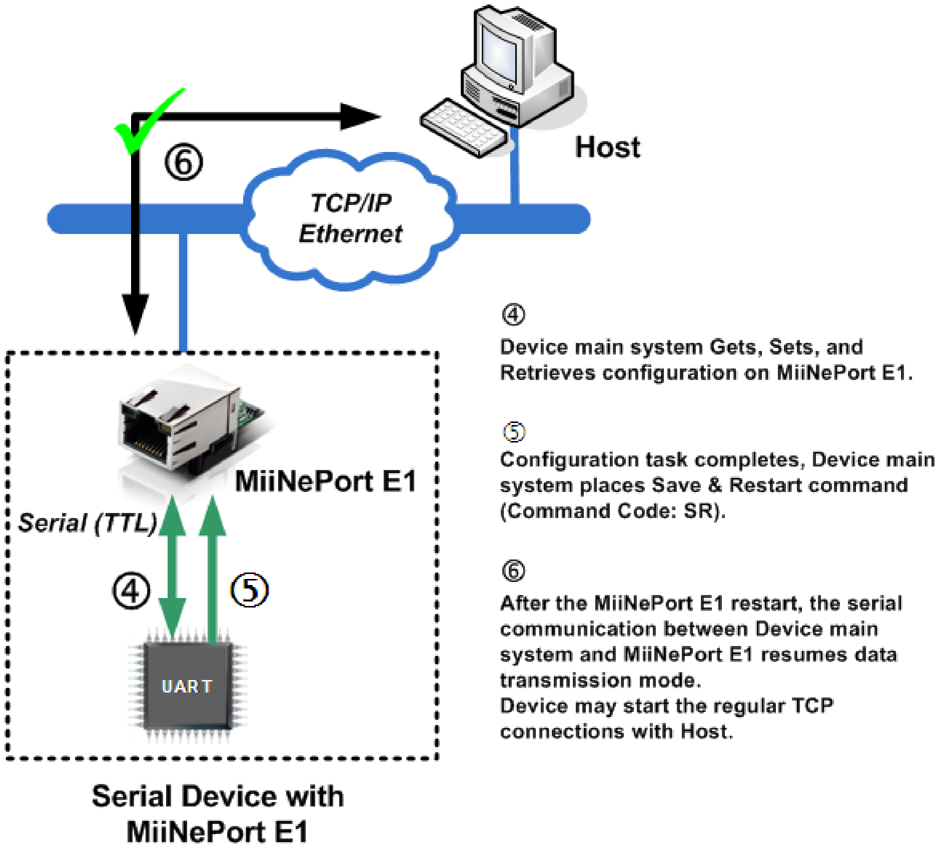
SCM is often used when your device has already been used in actual applications and you need to change the MiiNePort E1's configuration, such as changing the device's IP address by using your device's key pad. The details are shown below:



①
Device main system triggers the SCM.
(By
a. pulling low Pin 7 or,
b. sending assigned characters or,
c. sending break signal to MiiNePort E1)

②
Original connection will be stopped In the meantime.

③
Serial communication between Device main system and the MiiNePort E1 enters into Serial Command Mode.


Host
TCP/IP
Ethernet
⑥
④
Device main system Gets, Sets, and Retrieves configuration on MiiNePort E1.
⑤
Configuration task completes, Device main system places Save & Restart command (Command Code: SR).
⑥
After the MiiNePort E1 restart, the serial communication between Device main system and MiiNePort E1 resumes data transmission mode.
Device may start the regular TCP connections with Host.
MiiNePort E1
Serial (TTL)
④
⑤
UART
Serial Device with
MiiNePort E1

Refer to **Chapter 7: Web Console Configuration** for the most appropriate way to use SCM for your application.

Refer to **Appendix A: Introduction to SCM (Serial Command Mode) Command Set** for detailed instructions on using SCM commands.

EXTrigger (External Trigger)

The MiiNePort E1's EXTrigger has 2 functions: (1) Resetting the module to Moxa's default configuration, and (2) Restarting the module. EXTrigger is designed to provide you with an easy network troubleshooting tool that can be used without stopping the device's normal operation.

The Reset to Default function is enabled by default and cannot be disabled. This design prevents device deadlock in the future if you or your customer forgets the password and is unable to access the console to troubleshoot the device. To activate the EXTrigger's Reset to Default function, press the button inside the RJ45 jack for 5 seconds while your device and the MiiNePort E1 is starting up.

The Restart function simply shuts down the module and then turns it back on. This does not affect any settings in the module but is a commonly used method for initial troubleshooting. Simply hold the button for 5 seconds at any time during device operation to restart the MiiNePort E1 module.

NOTE Refer to Chapter 7: Web Console Configuration, Pin and IO Settings, External Reset Function to enable EXTrigger on your MiiNePort E1.

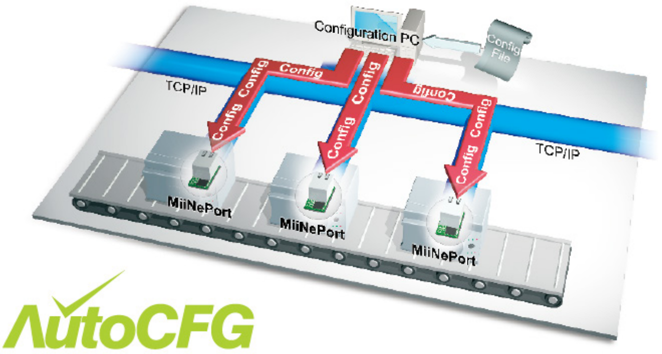
NOTE The Restart function of ExTrigger is disabled in Moxa's factory default settings.

Refer to the following table describing the LED interaction with EXTrigger.

EXTrigger	Fault/In Use LED Behavior	Blinking Duration	Trigger Timing	Note
Reset to Default	1 time/sec (Blinks amber quickly)	5 sec	Press the button, then power on the device.	When LED is Off (stops blinking) 1. Function active 2. Release button
Restart	0.5 time/sec (Blinks amber slowly)	5 sec	Press the button anytime during operation.	

EXTrigger is actually an external Digital Input (DI) that can be accessed by you or your device's end user. You can design your own application by leveraging this external DI. Contact Moxa for information about custom EXTrigger functions.

AutoCFG (Auto Configuration)



The MiiNePort E1's AutoCFG function is designed to allow users to realize true mass production. With AutoCFG, you no longer need to configure network modules one by one during the device manufacturing process. To use AutoCFG, follow the steps described below:

NOTE AutoCFG is enabled in Moxa's factory default settings. If you want to disable AutoCFG, see Chapter 7: Web Console Configuration, Maintenance, Configuration Tool, Auto Configuration.

Step 1: Export the configuration. Refer to **Chapter 7: Web Console Configuration → Maintenance → Configuration Tool → Configuration Export**. After you export your configuration, save the configuration to MiiNePortE1.txt (MiiNePortE1-H.txt), which contains all of the MiiNePort E1 module's settings.

Step 2: Set up a TFTP server on the network where you plan to conduct the AutoCFG task on your device production line. Set your TFTP server's IP address as 169.254.x.x/16. Next, save the previously saved MiiNePortE1.txt (MiiNePortE1-H.txt) file under the root directory of the TFTP server. At this point, the AutoCFG working environment should be ready.

NOTE Quite a bit of freeware that can help you easily set up a TFTP server can be found on the Internet.

Step 3: AutoIP protocol will automatically assign your MiiNePort E1 modules with a temporary IP address.

Step 4: Your MiiNePort E1 modules will actively send out a broadcast packet asking if there is a TFTP server on the same subnet that has a file named MiiNePortE1.txt (MiiNePortE1-H.txt).

Step 5: Your TFTP server will respond to the broadcast packet and your MiiNePort E1 modules will automatically download the MiiNePortE1.txt (MiiNePortE1-H.txt) from the TFTP server.

Step 6: Your MiiNePort E1 modules will import the configuration. Save the configuration. Once the MiiNePort E1 finishes the self-configuration process, the Fault/In-Use LED on the RJ45 connector will blink, alternating between Green and Amber. When you see the LED blinking, the AutoCFG task is completed and you can restart your device or power it off.

Step 7: When you need to change the configuration, just complete the revised configuration, export the new MiiNePortE1.txt (MiiNePortE1-H.txt), save the file to the TFTP server, choosing to replace the original MiiNePortE1.txt (MiiNePortE1-H.txt) under the root directory. Your MiiNePort E1 modules will then be able to retrieve the correctly revised configuration.

ATTENTION

Check the following if AutoCFG is not working properly on your device:

1. Make sure the AutoCFG function is enabled (the function is enabled by default).
2. Check to see if the TFTP Server is working properly.
3. Make sure the specific configuration filename is MiiNePortE1.txt (MiiNePortE1-H.txt).

ATTENTION

There is no theoretical limit to the number of MiiNePort E1 modules that can be connected at the same time to one subnet for conducting an AutoCFG task.

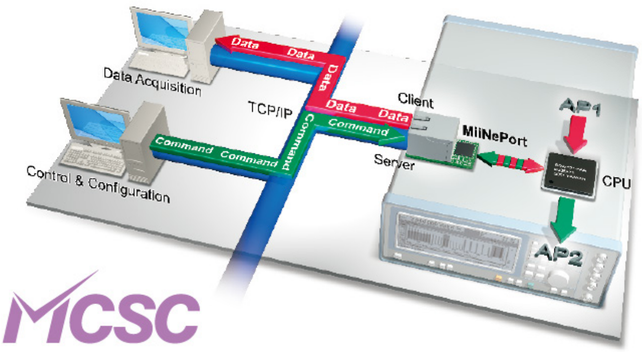
ATTENTION

Even if you do not want to use the auto configuration function, the module will still go through the process of searching the MiiNePortE1.txt (MiiNePortE1-H.txt) file on the network every time it starts up if you do not disable the AutoCFG function on your MiiNePort E1. The MiiNePort E1 will resume normal operation if it cannot find the correct TFTP server with the MiiNePortE1.txt (MiiNePortE1-H.txt) file on the network. On average, it will take less than 5 seconds to conduct a search. Your MiiNePort E1 will work properly with its original configuration even if it does not complete the AutoCFG task.

ATTENTION

If you do not want the AutoCFG function to remain active after you've successfully finished the auto-configuration, we suggest that you DISABLE the AutoCFG function when you are creating your MiiNePortE1.txt file. This will prevent the AutoCFG activity from recurring the next time your device is powered on. Refer to Chapter 7: Web Console Configuration for configuration details.

MCSC (Multiple Channel Serial Communication)



The Motivation Behind MCSC

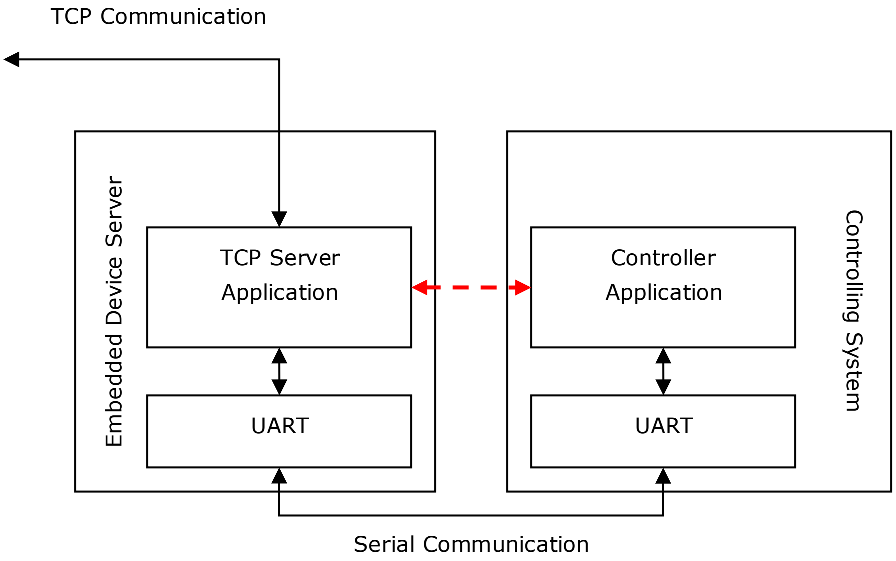
An embedded device server only provides a limited number of physical serial ports to communicate with the device's main system. As a result, users with one physical serial port are generally limited to a single application. If you want to implement a second, third, or more serial-to-Ethernet applications (note that the MiiNePort E1 has only one physical serial port), additional physical serial ports are needed. Unfortunately, providing additional physical serial ports increases hardware costs. Moxa's MiiNePort E1's MCSC is designed to solve this dilemma. MCSC (Multiple Channel Serial Communication) is a lightweight, pure software solution that enables multiple serial-to-Ethernet applications all through one physical serial port without any incremental hardware costs to you. In short, MCSC allows multiple application channels to share a single physical serial port at the same time.

Communication Model

In traditional serial communication models, communication is accomplished by connecting two devices over a single serial channel. At the embedded device server end, Operating Mode (OP Mode) enables the other end of the serial channel to transmit or receive data from the network. However, only one application, TCP Server transportation, TCP Client transportation, or UDP transportation is supported at a time. The communication model is depicted in the following figure:

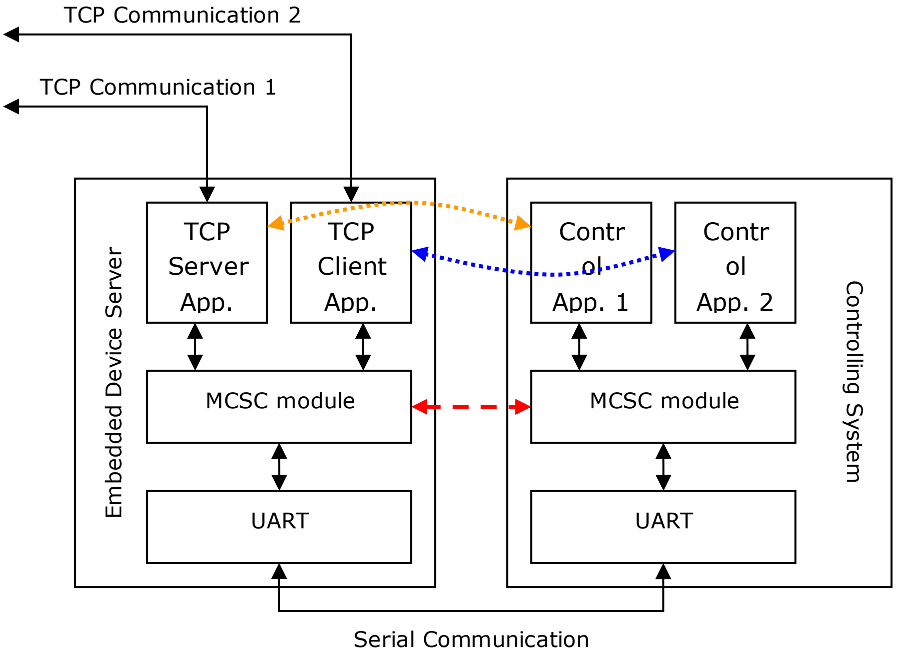
NOTE Refer to Chapter 3: Choosing the Proper Operation Mode for an introduction to the operation modes supported by the MiiNePort E1.

Although there are UARTs and even an OS and driver stacks between the TCP Server Application and Controller Application, the applications are considered to be connected by a logical bidirectional channel indicated in red in the following figure;



When using MCSC, each serial communication end is attached with an MCSC command encoding and decoding module (in this case, module refers to a software programming module). This module is used primarily for UART applications. It compresses the data transmitted by the application to the single physical serial channel, and dispatches the data received from the physical serial channel to the appropriate applications. With the help of the MCSC modules, the two applications at opposite ends are still connected to a logical bidirectional channel, but two or more application pairs are also connected by separate channels to the physical serial channel.

The communication model is depicted in the following figure.



The MCSC module is built into the MiiNePort E1. To enable MCSC, you need to set your MiiNePort E1 serial port operation mode to MCSC mode and then set the channels' operation mode individually according to your application. In addition, you also need to implement the MCSC module in your device's main system.

The following basic principles apply to all MCSC module implementations.

1. MCSC uses a command-based stream protocol. That is, transmissions that do not fit the MCSC command format are treated as application data. All MCSC commands begin with ASCII DLE characters. For the detailed command format, refer to the Command Packets section in following section.
2. Other than dealing with MCSC commands, an MCSC module simply transfers application data from the upper application channel to the underlying physical serial channel, and vice versa.
3. As with traditional serial applications, MCSC modules are logically connected to a bidirectional channel. To use the serial bandwidth more effectively, MCSC modules separate it into two unidirectional channels. Each MCSC module maintains its transmission application channel and reception application channel. That is, the transmission channel of one end is the reception channel of another.
4. Although logically all upper application channels share one physical serial port, only one channel can be connected to the physical serial port to prevent confusion. That is, the application channel connected to the transmission channel is called the active transmission channel, which transmits data to the serial channel. The application channel connected to the reception channel is called the active reception channel, which receives data from the serial channel.
5. An MCSC module decides the active transmission channel. In other words, the active reception channel is decided by the other MCSC module lying at the other end of the serial communication channel.
6. To avoid data loss, we recommend that MCSC is applied with flow control over serial communication.

Command Packets

An MCSC control packet begins with an ASCII DLE (Data Link Escape, 0x10) character. An end of serial communication (normally, an MCSC module) treats the DLE it receives as a special delimiter that indicates the other end of serial communication (normally, another MCSC module) that attempts to query or change the MCSC behavior.

If a serial communication end wants to send a 0x10 (DLE) data byte, it sends two DLEs, one after the other. The receiving end decodes these two DLEs into a single 0x10 byte.

Currently, there are 4 kinds of control packets defined under MCSC:

Channel Switch Command

DLE	**SOH**	***CHN***
0x10	0x01	0x??

This command is used for an end to query for its active serial reception channel. The CHN field indicates the new channel number. Note that the CHN index is zero based, so you will need to use 0x00 for CHN to switch to channel 1, 0x01 to switch to channel 2, and so on. For safety reasons, if one serial communication end receives a Channel Switch Command that switches to a channel it does not support, it will simply discard it after the data has been transmitted.

Note that when you begin using MCSC, both active communication channels are assumed to be channel 1. After serial communication has ended, data is transfered without directing Channel Switch Command to the channel 1 application.

In the MiiNePort E1, SCM will always be enabled when MCSC is enabled. A special channel indexed as 0xFF in MCSC is dedicated for SCM. For more information, refer to the Serial Command Mode section in this chapter.

Example

A>	10	01	01	11	22	33							10	01	00	44
B>							10	01	01	33	22	11				
A>	55	66														
B>			11	22	33	10	01	00	66	55	44	44	55	66		

This example shows that end A sends 3 bytes of data (11, 22, 33) using channel 1 and 3 bytes of data (44, 55, 66) using channel 2. On the other side, end B sends 6 bytes of data (33, 22, 11, 11, 22, 33) using channel 1 and 6 bytes of data (66, 55, 44, 44, 55, 66) using channel 2.

Channel Enquiry Command

DLE	**ENQ**
0x10	0x05

This command is used as a means to locate the active serial reception channel. By receiving this command, the other end will resend a Channel Switch Command to indicate the active transmission channel it is dealing with, which is the active reception channel the initiator of Channel Enquiry Command deals with.

Example

A>	10	05														
B>			10	01	01	11	22	33								

This examples shows that end A queries for the active transmission channel of end B, end B replies for channel 2 and then continues to send 3 bytes of data (11, 22, 33) using channel 2.

Data Escape Command

DLE	**DLE**
0x10	0x10

This is not actually an MCSC command. Since the DLE character is used by MCSC to escape its commands, a pair of DLEs will be transmitted to indicate a single 0x10 character of data.

Example

A>	10	01	00	00	02	04	06	08	0A	0C	0E	10	10	12		
B>																

This examples shows that end A sends 10 bytes of data (00, 02, 04, 06, 08, 0A, 0C, 0E, 10, 12) using channel 1.

Abnormal Packets

DLE	OTHER
0x10	???

Once the MiiNePort E1 receives a data stream with a header DLE character followed by characters that are not SOH, ENG, or DLE characters, the MiiNePort E1 will see this as an abnormal data packet caused by communication problems and will drop this data packet without sending data out through the Ethernet port. The MiiNePort E1 will return the channel enquiry command (10 05) to the serial main system through the serial channel, helping the main system to detect and troubleshoot the problem.

SCM (Serial Command Mode) under MCSC

When MCSC is enabled, a special application channel indexed as 0xFF (channel 256) is dedicated for Moxa SCM without additional configuration required. SCM enables the controlling system to configure, monitor, or control the attached embedded device server. For more information about what SCM can do, refer to Chapter 3: Choosing the Configuration Tool.

With MCSC, normal network communication continues uninterrupted when the controlling system is monitoring or diagnosing the embedded device server with SCM, providing SCM with greater flexibility.

9

Android API Instructions

The following topics are covered in this chapter:

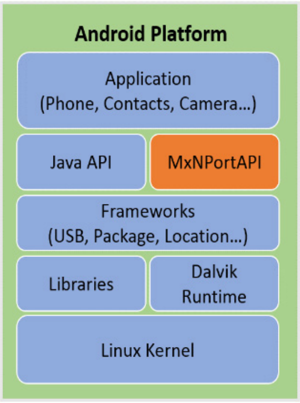
- ❒ **Overview**
  - ➢ How to Start MxNPortAPI
- ❒ **MxNPortAPI Function Groups**
- ❒ **Example Program**

Overview

If you want to remote control your serial devices on an Android platform, then the MxNPortAPI is a simple application programming tool that you can use. The MxNPortAPI helps programmers develop an Android application to access the device server by TCP/IP.

The MxNPortAPI provides frequently used serial command sets like port control, input/output, etc., and the style of developed Android application is similar to MOXA Driver Manager. For more details of the provided functions, please refer to the “MxNPortAPI Function Groups” section.

This MxNPortAPI is layered between the Android application and Android network manager framework. This Android library is compatible with Java 1.7, Android 3.1 (Honeycomb - API version 12), and later versions.

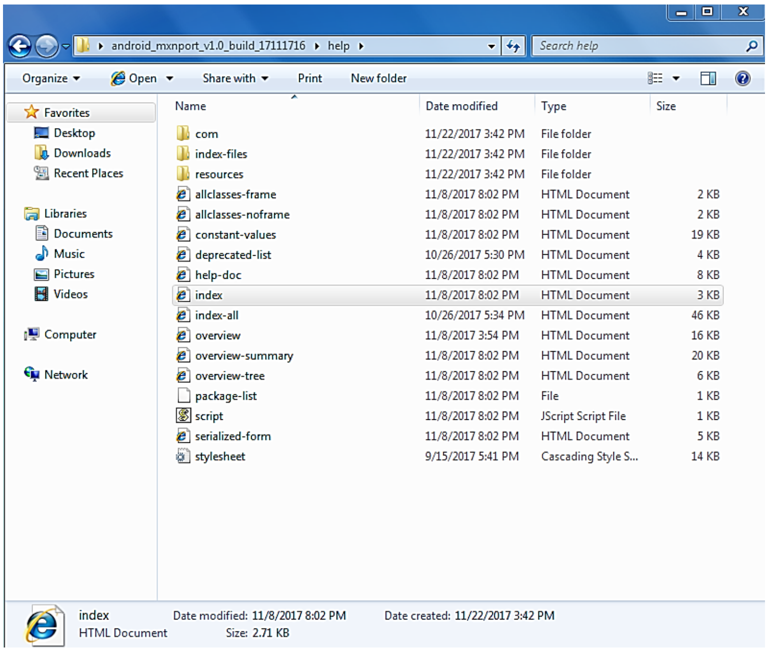


How to Start MxNPortAPI

You can download the MxNPortAPI from Moxa’s website at http://www.moxa.com and develop the application program in popular OSs, such as Windows, Linux, or Mac.

(You can refer the Android studio website to see the system requirements for development environment: https://developer.android.com/studio/index.html?hl=zh-tw#Requirements).

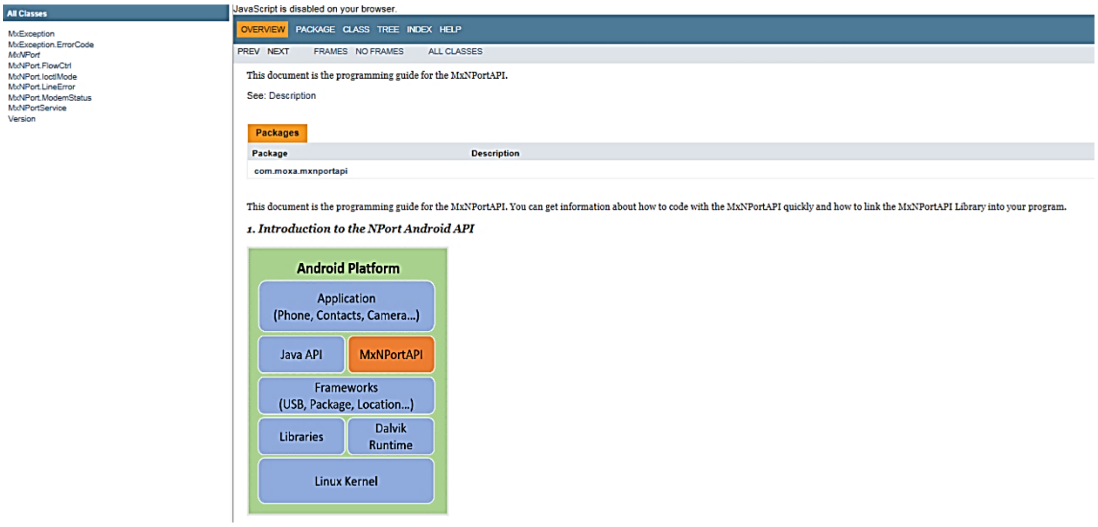
To start your application program, please unzip the MxNPortAPI file and refer to the index (.html) under the Help directory.



Name	Date modified	Type	Size
com	11/22/2017 3:42 PM	File folder	
index-files	11/22/2017 3:42 PM	File folder	
resources	11/22/2017 3:42 PM	File folder	
allclasses-frame	11/8/2017 8:02 PM	HTML Document	2 KB
allclasses-noframe	11/8/2017 8:02 PM	HTML Document	2 KB
constant-values	11/8/2017 8:02 PM	HTML Document	19 KB
deprecated-list	10/26/2017 5:30 PM	HTML Document	4 KB
help-doc	11/8/2017 8:02 PM	HTML Document	8 KB
index	11/8/2017 8:02 PM	HTML Document	3 KB
index-all	10/26/2017 5:34 PM	HTML Document	46 KB
overview	11/8/2017 3:54 PM	HTML Document	16 KB
overview-summary	11/8/2017 8:02 PM	HTML Document	20 KB
overview-tree	11/8/2017 8:02 PM	HTML Document	6 KB
package-list	11/8/2017 8:02 PM	File	1 KB
script	11/8/2017 8:02 PM	JScript Script File	1 KB
serialized-form	11/8/2017 8:02 PM	HTML Document	5 KB
stylesheet	9/15/2017 5:41 PM	Cascading Style S...	14 KB

index Date modified: 11/8/2017 8:02 PM Date created: 11/22/2017 3:42 PM
HTML Document Size: 2.71 KB

For more details about the installation, please refer to the Overview section.



JavaScript is disabled on your browser.

This document is the programming guide for the MxNPortAPI.

See: Description

Packages

Package	Description
com.moxa.mxnportapi	

This document is the programming guide for the MxNPortAPI. You can get information about how to code with the MxNPortAPI quickly and how to link the MxNPortAPI Library into your program.

1. Introduction to the NPort Android API

MxNPortAPI Function Groups

The supported functions in this API are listed below:

Port Control	Input/Output	Port Status Inquiry	Miscellaneous
open	read	getBaud	setBreak
close	write	getFlowCtrl	
setIoctlMode		getIoctlMode	
setFlowCtrl		getLineStatus	
setBaud		getModemStatus	
setRTS		getOQueue	
setDTR			
flush			

Example Program

To make sure this API is workable with the device server on an Android platform, see the example program below:

```
Thread thread = new Thread()
{
@Override
public void run() {
  /* Enumerate and initialize NPorts on system */
  List<MxNPort> NPortList = MxNPortService.getNPortInfoList();
  if(NPortList!=null){
        MxNPort.IoctlMode mode = new MxNPort.IoctlMode();
        mode.baudRate = 38400;
        mode.dataBits = MxNPort.DATA_BITS_8;
        mode.parity = MxNPort.PARITY_NONE;
        mode.stopBits = MxNPort.STOP_BITS_1;

        MxNPort mxNPort = NPortList.get(0); /* Get first NPort device */
        try {
              byte[] buf = {'H','e','l','l','o',' ','W','o','r','l','d'};
```

```
                    mxNPort.open(); /*open port*/
                    mxNPort.setIoctlMode(mode); /*serial parameters setting*/
    mxNPort.write(buf, buf.length); /*write data*/
                    mxNPort.close(); /*close port*/
                } catch (MxException e){
                    /*Error handling*/
                 }
               }
             }
        };
        thread.start();
```

Type section 1 content here.

A

Introduction to SCM (Serial Command Mode) Command Set

The following topics are covered in this appendix:

- ❒ **Command/Reply Format**
  - ➢ Single Line Command Format
  - ➢ Single Line Reply Format
  - ➢ Head and Tail Format
  - ➢ Operation Codes
  - ➢ Status Codes
  - ➢ Restriction
- ❒ **Command Code**
  - ➢ Command Code for Getting the Configuration
  - ➢ Command Codes for Setting the Configuration
  - ➢ Command Codes for Retrieving Running Configuration
  - ➢ Command Codes for Viewing the Status
  - ➢ Control Command Codes

Command/Reply Format

Single Line Command Format

Head	Op	Cmd.	Parameters	Tail
1 byte	1 byte	2 bytes	0 to n bytes	1 to 2 bytes

Single Line Reply Format

Head	Op	Cmd.	ST	Parameters	Tail
1 byte	1 byte	2 bytes	1 byte	0 to n bytes	1 to 2 bytes

Head and Tail Format

	Head 1 byte	---	**Tail** 1 or 2 bytes
Command	?	---	CR LF CR-LF
Reply	!	---	LF

Operation Codes

Operation Code	Meaning
G	Get configuration from MiiNePort's RAM
S	Set configuration to MiiNePort's RAM
R	Retrieve running configuration
V	View status
C	Control

Status Codes

Status Code	Meaning
E	Enter Serial Command Mode
0	Command was executed successfully
1	Unrecognized format
2	Operation is not valid
3	Command is not valid
4	Parameter is incorrect
5	Parameter is too long

Restriction

The total number of parameters in a single command cannot exceed 1024 characters.

Command Code

Command Code for Configuration

Device Name

Command code: BN

The command code for getting the configuration

Command parameters: N/A

Reply parameters: MiiNePort's name.

?GBN↵	Requests the configured device name for this MiiNePort.
!GBN0MiiNePort_E1_9527↵	The device name is MiiNePort_E1_9527

The command code for setting the configuration

Command parameters: The new device name for the MiiNePort.

Reply parameters: N/A

?SBNMiiNePort@Office↵	Sets the device name as 'MiiNePort@Office'.
!GBN0↵	The command was executed successfully.

The command code for retrieving the running configuration

Command parameters: N/A

Reply parameters: MiiNePort's name.

?RBN↵	Requests the configured device name for this MiiNePort.
!RBN0MiiNePort_9527↵	The device name is 'MiiNePort_9527'.

Console Password

Command code: BP

The command code for getting the configuration

Command parameters: N/A

Reply parameters: MiiNePort's console password.

?GBP↵	Requests the console's password for this MiiNePort.
!GBP01234↵	The console's password is '1234'.

The command code for setting the configuration

Command parameters: MiiNePort's console password.

Reply parameters: N/A

?SBP1234↵	Sets the console's password as '1234' for this MiiNePort.
!SBP0↵	The command was executed successfully.

The command code for retrieving the running configuration

Command parameters: N/A

Reply parameters: MiiNePort's console password.

?RBP↵	Requests the console password for this MiiNePort.
!RBP01234↵	The console password is '1234'.

Web Console

Command code: BH

The command code for getting the configuration

Command parameters: N/A

Reply parameters: 1 and the web console TCP port separated by a semicolon (;) if the web console is enabled, or a 0 if it is disabled.

?GBH↵	Requests the http console setting for this MiiNePort.
!GBH01;80↵	The http console is set as 'Enable' and the http port as '80'.

The command code for setting the configuration

Command parameters: 1 and the web console TCP port to enable the web console, or 0 to disable it.

Reply parameters: N/A

?SBH1;80↵	Sets the http console as 'Enable' and http port as '80' for this MiiNePort.
!SBH0↵	The command was executed successfully.

The command code for retrieving the running configuration

Command parameters: N/A

Reply parameters: 1 and the web console TCP port separated by a semicolon (;) if the web console is enabled, or a 0 if it is disabled.

?RBH↵	Requests the http console setting for this MiiNePort.
!RBH01;80↵	The http console is set as 'Enable' and the http port as '80'.

Telnet Console

Command code: BT

The command code for getting the configuration

Command parameters: N/A

Reply parameters: 1 and the Telnet console TCP port separated by a semicolon (;) if the Telnet console is enabled, or a 0 if it is disabled.

?GBT↵	Requests the Telnet console setting for this MiiNePort.
!GBT01;23↵	The Telnet console is set as 'Enable' and the Telnet port as '23'.

The command code for setting the configuration

Command parameters: 1 and the Telnet console TCP port to enable the Telnet console, or 0 to disable it.

Reply parameters: N/A

?SBT1;23↵	Sets the Telnet console as 'Enable' and the Telnet port as '23'.
!SBT0↵	The command was executed successfully.

The command code for retrieving the running configuration

Command parameters: N/A

Reply parameters: 1 and the Telnet console TCP port separated by a semicolon (;) if the Telnet console is enabled, or a 0 if it is disabled.

?RBT↵	Requests the Telnet console setting for this MiiNePort.
!RBT01;23↵	The MiiNePort reports the Telnet console as 'Enable' and the Telnet port as '23'.

Auto Configuration

Command code: BA

The command code for getting the configuration

Command parameters: N/A

Reply parameters: 1 if auto configuration is enabled, 0 otherwise.

?GBA↵	Requests the auto configuration setting for this MiiNePort.
!GBA01↵	Auto configuration is set as 'Enable'.

The command code for setting the configuration

Command parameters: 1 if auto configuration is enabled, 0 otherwise.

Reply parameters: N/A

?SBA1↵	Sets auto configuration as 'Enable'.
!SBA0↵	The command was executed successfully.

The command code for retrieving the running configuration

Command parameters: N/A

Reply parameters: 1 if auto configuration is enabled, 0 otherwise.

?RBA↵	Requests the auto configuration setting for this MiiNePort.
!RBA01↵	Auto configuration is set as 'Enable'.

External Reset Function

Command code: BE

The command code for getting the configuration

Command parameters: N/A

Reply parameters: 1 if the external reset function is enabled, 0 otherwise.

?GBE↵	Requests the external reset function setting for this MiiNePort.
!GBE01↵	The external reset function is set as 'Enable'.

The command code for setting the configuration

Command parameters: 1 if the external reset function is enabled, 0 otherwise.

Reply parameters: N/A

?SBE1↵	Sets the external reset function as 'Enable' for this MiiNePort.
!SBE0↵	The command was executed successfully.

The command code for retrieving the running configuration

Command parameters: N/A

Reply parameters: 1 if the external reset function is enabled, 0 otherwise.

?RBE↵	Requests the external reset function setting for this MiiNePort.
!RBE01↵	The external reset function is set as 'Enable'.

NECI (Utility Accessibility)

Command code: BU

The command code for getting the configuration

Command parameters: N/A

Reply parameters: 1 if NECI is enabled, 0 otherwise.

?GBU↵	Requests the NECI setting for this MiiNePort.
!GBU01↵	NECI is set as 'Enable'.

The command code for setting the configuration

Command parameters: 1 if NECI is enabled, 0 otherwise.

Reply parameters: N/A

?SBU1↵	Sets the NECI setting as 'Enable' for this MiiNePort.
!SBU0↵	The command was executed successfully.

The command code for retrieving the running configuration

Command parameters: N/A

Reply parameters: 1 if NECI is enabled, 0 otherwise.

?RBU↵	Requests the NECI setting for this MiiNePort.
!RBU01↵	NECI is set as 'Enable'.

IP Configuration

Command code: NC

IP Configuration Index

0	Static
1	DHCP
2	DHCP/BOOTP
3	BOOTP
4	AUTOIP

The command code for getting the configuration

Command parameters: N/A

Reply parameters: The MiiNePort's IP configuration index as in the above IP Configuration Index table

?GNC↵	Requests the IP configuration setting for this MiiNePort.
!GNC00↵	The IP configuration is set as 'Static'.

The command code for setting the configuration

Command parameters: The MiiNePort's IP configuration index as shown in the above IP configuration Index table

Reply parameters: N/A

?SNC0↵	Sets the IP configuration as 'Static'.
!SNC0↵	The command was executed successfully.

The command code for retrieving the running configuration

Command parameters: N/A

Reply parameters: The MiiNePort's IP configuration index as in the above IP Configuration Index table

?RNC↵	Requests the IP configuration setting for this MiiNePort.
!RNC00↵	The IP configuration is set as 'Static'.

IP Address

Command code: NI

The command code for getting the configuration

Command parameters: N/A

Reply parameters: The MiiNePort's IP address.

?GNI↵	Requests the IP address for this MiiNePort.
!GNI0192.168.127.254↵	The IP address is set as '192.168.127.254'.

The command code for setting the configuration

Command parameters: The MiiNePort's IP address.

Reply parameters: N/A

?SNI192.168.1.2↵	Sets the IP address as '192.168.1.2'.
!SNI0↵	The command was executed successfully.

The command code for retrieving the running configuration

Command parameters: N/A

Reply parameters: The MiiNePort's IP address.

?RNI↵	Requests the IP address for this MiiNePort.
!RNI0192.168.127.254↵	The IP address is set as '192.168.127.254'.

Netmask

Command code: NM

The command code for getting the configuration

Command parameters: N/A

Reply parameters: The MiiNePort's netmask address.

?GNM↵	Requests the netmask address for this MiiNePort.
!GNM0255.255.255.0↵	The netmask address is set as '255.255.255.0'.

The command code for setting the configuration

Command parameters: The MiiNePort's netmask address.

Reply parameters: N/A

?SNM255.255.255.0↵	The system sets the netmask address as '255.255.255.0'.
!SNM0↵	The command was executed successfully.

The command code for retrieving the running configuration

Command parameters: N/A

Reply parameters: The MiiNePort's netmask address.

?RNM↵	Requests the netmask address for this MiiNePort.
!RNM0255.255.255.0↵	The netmask address is set as '255.255.255.0'.

Gateway

Command code: NG

The command code for getting the configuration

Command parameters: N/A

Reply parameters: The MiiNePort's gateway address.

?GNG↵	Requests the gateway address for this MiiNePort.
!GNG0255.255.255.255↵	The gateway address is set as '255.255.255.255'.

The command code for setting the configuration

Command parameters: The MiiNePort's gateway address.

Reply parameters: N/A

?SNG192.168.1.254↵	Sets the gateway address as '192.168.1.254'.
!SNG0↵	The command was executed successfully.

The command code for retrieving the running configuration

Command parameters: N/A

Reply parameters: The MiiNePort's gateway address.

?RNG↵	Requests the gateway address for this MiiNePort.
!RNM0255.255.255.255↵	The gateway address is set as '255.255.255.255'.

DNS

Command code: ND

The command code for getting the configuration

Command parameters: The index (1 or 2) of the DNS server.

Reply parameters: The MiiNePort's DNS address.

?GND1↵	Requests DNS Server 1's address for this MiiNePort.
!GND0192.168.1.2↵	DNS Server 1's address is set as '192.168.1.2'.

The command code for setting the configuration

Command parameters: The index (1,2) and the DNS server's address, separated by a semicolon (;).

Reply parameters: N/A

?SND1;192.168.1.123↵	Sets DNS Server 1's address as '192.168.1.123'.
!SND0↵	The command was executed successfully.

The command code for retrieving the running configuration

Command parameters: The index (1 or 2) of DNS server.

Reply parameters: The MiiNePort's DNS address.

?RND1↵	Requests DNS Server 1's address for this MiiNePort.
!RND0192.168.1.2↵	NS Server 1's address is set as '192.168.1.2'.

PHY Speed

Command code: NS

PHY Speed Index

0	PHY auto negotiation is enabled.
10	PHY speed is forced to 10Mbps with half-duplex
11	PHY speed is forced to 10Mbps with full-duplex
100	PHY speed is forced to 100Mbps with half-duplex
101	PHY speed is forced to 100Mbps with full-duplex

The command code for getting the configuration

Command parameters: N/A

Reply parameters: The MiiNePort's PHY speed as in the above PHY Speed Index table

?GNS↵	Requests the PHY speed setting for this MiiNePort.
!GNS00↵	The PHY speed as determined by the PHY auto-negotiation function.

The command code for setting the configuration

Command parameters: N/A

Reply parameters: The MiiNePort's PHY speed as in the above PHY Speed Index table:

?SNS10↵	Forces the PHY speed to 10Mbps with half-duplex.
!SNS0↵	The command was executed successfully.

The command code for retrieving the running configuration

Command parameters: N/A

Reply parameters: The MiiNePort's PHY speed as in the above PHY Speed Index table:

?RNS↵	Requests the PHY speed setting for this MiiNePort.
!RNS00↵	The PHY speed is determined by the PHY auto-negotiation function.

Enable/Disable SNMP

Command code: MS

The command code for getting the configuration

Command parameters: N/A

Reply parameters: 1 (Enable) or 0 (Disable) the MiiNePort's SNMP agent.

?GMS↵	Requests to enable or disable the SNMP agent for this MiiNePort.
!GMS01↵	The SNMP agent is set as 'Enable'.

The command code for setting the configuration

Command parameters: 1 (Enable) or 0 (Disable) the MiiNePort's SNMP agent.

Reply parameters: N/A

?SMS1↵	Sets the SNMP agent as 'Enable'.
!SMS0↵	The command was executed successfully.

The command code for retrieving the running configuration

Command parameters: N/A

Reply parameters: 1 (Enable) or 0 (Disable) the MiiNePort's SNMP agent.

?RMS↵	Requests to enable or disable the SNMP agent for this MiiNePort.
!RMS01↵	The SNMP agent is set as 'Enable'.

Community String

Command code: MU

The command code for getting the configuration

Command parameters: N/A

Reply parameters: The MiiNePort's SNMP community string.

?GMU↵	Requests the SNMP community string for this MiiNePort.
!GMU0public↵	The SNMP community string is set as 'public'.

The command code for setting the configuration

Command parameters: The MiiNePort's SNMP community string.

Reply parameters: N/A

?SMUpublic↵	Sets the SNMP community string as 'public'.
!SMU0↵	The command was executed successfully.

The command code for retrieving the running configuration

Command parameters: N/A

Reply parameters: The MiiNePort's SNMP community string.

?RMU↵	Requests the SNMP community string for this MiiNePort.
!RMU0public↵	The SNMP community string is set as 'public'.

Contact Name

Command code: MN

The command code for getting the configuration

Command parameters: N/A

Reply parameters: The MiiNePort's SNMP contact name.

?GMN↵	Rquests the SNMP contact name for this MiiNePort.
!GMN0s_name↵	The SNMP contact name is set as 's_name'.

The command code for setting the configuration

Command parameters: The MiiNePort's SNMP contact name.

Reply parameters: N/A

?SMNcontact↵	Sets the SNMP contact name as 'contact'.
!SMN0↵	The command was executed successfully.

The command code for retrieving the running configuration

Command parameters: N/A

Reply parameters: The MiiNePort's SNMP contact name.

?RMN↵	Requests the SNMP contact name for this MiiNePort.
!RMN0s_name↵	The SNMP contact name is set as 's_name'.

Location

Command code: ML

The command code for getting the configuration

Command parameters: N/A

Reply parameters: The MiiNePort's SNMP location.

?GML↵	Requests the SNMP location for this MiiNePort.
!GML0s_location↵	The SNMP location is set as 's_location'.

The command code for setting the configuration

Command parameters: The MiiNePort's SNMP location.

Reply parameters: N/A

?SMLlocation↵	The system sets the SNMP contact name as 'location'.
!SML0↵	The command was executed successfully.

The command code for retrieving the running configuration

Command parameters: N/A

Reply parameters: The MiiNePort's SNMP location.

?RML↵	Requests the SNMP location for this MiiNePort.
!RML0s_location↵	The SNMP location is set as 's_location'.

Enable/Disable Accessible IP List

Command code: AS

The command code for getting the configuration

Command parameters: N/A

Reply parameters: Enable (1) or Disable (0) the MiiNePort's Accessible IP list.

	Requests to enable or disable the Accessible IP list for this MiiNePort.
!GNS01↵	The Accessible IP list is set as 'Enable'.

The command code for setting the configuration

Command parameters: 1 (Enable) or 0 (Disable) the MiiNePort's Accessible IP list.

Reply parameters: N/A

?SAS1↵	Sets the Accessible IP list as 'Enable'.
!SAS0↵	The command was executed successfully.

The command code for retrieving the running configuration

Command parameters: N/A

Reply parameters: Enable (1) or Disable (0) the MiiNePort's Accessible IP list.

?RNS↵	Requests to enable or disable the Accessible IP list for this MiiNePort.
!RNS01↵	The Accessible IP list is set as 'Enable'.

Accessible IP List

Command code: AI

The command code for getting the configuration

Command parameters: The index of the Accessible IP list (from 1 to 16)

Reply parameters: Return format Mode; IP; Netmask in the Accessible IP list. If mode equals 1, it is active; otherwise, it is inactive.

?GAI1↵	Requests the first Accessible IP list for this MiiNePort.
!GAI01;192.168.1.2;255.255.255.0↵	The first Accessible IP list is set as 'Active', the IP address as '192.167.1.2', and the netmask address as '255.255.255.0'.

The command code for setting the configuration

Command parameters: The format is "index;mode;IP;Netmask", with the index ranging from 1 to 16; mode is 1 if activated and 0 if not activated.

Reply parameters: N/A

?SAI1;1;192.168.1.2;255.255.255.0 ↵	Sets Accessible IP 1 as 'active', the IP address as '192.168.1.2,' and the netmask address as '255.255.255.0'.
!SAI0↵	The command was executed successfully.

The command code for retrieving the running configuration

Command parameters: The index of accessible IP list (from 1 to 16)

Reply parameters: Return format Mode;IP;Netmask in the accessible IP list. If mode equals 1, it's active; otherwise, it's inactive.

?RAI1↵	Requests first accessible IP list for this MiiNePort.
!RAI01;192.168.1.2;255.255.255.0↵	Set the first accessible IP list as 'Active', the IP address as '192.167.1.2', and the netmask address as '255.255.255.0' .

Auto IP Report

Command code: NR

The command code for getting the configuration

Command parameters: N/A

Reply parameters: The MiiNePort's Auto IP report setting.

?GNR↵	Requests the Auto IP report for this MiiNePort.
!GNR0192.168.1.250:4000;50↵	The Auto IP report server is set as '192.168.1.250', the UDP port as '4000', and the report period to '50'.

The command code for setting the configuration

Command parameters: Auto IP report destination and report period.

Reply parameters: N/A

?SNR192.168.1.123:4000;50↵	Sets the Auto report IP as '192.168.1.123', the UDP port as '4000', and the report period to '50' sec.
!SNR0↵	The command was executed successfully.

The command code for retrieving the running configuration

Command parameters: N/A

Reply parameters: The MiiNePort's Auto IP report setting.

?RNR↵	Requests the Auto IP report for this MiiNePort.
!RNR0192.168.1.250:4000;50↵	The Auto IP report server is set as '192.168.1.250', the UDP port as '4000', and the report period to '50'.

Port Alias

Command code: SA

The command code for getting the configuration

Command parameters: Port index.

Reply parameters: The MiiNePort's port alias.

?GSA1↵	Requests port's 1 alias for this MiiNePort.
!GSA0port1↵	The port alias is set as 'port1'.

The command code for setting the configuration

Command parameters: Port index and baudrate are separated by a semicolon (;).

Reply parameters: N/A

?SSA1;port 1↵	Sets port 1's alias name as 'port 1'.
!SSA0↵	The command was executed successfully.

The command code for retrieving the running configuration

Command parameters: N/A

Reply parameters: THE MiiNePort's port alias.

?RSA↵	Requests port alias for this MiiNePort.
!RSA0port1↵	The port alias is set as 'port1'.

Baudrate

Command code: SB

The command code for getting the configuration

Command parameters: Port index.

Reply parameters: The MiiNePort's baudrate.

?GSB1↵	Requests port 1's baudrate for this MiiNePort.
!GSB0115200↵	The baudrate is set as '115200'.

The command code for setting the configuration

Command parameters: Port index and baudrate are separated by a semicolon (;).

Reply parameters: N/A

?SSB1;115200↵	Sets port 1's baudrate as '115200'.
!SSB0↵	The command was executed successfully.

The command code for retrieving the running configuration

Command parameters: Port index.

Reply parameters: The MiiNePort's baudrate.

?RSB1↵	Requests port 1's baudrate for this MiiNePort.
!RSB0115200↵	Reports the baudrate as '115200'.

Data Bits

Command code: SD

The command code for getting the configuration

Command parameters: Port index.

Reply parameters: The MiiNePort’s data bits.

?GSD1↵	Requests port 1’s data bits for this MiiNePort.
!GSD08↵	Data bits is set as ‘8’.

The command code for setting the configuration

Command parameters: Port index.

Reply parameters: The MiiNePort’s data bits.

?RSD1↵	Requests port 1’s data bits for this MiiNePort.
!RSD08↵	Data bits is set as ‘8’.

The command code for retrieving the running configuration

Command parameters: Port index and data bits are separated by a semicolon (;).

Reply parameters: N/A

?SSD1;8↵	Sets port 1’s data bits as ‘8’.
!SSD0↵	The command was executed successfully.

Parity

Command code: SP

Parity Index

0	None
1	Odd
2	Even
3	Mark
4	Space

The command code for getting the configuration

Command parameters: Port index.

Reply parameters: The MiiNePort’s parity index as per above Parity Index table.

?GSP1↵	The system requests port 1’s parity for this MiiNePort.
!GSP00↵	MiiNePort reports parity as ‘None’.

The command code for setting the configuration

Command parameters: Port index and parity separated are by a semicolon (;). The MiiNePort’s parity index as shown in the above Parity Index table.

Reply parameters: N/A

?SSP1;0↵	Sets port 1’s parity bit as ‘None’.
!SSP0↵	The command was executed successfully.

The command code for retrieving the running configuration

Command parameters: Port index

Reply parameters: The MiiNePort's parity index as per above Parity Index table.

?RSP1↵	Requests port 1's parity for this MiiNePort.
!RSP00↵	Parity is set as 'None'.

Stop Bits

Command code: SP

Stop Bits Index

0	Stop bit is 1
1	Stop bit is 1.5
2	Stop bit is 2

The command code for getting the configuration

Command parameters: Port index.

Reply parameters: As in the above Stop Bits Index table.

?GSS1↵	Requests port 1's stop bits for this MiiNePort.
!GSS00↵	Stop bits is set as '1'.

The command code for setting the configuration

Command parameters: Port index and stop bits are separated by a semicolon (;). The MiiNePort's stop bits index as shown in the above Stop Bits Index table:

Reply parameters: N/A

?SSS1;0↵	Sets port 1's stop bits as '1'.
!SSS0↵	The command was executed successfully.

The command code for retrieving the running configuration

Command parameters: Port index.

Reply parameters: The MiiNePort's stop bits index as shown in the above Stop Bits Index table.

?RSS1↵	Requests port 1's stop bits for this MiiNePort.
!RSS00↵	Stop bits is set as '1'

Flow control

Command code: SL

Flow Control Index

0	None
1	RTS/CTS
2	XON/XOFF
3	DTR/DSR

The command code for getting the configuration

Command parameters: Port index.

Reply parameters: The MiiNePort's flow control index as in the above Flow Control Index table.

?GSL1↵	Requests port 1's flow control for this MiiNePort.
!GSL01↵	Flow control is set as 'RTS/CTS'.

The command code for setting the configuration

Command parameters: Port index and flow control are separated by a semicolon (;). The MiiNePort's flow control index as shown in the above Flow Control Index table.

Reply parameters: N/A

?SSL1;1↵	Sets port 1's flow control as 'RTS/CTS'.
!SSL0↵	The command was executed successfully.

The command code for retrieving the running configuration

Command parameters: Port index.

Reply parameters: The MiiNePort's flow control index as seen in the above Flow Control Index table.

?RSL1↵	Requests port 1's flow control for this MiiNePort.
!RSL01↵	Flow control is set as 'RTS/CTS'.

FIFO

Command code: SF

The command code for getting the configuration

Command parameters: Port index.

Reply parameters: 1 (Enable) or 0 (Disable) the MiiNePort's FIFO.

?GSF1↵	Requests port 1's FIFO for this MiiNePort.
!GSF01↵	FIFO is set as 'Enable'.

The command code for setting the configuration

Command parameters: Port index and FIFO setting are separated by a semicolon (;). The FIFO setting is 1 (Enable) or 0 (Disable).

Reply parameters: N/A.

?SSF1;1↵	Sets port 1's FIFO as 'Enable'.
!SSF0↵	The command was executed successfully.

The command code for retrieving the running configuration

Command parameters: Port index.

Reply parameters: 1 (Enable) or 0 (Disable) the MiiNePort’s FIFO.

?RSF1↵	Requests port 1’s FIFO for this MiiNePort.
!RSF01↵	FIFO is set as ‘Enable’.

Operation Mode

Command code: OM

Operation Mode Index

0	Disable
1	Real COM
2	TCP
3	UDP
4	MCSC (This value is unavailable for MCSC channels.)

The command code for getting the configuration

Command parameters: Port index and MCSC channel index are separated by a semicolon (;). For the port itself, the channel index is 0.

Reply parameters: The MiiNePort’s operation mode index is as shown in the above Operation Mode Index table.

?GOM1;0↵	Requests port 1’s operation mode for this MiiNePort.
!GOM04↵	The operation mode is set as ‘MCSC’.
?GOM1;2↵	When port 1 is set as MCSC, the operation mode for channel 2 of port 1 is requested.
!GOM02↵	The operation mode is set as `TCP’.

The command code for setting the configuration

Command parameters: Port index, MCSC channel index, and operation mode to set. Parameters are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0. The operation mode is as shown in above operation mode index table.

Reply parameters: N/A

?SOM1;0;2↵	Sets port 1’s operation mode as ‘TCP’.
!SOM0↵	The command was executed successfully.

The command code for retrieving the running configuration

Command parameters: The port index and MCSC channel index are separated by a semicolon (;). For the port itself, the channel index is 0.

Reply parameters: The MiiNePort’s operation mode index is as shown in the above Operation Mode Index table.

?ROM1;0↵	Requests port 1’s operation mode for this MiiNePort.
!ROM04↵	Flow control is set as ‘MCSC’.
?ROM1;2↵	When port 1 is set to MCSC, the operation mode for channel 2 of port 1 is requested.
!GOM02↵	The operation mode is set as`TCP’.

Data Packing Length

Command code: OL

The command code for getting the configuration

Command parameters: The port index and the MCSC channel index are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: The MiiNePort's data packing length is as follows:

?GOL1;0↵	Requests port 1's data packing length for this MiiNePort.
!GOL0256↵	The data packing length is set as '256'.
?GOL1;2	When port 1 is set to MCSC, the data packing length for channel 2 of port 1 is requested.
!GOL0128	The data packing length is set as `128'.

The command code for setting the configuration

Command parameters: Port index, MCSC channel index, and data packing length to set. Parameters are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: N/A

?SOL1;0;256↵	Sets port 1's data packing length as '256'.
!SOL0↵	The command was executed successfully.

The command code for retrieving the running configuration

Command parameters: The The port index and the MCSC channel index are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: The MiiNePort's data packing length is as follows:

?ROL1;0↵	The system requests port 1's data packing length for this MiiNePort.
!ROL0256↵	The MiiNePort reports the data packing length as '256'.

Delimiter

Command code: OD

The command code for getting the configuration

Command parameters: The port index and the MCSC channel index are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: The MiiNePort's delimiter setting. (delimiter1 enable/disable; hex1; delimiter2 enable/disable; hex2)

?GOD1;0↵	The system requests port 1's delimiter setting for this MiiNePort.
!GOD1;10;1;13↵	The MiiNePort reports delimiter 1 as 'Enable' and hex code as '10', delimiter 2 as 'Enable' and hex code as '13'.

The command code for setting the configuration

Command parameters: Port index, MCSC channel index, delimiter1 enable/disable, delimiter character 1 by hex, delimiter 2 enable/disable, and delimiter character 2 by hex. Parameters are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: N/A

?SOD1;2;1;10;1;13↵	Sets channel 2 of port 1's delimiter 1 as 'Enable' and hex code as '10'. Sets Delimiter 2 as 'Enable' and hex code as '13'.
!SOD0↵	The command was executed successfully.

The command code for the retrieving running configuration

Command parameters: The port index and the MCSC channel index are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: The MiiNePort's delimiter setting. (delimiter1 enable/disable; hex1; delimiter2 enable/disable; hex2)

?ROD1;0↵	Requests port 1's delimiter setting for this MiiNePort.
!ROD1;10;1;13↵	Delimiter 1 is set as 'Enable' and hex code as '10', Delimiter 2 as 'Enable' and hex code as '13'

Match Bytes

Command code: OY

The command code for getting the configuration

Command parameters: The port index and MCSC channel index are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: The MiiNePort's match bytes.

?GOY1;0↵	Requests port 1's match bytes for this MiiNePort.
!GOY02↵	Match bytes set as '2'.

The command code for setting the configuration

Command parameters: Port index, MCSC channel index, and match bytes. Parameters are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: N/A

?SOY1;0;2↵	The system sets port 1's match bytes as '2'.
!SOY0↵	The command was executed successfully.

The command code for retrieving the running configuration

Command parameters: The port index and the MCSC channel index are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: The MiiNePort's match bytes.

?ROY1;0↵	Requests port 1's match bytes for this MiiNePort.
!ROY02↵	Match bytes set as '2'.

Delimiter Process

Command code: OT

Delimiter Process Index.

0	Do nothing
1	Delimiter+1
2	Delimiter+2

The command code for getting the configuration

Command parameters: The port index and the MCSC channel index are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: The MiiNePort's delimiter process is as shown in the above Delimiter Process Index table.

?GOT1;0↵	Requests port 1's delimiter process for this MiiNePort.
!GOT02↵	The delimiter process is set as 'Delimiter+2'.

The command code for setting the configuration

Command parameters: N/A

Reply parameters: Port index, MCSC channel index, and delimiter process. Parameters are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

?SOT1;0;2↵	Sets port 1's delimiter process as 'Delimiter+2'.
!SOT0↵	The command was executed successfully.

The command code for retrieving running configuration

Command parameters: The port index and the MCSC channel index are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: The MiiNePort's delimiter process is as shown in the above Delimiter Process Index table.

?ROT1;0↵	Requests port 1's delimiter process for this MiiNePort.
!ROT02↵	The delimiter process is set as 'Delimiter+2'..

Force Transmit

Command code: OF

The command code for getting the configuration

Command parameters: The port index and the MCSC channel index are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: The MiiNePort's force transmit timeout.

?GOF1;0↵	Requests port 1's force transmit timeout for this MiiNePort.
!GOF00↵	The force transmit timeout is set as '0' sec.

The command code for setting the configuration

Command parameters: Port index, MCSC channel index, and force transmit timeout. Parameters are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: N/A

?SOF1;0;5↵	Sets port 1's force transmit timeout as '5'.
!SOF0↵	The command was executed successfully.

The command code for retrieving running configuration

Command parameters: The port index and the MCSC channel index are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: The MiiNePort's force transmit timeout.

?ROF1;0↵	Requests port 1's force transmit timeout for this MiiNePort.
!ROF00↵	The force transmit timeout is set to '0' sec.

Real COM TCP Alive Check Time

Command code: RA

The command code for getting the configuration

Command parameters: The port index and the MCSC channel index are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: The MiiNePort's TCP alive check time.

?GRA1;0↵	Requests port 1's TCP alive check time for port 1.
!GRA05↵	The alive check time is set to '5' min.

The command code for setting the configuration

Command parameters: Port index, MCSC channel index, and TCP alive check time. Parameters are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: N/A.

?SRA1;0;5↵	Sets port 1's TCP alive check time to '5' min. for port 1.
!SRA0↵	The command was executed successfully.

The command code for retrieving running configuration

Command parameters: Tport index and the MCSC channel index are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: The MiiNePort's TCP alive check time.

?RRA1;0↵	Requests port 1's TCP check alive time for port 1.
!RRA05↵	The TCP alive check time is set to '5' min.

Real COM Max Connection

Command code: RM

The command code for getting the configuration

Command parameters: The port index and MCSC channel index are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: The MiiNePort's TCP maximum connection number.

?GRM1;1↵	Requests the maximum connection number for port 1 and channel 1.
!GRM04↵	The maximum connection number is set as '4'.

The command code for setting the configuration

Command parameters: Port index, MCSC channel index, and maximum connection number. Parameters are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: N/A

?SRM1;1;4↵	Sets the maximum connection number as '4' for port 1 and channel 1.
!SRM0↵	The command was executed successfully.

The command code for retrieving the running configuration

Command parameters: The port index and MCSC channel index are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: MiiNePort's TCP maximum connection number.

?RRM1;1↵	Requests the maximum connection number for port 1 and channel 1.
!RRM03↵	The maximum connection number is set as '3'.

Real COM Ignore Jammed IP

Command code: RJ

The command code for getting the configuration

Command parameters: The port index and MCSC channel index are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: 1 (Enable) or 0 (Disable)

?GRJ1;2↵	Requests the jammed IP policy for port 1's channel 2.
!GRJ01↵	The ignore jammed IP is set as 'Enable'.

The command code for setting the configuration

Command parameters: Port index, MCSC channel index, and ignore jammed IP setting. Parameters are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0. Ignore jammed IP setting is 1 (Enable) or 0 (Disable).

Reply parameters: N/A

?SRJ1;2;0↵	The system sets ignore jammed IP policy as 'Disable' for port 1's channel 2.
!SRJ0↵	The command was executed successfully.

The command code for retrieving the running configuration

Command parameters: The port index and MCSC channel index are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: 1 (Enable) or 0 (Disable)

?RRJ1;2↵	Requests the jammed IP policy for port 1's channel 2.
!RRJ01↵	The MiiNePort reports ignore jammed IP as 'Enable'.

Real COM Allows Driver Control

Command code: RD

The command code for getting the configuration

Command parameters: The port index and MCSC channel index are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: 1 (Enable) or 0 (Disable)

?GRD1;0↵	Requests the allow driver control policy for port 1.
!GRD01↵	Allow driver control is set as ‘Enable’.

The command code for setting the configuration

Command parameters: Port index, MCSC channel index, and match bytes. Parameters are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0. Allow driver control is 1 (Enable) or 0 (Disable).

Reply parameters: N/A

?SRD1;0;0↵	Sets the allow driver control policy as ‘Disable’ for port 1.
!SRD0↵	The command was executed successfully.

The command code for retrieving the running configuration

Command parameters: The port index and MCSC channel index are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: 1 (Enable) or 0 (Disable)

?RRD1;0↵	Requests the allow driver control policy for port 1.
!RRD01↵	Allow driver control set as ‘Enable’.

TCP Role

Command code: TO

TCP Role Index

0	TCP server
1	TCP client
2	Mixed

The command code for getting the configuration

Command parameters: The port index and MCSC channel index are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: The TCP role is shown in the above TCP Role Index table.

?GTO1;0↵	Requests a TCP role for port 1.
!GTO00↵	The TCP role is set as TCP server.

The command code for setting the configuration

Command parameters: Port index, MCSC channel index, and TCP role. Parameters are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0. The TCP role is as in the above TCP Role Index table.

Reply parameters: N/A

?STO1;0;1↵	Sets port 1’s TCP role to TCP client’.
!STO0↵	The command was executed successfully.

The command code for retrieving the running configuration

Command parameters: The port index and MCSC channel index are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: The TCP role is shown in the above TCP Role Index table.

?RTO1;0↵	Requests a TCP role for port 1.
!RTO00↵	The TCP role is set as TCP server.

TCP Server Connection Control

Command code: TS

TCP Server Connection Index

0	Always accept
1	Accept with DSR on

The command code for getting the configuration

Command parameters: The port index and MCSC channel index are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: The TCP server connection control setting is as shown in the above TCP Server Connection Index table.

?GTS1;0↵	Requests the TCP server connection control for port 1.
!GTS00↵	The incoming policy is set as ‘Always accept’.

The command code for setting the configuration

Command parameters: Port index, MCSC channel index, and server connection control setting. Parameters are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0. The server connection control setting is as in the above TCP Server Connection Index table

Reply parameters: N/A

?STS1;0;0↵	Sets TCP server connection control as ‘Always accept’ for port 1.
!STS0↵	The command was executed successfully.

The command code for retrieving the running configuration

Command parameters: The port index and MCSC channel index are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: TCP server connection control as shown in the above TCP server connection index table.

?RTS1;0↵	Requests the accept incoming policy for port 1.
!RTS00↵	The incoming policy is set as ‘Always accept’.

TCP Password

Command code: TW

The command code for getting the configuration

Command parameters: The port index and MCSC channel index are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: 1 (Require) or 0 (Don't require) and password.

?GTW1;0↵	Requests port 1's password setting.
!GTW01;1234↵	The MiiNePort reports the password as 'required' and password as '1234'.

The command code for setting the configuration

Command parameters: Port index, MCSC channel index, require (1) TCP connection password or not (0), and TCP connection password if required. Parameters are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: N/A

?STW1;0;1;1234↵	Sets the password as 'required' and password as '1234' for port 1.
!STW0↵	The command was executed successfully.
?STW1;1;0↵	Sets that a password is `not required' for port 1's channel 1.
!STW0↵	The command was executed successfully.

The command code for retrieving running configuration

Command parameters: The port index and MCSC channel index are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: 1 (Require) or 0 (Don't require) and password.

?RTW1;0↵	Requests port 1's password setting for port 1.
!RTW01;1234↵	The MiiNePort reports the password as 'required' and password as '1234'.

TCP Communication Protocol

Command code: TR

The command code for getting the configuration

Command parameters: The port index and MCSC channel index are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: 0 (Raw TCP) or 1 (RFC 2217)

?GTR1;0↵	Requests the communication protocol for port 1.
!GTR00↵	The communication protocol is set as 'Raw TCP'.

The command code for setting the configuration

Command parameters: Port index, MCSC channel index, and communication protocol (0 for raw TCP and 1 for RFC-2217). Parameters are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: N/A

?STR1;0;0↵	Sets the communication protocol as 'Raw TCP' for port 1.
!STR0↵	The command was executed successfully.

The command code for retrieving the running configuration

Command parameters: The port index and MCSC channel index are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: 0 (Raw TCP) or 1 (RFC 2217)

?RTR1;0↵	Requests the communication protocol for port 1.
!RTR00↵	The communication protocol is set as 'Raw TCP'.

TCP Alive Check Time

Command code: TA

The command code for getting the configuration

Command parameters: The port index and MCSC channel index are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: TCP alive check time

?GTA1;0↵	Requests the TCP alive check time for port 1.
!GTA05↵	The TCP alive check time is set to '5' minutes.

The command code for setting the configuration

Command parameters: Port index, MCSC channel index, and TCP alive check time. Parameters are separated by a semicolon (;).

Reply parameters: N/A

?STA1;0;5↵	Sets port 1's TCP alive check time to `5' minutes.
!STA0↵	The command was executed successfully.

The command code for retrieving the running configuration

Command parameters: The port index and MCSC channel index are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: TCP alive check time.

?RTA1;0↵	Requests the TCP alive check time for port 1.
!RTA05↵	The TCP alive check time is set to '5' minutes.

TCP Port

Command code: TP

The command code for getting the configuration

Command parameters: The port index and MCSC channel index are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: TCP port

?GTP1;0↵	Requests the TCP port number for port 1.
!GTP04100↵	The TCP port number is set as '4100'.

The command code for setting the configuration

Command parameters: Port index, MCSC channel index, and TCP local port. Parameters are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: N/A

?STP1;0;4100↵	Sets the TCP port as ‘4100’ for port 1.
!STP0↵	The command was executed successfully.

The command code for retrieving the running configuration

Command parameters: The port index and MCSC channel index are separated by a semicolon (;). For an MCSC disabled port, the channel index is 0.

Reply parameters: TCP port.

?RTP1;0↵	Requests the TCP port number for port 1.
!RTP04100↵	The TCP port number is set as ‘4100’.

TCP Client Connection Control

Command code: TC

TCP Client Connection Control Index

0	Any character
1	Manual connection
2	Start up
3	With DSR on

The command code for getting the configuration

Command parameters: The port index and MCSC channel index are separated by a semicolon (;). For the MCSC-disabled port, the channel index is 0.

Reply parameters: As shown in the TCP Client Connection Control Index table above.

?GTC1;0↵	Requests TCP client connection control for port 1.
!GTC01↵	TCP client connection control is set as ‘Any character’.

The command code for setting the configuration

Command parameters: Port index, MCSC channel index, and client connection control setting. Parameters are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: N/A

?STC1;0;1↵	Sets TCP client connection control as ‘Manual connection’ for port 1.
!STC0↵	The command was executed successfully.

The command code for retrieving the running configuration

Command parameters: The port index and MCSC channel index are separated by a semicolon (;). For the MCSC-disabled port, the channel index is 0.

Reply parameters: As shown in the TCP Client Connection Control Index table.

?RTC1;0↵	Requests TCP client connection control for port 1.
!RTC01↵	TCP client connection control is set as ‘Any character’.

TCP Connection Response

Command code: TN

The command code for getting the configuration

Command parameters: The port index and MCSC channel index are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: 1 (Enable) or 0 (Disable)

?GTN1;0↵	Requests the connection response policy for port 1.
!GTN00↵	The connection response registered as 'Disable'.

Command code for setting the configuration

Command parameters: Port index, MCSC channel index, and client connection response (0 for disable and 1 for enable). Parameters are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: N/A

?STN1;0;1↵	Sets the connection response as 'Enable' for port 1.
!STN0↵	The command was executed successfully.

The command code for retrieving the running configuration

Command parameters: The port index and MCSC channel index are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: 1 (Enable) or 0 (Disable).

?RTN1;0↵	Requests the connection response policy for port 1.
!RTN00↵	The connection response is set as 'Disable'.

TCP Destination Address

Command code: TI

TCP Destination Address Index

0	Destination address
1	Alternate address 1
2	Alternate address 2
3	Alternate address 3

The command code for getting the configuration

Command parameters: The port index, the MCSC channel index, and the TCP destination address index are separated by a semicolon (;) denote. For the MCSC-disabled port, the channel index is 0. The destination address index is as shown in the above TCP Destination Address Index table.

Reply parameters: TCP destination address and port

?GTI1;0;0↵	Requests the destination address for port 1.
!GTI0192.168.1.2:4001↵	The destination address is set as '192.168.1.2' and the port number as '4001'.

The command code for setting the configuration

Command parameters: Port index, MCSC channel index, destination address index (0 for destination address and 1 to 3 for alternate addresses). The parameters are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: N/A

?STI1;0;0;192.168.1.2:4001↵	Sets the destination address as '192.168.1.2' and the port as '4001' for port 1.
!STT0↵	The command was executed successfully.

The command code for retrieving the running configuration

Command parameters: The port index, MCSC channel index, and destination address index are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0. The destination address index is as in the above TCP destination address index table.

Reply parameters: TCP destination address and port

?RTI1;0;1↵	Requests destination address 1 for port 1.
!RTI0192.168.1.2:4001↵	The destination address is set as '192.168.1.2' and the port as '4001'.

TCP Connect Retry Timeout

Command code: TT

The command code for getting the configuration

Command parameters: The port index and the MCSC channel index are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: TCP connect retry timeout

?GTT1;0↵	Requests the TCP connect retry timeout for port 1.
!GTI01500↵	TCP connect retry timeout is set as'1500'.

The command code for setting the configuration

Command parameters: Port index, MCSC channel index, and client connection retry timeout. The parameters are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: N/A

?STT1;0;1500↵	Sets TCP connect retry timeout as '1500' for port 1.
!STI0↵	The command was executed successfully.

The command code for retrieving the running configuration

Command parameters: The port index and the MCSC channel index are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: TCP connect retry timeout.

?RTT1;0↵	Requests the TCP connect retry timeout for port 1.
!RTI01500↵	The TCP connect retry timeout is '1500'.

TCP Max Connection

Command code: TM

The command code for getting the configuration

Command parameters: The port index and the MCSC channel index are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: MiiNePort's TCP maximum connection number.

?GTM1;1↵	The system requests the maximum connection number for port 1 and channel 1.
!GTM04↵	The maximum connection number is set as ‘4’.

The command code for setting the configuration

Command parameters: Port index, MCSC channel index, and maximum connection number. The parameters are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: N/A

?STM1;1;3↵	The system set the maximum connection number as ‘3’ for port 1 and channel 1.
!STM0↵	The command was executed successfully.

The command code for retrieving the running configuration

Command parameters: The port index and the MCSC channel index are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

?RTM1;1↵	The system requests the maximum connection number for port 1 and channel 1.
!RTM04↵	The maximum connection number is set as ‘4’.

TCP Ignore Jammed IP

Command code: TJ

The command code for getting the configuration

Command parameters: The port index and the MCSC channel index are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: 1 (Enable) or 0 (Disable)

?GTJ1;2↵	Requests the jammed IP setting for port 1's channel 2.
!GTJ01↵	The ignore jammed IP is set as ‘Enable’.

The command code for setting the configuration

Command parameters: Port index, MCSC channel index, and ignore jammed IP setting (0 for disable and 1 for enable). The parameters are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: N/A

?STJ1;2;0↵	Sets the ignore jammed IP policy as ‘Disable’ for port 1's channel 2.
!STJ0↵	The command was executed successfully.

The command code for retrieving the running configuration

Command parameters: The port index and the MCSC channel index are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: 1 (Enable) or 0 (Disable).

?RTJ1;2↵	Requests the jammed IP policy for port 1’s channel 2.
!RTJ01↵	The ignore jammed IP is set as ‘Enable’.

TCP Disconnect With DSR Off

Command code: TL

The command code for getting the configuration

Command parameters: The port index and the MCSC channel index are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: 1 (Enable) or 0 (Disable)

?GTL1;2↵	Requests the DSR off policy for port 1’s channel 2.
!GTL01↵	DSR Off is set as ‘Enable’.

The command code for setting the configuration

Command parameters: Port index, MCSC channel index, and disconnect with DSR off setting (0 for disable and 1 for enable). The parameters are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: N/A

?STL1;2;1↵	Sets the DSR off policy as ‘Enable’ for port 1’s channel 2.
!STL0↵	The command was executed successfully.

The command code for retrieving running configuration

Command parameters: The port index and the MCSC channel index are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: 1 (Enable) or 0 (Disable).

?RTL1;2↵	System requests DSR off policy for port 1’s channel 2.
!RTL01↵	DSR off is set as ‘Enable’.

TCP Check EOT Character

Command code: TE

The command code for getting the configuration

Command parameters: The port index and the MCSC channel index are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: 1 (Enable) or 0 (Disable), and EOT character.

?GTE1;0↵	For port 1, the setting for the check EOT character function is requested.
!GTE01;04↵	The check EOT character function is set as ‘Enable’, and the EOT character is set as ‘0x04’.

The command code for setting the configuration

Command parameters: Port index, MCSC channel index, enable EOT checking (1) or not (0), and EOT character by hex if enabled. The parameters are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: N/A

?STE1;0;1;04↵	The check EOT character function is set as 'Enable' and the EOT character is set as '0x04' for port 1.
!STE0↵	The command was executed successfully.
?STE1;0;0↵	EOT checking is disabled.
!STE0↵	The command was executed successfully.

The command code for retrieving the running configuration

Command parameters: The port index and the MCSC channel index are separated by a semicolon (;). For the MCSC-disabled port, the channel index is 0.

Reply parameters: 1 (Enable) or 0 (Disable), and EOT character.

?RTE1;0↵	For port 1, the setting for the check EOT character function is requested.
!RTE01;04↵	The check EOT character function is set as 'Enable', and the EOT character is set as '0x04'.

TCP Inactivity Time

Command code: TV

The command code for getting the configuration

Command parameters: The port index and the MCSC channel index are separated by a semicolon (;). For the MCSC-disabled port, the channel index is 0.

Reply parameters: TCP inactivity time.

?GTV1;0↵	The system requests the inactivity time for port 1.
!GTV00↵	The inactivity time is '0'.

The command code for setting the configuration

Command parameters: Port index, MCSC channel index, and inactivity timeout setting. The parameters are separated by a semicolon (;). For the MCSC-disabled port, the channel index is 0.

Reply parameters: N/A

?STV1;0;10↵	The system sets inactivity time at '10' ms for port 1.
!STV0↵	The command was executed successfully.

The command code for retrieving the running configuration

Command parameters: The port index and the MCSC channel index are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: TCP inactivity time.

?RTV1;0↵	The system requests the inactivity time for port 1.
!RTV00↵	The inactivity time is set as '0'.

UDP Serial Format

Command code: UT

UDP Serial Format Index

0	Raw data
1	Packet command

The command code for getting the configuration

Command parameters: The port index and the MCSC channel index are separated by a semicolon (;). For the MCSC-disabled port, the channel index is 0.

Reply parameters: Serial format as in the above UDP Serial Format Index table.

?GUT1;0↵	The system requests the UDP serial format for port 1.
!GUT00↵	UDP serial format is set as Raw data'.

The command code for setting the configuration

Command parameters: Port index, MCSC channel index, and UDP serial format. The parameters are separated by a semicolon (;). For the MCSC-disabled port, the channel index is 0. Serial format as in the UDP Serial Format Index table.

Reply parameters: N/A

?SUT1;0;0↵	Sets the UDP serial format as 'Raw data' for port 1.
!SUT0↵	The command was executed successfully.

The command code for retrieving the running configuration

Command parameters: The port index and the MCSC channel index are separated by a semicolon (;). For the MCSC-disabled port, the channel index is 0.

Reply parameters: Serial format as in the UDP Serial Format Index table.

?RUT1;0↵	Requests the UDP serial format for port 1.
!RUT00↵	UDP serial format is set as 'Raw data'.

UDP Destination Address

Command code: UD

The command code for getting the configuration

Command parameters: The port index, the MCSC channel index, and the UDP destination address index from 1 to 4 are separated by a semicolon (;). For the MCSC-disabled port, the channel index is 0.

Reply parameters: Begin address to End address: port

?GUD1;0;3↵	Requests the UDP destination address 3 for port 1.
!GUD0192.168.1.3-192.168.1.8:4001↵	The UDP destination address is from '192.168.1.3' to '192.168.1.8', and the port number is '4001'.

The command code for setting the configuration

Command parameters: Port index, MCSC channel index, destination address index (1 to 4), and destination addresses. The parameters are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0. The destination addresses are formatted as Begin address to End address: UDP port.

Reply parameters: N/A

?SUD1;0;3;192.168.1.3-192.168.1.8:4001↵	The system sets UDP destination address 3 from ‘192.168.1.3’ to ‘192.168.1.8’ and the port number as ‘4001’ for port 1.
!SUD0↵	MiiNePort reports command executed successfully.

The ommand code for retrieving the running configuration

Command parameters: the port index, the MCSC channel index, and the destination address index from 1 to 4 are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: Begin address to End address: port

?RUD1;0;3↵	The system requests the UDP destination address 3 for port 1.
!RUD0192.168.1.3-192.168.1.8:4001↵	The UDP destination address is from ‘192.168.1.3’ to ‘192.168.1.8’ and the port number is ‘4001’.

UDP Local Listen Port

Command code: UP

The command code for getting the configuration

Command parameters: The port index and the MCSC channel index are separated by the semicolon (;). For the MCSC-disabled port, the channel index is 0.

Reply parameters: local listen port

?GUP1;0↵	Requests the UDP local listen port for port 1.
!GUP4001↵	The UDP local listen port number is set as ‘4001’.

The command code for setting the configuration

Command parameters: Port index, MCSC channel index, and local UDP port. The parameters are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0.

Reply parameters: N/A

?SUP1;0;4001↵	Sets the UDP local listen port number as ‘4001’ for port 1.
!SUP0↵	The command was executed successfully.

The command code for retrieving the running configuration

Command parameters: The port index and the MCSC channel index are separated by a semicolon (;) denote. For the MCSC-disabled port, the channel index is 0.

Reply parameters: local listen port.

?RUP1;0↵	Requests the UDP local listen port number for port 1.
!RUP4001↵	The UDP local listen port number is set as ‘4001’.

The command code for getting the configuration

Command parameters: The port index, MCSC channel index, and the destination address index from 1 to 4 are separated by the semicolon (;) denote. For the MCSC-disabled port, the channel index is 0.

Reply parameters: Begin address to End address: port

?GUD1;0;3↵	Requests UDP destination address 3 for port 1.
!GUD0192.168.1.3-192.168.1.8:4001↵	The UDP destination address is from '192.168.1.3' to '192.168.1.8', and the port number is '4001'.

Command code for setting the configuration

Command parameters: Port index, MCSC channel index, destination address index (1 to 4), and destination addresses. Parameters are separated by a semicolon (;). For an MCSC-disabled port, the channel index is 0. The destination addresses are formatted as Begin address to End address: UDP port.

Reply parameters: N/A

?SUD1;0;3;192.168.1.3-192.168.1.8:4001↵	Sets UDP destination address 3 from '192.168.1.3' to '192.168.1.8', and the port number as '4001' for port 1.
!SUD0↵	The command was executed successfully.

The command code for retrieving the running configuration

Command parameters: The port index, MCSC channel index, and destination address index from 1 to 4 are separated by the semicolons (;). For the MCSC-disabled port, the channel index is 0.

Reply parameters: Begin address – End address: port

?RUD1;0;3↵	Requests the UDP destination address 3 for port 1.
!RUD0192.168.1.3-192.168.1.8:4001↵	The UDP destination address is from '192.168.1.3' to '192.168.1.8', and the port number is '4001'.

Pin Function

Command code: PF

PIN Function Index

Value	Meaning	MiiNePort E1 constraint
0	GND	
1	VCC	
2	RST	
3	Tx	
4	Rx	
5	Ready/RTS	Available for PIN 6 only.
6	CTS	Available for PIN 8 only.
7	DTR	Available for PIN 7 only.
8	DSR	Available for PIN 8 only.
9	485_en	Available for PIN 6 or PIN 7 only, and unable to set to both PINs.
10	DIO	
11	Reset to default	Available for PIN 7 only.

The command code for getting the configuration

Command parameters: PIN index from 1 to 8.

Reply parameters: The PIN function is shown in the above PIN Function Index table.

?GPF1↵	Requests PIN 1's function
!GPF00↵	The PIN function is set as 'GND'.
?GPF7↵	Requests PIN 7's function.
!GPF011↵	The PIN function is set as `Reset to default'.

The command code for setting the configuration

Command parameters: PIN index 1 to 8 and the PIN function are separated by a semicolon (;). The PIN function is as in the PIN Function Index table.

Note that for the MiiNePort E1, only PIN 6 to 8 are configurable.

Reply parameters: N/A

?SPF6;9↵	Sets PIN 6's function as '485_en'.
!SPF0↵	The command was executed successfully.

The command code for retrieving the running configuration

Command parameters: PIN index from 1 to 8.

Reply parameters: The PIN function is shown in the PIN Function Index table.

?RPF1↵	Requests PIN 1's function.
!RPF00↵	The PIN function is 'GND'.
?GPF7↵	Requests PIN 7's function.
!GPF011↵	The PIN function is set as `Reset to default'.

DIO Initial Mode

Command code: PM

DIO Initial Mode Index.

Index	DIO Port	MiiNePort E1 PIN Number
0	DIO0	PIN6, can only be output
1	DIO1	PIN7
2	DIO2	PIN 8, can only be input

Note that for the MiiNePort E1, this configuration is available only if a specific PIN function is set as DIO.

The command code for getting the configuration

Command parameters: DIO port index. For the MiiNePort E1, the index is as in the above DIO Initial Mode Index table. Reply parameters: 1 (output) or 0 (input);

?GPM1↵	Requests DIO1's initial mode.
!GPM01↵	DIO1's initial mode is set as 'output'.

The command code for setting the configuration

Command parameters: The DIO port index and initial mode (0 for input and 1 for output) are separated by a semicolon (;).

Note that this configuration is valid only if the related PIN function is set as DIO.

Reply parameters: N/A

?SPM0;1↵	Sets DIO0 mode as 'output'.
!SPM0↵	The command was executed successfully.

The command code for retrieving the running configuration

Command parameters: DIO port index. For the MiiNePort E1, the index is as in the DIO Initial Mode Index table.

Reply parameters: 1 (output) or 0 (input).

?RPM1↵	Requests DIO1's initial mode.
!RPM01↵	DIO1's initial mode is set as 'output'.

DIO Initial State

Command code: PS

DIO Initial State Index Table:

Index	DIO Port	MiiNePort E1 PIN Number
0	DIO0	PIN6
1	DIO1	PIN7
2	DIO2	PIN8

Note that this function is for configuration only if a specific DIO port is set to initial output.

The command code for getting the configuration

Command parameters: DIO port index. For the MiiNePort E1, the index is as in the above DIO Initial State Index table:

Reply parameters: 1 (high) or 0 (low);

?GPS0↵	Requests DIO0's initial state.
!GPS00↵	DIO0's initial state is set as 'low'.

The command code for setting the configuration

Command parameters: DIO port index and initial output state (0 for low and 1 for high).

Reply parameters: N/A

?SPS0;1↵	Sets DIO0 state as 'high'.
!SPS0↵	The command was executed successfully.

The command code for retrieving the running configuration

Command parameters: DIO port index. For the MiiNePort E1, the index is as in the above DIO initial state index table:

Reply parameters: 1 (high) or 0 (low).

?RPS0↵	Requests DIO0's initial state.
!RPS00↵	DIO0's initial state is set as 'low'.

Serial Command Mode Trigger

Command code: CT

Serial Command Mode Trigger Index

0	Disable
1	H/W control pin (DIO1)
2	Activated by characters
3	Activated by a break signal

Note that this configuration is valid only if port 1's operation mode is not set to MCSC.

The command code for getting the configuration

Command parameters: N/A

Reply parameters: The serial command mode trigger index is shown in the above Serial Command Mode Trigger Index.

?GCT↵	Requests the serial command mode trigger method.
!GCT01↵	The serial command mode is triggered by DIO1.

The command code for setting the configuration

Command parameters: As shown in the Serial Command Mode Trigger Index table.

Reply parameters: N/A

?SCT2↵	Sets the serial command mode as 'Activate by characters'.
!SCT0↵	The command was executed successfully.

The command code for retrieving the running configuration

Command parameters: N/A

Reply parameters: As shown in the Serial Command Mode Trigger Index table.

?RCT↵	Requests the serial command mode trigger method.
!RCT01↵	The serial command mode is triggered by DIO1.

Serial Command Mode Trigger Characters

Command code: CC

Note that this configuration is valid only if the serial command mode trigger method is set to 'Activated' by characters.

The command code for getting the configuration

Command parameters: N/A

Reply parameters: Serial command mode trigger characters, in hex, separated by a semicolon (;).

?GCC↵	Requests serial command mode trigger characters.
!GCC058;58;58↵	The serial command mode is triggered by `XXX' (0x585858).

The command code for setting the configuration

Command parameters: Serial command mode trigger characters, in hex, separated by a semicolon (;).

Reply parameters: N/A

?SCC45;66;67↵	Sets the serial command mode trigger characters as `ABC' (0x656667).
!SCC0↵	The command was executed successfully.

The ommand code for retrieving the running configuration

Command parameters: N/A

Reply parameters: Serial command mode trigger characters, in hex, separated by a semicolon (;).

?RCC↵	Requests the serial command mode trigger characters.
!RCC058;58;58↵	The serial command mode is triggered by `XXX' (0x585858).

Serial Command Mode Triggered Only at Boot Up

Command code: CB

Note that this configuration is only valid if the serial command mode trigger method is set to 'Activated' by characters.

The command code for getting the configuration

Command parameters: N/A

Reply parameters: 0 if the serial command mode can be triggered at any time, or 1 if it can only be triggered at boot up.

?GCB↵	Requests the serial command mode.
!GCB01↵	The serial command mode can only be triggered by characters at boot up.

The command code for setting the configuration

Command parameters: 0 if the serial command mode can be triggered at any time, or 1 if it can only be triggered at boot up.

Reply parameters: N/A

?SCB1↵	Sets that the serial command mode can only be triggered by characters at boot up.
!SCB0↵	The command was executed successfully.

The command code for retrieving the running configuration

Command parameters: N/A

Reply parameters: 0 if serial command mode can be triggered at any time, or 1 if it can only be triggered at boot up.

?RCB↵	Requests the serial command mode.
!RCB01↵	The serial command mode can only be triggered by characters at boot up.

Gratuitous ARP

Command code: VA

The command code for getting the configuration

Command parameters: N/A.

Reply parameters: 1 and send period if the gratuitous ARP is enabled; otherwise, reply 0.

?GVA↵	Requests the gratuitous ARP status for the alert mail server.
!GVA1;300↵	The gratuitous ARP is set as 'Enable' and the send period at 300 sec.

The command code for setting the configuration

Command parameters: 1 and send period if the gratuitous ARP is enabled; otherwise, 0.

Reply parameters: N/A

?SVA1;300↵	Sets gratuitous ARP as 'Enable' and the send period at 300 sec.
!SVA0↵	The command was executed successfully.

The Command code for retrieving the running configuration

Command parameters: N/A.

Reply parameters: 1 and send period if the gratuitous ARP is enabled; otherwise, reply 0.

?RVA↵	The system requests the gratuitous ARP status for the alert mail server.
!RVA1;300↵	The gratuitous ARP is set as 'Enable' and the send period at 300 sec.

Command Codes for Viewing the Status

Ethernet Speed and Duplex

Command code: NS

Command parameters: N/A

Reply parameters: The speed and duplex status of the Ethernet. Possible values are 0 (Unlink), 10H (10Mbps half-duplex), 10F (10Mbps full-duplex), 100H (100Mbps half-duplex), 100F (100Mbps full duplex).

?VNS↵	Requests the current Ethernet status.
!VNS0100F↵	Ethernet is in 100Mbps full-duplex mode.

TCP Connection

Command code: CS

Command parameters: Port index, MCSC channel index, and connection index. Parameters are separated by a semicolon (;). For an MCSC-disable port, the channel index is 0. When the connection index is 0, the MiiNePort returns the number of TCP connections. Please note that for the mixed mode of MiiNePort E1, the returned number of the TCP connection will only show the established TCP client connection(s).

?VCS1;1;0↵	Requests the number of TCP connections on channel 1 of port 1.
!VCS03↵	The number of the TCP connection is 3.
?VCS1;1;2↵	Requests the IP address of TCP connection no. 2.
!VCS0192.168.32.1↵	The IP address of TCP connection no. 2 is 192.168.32.1.

DIO Mode

Command code: PM

Command parameters: DIO port index. For the MiiNePort E1, the index is as follows:

Index	DIO Port	MiiNePort E1 PIN Number
0	DIO0	PIN6
1	DIO1	PIN7
2	DIO2	PIN8

Reply parameters: 1 (output) or 0 (input). For the MiiNePort E1, this configuration is available if a specific PIN function is set as DIO.

?GPM1↵	Requests DIO1's initial mode.
!GPM01↵	DIO1's initial mode is set as 'output'.

DIO State

Command code: PS

Command parameters: DIO port index.
Reply parameters: If the DIO port is in output mode, 0 denotes that it is outputting a low signal, and 1 that it is outputting a high signal. If the DIO port is in input mode, 0 denotes that it is receiving a low signal, and 1 that it is receiving a high signal.

?VPS2↵	Requests the DIO state of DIO port 2.
!VPS01↵	DIO port's state is set to high.

Control Command Codes

PING

Command code: NP

Command parameters: Target host name or IP address.
Reply parameters: A single minus symbol indicates the target host did not reply in 1000 milliseconds. Otherwise, one decimal number indicating the reply latency in milliseconds is returned.

?CNPwww.moxa.com↵	Requests to PING www.moxa.com.
!CNP010↵	The target host replies in 10 milliseconds.
?CNP192.168.1.1↵	Requests to PING 192.168.1.1.
!CNP0-↵	The target host did not reply in 1000 milliseconds.

DIO mode

Command code: PM

Command parameters: DIO port index and IO mode (0 for input and 1 for output) are separated by a semicolon (;). For the MiiNePort E1, the index is as follows:

Index	DIO Port	MiiNePort E1 PIN Number
0	DIO 0	PIN 6, can only be output
1	DIO 1	PIN 7
2	DIO 2	PIN 8, can only be input

This function works only if the related PIN function is set to DIO.

Reply parameters: N/A

?CPM1;0↵	Sets DIO 1's mode to 'input'.
!CPM0↵	The command was executed successfully.

DIO state

Command code: PS

Command parameters: DIO port index and output state (0 for low and 1 for high).

This function works only if the DIO mode is set to output.

Reply parameters: N/A

?CPS0;1↵	Sets the DIO 0's state to 'high'.
!CPS0↵	The command was executed successfully.

Save & Restart

Command code: SR

Command parameters: 1 for "Save & Restart", 0 for "Restart only".

Reply parameters: N/A

?CSR1↵	Requests to save the configuration and restart.
!CSR0↵	Your configuration is saved; restart the MiiNePort server now.

Load Factory Default

Command code: LD

This command is not applied to the configuration until you save and restart the MiiNePort.

Command parameters: 1 for "All setting", '0' for "Keep IP setting".

Reply parameters: N/A

?CLD0↵	Requests to load factory default.
!CLD0↵	Factory default was successfully loaded.

B

Well-Known Port Numbers

This appendix is included for your reference. Listed below are port numbers that already have a well-established use. These port numbers should be avoided when assigning a port number to your MiiNePort E1 Series module; otherwise you may experience network problems. Refer to the RFC 1700 standard for Well-Known Port Numbers or refer to the following introduction from IANA.

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

The Well-Known Ports are assigned by IANA, and, on most systems, can only be used by system processes or by programs executed by privileged users. The following table shows famous port numbers among the well-known port numbers. For more details, visit the IANA website at http://www.iana.org/assignments/port-numbers.

TCP Socket	Application Service
0	reserved
1	TCP Port Service Multiplexor
2	Management Utility
7	Echo
9	Discard
11	Active Users (systat)
13	Daytime
15	Netstat
20	FTP data port
21	FTP CONTROL port
23	Telnet
25	SMTP (Simple Mail Transfer Protocol)
37	Time (Time Server)
42	Host name server (names server)
43	Whois (nickname)
49	(Login Host Protocol) (Login)
53	Domain Name Server (domain)
79	Finger protocol (Finger)

TCP Socket	Application Service
80	World Wide Web HTTP
119	Network News Transfer Protocol (NNTP)
123	Network Time Protocol
213	IPX
160 – 223	Reserved for future use

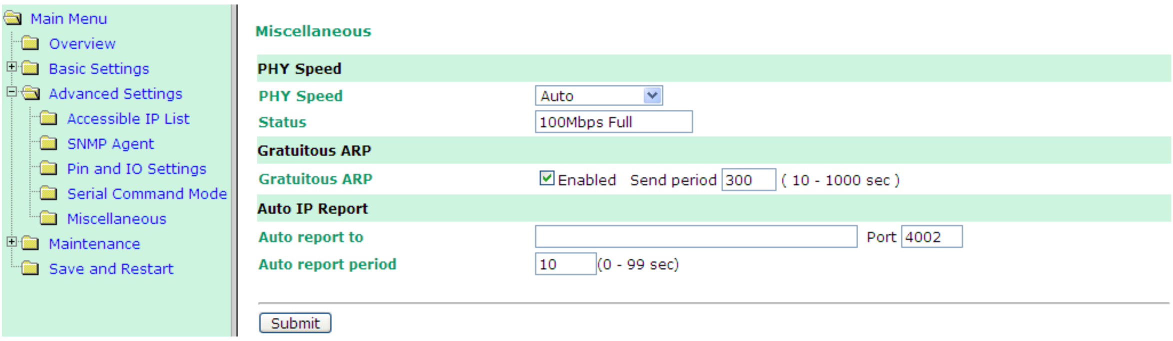
UDP Socket	Application Service
0	reserved
2	Management Utility
7	Echo
9	Discard
11	Active Users (systat)
13	Daytime
35	Any private printer server
39	Resource Location Protocol
42	Host name server (names server)
43	Whois (nickname)
49	(Login Host Protocol) (Login)
53	Domain Name Server (domain)
69	Trivial Transfer Protocol (TETP)
70	Gopler Protocol
79	Finger Protocol
80	World Wide Web HTTP
107	Remote Telnet Service
111	Sun Remote Procedure Call (Sunrpc)
119	Network news Tcanster Protocol (NNTP)
123	Network Time protocol (nnp)
161	SNMP (Simple Network Mail Protocol)
162	SNMP Traps
213	IPX (Used for IP Tunneling)

C

Auto IP Report Protocol

There are several ways to configure the IP address of an MiiNePort E1 Series module. One way is with DHCP Client. When you set up the module to use DHCP Client for IP address configuration, it will automatically send a DHCP request over the network to find the DHCP server. The DHCP server will then send an available IP address to the module with an expiration time. The module will use this IP address until the expiration time has been reached. When the expiration time has been reached, the process will repeat, and module will send another DHCP request to the DHCP server. Therefore, a module may end up using more than one IP address while it is connected to the network.

The module has a built-in IP address report function that will send its IP data to a specific IP address and port number when it is not using a Static or fixed IP address. The IP address report parameters may be configured in the Network Settings in the web console. Enter the IP address and the port number of the PC that is to receive this information as shown below:



Miscellaneous

PHY Speed

PHY Speed: Auto

Status: 100Mbps Full

Gratuitous ARP

Gratuitous ARP: ☑ Enabled Send period 300 (10 - 1000 sec)

Auto IP Report

Auto report to: Port 4002

Auto report period: 10 (0 - 99 sec)

IP Address Report Structure

The first 4 bytes of the module's IP address report are the characters MOXA. The rest of the report is composed of 9 items, with each item preceded by a 2-byte header indicating the item ID and item length.

Header (Item ID)	Header (Item Length)	Item
(none)	(none)	"Moxa" (text string)
1	(varies)	server name (text string)
2	2	hardware ID (little endian, see table below)
3	6	MAC address (00-90-E8-01-02-03 would be sent in sequence as 0x00, 0x90, 0xE8, 0x01, 0x02, 0x03)
4	4	serial number (little endian DWORD)
5	4	IP address
6	4	Netmask
7	4	default gateway
8	4	firmware version (little endian DWORD, Version 4.3.1= 0x04030100)
9	4	AP ID (little endian DWORD, see table below)

Example

The following example shows the first 22 bytes of a typical IP address report:

	report header “Moxa”				item ID	item length	server name “TEST”				item ID	item length	hardware ID 0x4119			item ID	item length	MAC address 00-90-E8-01-02-03				
HEX	4D	4F	58	41	01	04	54	45	53	54	02	02	19	41	03	06	00	90	E8	01	02	03
ASCII	“M”	“O”	“X”	“A”			“T”	“E”	“S”	“T”												

D

DIO Commands

In this appendix, we provide information on sending commands to the module's DIO channels over an Ethernet network. Digital I/O commands and responses are accessed using a specific TCP port (default 5001) on the module. Each command is initiated by the host and is followed by a response from the module. A utility on the CD-ROM can be used to test the DIO access commands.

The following topics are covered in this appendix:

- ❒ **Overview**
- ❒ **C Code Example**
- ❒ **Read Single DIO**
  - ➢ Command
  - ➢ Response
  - ➢ C Code Example
- ❒ **Write Single DIO**
  - ➢ Command
  - ➢ Response
  - ➢ C Code Example
- ❒ **Read Multiple DIOs**
  - ➢ Command
  - ➢ Response
  - ➢ C Code Example
- ❒ **Write Multiple DIOs**
  - ➢ Command
  - ➢ Response
  - ➢ C Code Example

Overview

Each DIO command and response consists of a 4-byte header and up to 255 bytes of data. The first byte of the header indicates the command. The second byte indicates the version, which is "2" for current firmware versions. The third byte is a code that is used by the module to report errors. The fourth byte is the number of bytes that follows the header, and will depend on the command or response.

In the event of an error, the module will return the entire command as its response, but with the third byte changed according to the following status/error codes:

1:	Command error; may be unknown
2:	Version error; not supported by this version
3:	Length error; the length member does not match the attached data
4:	Operation error; invalid status or invalid mode
5:	"Packet too short" error
6:	DIO number error; might not support requested DIO number
0xFF:	other unknown error

ATTENTION

DIO command data is transmitted as values rather than text strings. A value of 1 would be transmitted as 0x01.

C Code Example

```
//define DIO Header format
typedef struct _DIO_Header_Struct {
char command;
char version; /* This specification is version 2 */
char status;
char length;
} DIOHeaderStruct, *pDIOHeaderStruct;
//define DIO Packet format
//Used for Command and ACK packet
typedef struct _DIO_Packet_Struct {
DIOHeaderStruct header;
Char data[ 255];
} DIOPacketStruct, *pDIOPacketStruct;
```

Read Single DIO

Command

Byte #	Descriptor	Value	Description
1	Header	1	command #, fixed
2	Header	2	version, fixed
3	Header	(any)	this byte is only used in the module's response
4	Header	1	data length, fixed
5	Data	0, 1, 2, 3	desired DIO channel #

For example, the 5-byte command sequence 1-2-0-1-0 requests the status of DIO 0.

Response

Byte #	Descriptor	Value	Description
1	Header	1	command #, fixed
2	Header	2	version, fixed
3	Header	0, 1, 2, 3, 4, 5, 6, 0xFF	command status/error code (0 = okay)
4	Header	3	data length, fixed
5	Data	0, 1, 2, 3	desired DIO channel #
6	Data	0, 1	0: channel is in input mode 1: channel is in output mode
7	Data	0, 1	0: channel status is low 1: channel status is high

For example, the 7-byte response sequence 1-2-0-3-0-0-0 indicates that DIO 0 is in input mode.

C Code Example

```
BOOL ReadSingleDIO(int port, int *mode, int *status)
{
DIOPacketStruct packet;
packet.header.command = 1; // read single DIO command
packet.header.version = 2; // DIO protocol version
packet.header.length = 1; // data length
packet.data[0] = (char)port; // Number of the DIO
send(SocketFd, (char *)&packet, sizeof(DIOHeaderStruct)+1, 0);
//Send TCP Packet
// Process the returned data here.
return TRUE;
}
```

Write Single DIO

Command

Byte #	Descriptor	Value	Description
1	Header	2	command number, fixed
2	Header	2	version, fixed
3	Header	(any)	this byte is only used in the module’s response
4	Header	3	data length, fixed
5	Data	0, 1, 2, 3	desired DIO channel number
6	Data	0, 1	0: set to input mode 1: set to output mode
7	Data	0, 1	this byte is ignored for input mode 0: set to low 1: set to high

For example, the 7-byte command sequence 2-2-0-3-0-0-0 requests that DIO 0 be set to digital input mode.

Response

Byte #	Descriptor	Value	Description
1	Header	2	command number, fixed
2	Header	2	version, fixed
3	Header	0, 1, 2, 3, 4, 5, 6, 0xFF	command status/error code (0 = okay)
4	Header	3	data length, fixed
5	Data	0, 1, 2, 3	desired DIO channel #
6	Data	0, 1	0: channel has been changed to input mode 1: channel has been changed to output mode
7	Data	0, 1	this byte is ignored for input mode 0: channel status has been changed to low 1: channel status has been changed to high

For example, the 7-byte response sequence 2-2-0-3-0-0-0 indicates that DIO 0 has been changed to input mode.

C Code Example

```
void WriteSingleDIO(int port, int mode, int status)
{
DIOPacketStruct packet;
packet.header.command = 2; // write single DIO command
packet.header.version = 2; // DIO protocol version
packet.header.length = 3; // data length
packet.data[0] = (char)port; // number of the DIO
packet.data[1] = (char)mode; // DIO mode
packet.data[2] = (char)status; // DIO status;
send(SocketFd, (char *)&packet, sizeof(DIOHeaderStruct)+3, 0);
//Send TCP packet
//Process the returned data here
}
```

Read Multiple DIOs

Command

Byte #	Descriptor	Value	Description
1	Header	5	command number, fixed
2	Header	2	version, fixed
3	Header	(any)	this byte is only used in the module’s response
4	Header	2	data length, fixed
5	Data	0, 1, 2	starting DIO channel number
6	Data	1, 2, 3	ending DIO channel number

This command requests the status of a range of DIO channels, specified in bytes 5 and 6. For example, the 6-byte command sequence 5-2-0-2-0-2 requests the status of DIO channels 0 through 2.

Response

Byte #	Descriptor	Value	Description
1	Header	5	command number, fixed
2	Header	2	version, fixed
3	Header	0, 1, 2, 3, 4, 5, 6, 0xFF	command status/error code (0 = okay)
4	Header	4, 6, 8	data length, depends on the number of DIO channels requested
5	Data	0, 1	1st requested DIO channel 0: channel is in input mode 1: channel is in output mode
6	Data	0, 1	1st requested DIO channel 0: channel status is low 1: channel status is high
7	Data	0, 1	2nd requested DIO channel 0: channel is in input mode 1: channel is in output mode
8	Data	0, 1	2nd requested DIO channel 0: channel status is low 1: channel status is high
9	Data	0, 1	3rd requested DIO channel, optional 0: channel is in input mode 1: channel is in output mode
10	Data	0, 1	3rd requested DIO channel, optional 0: channel status is low 1: channel status is high
11	Data	0, 1	4th requested DIO channel, optional 0: channel is in input mode 1: channel is in output mode
12	Data	0, 1	4th requested DIO channel, optional 0: channel status is low 1: channel status is high

For example, the 10-byte response 5-2-0-6-0-0-1-1-0-1 indicates that DIO 0 is in input mode and "low" status, DIO 1 is in output mode and "high" status, and DIO 2 is in input mode and "high" status.

C Code Example

```
BOOL ReadMultipleDIO(int start, int end, int *mode, int *status)
{
DIOPacketStruct packet;
packet.header.command = 5; // Read Multiple DIO Commands
packet.header.version = 2; // DIO protocol command version
packet.header.length = 2; // data length
packet.data[0] = start; // start of the DIO number
packet.data[1] = end; // end of the DIO number
send(SocketFd, (char *)&packet, sizeof(DIOHeaderStruct)+2, 0);
//Send TCP packet
//Process the returned data here
return TRUE;
}
```

Write Multiple DIOs

Command

Byte #	Descriptor	Value	Description
1	Header	6	command number, fixed
2	Header	2	version, fixed
3	Header	(any)	this byte is only used in the module’s response
4	Header	6, 8, 10	data length, depends on the number of channels being written (6 bytes for 2 channels, 8 bytes for 3 channels, 10 bytes for 4 channels)
5	Data	0, 1, 2	starting DIO channel number
6	Data	1, 2, 3	ending DIO channel number
7	Data	0,1	1st DIO channel to be written 0: set to input mode 1: set to output mode
8	Data	0, 1	1st DIO channel to be written 0: set to low 1: set to high
9	Data	0, 1	2nd DIO channel to be written 0: set to input mode 1: set to output mode
10	Data	0, 1	2nd DIO channel to be written 0: set to low 1: set to high
11	Data	0, 1	3rd DIO channel to be written, optional 0: set to input mode 1: set to output mode
12	Data	0, 1	3rd DIO channel to be written, optional 0: set to low 1: set to high
13	Data	0, 1	4th DIO channel to be written, optional 0: set to input mode 1: set to output mode
14	Data	0, 1	4th DIO channel to be written, optional 0: set to low 1: set to high

This command writes the status of a range of DIO channels, specified in bytes 5 and 6. The length of the command depends on the number of channels to be written. For example, the 10-byte command 6-2-0-6-0-1-0-0-1-1 requests DIO 0 be set to digital input mode and “low” status and DIO 1 be set to digital output mode and “high” status. If you wanted to include a change of DIO 2 to digital output mode and “low” status, the 12-bye command sequence would be 6-2-0-8-0-2-0-0-1-1-1-0.

Response

Byte #	Descriptor	Value	Description
1	Header	6	command number, fixed
2	Header	2	version, fixed
3	Header	0, 1, 2, 3, 4, 5, 6, 0xFF	command status/error code (0 = okay)

Byte #	Descriptor	Value	Description
4	Header	4, 6, 8	data length, depends on the number of DIO channels requested
5	Data	0, 1	1st requested DIO channel, 0: channel has been changed to input mode, 1: channel has been changed to output mode
6	Data	0, 1	1st requested DIO channel, 0: channel status has been changed to low, 1: channel status has been changed to high
7	Data	0, 1	2nd requested DIO channel, 0: channel has been changed to input mode, 1: channel has been changed to output mode
8	Data	0, 1	2nd requested DIO channel, 0: channel status has been changed to low, 1: channel status has been changed to high
9	Data	0, 1	3rd requested DIO channel, optional, 0: channel has been changed to input mode, 1: channel has been changed to output mode
10	Data	0, 1	3rd requested DIO channel, optional, 0: channel status has been changed to low, 1: channel status has been changed to high
11	Data	0, 1	4th requested DIO channel, optional, 0: channel has been changed to input mode, 1: channel has been changed to output mode
12	Data	0, 1	4th requested DIO channel, optional, 0: channel status has been changed to low, 1: channel status has been changed to high

For example, the 8-byte response 6-2-0-4-0-0-1-1 indicates that DIO 0 has been changed to input mode and "low" status and DIO 1 has been changed to output mode and "high" status.

C Code Example

```
void WriteMultipleDIO(int start, int end, int* mode, int* status)
{
DIOPacketStruct packet;
packet.header.command = 6; // Write Multiple DIO Command Codes
packet.header.version = 2; // DIO protocol version
packet.header.length = (end-start+1)*2+2; // data length
packet.data[0] = start; // start DIO number
packet.data[1] = end; // end DIO number
int i, len;
for ( i=0; i<(end-start+1);i++ ) {
packet.data[i+2] = mode[i];
packet.data[i+3] = status[i];
}
send(SocketFd, )(char*)&packet,(end-start+1)*2+2+sizeof(DIOHeaderStruct), 0);
//Send TCP packet
//Process the returned data here
}
```

E

SNMP Agent with MIB II and RS-232 Like Groups

MiiNePort E1 Series modules have SNMP (Simple Network Management Protocol) agent software built in. The software supports RFC1317 RS-232 like groups and RFC 1213 MIB-II. The following table lists the standard MIB-II groups, as well as the variable implementations for the MiiNePort E1 Series modules.

RFC1 213 MIB-II supported SNMP variables:

System MIB	Interfaces MIB	IP MIB	ICMP MIB
SysDescr	itNumber	ipForwarding	IcmpInMsgs
SysObjectID	ifIndex	ipDefaultTTL	IcmpInErrors
SysUpTime	ifDescr	ipInreceives	IcmpInDestUnreachs
SysContact	ifType	ipInHdrErrors	IcmpInTimeExcds
SysName	ifMtu	ipInAddrErrors	IcmpInParmProbs
SysLocation	ifSpeed	ipForwDatagrams	IcmpInSrcQuenchs
SysServices	ifPhysAddress	ipInUnknownProtos	IcmpInRedirects
	ifAdminStatus	ipInDiscards	IcmpInEchos
	ifOperStatus	ipInDelivers	IcmpInEchoReps
	ifLastChange	ipOutRequests	IcmpInTimestamps
	ifInOctets	ipOutDiscards	IcmpTimestampReps
	ifInUcastPkts	ipOutNoRoutes	IcmpInAddrMasks
	ifInNUcastPkts	ipReasmTimeout	IcmpOutMsgs
	ifInDiscards	ipReasmReqds	IcmpOutErrors
	ifInErrors	ipReasmOKs	IcmpOutDestUnreachs
	ifInUnknownProtos	ipReasmFails	IcmpOutTimeExcds
	ifOutOctets	ipFragOKs	IcmpOutParmProbs
	ifOutUcastPkts	ipFragFails	IcmpOutSrcQuenchs
	ifOutNUcastPkts	ipFragCreates	IcmpOutRedirects
	ifOutDiscards	ipAdEntAddr	IcmpOutEchos
	ifOutErrors	ipAdEntIfIndex	IcmpOutEchoReps
	ifOutQLen	ipAdEntNetMask	IcmpOutTimestamps
	ifSpecific	ipAdEntBcastAddr	IcmpOutTimestampReps
		ipAdEntReasmMaxSize	IcmpOutAddrMasks
		IpNetToMediaIfIndex	IcmpOutAddrMaskReps
		IpNetToMediaPhysAddres s	
		IpNetToMediaNetAddress	
		IpNetToMediaType	
		IpRoutingDiscards	

UDP MIB	TCP MIB	SNMP MIB
UdpInDatagrams	tcpRtoAlgorithm	snmpInPkts
UdpNoPorts	tcpRtoMin	snmpOutPkts
UdpInErrors	tcpRtoMax	snmpInBadVersions
UdpOutDatagrams	tcpMaxConn	snmpInBadCommunityNames
UdpLocalAddress	tcpActiveOpens	snmpInASNParseErrs
UdpLocalPort	tcpPassiveOpens	snmpInTooBigs
	tcpAttempFails	snmpInNoSuchNames
Address Translation MIB	tcpEstabResets	snmpInBadValues
AtIfIndex	tcpCurrEstab	snmpInReadOnlys
AtPhysAddress	tcpInSegs	snmpInGenErrs
AtNetAddress	tcpOutSegs	snmpInTotalReqVars

F

NECI Library

NECI (Network Enabler Configuration Interface) is a set of APIs that run on Windows systems (95 and above) to search, locate, and configure MiiNePort Series modules over the network. The MiiNePort Series library can be found in the folder **.\Software\Library\NECI** on the Documentation and Software CD included with each module. For more information, refer to **NECI.chm** in that directory as well as examples located in **.\Software\Library\NECI\C\** or **.\Software\Library\NECI\Java**.