MGate MB3660 Modbus Gateway
User Manual

Version 6.4, December 2023

www.moxa.com/products

MGate MB3660 Modbus Gateway User Manual

Copyright Notice

Trademarks

Disclaimer

Technical Support Contact Information

www.moxa.com/support

Table of Contents

1. Introduction 5
Overview 5
Package Checklist 6
Product Features 7
2. Getting Started 8
Connecting the Power 8
Connecting Serial Devices 8
Connecting to a Host or the Network 8
Wiring Requirements 9
LED Indicators 9
Dimensions 10
Adjustable Pull High/Low Resistors for the RS-485 Port 11
SD card 12
Pin Assignments 13
Power Input 13
Relay Output 13
Rackmount 14
Specifications 14
3. Device Search Utility 15
Installing the Software 15
Starting Device Search Utility (DSU) 19
Connecting to the Unit 19
Broadcast Search 20
Search IP 22
Locate 22
Upgrading the Firmware 23
4. Web Console Configuration 24
Logging into the Web Console 24
Basic Settings 25
Network Settings 26
Serial Settings 27
Protocol Settings 28
Transparent Mode 28
Agent Mode 38
System Management 52
System Monitoring 62
Serial Redirector Settings 65
Save/Restart 72
Logout 72
MXview 72
MXconfig 74
5. Typical Applications 76
Ethernet Clients/Masters With Multiple Serial Servers/Slaves 76
Serial Clients/Masters with Multiple Ethernet Servers/Slaves 77
Modbus TCP Clients/Masters with ASCII and RTU Servers/Slaves 77
Serial Client(s)/Master(s) with Serial Servers/Slaves 78
6. Case Studies 79
Introduction 79
Replace Serial Clients/Masters with Ethernet Client(s)/Master(s), Configurable Slave IDs 79
Replace Serial Clients/Masters with Ethernet Client(s)/Master(s), Fixed Slave IDs 80
Keep Serial Client/Master and Add Ethernet Client(s)/Master(s) 80
Serial Redirector 81
Integrate Modbus RTU, ASCII, and TCP at the Same Time 81
A. Modbus Overview 82
Introduction 82
Devices are Either Clients/Masters or Servers/Slaves 82
Slaves are Identified by ID 82
Communication is by Request and Response 82

Requests Need a Time Limit 83
Modbus Ethernet vs. Modbus Serial 84
Integrate Modbus Serial and Ethernet with Gateways 84
B. SNMP Agents with MIB II and RS-232-Like Groups 85
RFC1213 MIB-II Supported SNMP Variable 85
RFC1317 RS-232-Like Groups 86

1. Introduction

Welcome to the MGate MB3660 Series of 8- or 16-port Modbus gateways that convert between Modbus TCP and Modbus RTU/ASCII protocols.

All MB3660 gateways (MB3660-8, MB3660-16) have dual AC/DC power inputs and dual IP addresses built in for redundancy. Magnetic serial port isolation is also provided for "-I" models.

In this chapter, we introduce the MGate MB3660. The following topics are covered:

Overview

The MGate MB3660 (MB3660-8 and MB3660-16) Series comprises redundant Modbus gateways that convert between Modbus TCP and Modbus RTU/ASCII protocols. They can be accessed by up to 256 Modbus TCP client/master devices or connect to 128 Modbus TCP server/slave devices. The MGate MB3660 isolation model provides 2 kV isolation protection suitable for power substation applications. The MGate MB3660 gateways are designed to easily integrate Modbus TCP and RTU/ASCII networks. The MGate MB3660 gateways offer features that make network integration easy, customizable, and compatible with almost any Modbus network.

For large-scale Modbus deployments, MGate MB3660 gateways can effectively connect many Modbus nodes to the same network. The MB3660 Series can physically manage up to 248 serial server/slave nodes for 8-port models or 496 serial server/slave nodes for 16-port models (the Modbus standard only defines Modbus IDs from 1 to 247). Each RS-232/422/485 serial port can be configured individually for Modbus RTU or Modbus ASCII operation and for different baudrates, allowing both types of networks to be integrated with Modbus TCP through one Modbus gateway.

High Performance with Innovative Command Learning

The MGate MB3660 gateways support two communication modes: transparent mode and agent mode. For transparent mode, the gateway converts Modbus commands from Modbus TCP to Modbus RTU/ASCII, and vice versa, or from serial Client/Master to serial Server/Slave. However, since only one Modbus protocol request-response action can be executed at any given time, each Modbus device must wait its turn, resulting in poorer performance. Agent mode is designed to overcome this performance weakness. By allowing users to manually key in Modbus commands, the gateway can send Modbus commands to multiple Modbus devices at the same time. Since the gateway actively and continuously retrieves data from Modbus devices simultaneously through the different serial ports, users will see a dramatic reduction in the amount of time a Modbus device needs to wait to be accessed. SCADA systems can retrieve Modbus device data directly from the gateway's memory, instead of waiting for the gateway to pass commands to the serial ports, enhancing the Modbus gateway's communication performance.

Transparent mode helps users adopt existing SCADA programs, but with reduced communication performance, whereas agent mode is characterized by high performance, but it requires users to go through the trouble of keying in Modbus commands. To provide better performance, without requiring users to key in a lot of Modbus commands, the MGate MB3660 gateways are designed with an innovative Command Learning function, which can be activated with a single mouse click. Once activated, the gateway will learn and memorize the Modbus commands it receives, and once a command has been learned, the gateway will act as though it were in agent mode and actively send Modbus requests to the relevant Modbus devices. Since the data is saved in a different memory space that can be accessed by the SCADA system, the SCADA system can retrieve Modbus response data directly from the gateway's memory, instead of waiting for the data to pass through the Modbus devices, dramatically increasing communication performance.

Windows-Based Utility and Web Console for Easy Setup

A Windows-based utility (refer to Chapter 3) is provided to make it easy to search for and locate devices, assign IP addresses, import/export configuration files, and upgrade the MGate MB3660's firmware. The utility automatically connects to all available MGate MB3660 units on the LAN. A user-friendly web console (refer to Chapter 4) is provided to configure the device from a web browser.

Package Checklist

All models in the MGate MB3660 Series are shipped with the following items:

Standard Accessories

- MGate MB3660 Modbus gateway
- 8-pin RJ45-to-DB9 female serial cable for console setting
- Rack-mounting kit (includes 2 L-shaped plates and 8 screws)
- Two AC power cord (for AC models); two terminal blocks (for DC models)
- One dust cover for the SD slot
- Quick installation guide (printed)
- Warranty card

NOTE

The dust cover for the SD slot should be used in accordance with the MGate's operating temperature, which is from 0 to 60°C.

Optional Accessories

- **Mini DB9F-to-TB Adapter:** DB9 female to terminal block adapter for RS-422/485 applications
- **CBL-RJ45M9-150:** 8-pin RJ45 to DB9 male cable, 150 cm
- **CBL-RJ45F9-150:** 8-pin RJ45 to DB9 female cable, 150 cm
- **CBL-F9M9-20:** RJ45-to-DB9 female serial cable, 150 cm
- **CBL-RJ45SF9-150:** RJ45-to-DB9 female serial shielded cable, 150 cm
- **WK-45-01:** Rackmounting kit, 2 L-shaped plates, 8 screws, 45 x 57 x 2.5 mm
- **PWC-C13AU-3B-183:** Power cord with Australian (AU) plug, 183 cm
- **PWC-C13CN-3B-183:** Power cord with three-prong China (CN) plug, 183 cm
- **PWC-C13EU-3B-183:** Power cord with Continental Europe (EU) plug, 183 cm
- **PWC-C13JP-3B-183:** Power cord with Japan (JP) plug, 7 A/125 V, 183 cm
- **PWC-C13UK-3B-183:** Power cord with United Kingdom (UK) plug, 183 cm
- **PWC-C13US-3B-183:** Power cord with United States (US) plug, 183 cm
- **CBL-PJTB-10:** Nonlocking barrel plug to bare-wire cable

NOTE

Notify your sales representative if any of the above items are missing or damaged.

Product Features

- Innovative Command Learning eliminates the need to key-in SCADA Modbus commands (acts as an agent gateway)
- Auto device routing (patented)
- High performance through active and parallel polling of serial devices
- Supports serial (Client/Master) to serial (Server/Slave) communication
- 2 Ethernet ports with the same IP or dual IP addresses
- SD card for configuration backup
- Access by up to 256 Modbus TCP client/master devices, or connect to 128 Modbus TCP server (server/slave) devices
- Serial port routing by designated TCP port and designated IP address allow access for up to 4 TCP clients/masters, while routing by slave ID mapping (slave ID table) allows access for up to 256 TCP clients/masters.
- Dual VDC or VAC power inputs with a wide power input range
- 3-pin fault relay circuit for event alarms
- 2 kV isolation protection (for "-I" models)

2. Getting Started

This chapter provides basic instructions for installing the MGate MB3660.

Connecting the Power

The unit can be powered by connecting a power source to the terminal block for DC models or power connector for AC models.

For DC power input models:

1. Loosen or remove the screws on the terminal block.
2. Connect the 20-60 VDC power line to the terminal block.
3. Tighten the connections using the screws on the terminal block.

For AC power input models:

- Connect the 100-240 VAC power line to the AC connector.

Note that the unit does not have an on/off switch. It automatically turns on when it receives power. The PWR LED on the front panel will glow to show that the unit is receiving power. There are two DC power inputs for redundancy.

Connecting Serial Devices

The unit's serial port(s) are located on the back panel. If you are connecting an RS-485 multidrop network with multiple devices, note the following:

- All devices that are connected to a single serial port must use the same protocol (i.e., either Modbus RTU or Modbus ASCII).
- Each client/master device must connect to its own port on the unit. If you are connecting to a network with both client/master and server/slave devices, the client/master must be connected to a separate port from the servers/slaves.

For serial port pin assignments, refer to the Pin Assignments section.

Connecting to a Host or the Network

Two 10/100BaseT Ethernet ports are located on the gateway's back panel. These ports are used to connect the unit to a host or Ethernet network, as follows:

- For normal operation, use a standard straight-through Ethernet cable to connect the unit to your Modbus TCP network.
- For initial configuration or for troubleshooting purposes, you may connect the unit directly to a PC.

The unit's Link LED will light up to indicate a live Ethernet connection.

The MGate MB3660 has two Ethernet ports with two MAC addresses. Hence, the unit can be connected by two different IP addresses.

Wiring Requirements

ATTENTION

Safety First!

Be sure to disconnect the power cord before installing and/or wiring your MGate MB3660.

Wiring Caution!

Calculate the maximum possible current in each power wire and common wire. Observe all electrical codes dictating the maximum allowed current for each wire size.

If the current goes over the allowed maximum, the wiring could overheat, causing serious damage to your equipment.

Temperature Caution!

Be careful when handling the MGate MB3660. When plugged in, the MGate MB3660's internal components generate heat, and consequently the board may feel too hot to touch.

You should also observe the following common wiring rules:

- Use separate paths to route wiring for power and devices. If power wiring and device wiring paths must cross, make sure the wires are perpendicular at the point of intersection.

NOTE

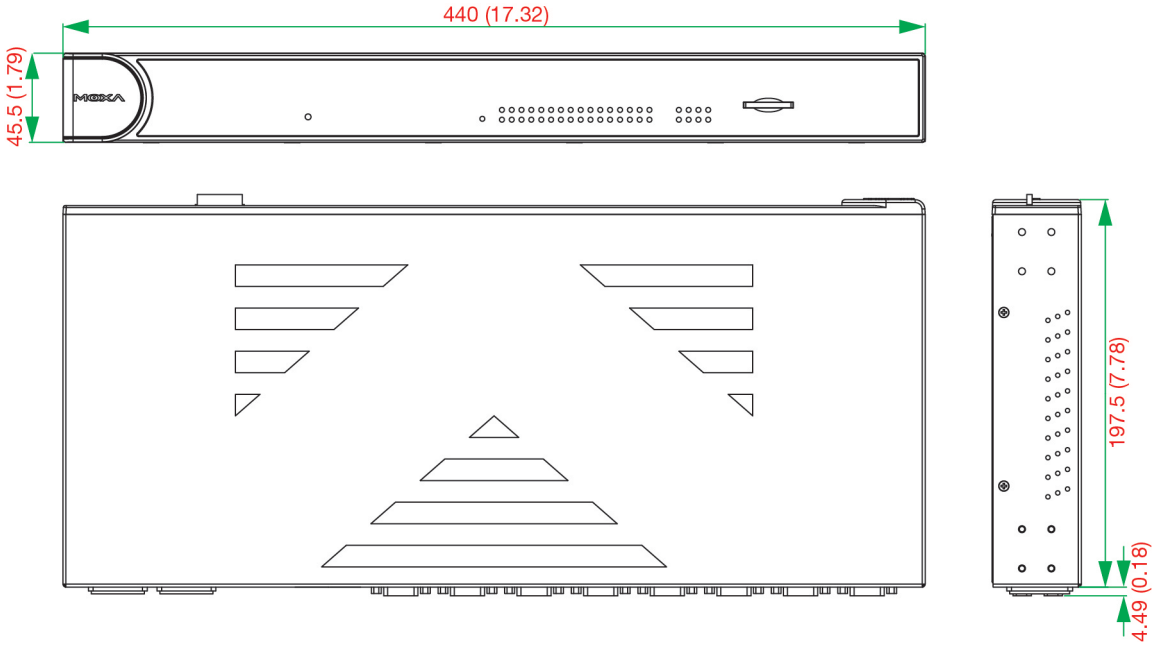
Do not run signal or communication wiring and power wiring in the same wire conduit. To avoid interference, wires with different signal characteristics should be routed separately.

- 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 wiring that shares similar electrical characteristics can be bundled together.
- Keep input wiring and output wiring separate.
- When necessary, we strongly advise labeling wiring on all devices in the system.

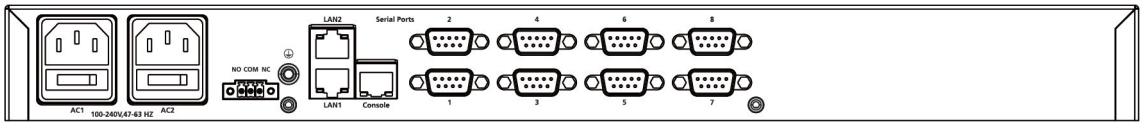
LED Indicators

Item	Description	
Reset Button	Press the Reset button for five seconds to load factory defaults. The MGate MB3660 will beep twice when the configuration has been reset.	
(LEDs)		
PWR 1, PWR 2	Red	Power connection
	Off	Power cable is not connected
Ready	Red	Steady on: Power is on, and unit is booting up Blinking: IP conflict, the DHCP or BOOTP server did not respond properly, or a relay output occurred
	Green	Steady on: Power is on, and unit is functioning normally Blinking: Unit is responding to locate function
	Off	Power is off, or power error condition exists
Tx 1–8 (16)	Green	Serial port is transmitting data
Rx 1–8 (16)	Amber	Serial port is receiving data
LAN 1, LAN 2	Green	Indicates 100 Mbps Ethernet connection
	Amber	Indicates 10 Mbps Ethernet connection
	Off	Ethernet cable is disconnected

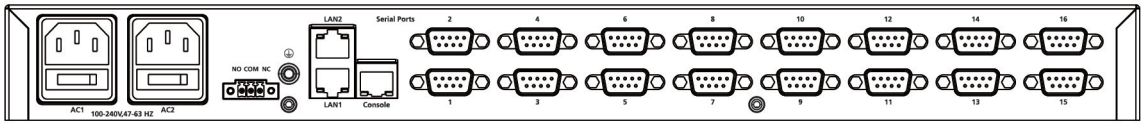
Dimensions



AC-DB9 Models

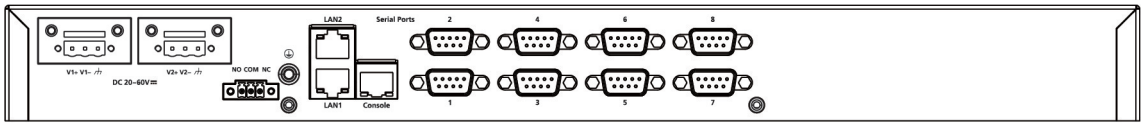


MGate MB3660-8-2AC

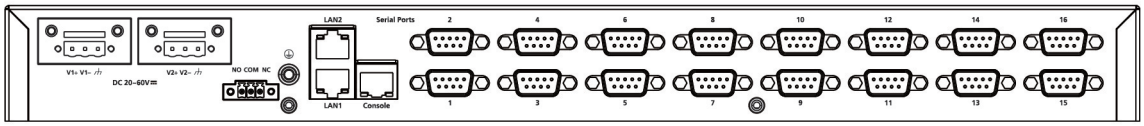


MGate MB3660-16-2AC

DC-DB9 Models

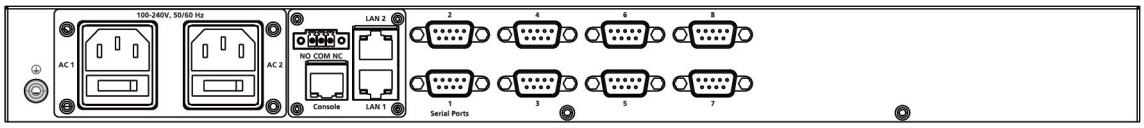


MGate MB3660-8-2DC

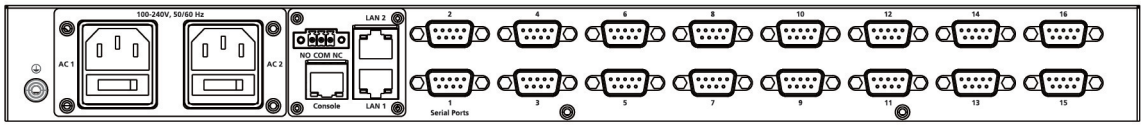


MGate MB3660-16-2DC

AC-DB9-I Models

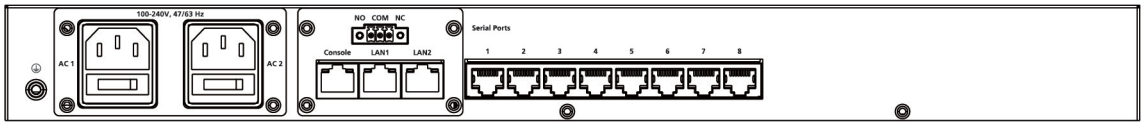


MGate MB3660I-8-2AC

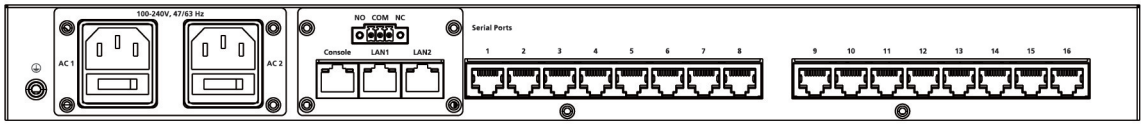


MGate MB3660I-16-2AC

AC-RJ45 Models



MGate MB3660-8-J-2AC



MGate MB3660-16-J-2AC

Adjustable Pull High/Low Resistors for the RS-485 Port

In some critical environments, you may need to add termination resistors to prevent the reflection of serial signals. When using termination resistors, it is important to set the pull high/low resistors correctly so that the electrical signal is not corrupted. The MGate MB3660 uses DIP switches to set the pull high/low resistor values for each serial port. Tear open the screws and find the DIP switches at the back side of the PCB.

To add a 120 (termination resistor, set switch 3 on the port's assigned DIP switch to ON; set switch 3 to OFF (the default setting) to disable the termination resistor.

To set the pull high/low resistors to 150 KΩ, set switches 1 and 2 on the port's assigned DIP switch to OFF. This is the default setting.

To set the pull high/low resistors to 1 KΩ, set switches 1 and 2 on the port's assigned DIP switch to ON.

ATTENTION

Do not use the 1 KΩ pull high/low setting on the MGate MB3660 when using the RS-232 interface. Doing so will degrade the RS-232 signals and reduce the effective communication distance.

SD card

The MGate MB3660 provides users with an easy way to back up, copy, replace, or deploy. The MGate is equipped with an SD card slot.

A First time using the MGate gateway with a new SD card

1. Format the SD card as a FAT file system through a PC.
2. Power off the MGate and insert the SD card (ensure that the SD card is empty).
3. Power on the MGate. The default settings will be copied to the SD card.
4. Manually configure the MGate via MGate Manager or the web console, and all the stored changes will be copied to the SD card for synchronization.

First time using the MGate with a SD card containing a configuration file

1. Power off the MGate and insert the SD card.
2. Power on the MGate.
3. The configuration file stored in the SD card will automatically be copied to the MGate.

Duplicating current configurations to another MGate gateway

1. Power off the MGate and insert a new SD card.
2. Power on the MGate.
3. The configuration will be copied from the MGate to the SD card.
4. Power off the MGate and insert the SD card in the other MGate.
5. Power on the second MGate.
6. The configuration file stored in the SD card will automatically be copied to the MGate.

Replacing a malfunctioning MGate

1. Replace the malfunctioning MGate with a new MGate.
2. Insert the SD card into the new MGate.
3. Power on the MGate.
4. The configuration file stored on the SD card will automatically be copied to the MGate.

SD card writing failure

The following circumstances may cause the SD card to experience a writing failure:

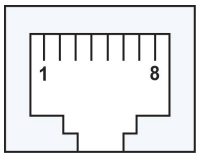
1. The SD card has less than 256 Mbytes of free space remaining.
2. The SD card is write-protected.
3. The file system is corrupted.
4. The SD card is damaged.

The MGate will stop for the abovementioned events, accompanied by a flashing Ready LED and beeping alarm. When you replace the MGate gateway's SD card, the SD card will synchronize the configurations stored on the MGate gateway. Note that the replacement SD card should not contain any configuration files on it; otherwise, the out-of-date configuration will be copied to the MGate device.

Pin Assignments

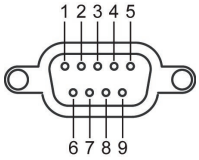
The MGate MB3660 uses DB9 serial ports to connect to Modbus RTU or ASCII devices. Each port supports three serial interfaces that are selected by software: RS-232, RS-422, and RS-485 (both 2 and 4-wire).

RJ45 (Ethernet, Console)



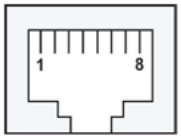
Pin	Ethernet	Console (RS-232)
1	Tx+	DSR
2	Tx-	RTS
3	Rx+	GND
4	–	TxD
5	–	RxD
6	Rx-	DCD
7	–	CTS
8	–	DTR

Male DB9 (Serial Ports)



Pin	RS-232	RS-422/RS-485-4W	RS-485-2W
1	DCD	TxD-(A)	–
2	RxD	TxD+(B)	–
3	TxD	RxD+(B)	Data+(B)
4	DTR	RxD-(A)	Data-(A)
5	GND	GND	GND
6	DSR	–	–
7	RTS	–	–
8	CTS	–	–
9	–	–	–

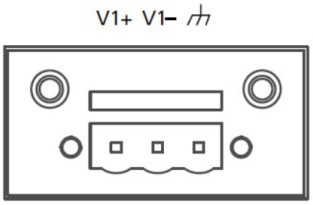
RJ45 (Serial Ports)



Pin	RS-232	RS-422/RS-485-4W	RS-485-2W
1	DSR	-	-
2	RTS	TxD+(B)	-
3	GND	GND	GND
4	TxD	TxD-(A)	-
5	RxD	RxD+(B)	Data+(B)
6	DCD	RxD-(A)	Data-(A)
7	CTS	-	-
8	DTR	-	-

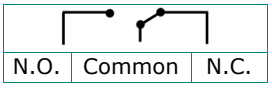
Power Input

DC Model (20 to 60 VDC)



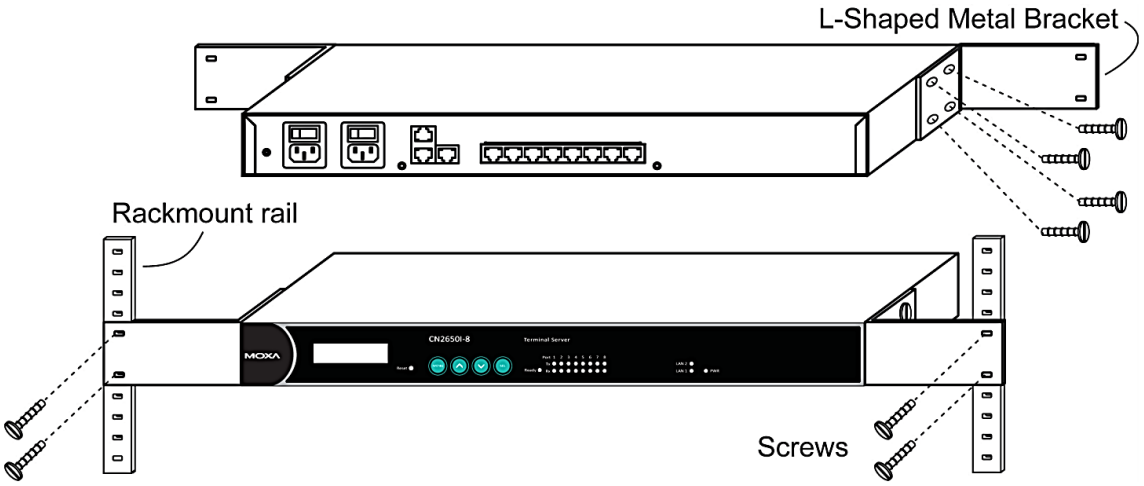
AC Model (100 to 240 VAC)

Relay Output



Rackmount

The MGate MB3660 is designed to be mounted on a standard 19-inch rack. Use the enclosed pair of L-shaped metal brackets and screws to fasten your MGate MB3660 to the rack cabinet. Each L-shaped bracket has six holes, leaving two outer or inner holes available for other uses. You have two options. You can lock either the front or rear panel of the MGate MB3660 to the front of the rack. Locking the front panel is shown in the following figure.



Specifications

NOTE

The latest specifications for Moxa's products can be found at https://www.moxa.com.

3. Device Search Utility

Installing the Software

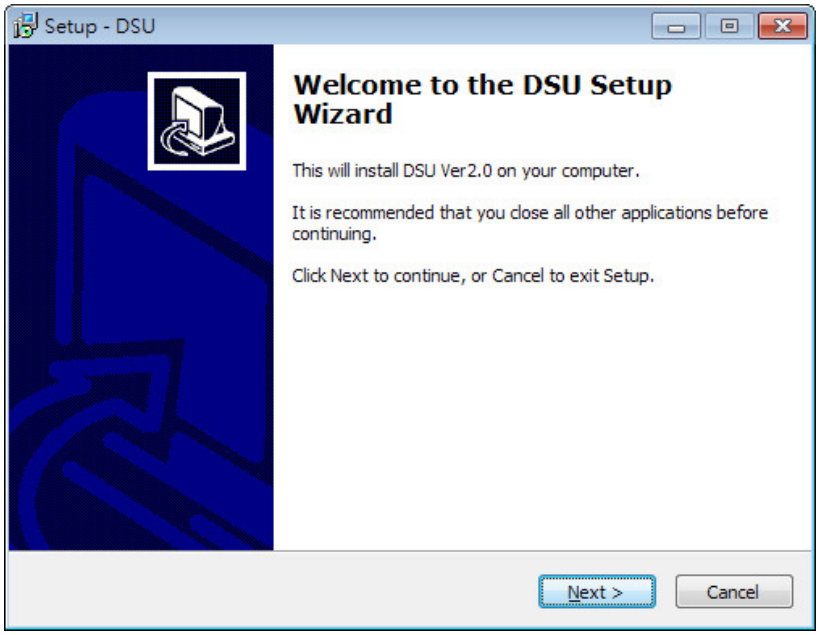
The following instructions explain how to install the Device Search Utility (abbreviated **DSU**), a utility for configuring and monitoring MGate MB3660 units over the network.

1. Please download the DSU from Moxa's website at http://www.moxa.com. Then, run the following setup program to begin the installation process:

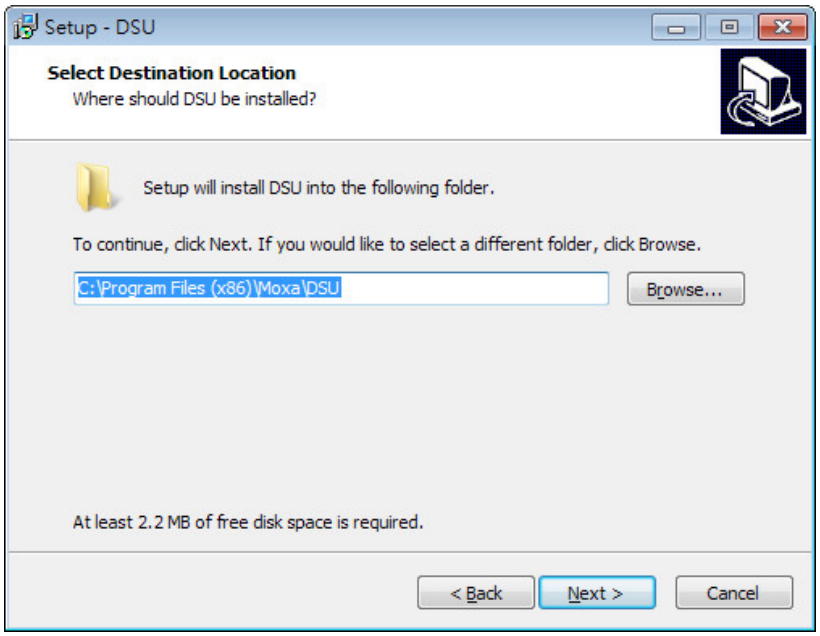
   dsu_setup_[*Version*]**_Build_**[*DateTime*]**.exe**

   The version might be named **dsu_setup_Ver2.x_Build_xxxxxxxx.exe**, for example:

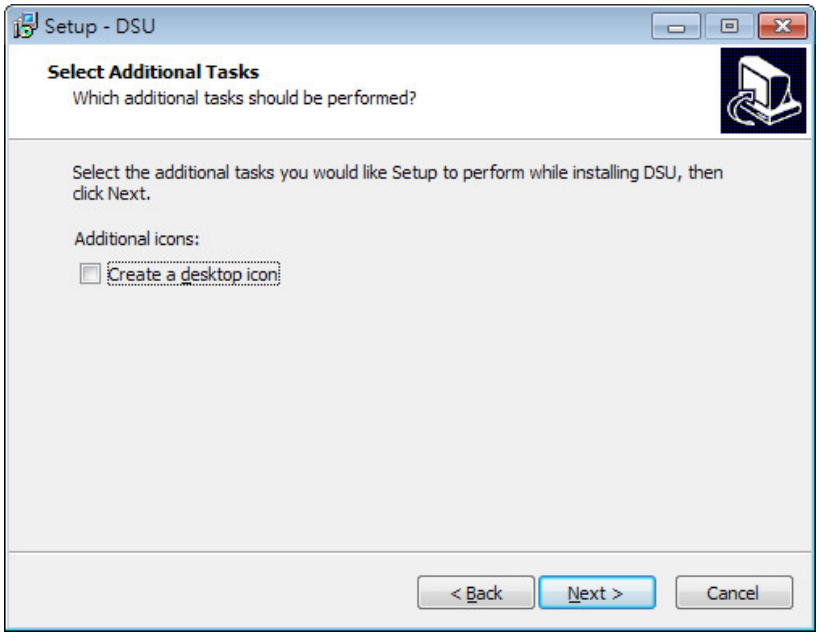
2. You will be greeted by the Welcome window. Click **Next** to continue.



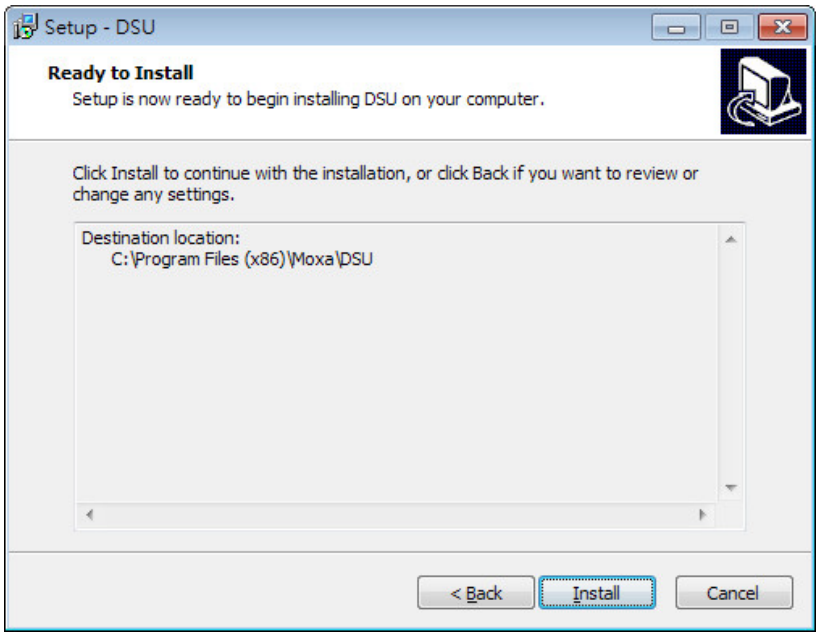
3. When the **Select Destination Location** window appears, click **Next** to continue. You may change the destination directory by first clicking on **Browse...**.



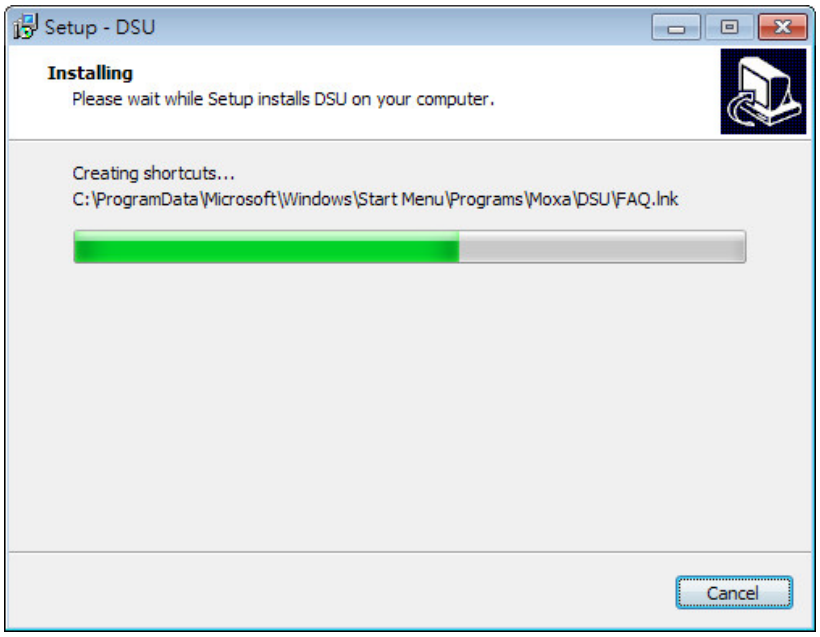
4. When the **Select Additional Tasks** window appears, click **Next** to continue. You may select **Create a desktop icon** if you would like a shortcut to DSU on your desktop.



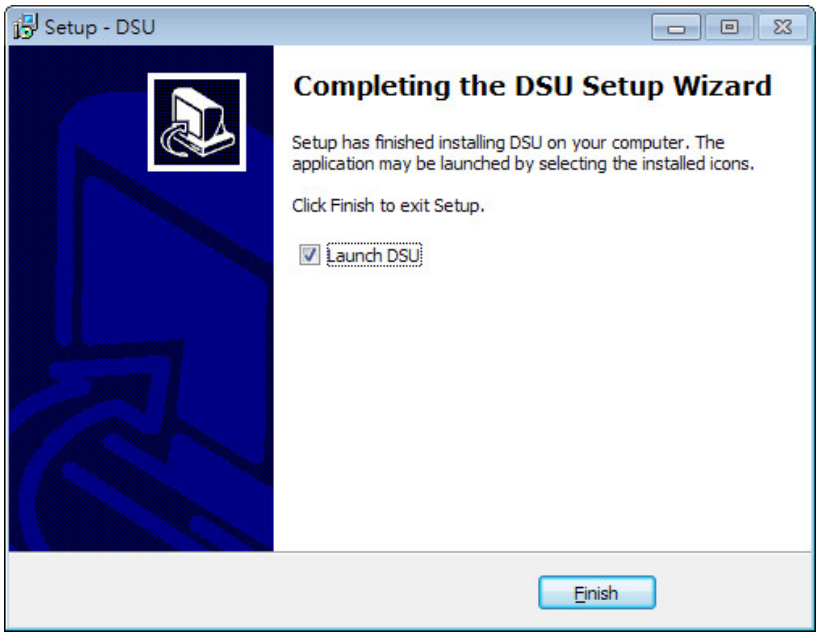
5. Click **Install** to copy the software files.



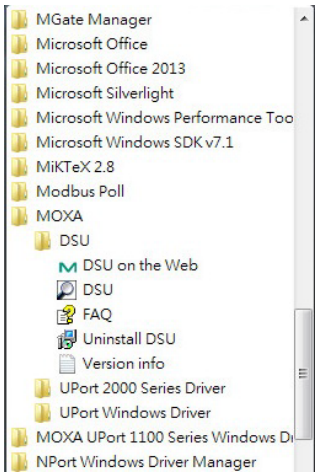
6. A progress bar will appear. The procedure should take only a few seconds to complete.



7. A message will show that DSU is successfully installed. You may choose to run it immediately by selecting **Launch DSU.**



8. You may also open DSU through **Start > Programs > MOXA > DSU**, as shown below.

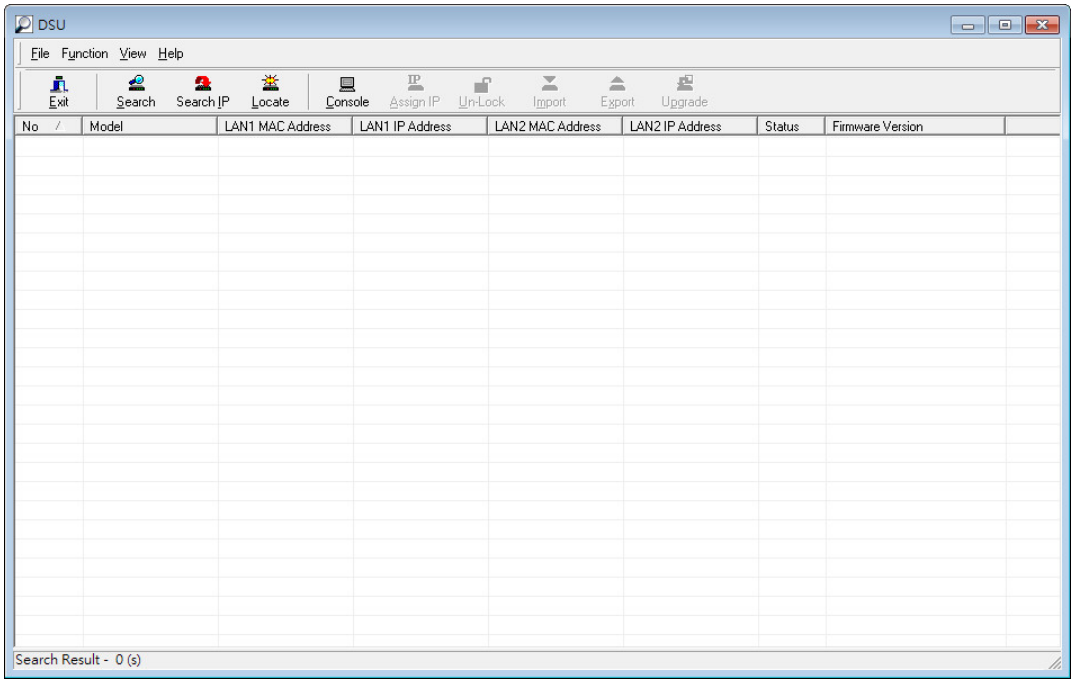


Starting Device Search Utility (DSU)

DSU is a Windows-based utility that is used to configure the MGate MB3660 Series.

Before running DSU, make sure that your PC and the MGate MB3660 are connected to the same network. Alternatively, the MGate MB3660 Series may be connected directly to the PC for configuration. Refer to Chapter 2 for more details.

You may open DSU from the Windows Start menu by clicking **Start > Programs > MOXA > DSU.** The DSU window should appear as shown below.

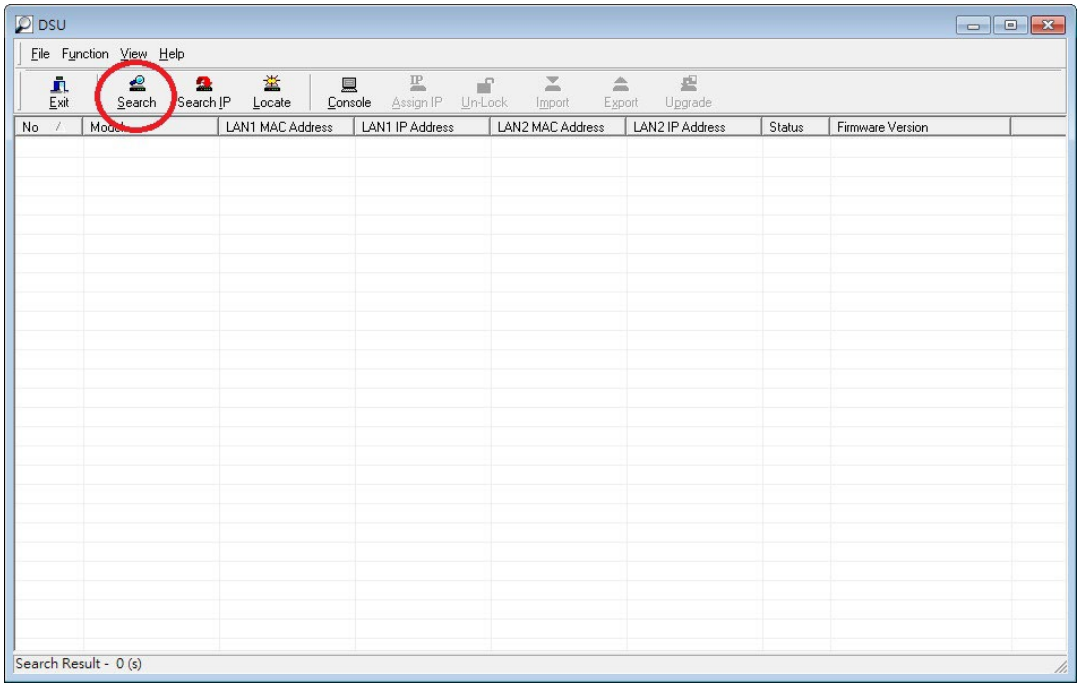


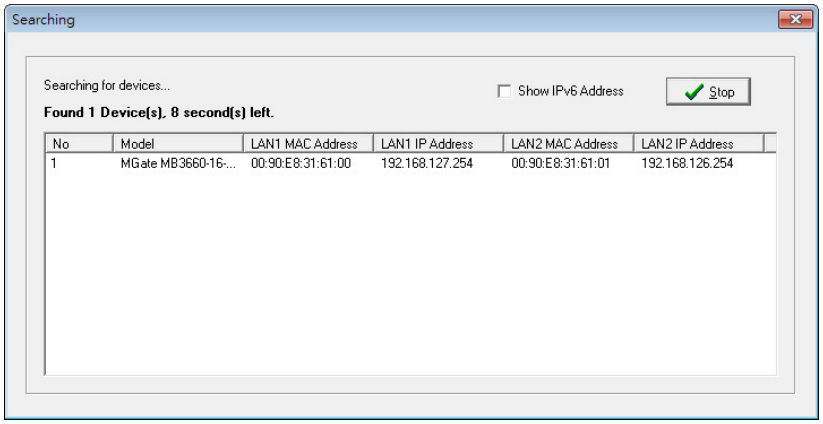
Connecting to the Unit

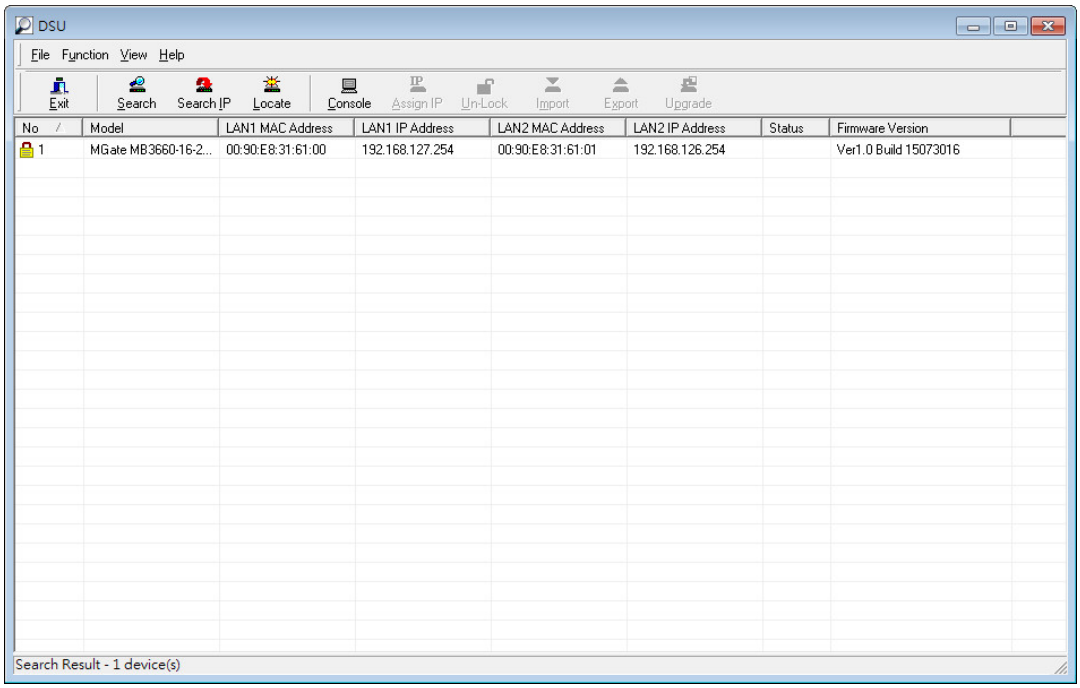
The DSU needs to connect to the unit before the unit can be configured. There are two methods to connect to the unit. **Broadcast Search** is used to find all MGate MB3660 units on the LAN. **Search IP** attempts to connect to a specific unit by IP address, which is useful if the unit is located outside the LAN or can only be accessed by going through a router.

Broadcast Search

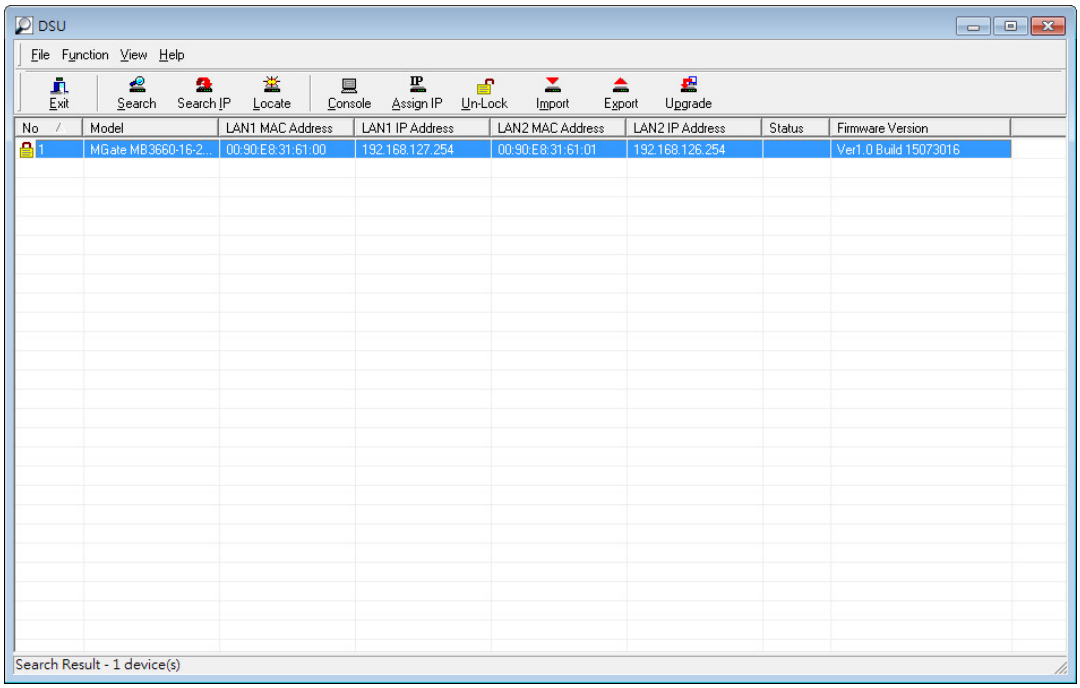
Click **Search** and a new Search window will pop up.







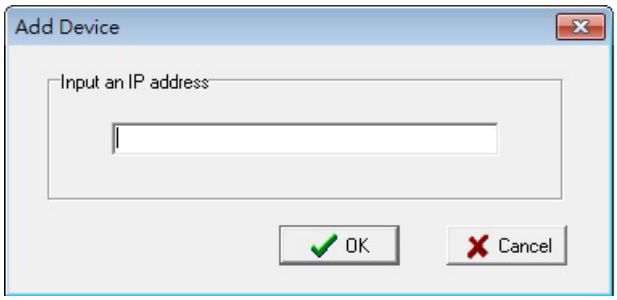
When the search is complete, every MGate MB3660 found on the LAN will appear in the DSU window. The MAC address, IP address, and Firmware version of each unit will be shown. Select the one you would like to configure.



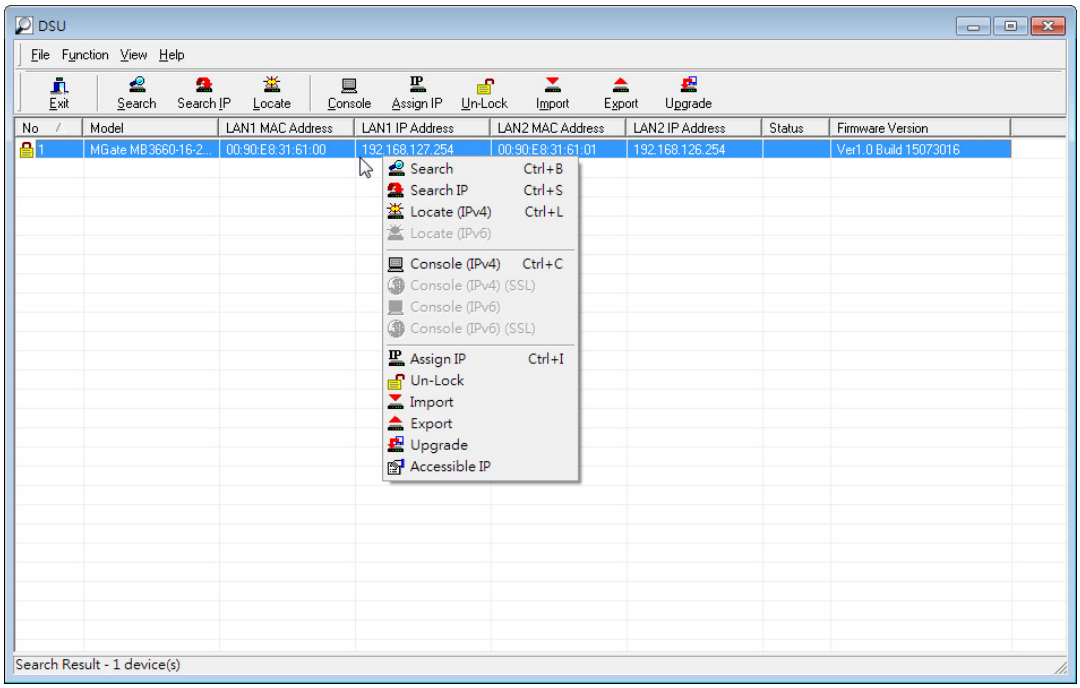
Search IP

Click **Search IP** if you know the IP address of the unit and wish to connect to it directly.

Enter the unit's IP address and click **OK**.

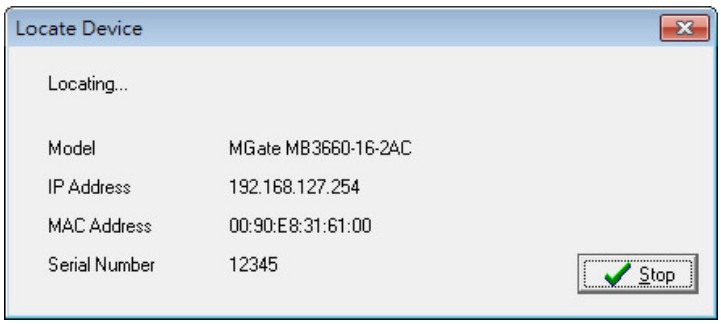


If the search is successful, the unit will be listed in the DSU window. Right click the unit to open a pop-up list of possible actions or double-click a unit to open the web console.



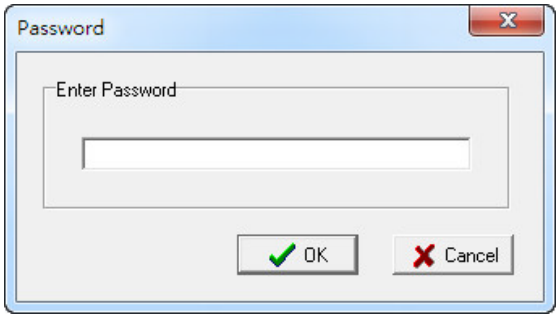
Locate

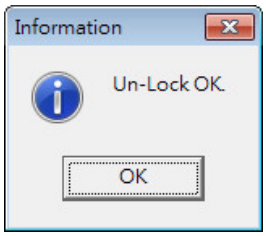
The **Locate** function will cause the unit to beep, so you can determine which unit is the target.



The **Assign IP** function allows you to change the unit's IP addresses.

Use the **Un-Lock** function to execute Import, Export, and Upgrade actions. The default password is **moxa**.





To **Import** or **Export** the configuration file, click the icons to import the configuration file from a laptop or export the currently used unit's configuration file to a laptop.

ATTENTION

If Search IP fails to locate the MGate MB3660, the IP address that you entered might be incorrect. Try doing the search again and re-entering the IP address carefully.

Another possibility is that the MGate MB3660 is on the same LAN as your PC, but on a different subnet. In this case, you can change your PC's IP address and/or netmask so that it is on the same subnet as the MGate MB3660. After your PC and the MGate MB3660 are on the same subnet, DSU should be able to find the unit.

Upgrading the Firmware

You can get the latest firmware for the MGate MB3660 from www.moxa.com. After downloading the new firmware file to your PC, you can use the DSU to write it to your MGate MB3660. Select the desired unit from the DSU list and then click **Upgrade** to begin the process.

4. Web Console Configuration

The MGate MB3660 provides a web console for easy configuration through a web browser such as Microsoft Internet Explorer or Google Chrome.

Logging into the Web Console

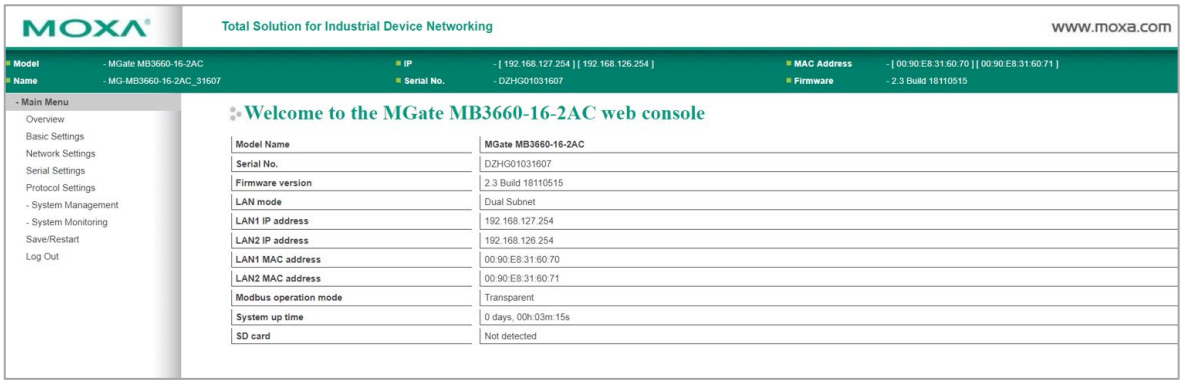
To connect to the MGate web console, open a web browser and enter the MGate gateway's IP address. If you cannot log in to the unit, you can use the Device Search Utility (DSU) to first search for the unit.

Default IP address: **LAN1 - 192.168.127.254**
LAN2 - 192.168.126.254

Default account: **admin**

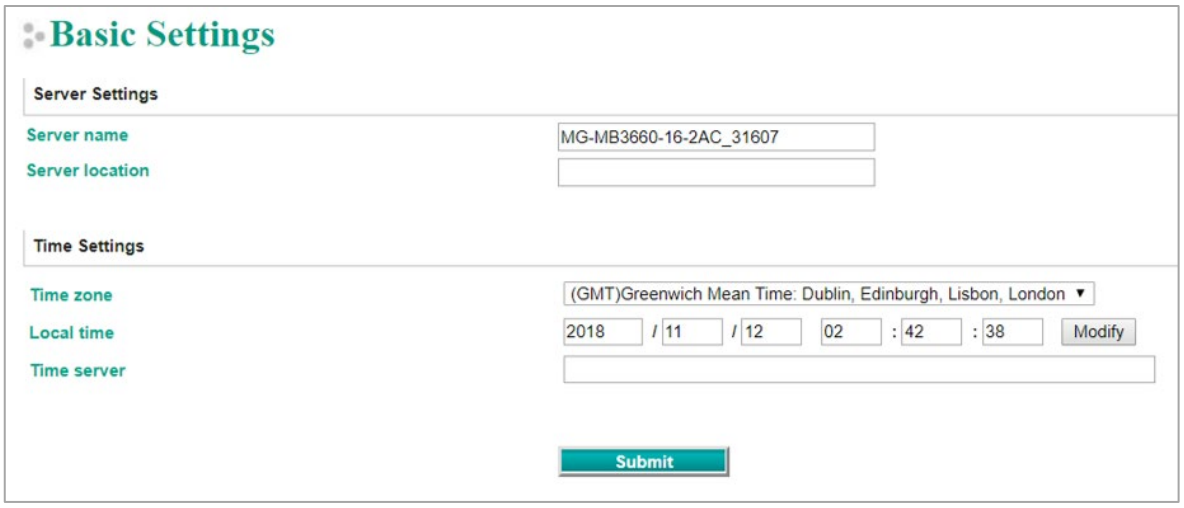
Default password: **moxa**

The welcome page shows information relevant to the MGate MB3660, which is also considered as the **Overview** page.



Basic Settings

Server Settings and **Time Settings** are shown on the **Basic Settings** page. Click **Submit** to save the current changes to the unit and click **Save/Restart** once all the settings have been changed. The unit will reboot immediately to use the new settings.



Server Setting

Parameter	Value	Notes
Server Name	(an alphanumeric string)	You can enter a name to help you identify the unit, such as the function, etc.
Server Location	(an alphanumeric string)	You can enter a name to help you identify the unit location, such as "Cabinet A001."

Time Settings

The MGate MB3660 has a built-in real-time clock for time calibration functions. Functions such as the log function can add real-time information to the message. You can do time calibration either followed by local time or time server.

ATTENTION

First-time users should select the time zone first. The console will display the realtime according to the relevant GMT time zone. If you would like to change the real-time clock, select **Local time**. The MGate firmware will change the GMT time according to the time zone.

Parameter	Value	Notes
Time Zone	User selectable time zone	This field shows the currently selected time zone and allows you to select a different time zone.
Local Time	User adjustable time. (1900/1/1-2037/12/31)	
Time Server	IP or Domain address (e.g., 192.168.1.1 or time.stdtime.gov.tw)	This optional field specifies your time server's IP address or domain name if a time server is used on your network. The module supports SNTP (RFC-1769) for automatic time calibration. MGate will request time information from the specified time server every 10 minutes.

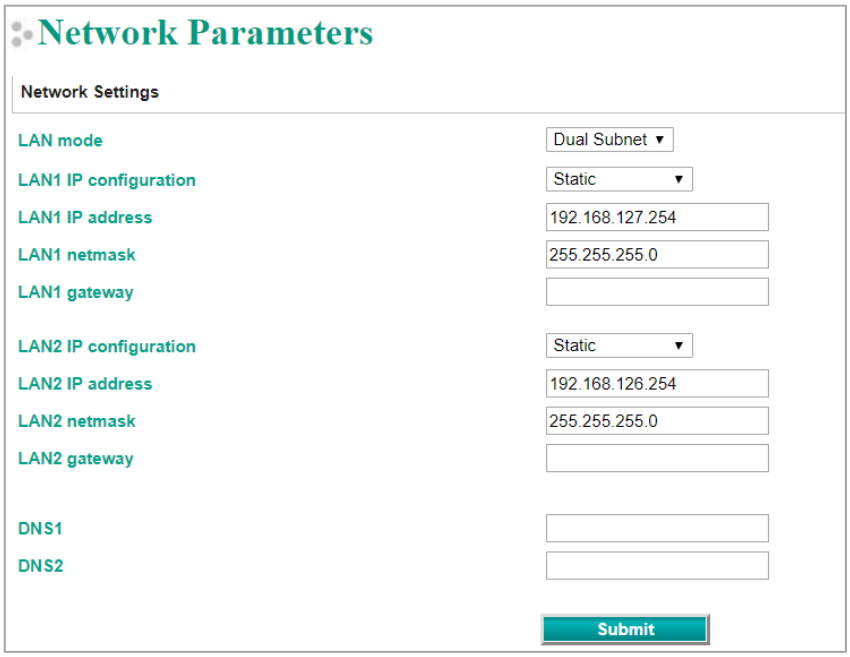
ATTENTION

When changing the local time, select the time zone first. The time display will be updated to reflect the specified time zone.

Network Settings

The **Network** tab is where the unit's network settings are configured. You can change the **LAN mode**, **Network Configuration**, **IP Address**, **Netmask**, **Default Gateway**, and **DNS**.

The MGate MB3660 gateways have dual Ethernet ports with dual MACs. There are two LAN modes: **Dual Subnet** and **Single IP**.



Dual Subnet mode allows the gateway to have two different IP addresses, each with distinct netmask and gateway settings. **Single IP** mode allows users to use the same IP address on both Ethernet ports.

You need to choose which LAN port will be active when the device boots up. The MGate MB3660 will continuously send PING requests to the assigned host to determine the network status. If the active LAN cannot respond, the unit will automatically be handed over to the backup LAN. This mechanism not only detects a physical link down situation but also the actual network status via the PING function. You can plug in both Ethernet cables into the two Ethernet ports using the same IP address. The MGate MB3660 gateway will detect and hand over to the active/backup LAN automatically.

Ethernet Settings

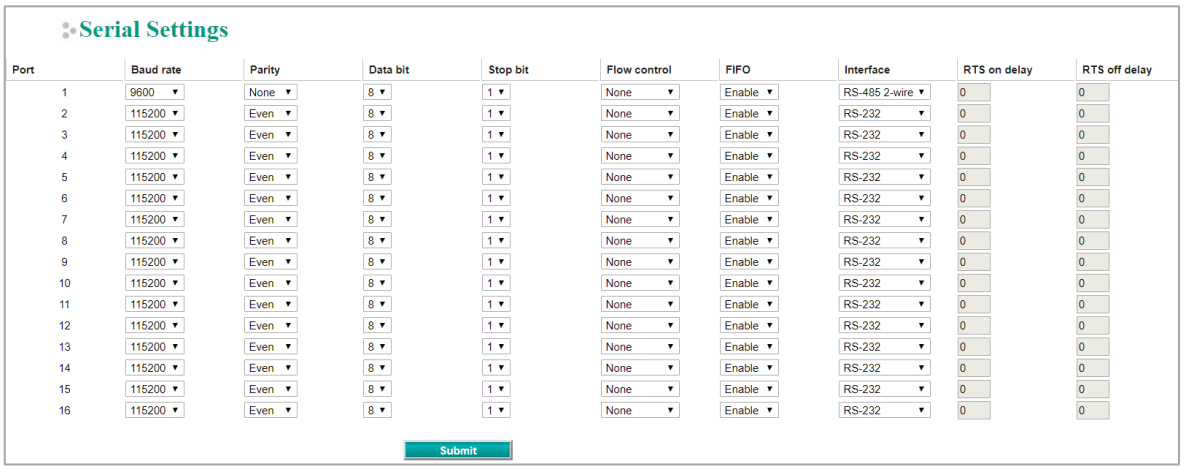
Parameter	Value	Notes
IP Configuration	Static IP, DHCP, BOOTP	Select Static IP if you are using a fixed IP address. Select one of the other options if the IP address is set dynamically.
IP Address	192.168.127.254 (or other 32-bit number)	The IP (Internet Protocol) address identifies the server on the TCP/IP network.
Netmask	255.255.255.0 (or other 32-bit number)	This identifies the server as belonging to a Class A, B, or C network.
Gateway	0.0.0.0 (or other 32-bit number)	This is the IP address of the router that provides network access outside the server's LAN.

DNS Server

Parameter	Value	Notes
DNS Server 1	0.0.0.0 (or other 32-bit number)	This is the IP address of the primary domain name server.
DNS Server 2	0.0.0.0 (or other 32-bit number)	This is the IP address of the secondary domain name server.

Serial Settings

The **Serial** tab is where each serial port's communication parameters are configured. You can configure **Baudrate**, **Parity**, **Stop Bit**, **Flow Control**, **FIFO**, **Interface**, **RTS on delay**, and **RTS off delay**.



Serial settings

Parameter	Value	Notes
Baudrate	Support standard baudrates (bps): 50/ 75/ 110/ 134/ 150/ 300/ 600/ 1200 1800/ 2400/ 4800/ 7200/ 9600/ 19200/ 38400/ 57600/ 115200/ 230.4k/ 460.8k/ 921.6k	
Parity	None, Odd, Even, Mark, Space	
Data bits	8	
Stop bits	1, 2	
Flow control	None, RTS/CTS, RTS Toggle	RTS Toggle will turn off RTS signals when there is no data to be sent. If there is data to be sent, RTS will turn on before data transmission and off after the transmission completes.
FIFO	Enable, Disable	The internal buffer of UART. Disabling FIFO can reduce the latency time when receiving data from serial communications, but this will also slow down the throughput.
Interface	RS-232 RS-422 RS-485 2-wire RS-485 4-wire	
RTS on delay	0 to 100 ms	Only available for RTS Toggle
RTS off delay	0 to 100 ms	Only available for RTS Toggle

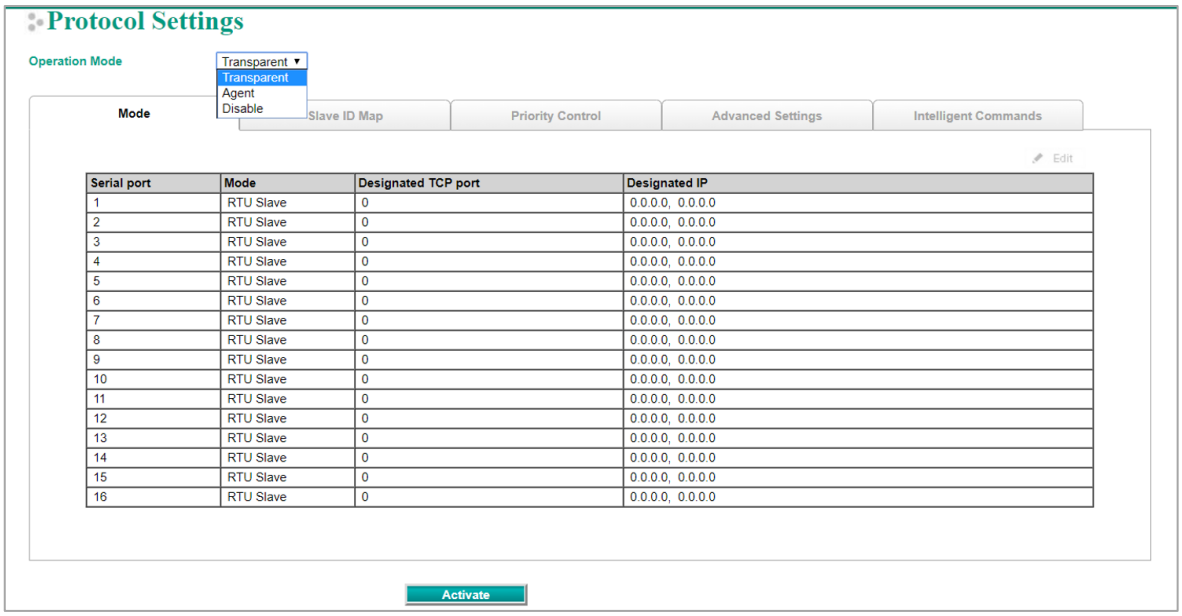
RTS Toggle

The **RTS Toggle** function in flow control is used for **RS-232** mode only. This flow-control mechanism is achieved by toggling the RTS pin in the transmission direction. When activated, data will be sent after the RTS pin is toggled ON for the specified time interval. After data transmission is finished, the RTS pin will toggle OFF for the specified time interval.

Protocol Settings

The MGate MB3660 provides two operation modes for Modbus communication: **Transparent mode** and **Agent Mode**; and can select **Disable** to deactivate it.

In **Transparent mode**, the gateway will bypass and translate Modbus commands between Modbus TCP and RTU/ASCII. In **Agent mode**, the gateway will actively poll the Modbus server/slave devices and store the data in the gateway's memory. The Modbus client/master can retrieve Modbus server/slave devices' data via the gateway's memory.

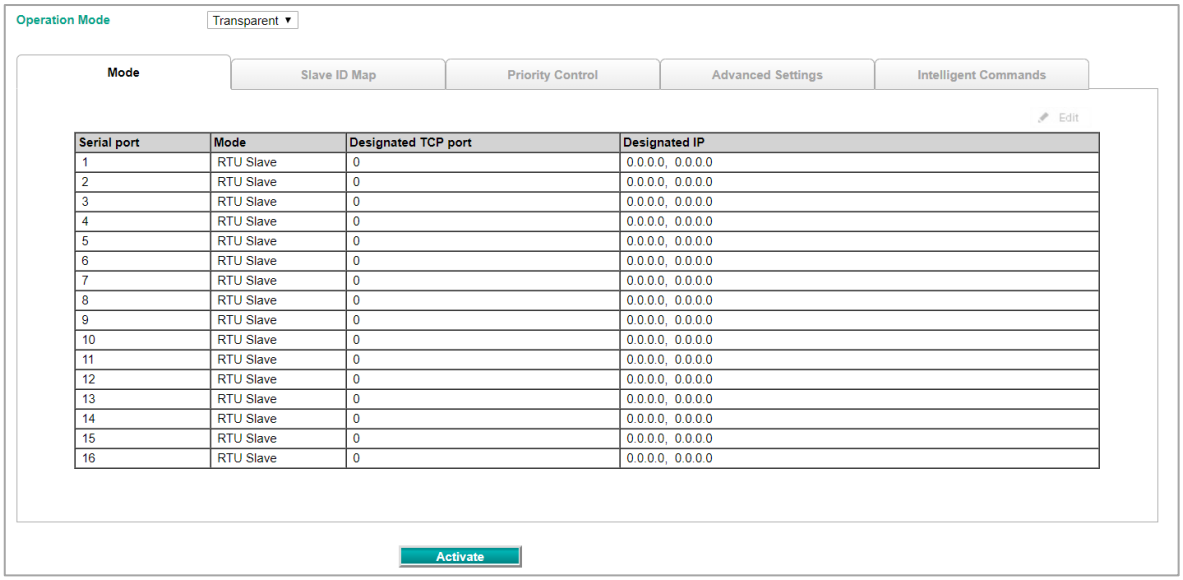


Serial port	Mode	Designated TCP port	Designated IP
1	RTU Slave	0	0.0.0.0, 0.0.0.0
2	RTU Slave	0	0.0.0.0, 0.0.0.0
3	RTU Slave	0	0.0.0.0, 0.0.0.0
4	RTU Slave	0	0.0.0.0, 0.0.0.0
5	RTU Slave	0	0.0.0.0, 0.0.0.0
6	RTU Slave	0	0.0.0.0, 0.0.0.0
7	RTU Slave	0	0.0.0.0, 0.0.0.0
8	RTU Slave	0	0.0.0.0, 0.0.0.0
9	RTU Slave	0	0.0.0.0, 0.0.0.0
10	RTU Slave	0	0.0.0.0, 0.0.0.0
11	RTU Slave	0	0.0.0.0, 0.0.0.0
12	RTU Slave	0	0.0.0.0, 0.0.0.0
13	RTU Slave	0	0.0.0.0, 0.0.0.0
14	RTU Slave	0	0.0.0.0, 0.0.0.0
15	RTU Slave	0	0.0.0.0, 0.0.0.0
16	RTU Slave	0	0.0.0.0, 0.0.0.0

Transparent Mode

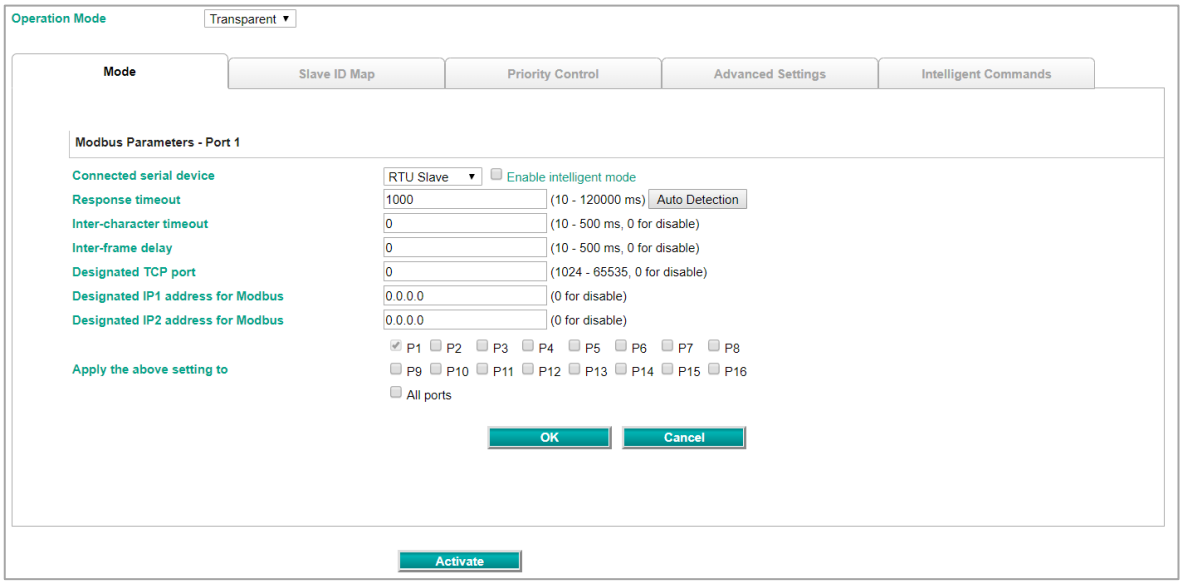
Mode

Select **Transparent** under Operation Mode, and the **Mode** would be listed on the first tab.

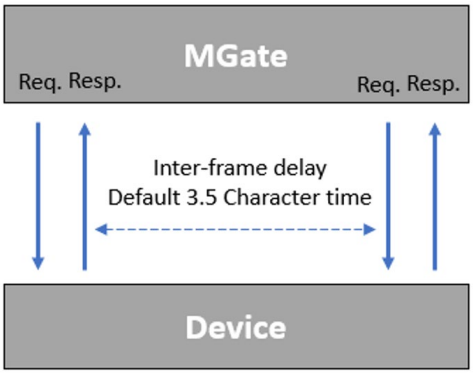


Serial port	Mode	Designated TCP port	Designated IP
1	RTU Slave	0	0.0.0.0, 0.0.0.0
2	RTU Slave	0	0.0.0.0, 0.0.0.0
3	RTU Slave	0	0.0.0.0, 0.0.0.0
4	RTU Slave	0	0.0.0.0, 0.0.0.0
5	RTU Slave	0	0.0.0.0, 0.0.0.0
6	RTU Slave	0	0.0.0.0, 0.0.0.0
7	RTU Slave	0	0.0.0.0, 0.0.0.0
8	RTU Slave	0	0.0.0.0, 0.0.0.0
9	RTU Slave	0	0.0.0.0, 0.0.0.0
10	RTU Slave	0	0.0.0.0, 0.0.0.0
11	RTU Slave	0	0.0.0.0, 0.0.0.0
12	RTU Slave	0	0.0.0.0, 0.0.0.0
13	RTU Slave	0	0.0.0.0, 0.0.0.0
14	RTU Slave	0	0.0.0.0, 0.0.0.0
15	RTU Slave	0	0.0.0.0, 0.0.0.0
16	RTU Slave	0	0.0.0.0, 0.0.0.0

Double-click the intended serial port column to set detailed Modbus parameters or click **Edit** to enter the settings page.



Parameters	Description
Connected serial device	Select the role of the device that is connected to the serial port.
Response timeout	According to the Modbus standard, the time it takes for a server/slave device to respond to a request is defined by the device manufacturer. Based on this response time, a client/master can be configured to wait a certain amount of time for a server/slave's response. If no response is received within the specified time, the client/master will disregard the request and continue operation. This allows the Modbus system to continue operation, even if a server/slave device is disconnected or faulty. The MGate MB3660 can also auto-detect the response timeout. Instead of manually figuring out the appropriate setting, you can click Auto Detection to have MGate figure out the setting for you. Once a value has been recommended, you can fine-tune it to get the best performance.
Inter-character timeout (only for Modbus RTU)	In Modbus RTU slave mode, use this function to define the time-out interval between characters in one frame. When the serial side of MGate receives one character, and the next one comes after the "inter-character timeout" defined, the frame will be discarded because of time-out. The inter-character timeout can be adjusted within the range of 10 to 500 ms or use the default value. The default value in this field is 0 ms, but the default inter-character timeout varies depending on the baudrate setting. When the baudrate is configured below 19200 bps, the default inter-character time-out is set to 1.5 character times. When the baudrate is configured equal to or larger than 19200 bps, a predefined fixed value is used.
Inter-frame delay (only for Modbus RTU)	In Modbus RTU slave mode, use this function to define the time interval between a Modbus RTU response and the next Modbus RTU request. The reason for enabling manual configuration of this value is to accommodate certain scenarios where legacy Modbus devices may not be capable of promptly processing Modbus RTU requests. By setting a longer interval value, you can increase tolerance for delays in serial reception and transmission. The inter-frame delay can be adjusted within the range of 10 to 500 ms or use the default value. The default value in this field is 0 ms, but the default inter-frame delay varies depending on the baudrate setting. When the baudrate is configured below 19200 bps, the default inter-frame delay is set to 3.5 character times. When the baudrate is configured equal to or larger than 19200 bps, a predefined fixed value is used. How to calculate Modbus character time? For example, if the baudrate is 9600 bps, 1 character time is about 1 ms. In a serial frame (11 bits, including start bit, data, parity bit and stop bit), 9,600 bps approximately equals to 960 characters/s, so transmitting 1 character needs about 1/960 = 1 ms.

Parameters	Description
	
Designated TCP port	In RTU/ASCII slave mode, a Modbus command from a specified TCP port can be routed to a specified serial port. It is the destination TCP port from the viewpoint of a Modbus TCP client/master. Under this routing mechanism, each serial port can be accessed by up to 4 TCP clients/masters.
Designated IP 1/2 address for Modbus	In RTU/ASCII slave mode, a Modbus command sent to a specified IP address can be routed to a specified serial port. It is the destination IP address from the viewpoint of a Modbus TCP client/master. If the command comes from LAN1 and LAN2 respectively, set different IP addresses accordingly. Under this routing mechanism, each serial port can be accessed by up to 4 TCP clients/masters.

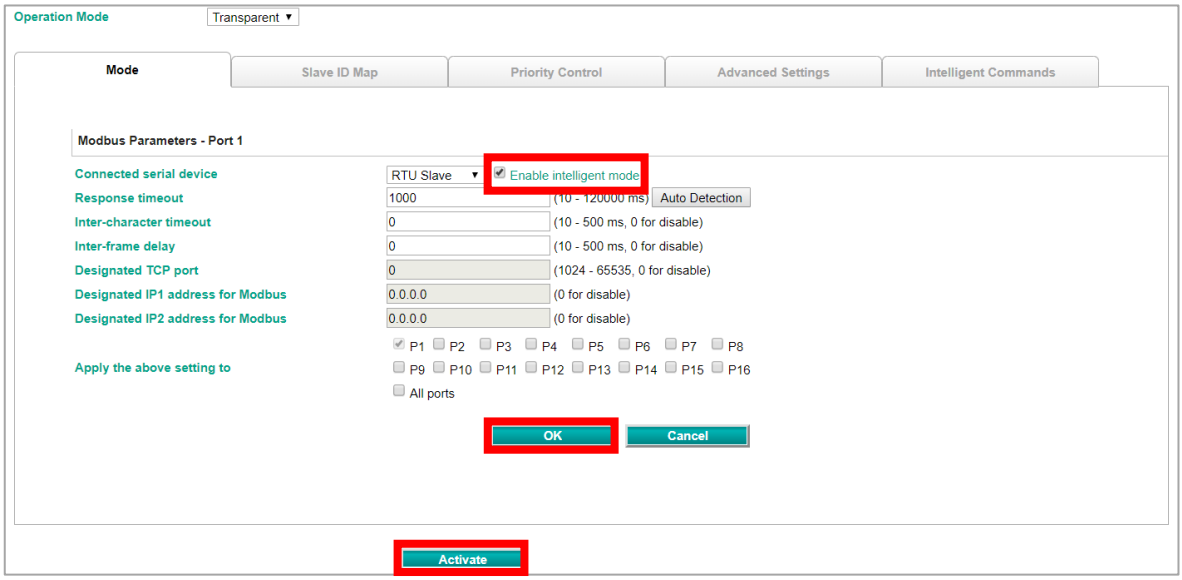
For convenience, you can apply the setting to other serial ports by checking the desired ports or to all ports by selecting the **All ports** checkbox. This feature can dramatically reduce the time needed to configure Modbus gateways that service many serial ports.

Enable Intelligent Mode

In order to provide better performance as an agent gateway under transparent mode, but without requiring users to key in a lot of Modbus commands, the MGate MB3660 series of gateways are designed with an innovative command learning function, which can be enabled with a single mouseclick on **Enable Intelligent Mode**. Then, press **OK** and **Activate**, and the function will be activated.

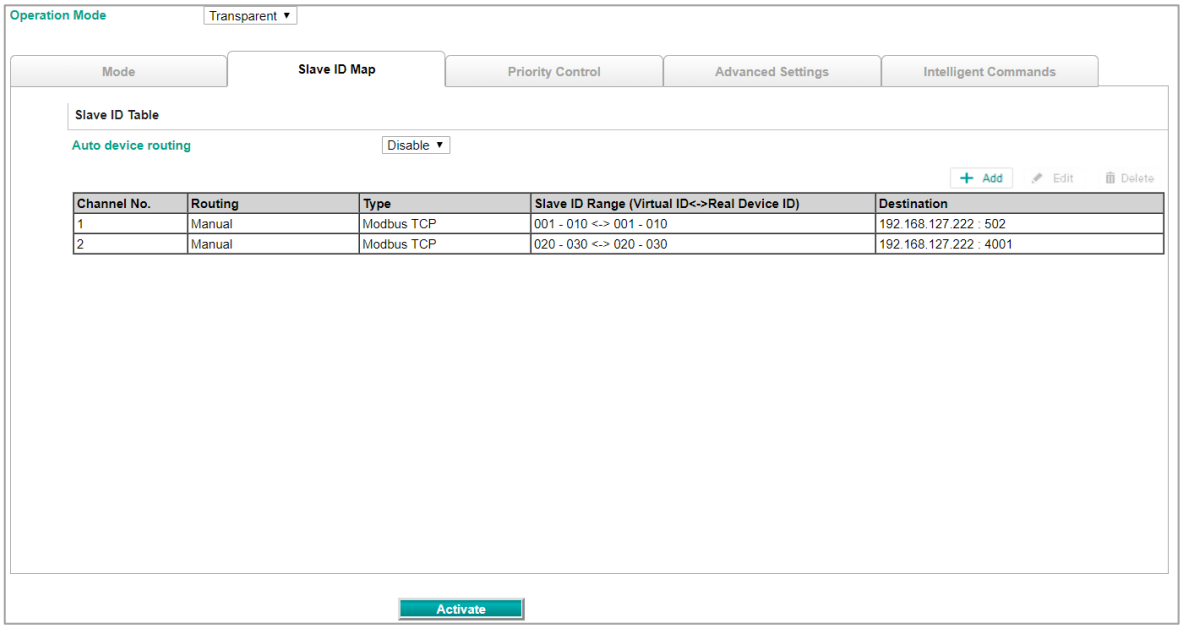
Once activated, the gateway will learn and memorize the Modbus commands it receives. While a command has been learned, the gateway will act as though it were in agent mode and actively send Modbus requests to the relevant Modbus devices. Since the data is saved in a different memory space that can be accessed by the SCADA system, the SCADA system can retrieve Modbus response data directly from the gateway's memory, instead of waiting for the data to pass through the Modbus devices, dramatically increasing communication performance.

To complete **Intelligent Mode**, please check the session of **Intelligent Commands**.

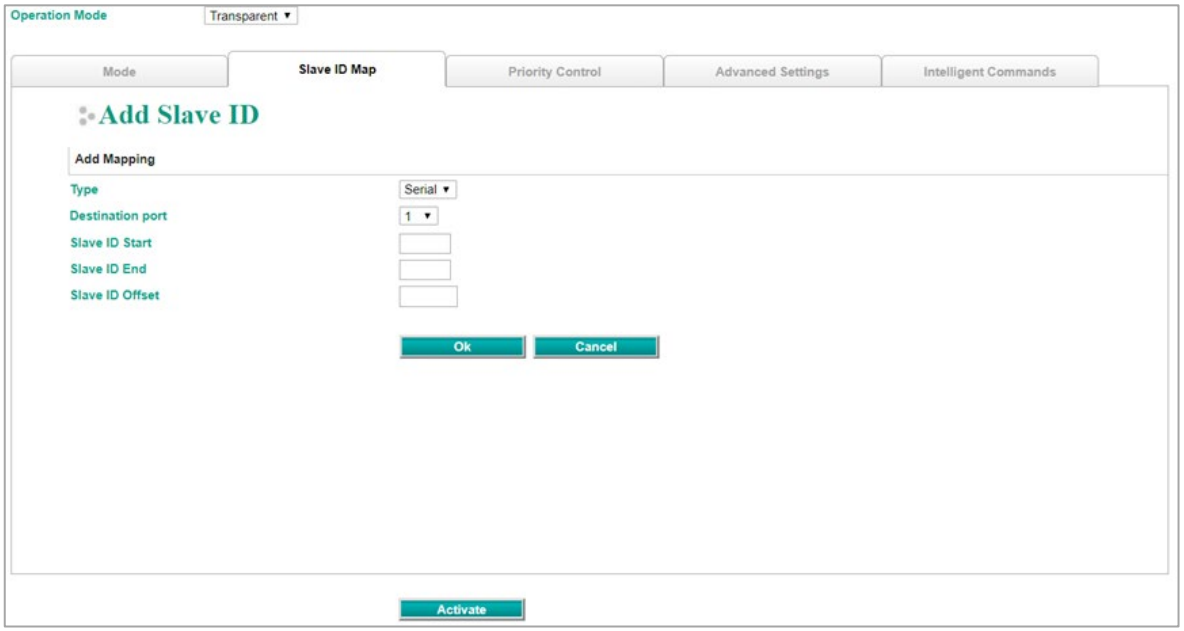


Slave ID Map

The **Slave ID Map** tab is where slave IDs are managed. The definitions on this tab determine how Modbus requests will be routed by the unit. With the slave ID table, a routing mechanism is achieved for gateways with two or more serial ports. Since the Modbus devices (all with different slave IDs) are connected to the different serial ports of a gateway, the Modbus requests should be routed to the specific serial port that is connected to the targeted Modbus server/slave device.



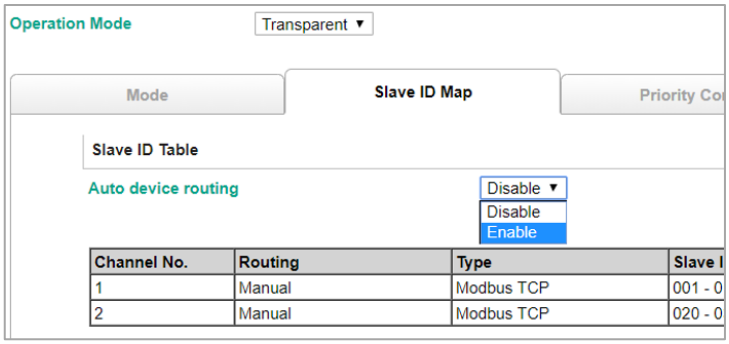
Traditionally, there is a factory default routing. For example, the Modbus requests with slave ID 001 to 005 will be routed to serial port1, and the Modbus requests with slave ID 006~010 will be routed to serial port2. Users must set their own customized routing. Select the one you want to set and click **Add/Edit/Delete** buttons to change the existing routing. Set each port one by one.

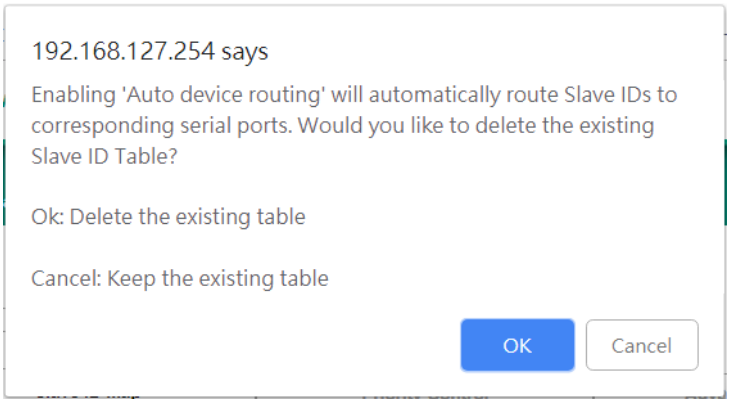


Auto Device Routing (Patented)

The Moxa Modbus gateways provide an auto routing mechanism that eliminates the burdensome task of setting the slave ID table manually. Now, users no longer need to set the routing table. The Moxa Modbus gateways will help detect and route correctly.

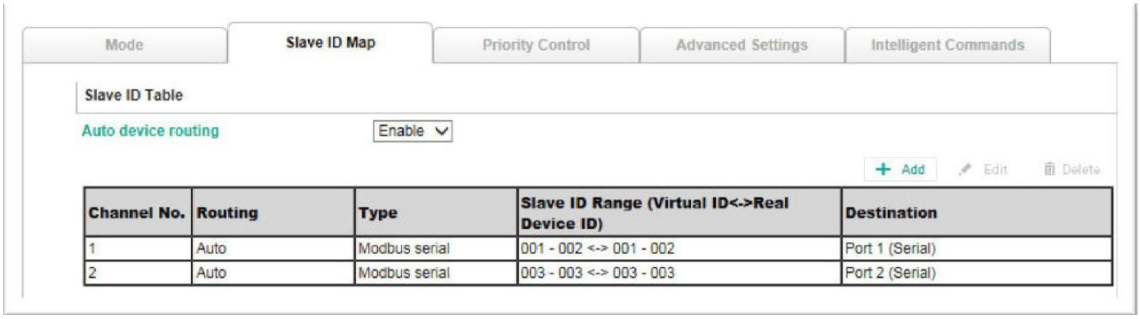
Enable **Auto Device Routing**, and a message window will pop up.





Click **OK** to delete the existing (factory default or user-set) routing table; the auto routing mechanism will automatically find the correct serial port that connects to the target Modbus device. Moreover, if a device is added to the gateway later, the gateway can also route it correctly.

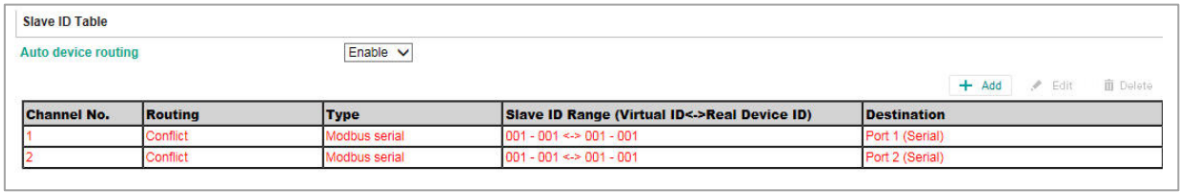
Once the Modbus Client/Master sends Modbus requests, the gateway will auto-detect the routing and show results in the Slave ID table.



This snapshot shows the routing mechanism is in Auto mode, and the gateway detects that slave ID 1 and 2 are connected to port 1 and slave ID 3 is connected to port 2.

If a *conflict* exists, the table will show the error in red for notification purposes.

For example, two Modbus devices with the same slave ID are connected to serial port 1 and port 2.



Channel No.	Routing	Type	Slave ID Range (Virtual ID<->Real Device ID)	Destination
1	Conflict	Modbus serial	001 - 001 <-> 001 - 001	Port 1 (Serial)
2	Conflict	Modbus serial	001 - 001 <-> 001 - 001	Port 2 (Serial)

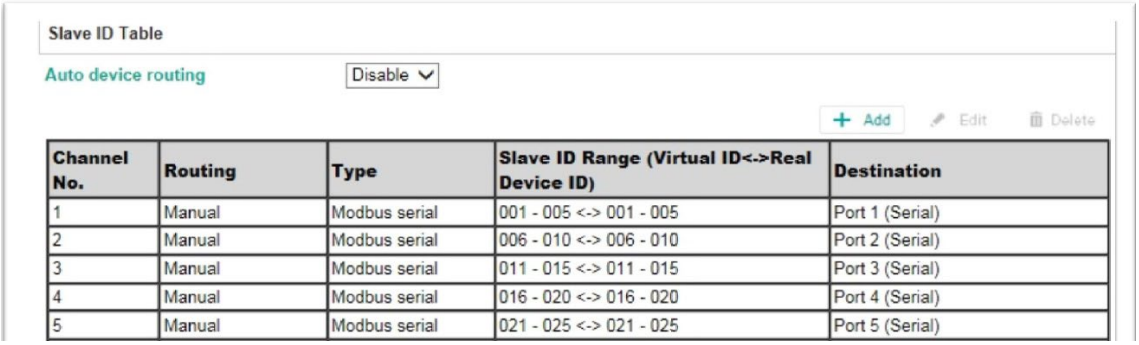
On the other hand, if you have manually set routing table already and would like to enable the auto routing mechanism for the newly added devices, click **Cancel** to keep the existing routing table. The gateway will keep the existing user-set routing table and automatically route the new devices. Note that if a newly added device cannot be polled by the Modbus client/master correctly; the slave ID of this newly added device might be set in the existing user-set table. Users must change the existing user-set table.

Another scenario is when legacy Modbus devices cannot set slave ID arbitrarily. It has a fixed slave ID or a short range of slave IDs. Then, the slave IDs of the Modbus devices connected to different serial ports will be in conflict. The virtual-to-real slave ID function can help you connect the same slave ID devices to different serial ports of a gateway.

Let's assume there are two legacy Modbus devices (named Device A, and Device B) with the same slave ID 1. Device A is connected to serial port 1 and Device B is connected to serial port 2. On the Modbus Client/Master side, the Modbus request for Device A is recognized by slave ID 1, and the request for Device B is recognized by slave ID 2. You must set the offset for the duplicate slave ID.

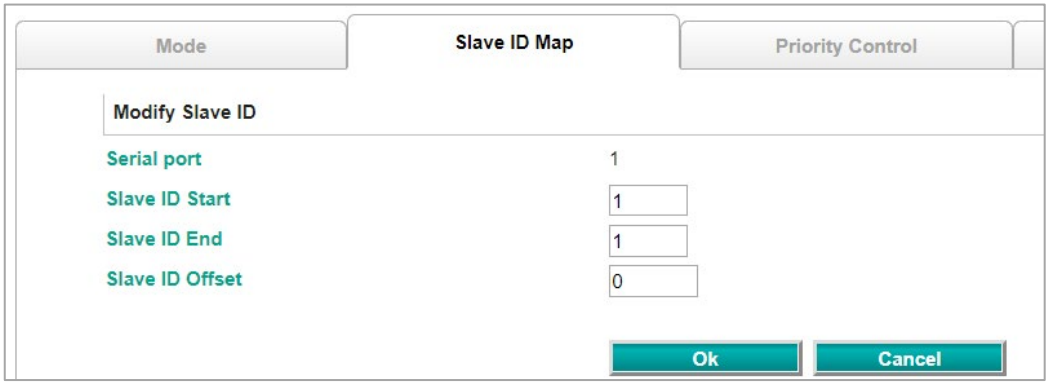
	Real device ID	Virtual ID(Device ID recognized by Modbus Client/Master)	Offset
Device A	1	1	0
Device B	1	2	-1

Original Setting

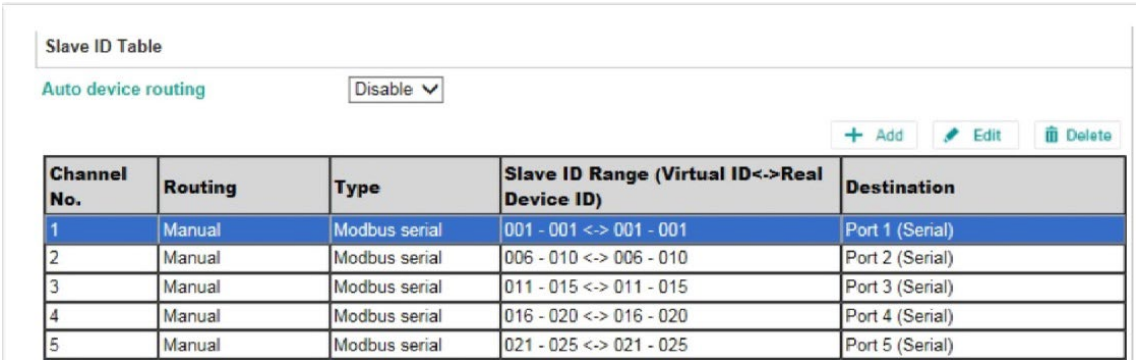


Channel No.	Routing	Type	Slave ID Range (Virtual ID<->Real Device ID)	Destination
1	Manual	Modbus serial	001 - 005 <-> 001 - 005	Port 1 (Serial)
2	Manual	Modbus serial	006 - 010 <-> 006 - 010	Port 2 (Serial)
3	Manual	Modbus serial	011 - 015 <-> 011 - 015	Port 3 (Serial)
4	Manual	Modbus serial	016 - 020 <-> 016 - 020	Port 4 (Serial)
5	Manual	Modbus serial	021 - 025 <-> 021 - 025	Port 5 (Serial)

Select the first channel and click **Modify**. The *Slave ID* here represents the Virtual ID recognized by the Modbus client/master.



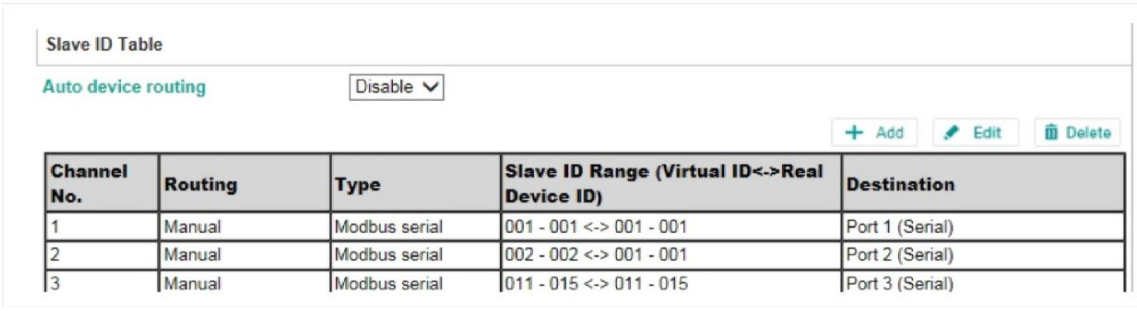
The routing table will be as follows:



Channel No.	Routing	Type	Slave ID Range (Virtual ID<->Real Device ID)	Destination
1	Manual	Modbus serial	001 - 001 <-> 001 - 001	Port 1 (Serial)
2	Manual	Modbus serial	006 - 010 <-> 006 - 010	Port 2 (Serial)
3	Manual	Modbus serial	011 - 015 <-> 011 - 015	Port 3 (Serial)
4	Manual	Modbus serial	016 - 020 <-> 016 - 020	Port 4 (Serial)
5	Manual	Modbus serial	021 - 025 <-> 021 - 025	Port 5 (Serial)

Then, select the second channel and click **Edit**. Since the virtual ID recognized by the Modbus client/master side is 2, and the real slave ID of the device B is ID 1, the offset should be set as -1.

The routing table will be as follows:

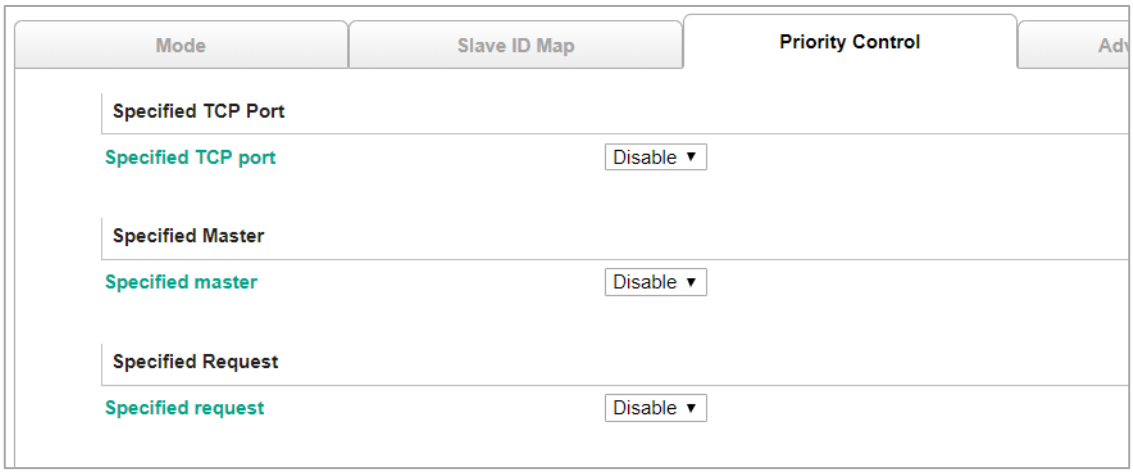


Channel No.	Routing	Type	Slave ID Range (Virtual ID<->Real Device ID)	Destination
1	Manual	Modbus serial	001 - 001 <-> 001 - 001	Port 1 (Serial)
2	Manual	Modbus serial	002 - 002 <-> 001 - 001	Port 2 (Serial)
3	Manual	Modbus serial	011 - 015 <-> 011 - 015	Port 3 (Serial)

Now, the Modbus client/master can send a request with slave ID 1 to the Modbus device A connected to serial port 1 as well as send a request with slave ID 2 to the Modbus device B connected in serial port 2.

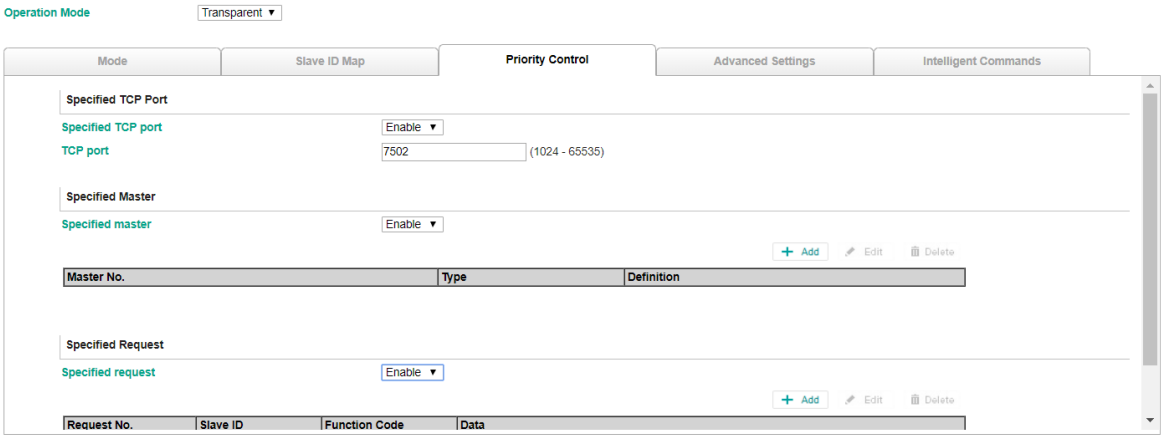
Priority Control

The **Priority Control** tab is where emergency requests are enabled and configured.



Priority control is designed for requests that are sent to Modbus RTU/ASCII servers/slaves. Since Modbus RTU/ASCII servers/slaves cannot handle multiple requests, the Modbus gateway must send each request individually and wait for the response before sending the next request. As requests stack up, the response time can suffer. This can cause problems for certain critical requests that require an immediate response.

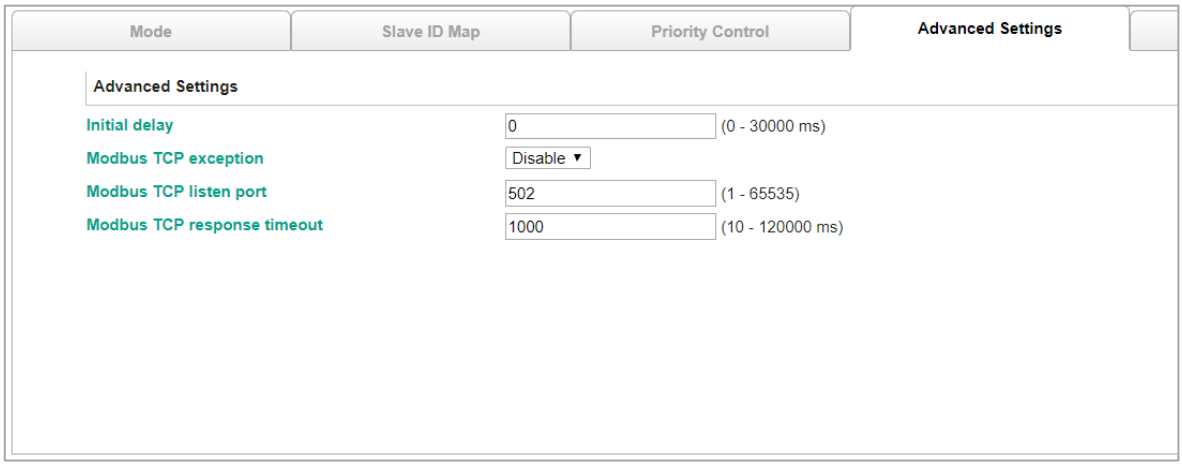
With priority control, you can specify that certain requests are sent to the front of the queue for more immediate response times. Priority requests can be specified by client/master (IP address or serial port), TCP port, or command type (slave ID, function code, or data). When the Modbus gateway identifies a priority request, the request will immediately be placed at the front of the queue.



To define a priority request, enable the appropriate priority scheme (i.e., **Specified Masters**, **Specified TCP Port**, or **Specified Requests**). Then, specify the parameter(s) that will show a priority request. Finally, click **Add/Modify** to apply this definition. (This last step is unnecessary for **Specified TCP Port**.)

Advanced Settings

The **Advanced Modbus** tab is where certain adjustments can be made to fine-tune the communication between different Modbus networks. You can configure **Initial Delay, Modbus TCP Exception, Modbus TCP listen port,** and **Modbus TCP Response Time-out.**



Parameter	Value
Initial delay	0 to 30000 ms
Modbus TCP exception	Enable or Disable
Modbus TCP listen port	1 to 65535
Modbus TCP response timeout	10 to 120000 ms

Initial Delay

Some Modbus servers/slaves may take more time to boot up than other devices. For certain environments, this may cause the entire system to suffer from repeated exceptions during the initial boot-up. You can force the MGate to wait after booting up before sending the first request with the **Initial Delay** setting.

Modbus TCP Exception

The MGate MB3660 is a protocol gateway that transparently passes requests and responses between the Ethernet and serial interfaces. In some situations, it may be necessary for the gateway to return an exception in response to a request from a Modbus TCP client/master. This is enabled or disabled with the **Modbus TCP Exception** setting. When enabled, the unit can return two types of exception:

Exception	Conditions
Timeout	There is no response from the server/slave. Maybe the device is offline, or the serial cable is broken.
Request dropped	There are two situations that will result in this exception: The request queue is full (32 request queue for each client/master) The destination ID is not included in the slave ID map.

Not all Modbus TCP clients/masters require this exception, so it is up to you to determine if this setting should be enabled.

Modbus TCP Listen Port

Allow you to change Modbus TCP listen port from the default value (502).

Modbus TCP Response Timeout

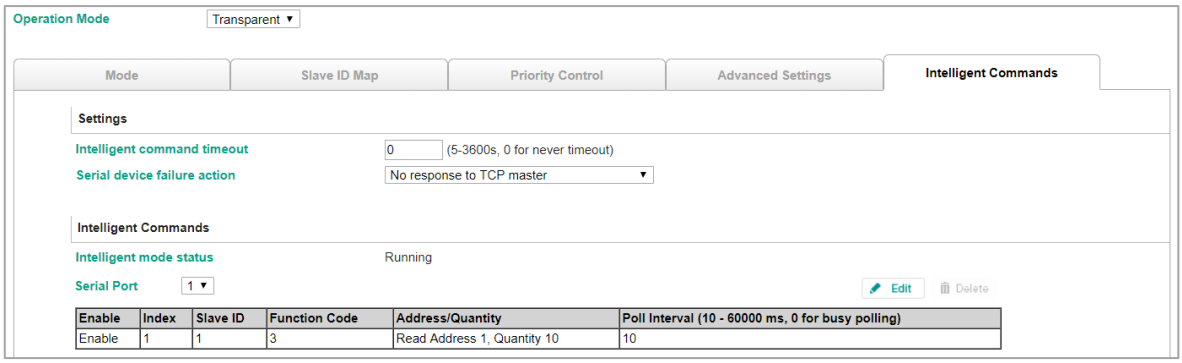
According to the Modbus standard, the time that it takes for a server/slave device to respond to a request is defined by the device manufacturer (refer to Appendix A for details). Based on this response time, a client/master can be configured to wait a certain amount of time for a server/slave's response. If no response is received within the specified time, the client/master will disregard the request and continue operation. This allows the Modbus system to continue operation even if a server/slave device is disconnected or faulty.

On the MGate MB3660, the **Modbus TCP response timeout** field is used to configure how long the gateway will wait for a response from a Modbus ASCII or RTU server/slave. Refer to your device manufacturer's documentation to manually set the response timeout.

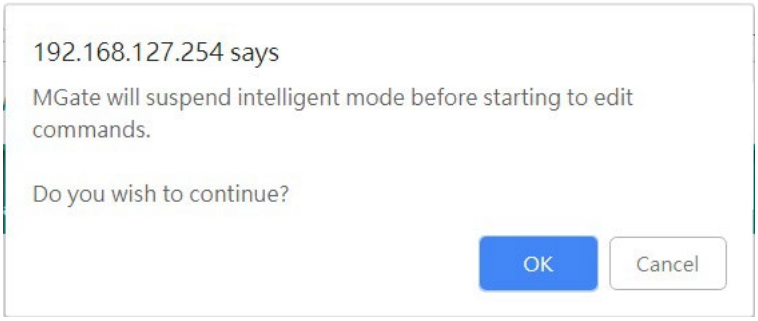
Intelligent Commands

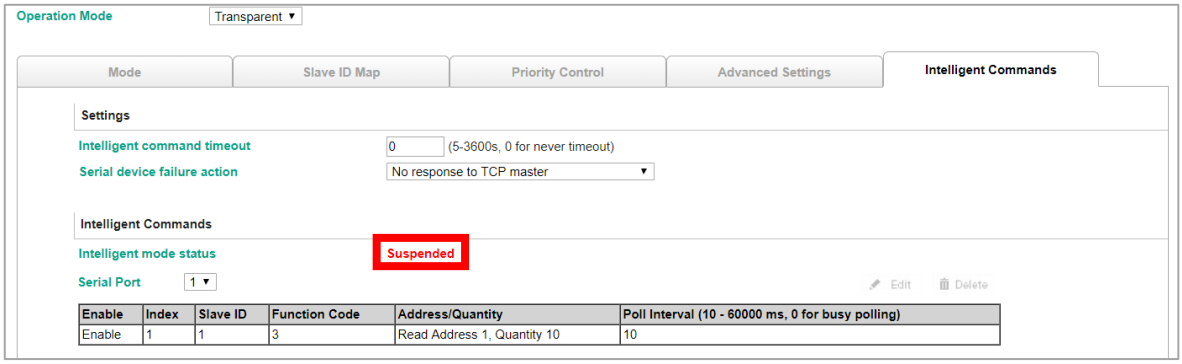
When the MGate MB3660 runs under transparent mode with the serial device as a Modbus server/slave, you do not need to type Modbus commands (copied from the SCADA system) into the gateway. However, transparent mode uses a traditional round-robin polling mechanism, which supports only one request-response action at a time, resulting in poor performance. For applications that use large numbers of Modbus devices, the inherent latency is unacceptable from the SCADA system's point of view. Agent mode provides better performance, since the gateway actively polls the devices to retrieve data from the remote site. SCADA systems can retrieve Modbus device data directly from the gateway's memory, instead of waiting for the gateway to pass commands to the serial ports.

In order to activate **Intelligent Mode**, the feature of **Enable Intelligent Mode** is needed to be preset under **Mode (Modbus Parameter)** settings.

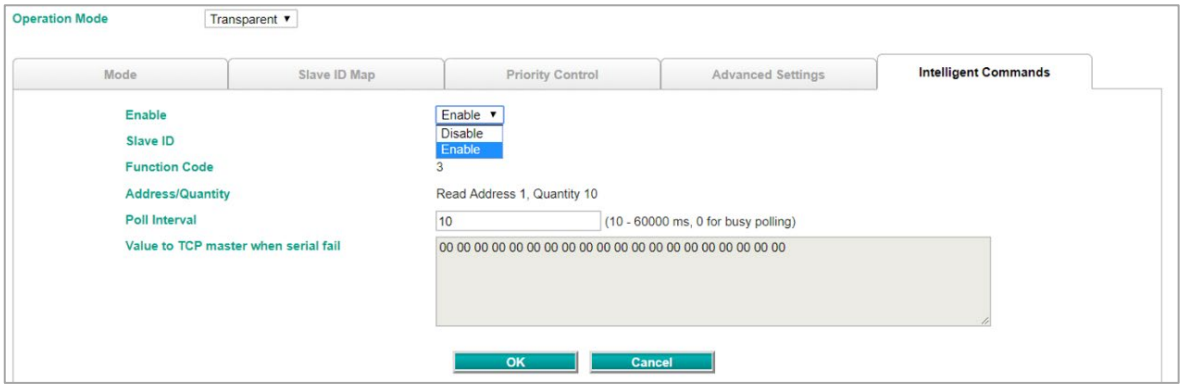


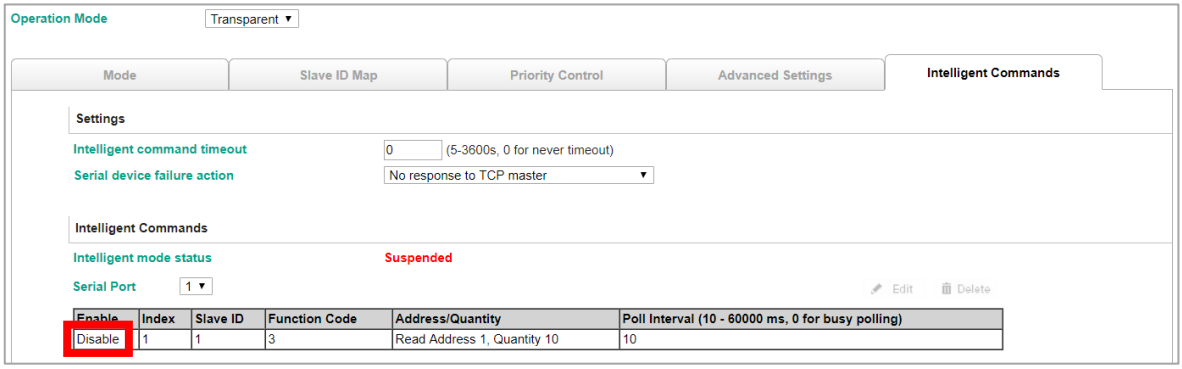
The learned Modbus commands will be shown on the **Intelligent Commands** tab. The gateway will act as in agent mode when intelligent mode is activated. Once the Modbus command is learned, the gateway will start to actively poll the Modbus device according to the command learned from the SCADA system. You can edit the learned Modbus commands received from the Modbus client/master by clicking the **Edit** button. Once you click it, the pop-up message regarding suspending intelligent mode will appear. To continue this process, the active polling function will be activated. Also, the status of intelligent mode will change to **Suspended**.





Click **Edit** to open the edit page in order to disable the Modbus command or reactivate it (enable) when needed. By clicking the **delete button**, the intelligent commands can be deleted directly.





The gateway now acts as though it were in agent mode and actively polls the Modbus server/slave devices. The Modbus Client/Master will retrieve the Modbus device's data directly from the gateway's memory. If the serial device fails, the Modbus client/master will not be aware of the failure since it is still getting the Modbus server/slave's data from the gateway's memory. The gateway is designed with a failed report mechanism to inform the Modbus Client/Master. You may set a pre-defined value for the serial port abnormality warning in the **Value to TCP master when serial fail** text box. When the serial device fails, the gateway will automatically write this predefined value to memory. The Modbus client/master will be aware of the serial device failure when it receives this predefined value.

Agent Mode

When running in agent mode, two Modbus roles must be set. One is the Ethernet side (Modbus TCP), and the other is the serial side (Modbus RTU/ASCII).

Modbus TCP

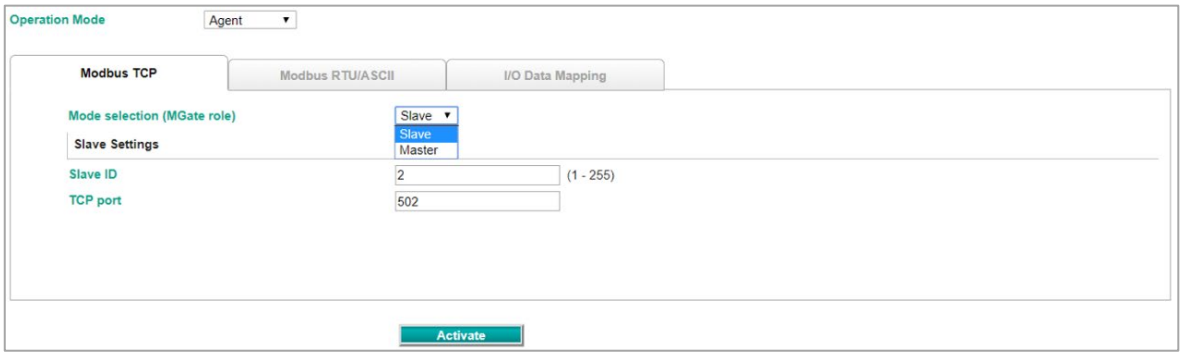
The MGate MB3660 supports a Modbus TCP function with Master and Slave modes. For slave mode, the MGate works as a server/slave and waits for incoming connections from the Modbus TCP client/master. In master mode, the MGate works as a client/master and tries to build a TCP connection with the remote Modbus TCP server/slave device.

NOTE

Under Modbus TCP slave mode, the recommended polling interval of each Modbus request is 2000 ms when establishing 256 Modbus TCP connections. For a requirement that needs a shorter polling interval than 2000 ms, the number of connections should be adjusted accordingly.

Slave Mode Settings

The MGate MB3660 supports Modbus slave mode, which means the MGate will work as a server/slave and wait for incoming connection requests. The default TCP listen port is 502. In this mode, the MGate will wait for incoming Modbus TCP requests and use the internal memory as the server/slave register to respond.

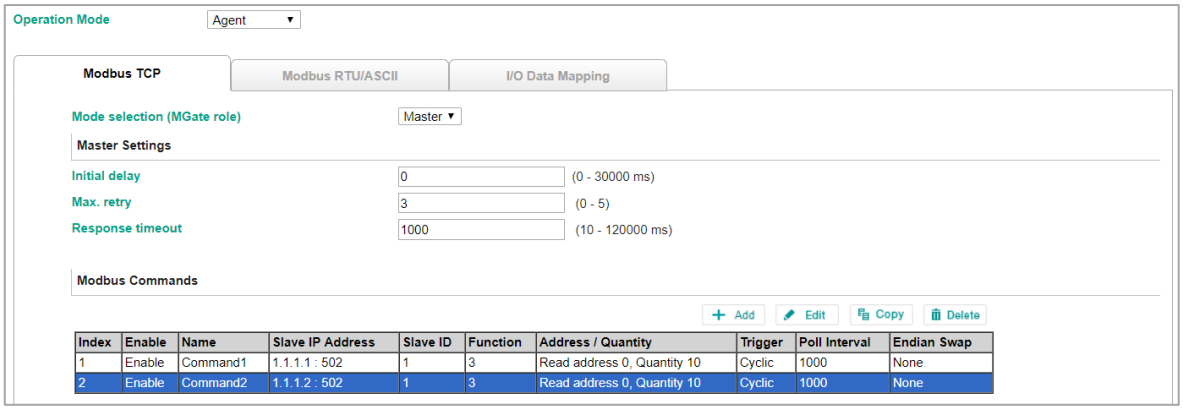


Change the Slave ID settings to match the system requirements. The default TCP port for Modbus TCP is 502, so you may need to change if there is a firewall in place.

Parameters	Value	Description
Slave ID	1 to 255	The Modbus address of the MGate.
TCP port	1 to 65535	The local TCP port for the MGate.

Master Mode Settings

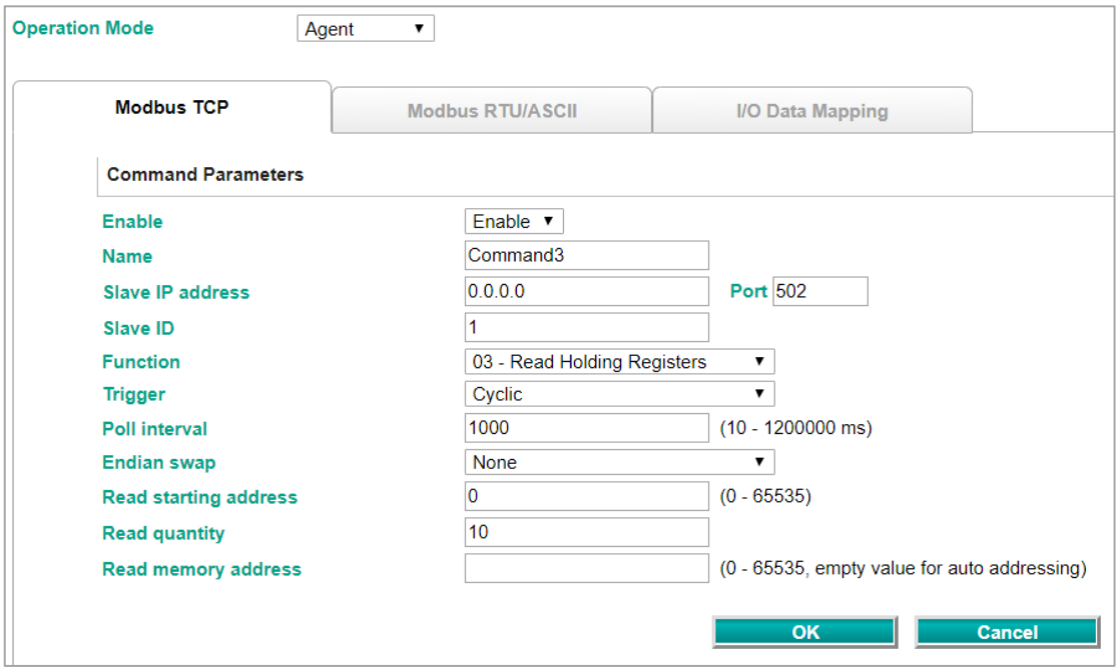
The MGate MB3660 supports Modbus TCP master mode, which means the MGate will work as a client/master and send the Modbus command request to the server/slave device actively. You will need to configure each Modbus command manually. On this page, users can see all the commands listed in the table.



Parameters	Value	Description
Initial delay	0 to 30000 ms	Some Modbus servers/slaves may take more time to boot up than other devices. In some environments, this may cause the entire system to suffer from repeated exceptions during the initial boot-up. You can force the MGate to wait after booting up before sending the first request with the Initial Delay setting.
Max. retry	0 to 5	This is used to configure how many times the MGate will try to communicate with the Modbus server/slave.
Response timeout	10 to 12000 ms	This is used to configure how long the MGate will wait for a response from a Modbus server/slave.

To add a new command or change the existing one, click the **Add** button or **Modify** button and a new dialog box will appear. To remove Modbus commands, select the specific command and then click the **Remove** button.

To communicate with remote Modbus TCP server/slave devices, specify the Modbus command for each device. For each Modbus read/write command, specify the internal memory address for data exchange. For the read command, the information received from remote devices will be updated to the specified internal memory address. For the write command, the data in the specified internal memory address will be sent to the remote device. The data will be used to update the remote device register.



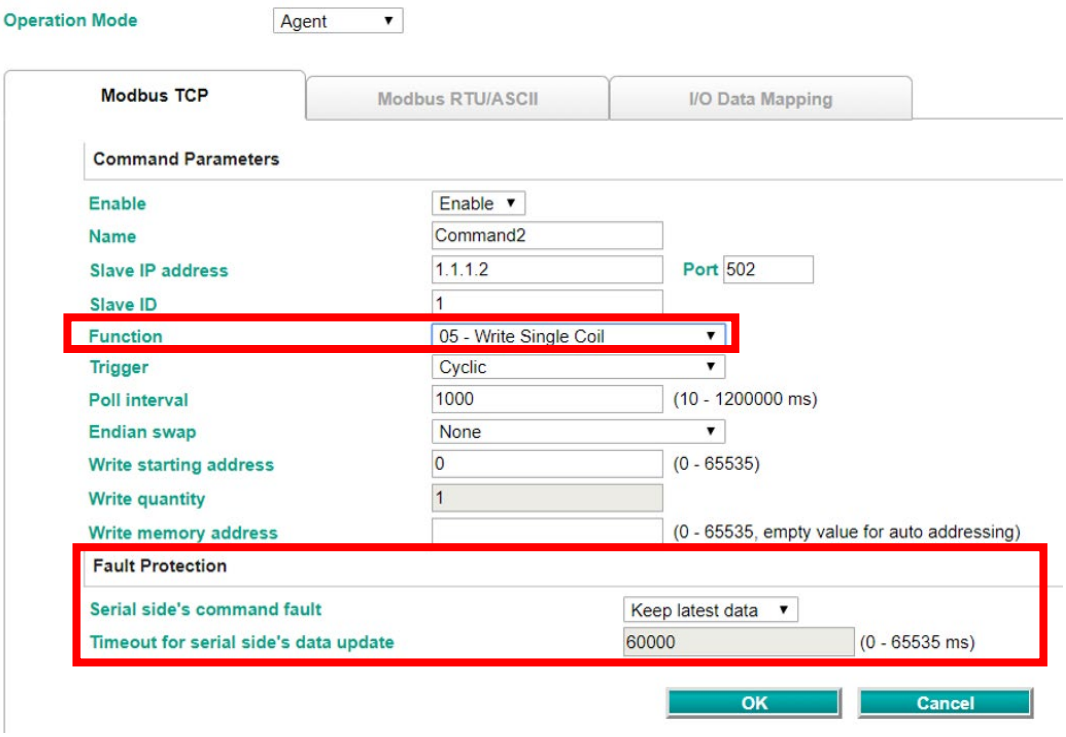
Each remote device may need more than one command for communication, so you will need to input all the commands manually.

Parameters	Description
Enable	To enable/disable this Modbus command
Name	Enter a name to help identify the command, such as the location, function, etc.
Slave IP address	The IP address of remote server/slave device.
Port	The TCP port number of remote server/slave devices. 1 to 65535
Slave ID	The Modbus slave ID that this server/slave module will accept. 0: Broadcasting 1 to 255: Device specific.
Function	When a message is sent from a Client/Master to a Server/Slave device the function code field tells the server/slave what kind of action to perform. We support the following function codes so far: 01: Read coils 02: Read discrete inputs 03: Read holding registers 04: Read input register 05: Write single coil 06: Write single register 15: Write multiple coils 16: Write multiple registers 23: Read/Write multiple registers
Trigger	Disable: The command is never sent Cyclic: The command is sent cyclically at the interval specified in the **Poll Interval** parameter. Data change: The data area is polled for changes at the time interval defined by Poll Interval. A command is issued when a change in data is detected.
Poll interval	Polling intervals are in milliseconds; since the module sends all requests in turns, the actual polling interval also depends on the number of requests in the queue and their parameters. The range is from 10 to 1,200,000 ms.

Parameters	Description
Endian swap	Data Byte Swapping **None**: Don't need to swap **Byte**: 0x0A, 0x0B, 0x0C, 0x0D becomes 0x0B, 0x0A, 0x0D, 0x0C. **Word**: 0x0A, 0x0B, 0x0C, 0x0D becomes 0x0C, 0x0D, 0x0A, 0x0B. **ByteWord**: 0x0A, 0x0B, 0x0C, 0x0D becomes 0x0D, 0x0C, 0x0B, 0x0A.
Read starting address	Station Address. The range is from 0 to 65535
Read quantity	Specifying how many quantities to write. There are two kinds of quantity units, bit and 16bits, which are associated with the function field. The range is from 1 to 125.
Write a starting address	Station Address. The range is from 0 to 65535
Write quantity	Specifying how many quantities to write. There are two kinds of quantity units, bit and 16bits, which are associated with the function field. The range is from 1 to 123.
Read/Write memory address	Manually decide the read or write starting address in the gateway's memory. The range is from 0 to 65535. Set this value as empty for auto addressing by the system.
Opposite side's command fault	For Modbus TCP master mode, the opposite side refers to the serial port side. The Modbus writer command is sent from the serial port to the TCP side. Once the serial connection fails, the gateway will not be able to receive the serial Client/Master's write command, but the gateway will continuously send Modbus write commands to the Modbus TCP server/slave device. To avoid problems when the serial side fails, the MB3660 can be configured to react in one of three ways: keep the latest data, clear data to zero, and user-defined value.
Timeout for opposite side's data update	Defines the timeout for the serial side. The range is from 0 to 65535 ms.

Fault Protection

To diagnose the upstream connection that is lost, the **Fault Protection** function would be quite useful by sending a predefined setting to field devices to prevent incorrect actions. This feature will appear while setting the "Write" function codes under **Command Parameters**.



The MB3660 supports a Fault Protection function when in agent mode. You can configure the criteria used to determine what to do when the write command is no longer received from the client/master side. For example, when a cable comes loose accidentally, the most up-to-date write command from the client/master side will not be received by the gateway. Hence, the server/slave device will use the latest command from the gateway, which is now out-of-date, creating an inconsistency between the client/master and server/slave devices. To avoid this problem, MB3660 supports options to determine which actions should be taken when the client/master's side is disconnected from the gateway.

The **Serial side's command fault** offers several options to be selected.

Options	Description
Keep latest data	The gateway will write the same data to the server/slave device.
Clear data to zero	The gateway will write zero values to the server/slave device.
User-define value	A user-defined value will be written to the server/slave device.

Use the **Timeout for serial side's data update** item to set how long the gateway will wait to activate this function.

Status Monitoring

The **Fault Protection** function assists to detect upstream connection failure, whereas the **Status Monitoring** function supports diagnosing downstream connection lost. The Status Monitoring function supports and built-in MGate MB3660 Series, which provides status information of field devices when the MGate is being used as a client/master; information includes an alive list, counter, the result of commands issued, etc.

For Modbus gateways in agent mode, if a server/slave device fails, or a cable comes loose, generally the gateway won't be able to receive up-to-date data from the server/slave device. The out-of-date data will be stored in gateway's memory and will be retrieved by the Modbus client/master, which will not be aware that the server/slave device is not providing up-to-date data. The MB3660 supports the Status Monitoring function, which provides a warning mechanism to report the list of server/slave devices that are still "alive."

In agent mode, each serial port supports 256 Modbus commands. Hence, there are at most 2,048 and 4,096 Modbus commands for all serial devices in 8 ports and 16 ports, respectively. The MB3660 allocates 1 bit of the gateway's specified memory address to indicate the status of each Modbus command as being normal or abnormal. In other words, the MB3660 allocates 4,096 bits of memory to show the status of all Modbus commands. If a command has run successfully, the status value will continuously be 0. On the contrary, if a command has failed, the status will be set to 1. In this case, the Modbus TCP client/master will be aware of the failure status of the server/slave device in relation to the Modbus command.

In agent mode, when the Modbus TCP client/master queries Modbus serial server/slave devices, the MB3660 plays the role of TCP server/slave on the Ethernet side, and consequently is assigned a Modbus TCP slave ID. The Modbus TCP client/master can retrieve the Modbus server/slave devices' status via Modbus command with the following information.

Slave ID	[MB3660's Modbus TCP slave ID]
Function	0x03 or 0x04
Address	40001-
Quantity	16 registers per port (128 registers supported in MB3660-8, 256 registers supported in MB3660-16)

Here is an example of 2 Modbus RTU commands per port under Modbus RTU to Modbus TCP Agent mode:

<Port 1>

Index	Enable	Name	Slave ID	Function	Address / Quantity	Trigger	Poll Interval	Endian Swap
1	Enable	Port1_ID1	1	3	Read address 0, Quantity 1	Cyclic	1000	None
2	Enable	Port1_ID2	2	3	Read address 0, Quantity 1	Cyclic	1000	None

<Port 2>

Index	Enable	Name	Slave ID	Function	Address / Quantity	Trigger	Poll Interval	Endian Swap
1	Enable	Port2_ID1	1	3	Read address 0, Quantity 1	Cyclic	1000	None
2	Enable	Port2_ID2	2	3	Read address 0, Quantity 1	Cyclic	1000	None

.....

<Port 16>

Index	Enable	Name	Slave ID	Function	Address / Quantity	Trigger	Poll Interval	Endian Swap
1	Enable	Port16_ID1	1	3	Read address 0, Quantity 1	Cyclic	1000	None
2	Enable	Port16_ID2	2	3	Read address 0, Quantity 1	Cyclic	1000	None

The status register request command (Function Code 3/4) mapping as below:

<Port 1>				
Command No.	1 to 16	17 to 32	...	241 to 256
Starting Address	40001	40002	...	40016
<Port 2>				
Command No.	1 to 16	17 to 32	...	241 to 256
Starting Address	40017	40018	...	40032
... ... <Port 16>				
Command No.	1 to 16	17 to 32	...	241 to 256
Starting Address	40241	40242	...	40256

Please note that the No. Address is the "PLC Addresses (Base-1)".

If the commands are working properly under Modbus polling and responding mechanism, the Modbus TCP client/master would receive the status register response as 0x0000 hex (0000 0000 0000 0000 binary) from each port.

However, for some commands are not responsive; the status register will be updated to Modbus TCP client/master for notification. Take the below status register outcome, for example.

Address:	40018															
Bit:	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0
Command No.:	24	23	22	21	20	19	18	17	32	31	30	29	28	26	26	25

The commands of No. 20, 26, and 30 from Port 2 do not respond under status monitoring, which will lead you to further identify those commands set by the devices connected to Port 2.

Status monitoring is not limited to monitoring the status of Modbus serial devices. When a Modbus serial client/master queries the Modbus TCP server/slave devices, the MB3660 plays the role of serial server/slave on the serial side, and consequently is assigned a Modbus serial slave ID. The Modbus serial client/master can retrieve the Modbus TCP server/slave devices' status via Modbus command with the following information.

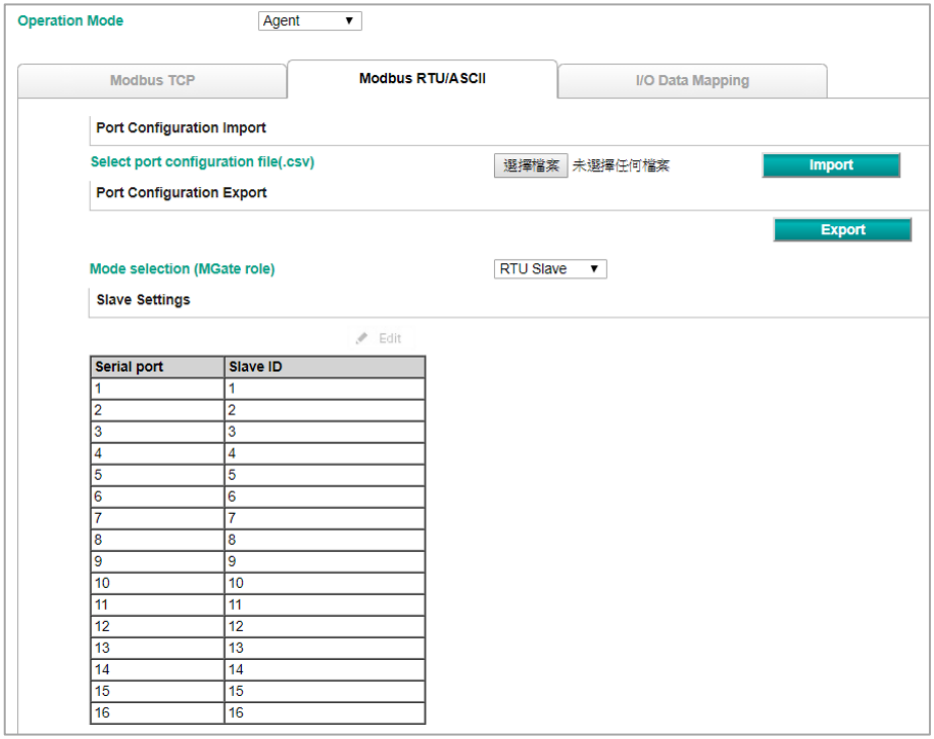
Slave ID	[MB3660's Modbus RTU/ASCII slave ID]
Function	0x03 or 0x04
Address	41001-
Quantity	16 registers

Modbus RTU/ASCII

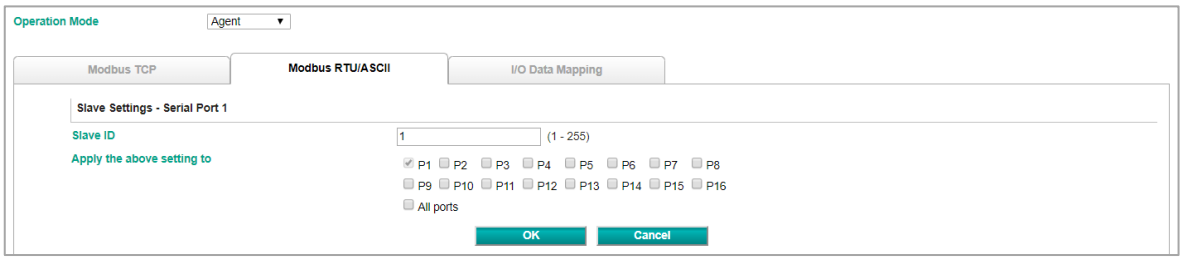
According to the Modbus RTU/ASCII settings, the MGate MB3660 will act as a Modbus client/master or Modbus server/slave to communicate with your Modbus RTU/ASCII devices. For Slave mode, MGate acts as a server/slave and waits for the incoming connection from the Modbus client/master. In this mode, you only need to specify the slave ID for the MGate gateway. For Master mode, the MGate works as a client/master and will try to send Modbus commands to the Modbus server/slave devices, so you will need to specify the slave device IDs and the relative Modbus commands.

Slave Mode Settings

You will need to specify which Modbus protocols will run in Slave (MGate role) mode. The MGate MB3660 supports Modbus RTU and Modbus ASCII protocols in Slave mode.



Double-click a serial port for additional settings or click the intended serial port and then click **Edit**.

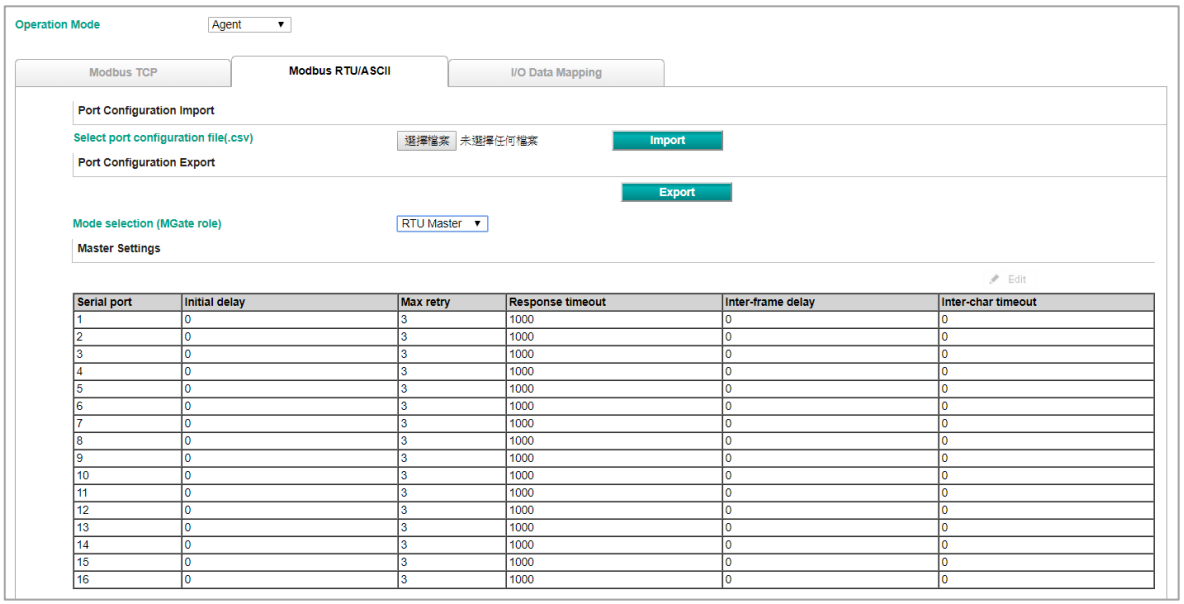


Parameters	Value	Description
Slave ID	1 to 255	The Modbus Slave ID that this server/slave module will accept. 1 to 255: Device specific.

You can change the Modbus slave ID on this page. If two or more serial ports use the same slave ID, you can check to see if they use the same setting. If two or more server/slave devices are using the same slave ID connected to different serial ports, you can click the intended serial ports for the same slave ID setting.

Master Mode Settings

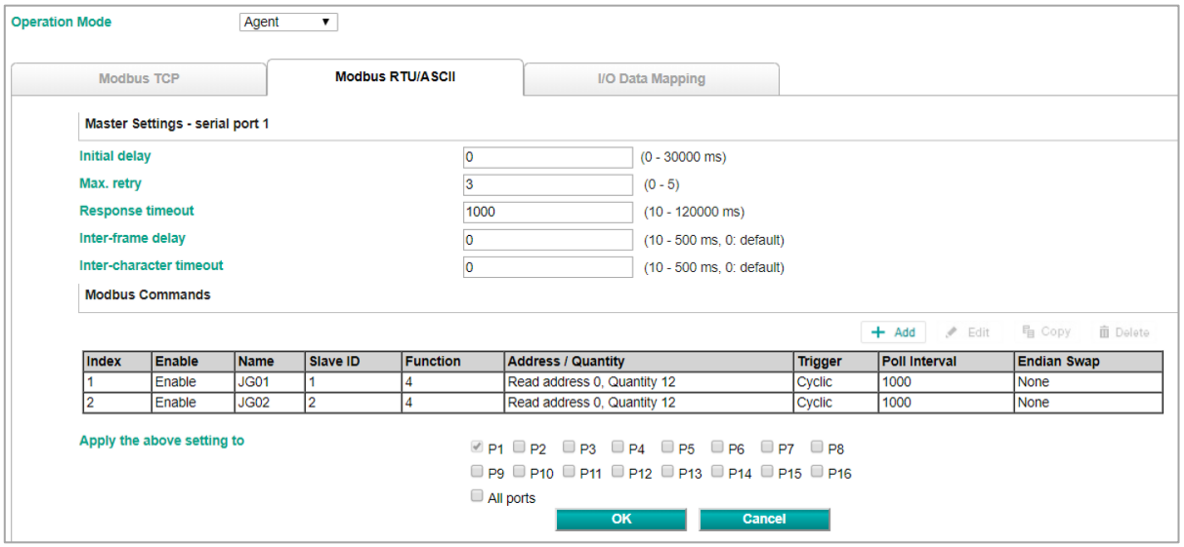
You will need to specify which Modbus protocols will run in Master mode. The MGate MB3660 supports Modbus RTU and Modbus ASCII protocols in Master mode.



Serial port	Initial delay	Max retry	Response timeout	Inter-frame delay	Inter-char timeout
1	0	3	1000	0	0
2	0	3	1000	0	0
3	0	3	1000	0	0
4	0	3	1000	0	0
5	0	3	1000	0	0
6	0	3	1000	0	0
7	0	3	1000	0	0
8	0	3	1000	0	0
9	0	3	1000	0	0
10	0	3	1000	0	0
11	0	3	1000	0	0
12	0	3	1000	0	0
13	0	3	1000	0	0
14	0	3	1000	0	0
15	0	3	1000	0	0
16	0	3	1000	0	0

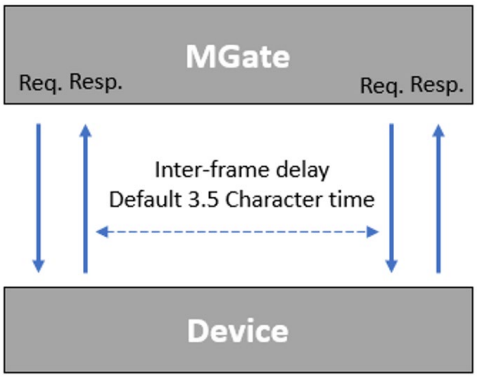
The MGate MB3660 also provides several advanced settings for specific application requirements. The following settings are optional for most applications. It is suggested to use the default settings to test the MGate MB3660.

Double-click the intended serial port to configure additional settings.



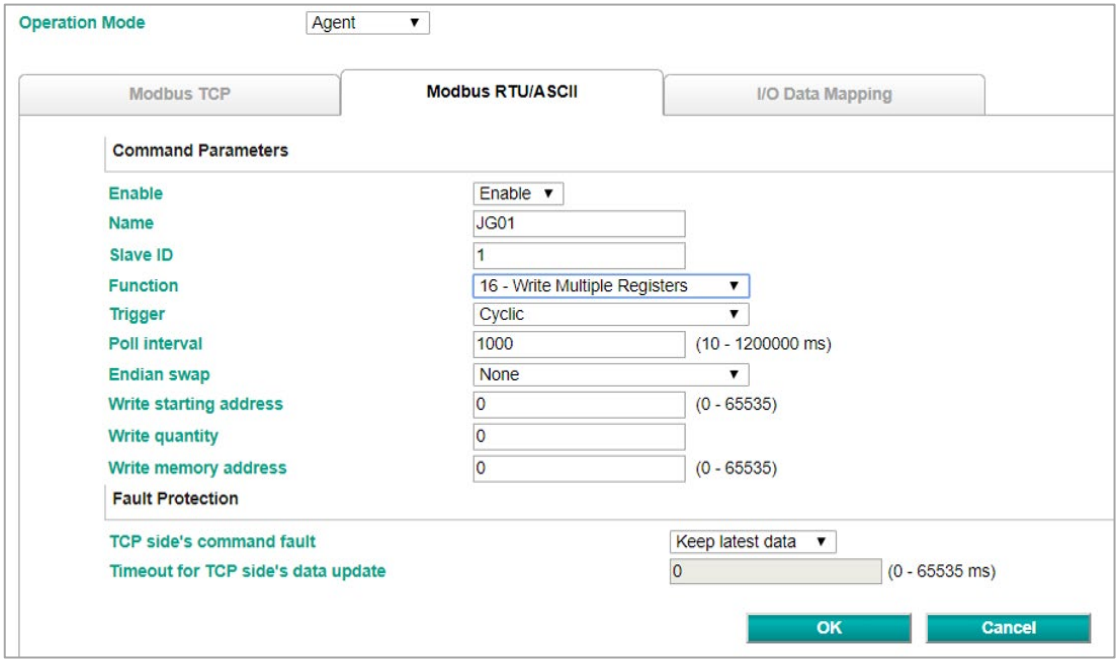
Index	Enable	Name	Slave ID	Function	Address / Quantity	Trigger	Poll Interval	Endian Swap
1	Enable	JG01	1	4	Read address 0, Quantity 12	Cyclic	1000	None
2	Enable	JG02	2	4	Read address 0, Quantity 12	Cyclic	1000	None

Parameters	Description
Initial delay	Some Modbus servers/slaves may take more time to boot up than other devices. In some environments, this may cause the entire system to suffer from repeated exceptions during the initial boot-up. You can force the MGate to wait after booting up before sending the first request with the **Initial Delay** setting.
Max. retry	The number of times the client/master will retry the same request when the response times out.

Parameters	Description
Response timeout	According to the Modbus standard, the time it takes for a server/slave device to respond to a request is defined by the device manufacturer. Based on this response time, a client/master can be configured to wait a certain amount of time for a server/slave's response. If no response is received within the specified time, the client/master will disregard the request and continue operation. This allows the Modbus system to continue operation, even if a server/slave device is disconnected or faulty. On the MGate 5101-MB-EIP, the **Response timeout** field is used to configure how long the gateway will wait for a response from a Modbus ASCII or RTU server/slave. Refer to your device manufacturer's documentation to manually set the response time.
Inter-frame delay	In Modbus RTU slave mode, use this function to define the time interval between a Modbus RTU response and the next Modbus RTU request. The reason for enabling manual configuration of this value is to accommodate certain scenarios where legacy Modbus devices may not be capable of promptly processing Modbus RTU requests. By setting a longer interval value, you can increase tolerance for delays in serial reception and transmission. The inter-frame delay can be adjusted within the range of 10 to 500 ms or use the default value. The default value in this field is 0 ms, but the default inter-frame delay varies depending on the baudrate setting. When the baudrate is configured below 19200 bps, the default inter-frame delay is set to 3.5 character times. When the baudrate is configured equal to or larger than 19200 bps, a predefined fixed value is used. How to calculate Modbus character time? For example, if the baudrate is 9,600 bps, 1 character time is about 1 ms. In a serial frame (11 bits, including start bit, data, parity bit and stop bit), 9,600 bps approximately equals to 960 characters/s, so transmitting 1 character needs about 1/960 = 1 ms. 
Inter-character timeout	In Modbus RTU slave mode, use this function to define the time-out interval between characters in one frame. When the serial side of MGate receives one character, and the next one comes after the "inter-character timeout" defined, the frame will be discarded because of time-out. The inter-character timeout can be adjusted within the range of 10 to 500 ms or use the default value. The default value in this field is 0 ms, but the default inter-character timeout varies depending on the baudrate setting. When the baudrate is configured below 19200 bps, the default inter-character time-out is set to 1.5 character times. When the baudrate is configured equal to or larger than 19200 bps, a predefined fixed value is used.

For Master mode, you must identify which Modbus requests need to be sent to the Modbus server/slave devices through serial interface. The data will be exchanged between server/slave devices and the MGate gateway's internal memory. To do this, manually add all Modbus commands that will handle the data exchange.

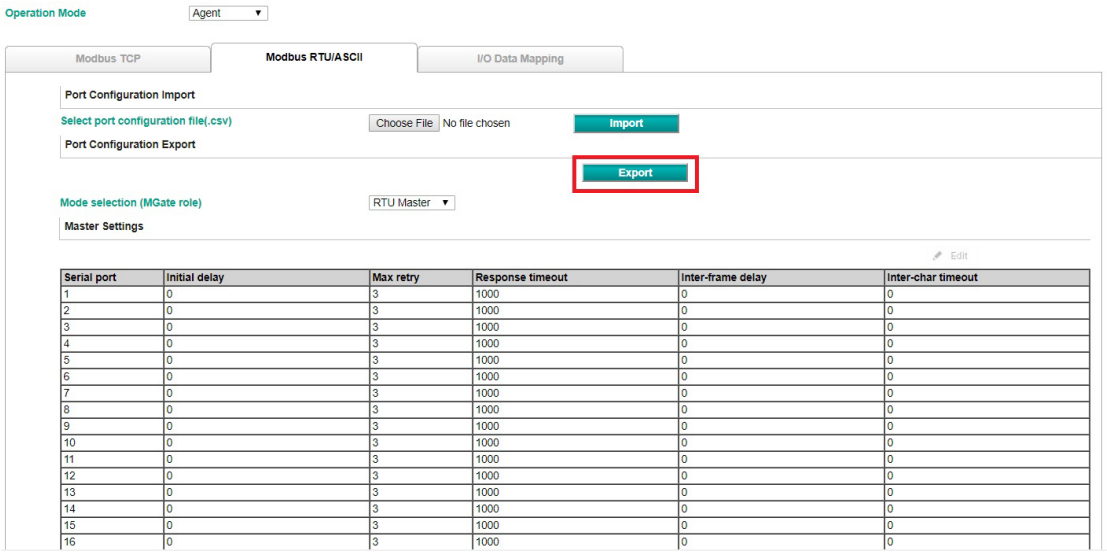
The **Add**, **Edit**, **Copy**, and **Delete** buttons support the Modbus command arrangement. When you click on the **Add** and **Edit** buttons, the following dialog box will be displayed.



The description of all the fields can refer to Agent mode---**Master Mode Settings**.

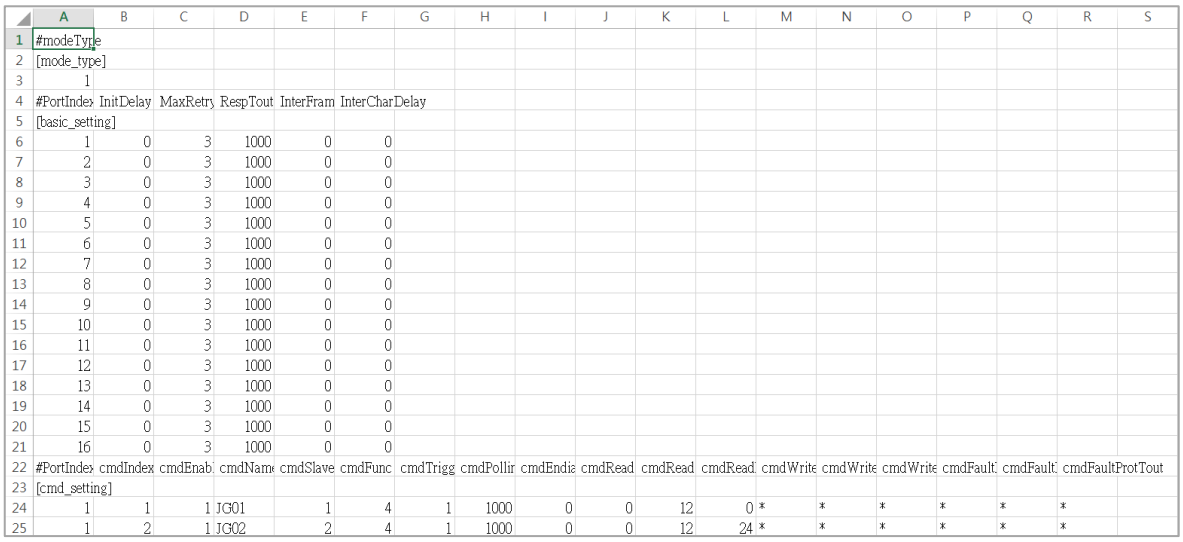
Port Configuration

A Modus RTU/ASCII client/master may set several commands that are sent to server/slave devices. Some users are familiar with CSV file format to edit these commands. Therefore, the MGate MB3660 supports the import or export functions of CSV files. First, click **Export** to generate the template file. Then open the exported CSV file to further configure.



Serial port	Initial delay	Max retry	Response timeout	Inter-frame delay	Inter-char timeout
1	0	3	1000	0	0
2	0	3	1000	0	0
3	0	3	1000	0	0
4	0	3	1000	0	0
5	0	3	1000	0	0
6	0	3	1000	0	0
7	0	3	1000	0	0
8	0	3	1000	0	0
9	0	3	1000	0	0
10	0	3	1000	0	0
11	0	3	1000	0	0
12	0	3	1000	0	0
13	0	3	1000	0	0
14	0	3	1000	0	0
15	0	3	1000	0	0
16	0	3	1000	0	0

Then open the exported CSV file to further configure.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	#modeType																		
2	[mode_type]																		
3	1																		
4	#PortIndex	InitDelay	MaxRetry	RespTout	InterFram	InterCharDelay													
5	[basic_setting]																		
6	1	0	3	1000	0	0													
7	2	0	3	1000	0	0													
8	3	0	3	1000	0	0													
9	4	0	3	1000	0	0													
10	5	0	3	1000	0	0													
11	6	0	3	1000	0	0													
12	7	0	3	1000	0	0													
13	8	0	3	1000	0	0													
14	9	0	3	1000	0	0													
15	10	0	3	1000	0	0													
16	11	0	3	1000	0	0													
17	12	0	3	1000	0	0													
18	13	0	3	1000	0	0													
19	14	0	3	1000	0	0													
20	15	0	3	1000	0	0													
21	16	0	3	1000	0	0													
22	#PortIndex	cmdIndex	cmdEnab	cmdNam	cmdSlave	cmdFunc	cmdTrigg	cmdPollir	cmdEndia	cmdRead	cmdRead	cmdRead	cmdWrite	cmdWrite	cmdWrite	cmdFault	cmdFault	cmdFaultProtTout	
23	[cmd_setting]																		
24	1	1	1	JG01	1	4	1	1000	0	0	12	0	*	*	*	*	*	*	
25	1	2	1	JG02	2	4	1	1000	0	0	12	24	*	*	*	*	*	*	

Follow the format or an error will occur when the file is imported. The detailed description of the format is shown below.

Item		Value	Note
mode_type		0, 1, 2, 3	0: RTU Slave 1: RTU Master 2: ASCII Slave 3: ASCII Master
basic_setting (RTU/ASCII Slave mode only)	PortIndex	1 to 8 (the MGate MB3660-8 Series) 1 to 16 1 to 8 (the MGate MB3660-16 Series)	Serial port number
	SlaveID (Slave ID)	1 to 255	
basic_setting (Master mode only)	PortIndex	1 to 8	Serial port number
	InitDelay (Initial delay)	0 to 30000 ms	
	MaxRetry (Max. retry)	0 to 5	
	RespTout (Response timeout)	10 to 120000 ms	
	InterFrameDelay (Inter-frame delay)	10 to 500 ms, Default: 0 ms (baudrate < 19200 bps: 3.5 character times ; baudrate >= 19200 bps: fixed value)	
	InterCharDelay (Inter-character timeout)	10 to 500 ms, Default: 0 ms (baudrate < 19200 bps: 1.5 character times ; baudrate >= 19200 bps: fixed value)	
cmd_setting (Master mode)	PortIndex	1 to 8 (the MGate MB3660-8 Series) 1 to 16 1 to 8 (the MGate MB3660-16 Series)	Serial port number
	cmdIndex (Command index)	1 to 32	Max. up to 32 commands per port
	cmdEnable (Enable)	0, 1	0: disable 1: enable
	cmdName (Name)	Name	Max. up to 40 characters
	cmdSlaveId (Slave ID)	1 to 255	

Item		Value	Note
	cmdFunc (Function)	1 to 6,15,16,23	
	cmdTrigger (Trigger)	1, 2	1:Cyclic 2:Data Change
	cmdPollinterval (Poll interval)	10 to 1200000 ms	
	cmdEndianSwap (Endian swap)	0, 1, ,2, 3	0:None 1:Byte 2:Word 3:Byte and Word
	cmdReadStartAddr (Read starting address)	0 to 65535	
	cmdReadQuan (Read quantity)	1 to 123	
	cmdReadMemAddr (Read memory address)	0 to 65535	
	cmdWriteStartAddr (Write starting address)	0 to 65535	
	cmdWriteQuan (Write quantity)	1 to 123	
	cmdFaultProtType (Opposite side's command fault)	0, 1, 2	0:keep latest data 1:clear data to zero 2:user-defined value
	cmdFaultProtValue (Fault protection value)	00 to FF	
	cmdFaultProtTout	0 to 65535 ms	Timeout for opposite side's
	(Timeout for opposite side's data update)		data update

NOTE

In [basic_setting], the value of "portIndex" must be bigger than the previous row.
In [cmd_setting], the value of "portIndex" must be equal or bigger than the previous row.
In [cmd_setting], the value of "cmdIndex" must be bigger than the previous row.

Content that appears after the "#" character will be ignored. It is used to write notes on the CSV file.

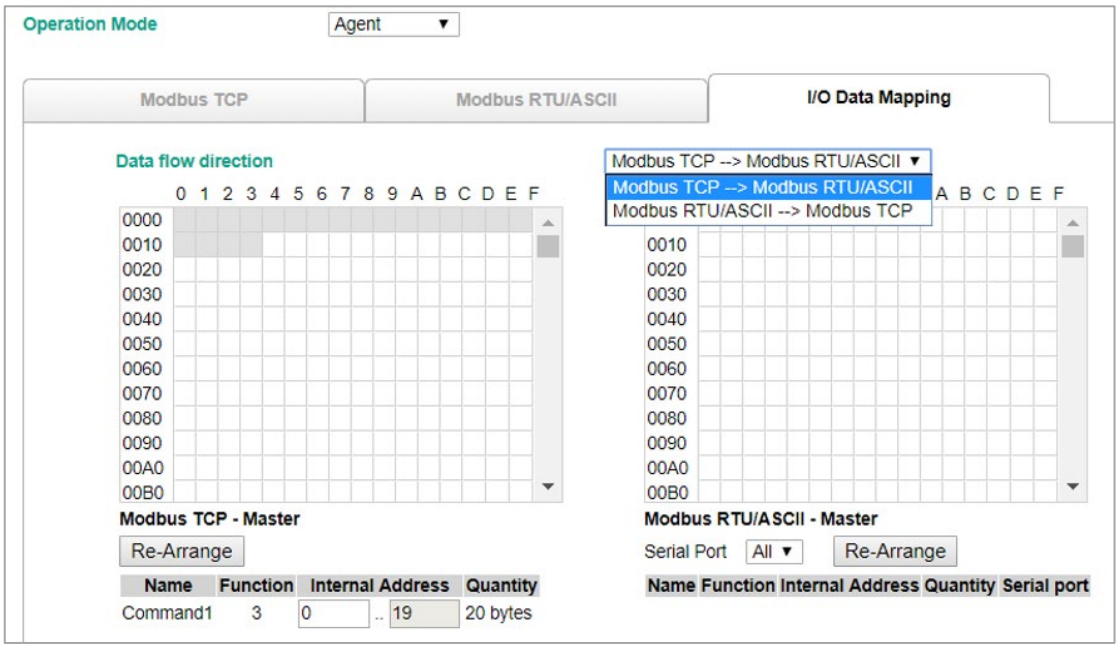
Error Message

If you import an invalid format of a configuration file, a notification message will pop up to show which columns and rows are incorrect on the web console. Two types of errors should be avoided.

Format Error	invalid character, absent/additional columns/rows below data block
Data Range Error	value is out of range (Ref Port Configuration Format)

I/O Data Mapping

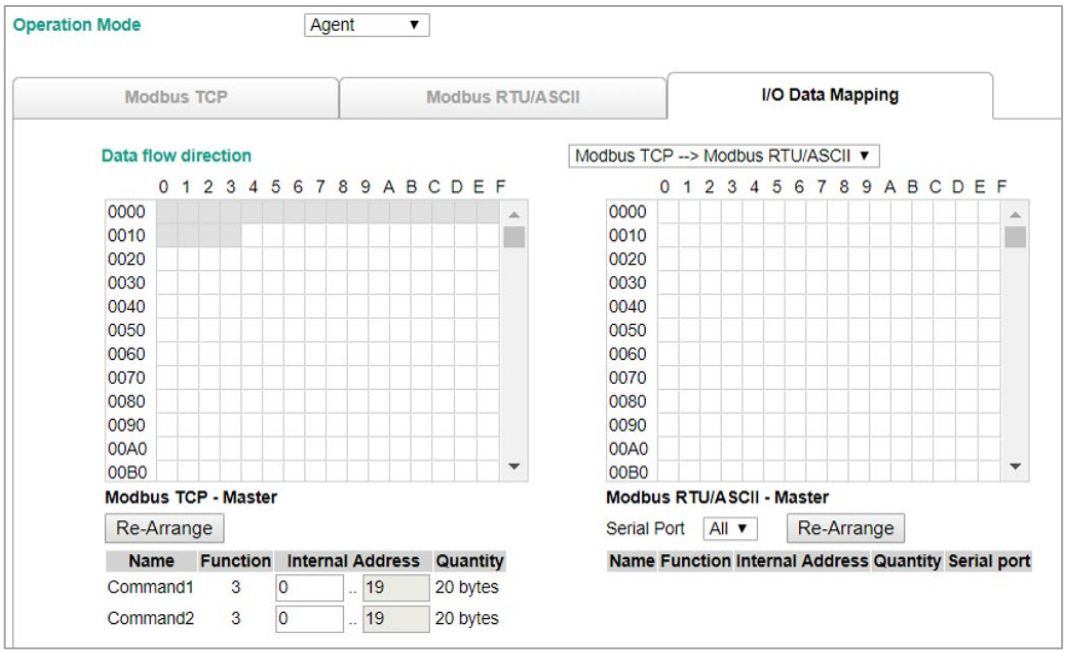
You can verify the gateway's memory allocation on the **I/O Data Mapping** page. First, select the Modbus data flow you want to see.



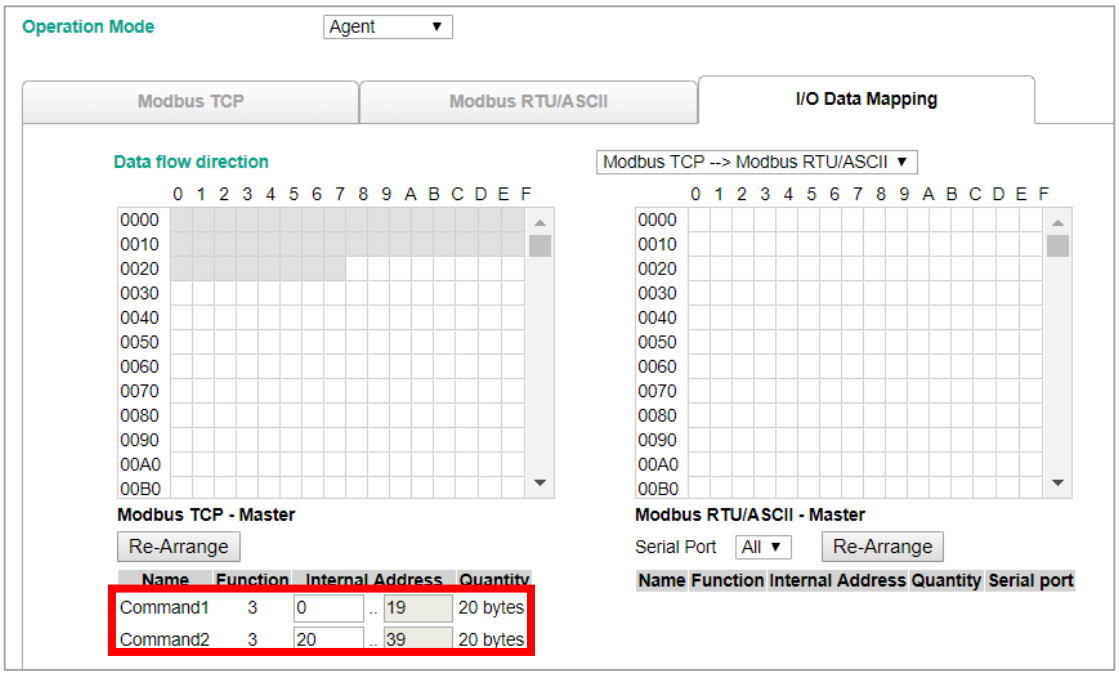
In agent mode, you need to manually set Modbus commands one-by-one and assign a gateway memory address for storing this data. We recommend using **I/O Data Mapping** to check the memory address of each command. The commands will be shown under the Modbus Master (MGate role) mode. While using I/O Data mapping function, the MGate MB3660 Series supports both manual and automatic ways for mapping to an internal address.

Automatic Arrangement

If there are two commands with the same internal address, as shown in the figure below.

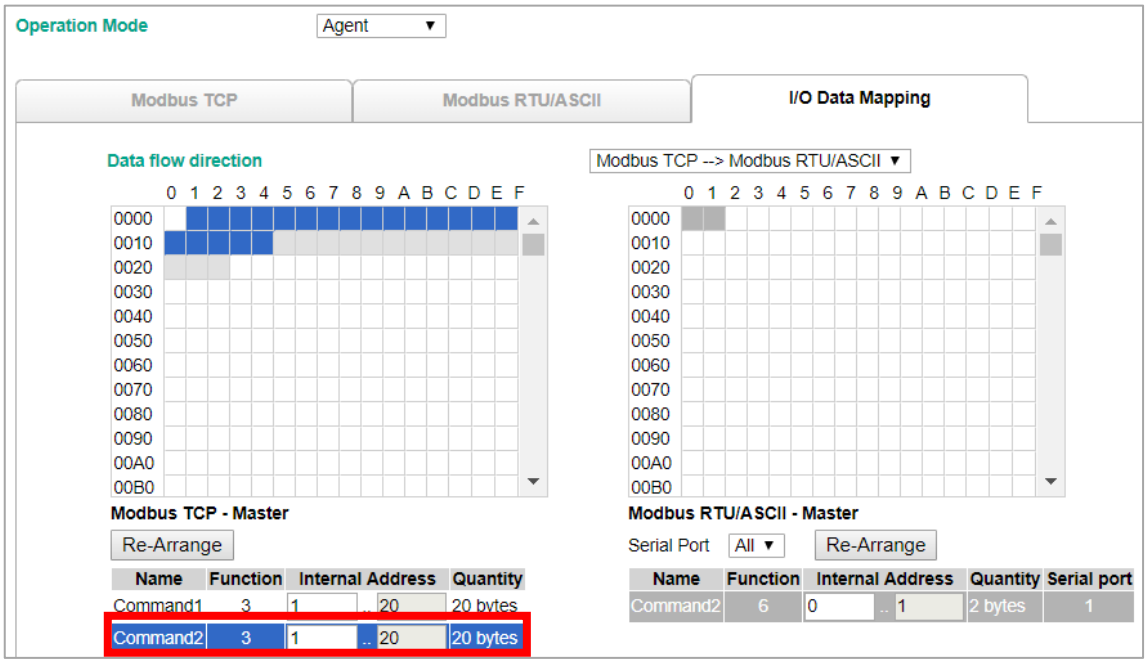


You can click on the **Re-Arrange** button to automatically address the internal address. The updated internal address will become as follows:

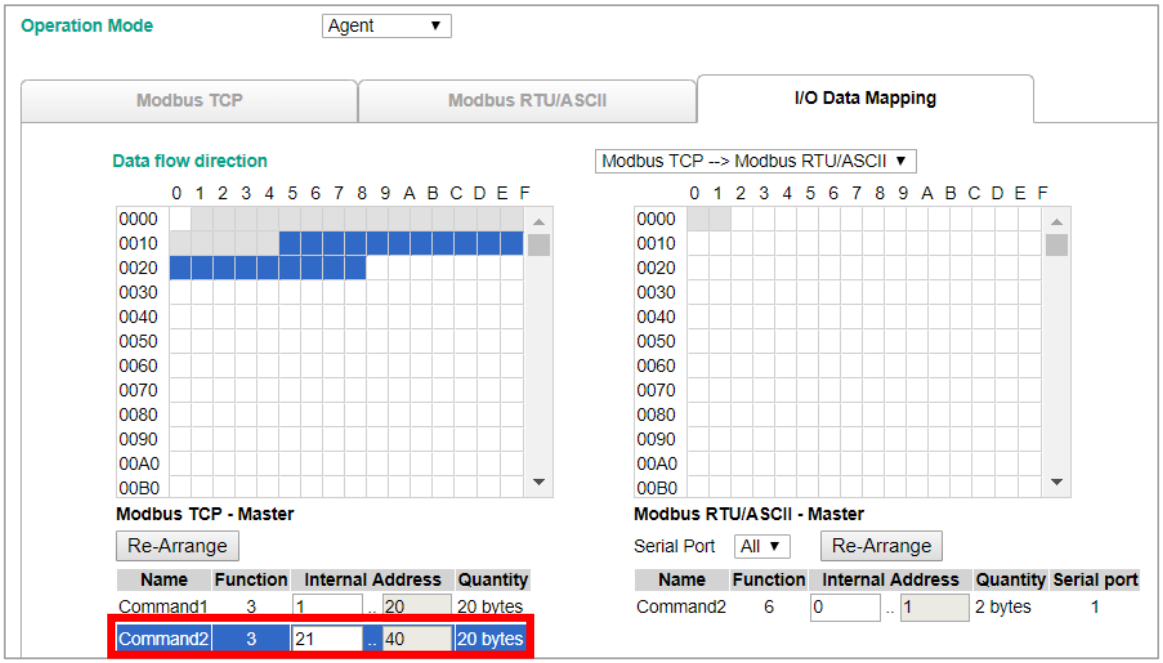


Manual Arrangement

On the contrary, you can also set the internal address manually. For example, if you add two Modbus commands that Command1 uses addresses 1 to 20, whereas Command2 uses addresses 1 to 20, then obviously, a memory overlap exists.



To rectify the error, click **Command2** to change its starting address from 21 to 40. The internal address allocation blocks are illustrated above to assist your manual setting.



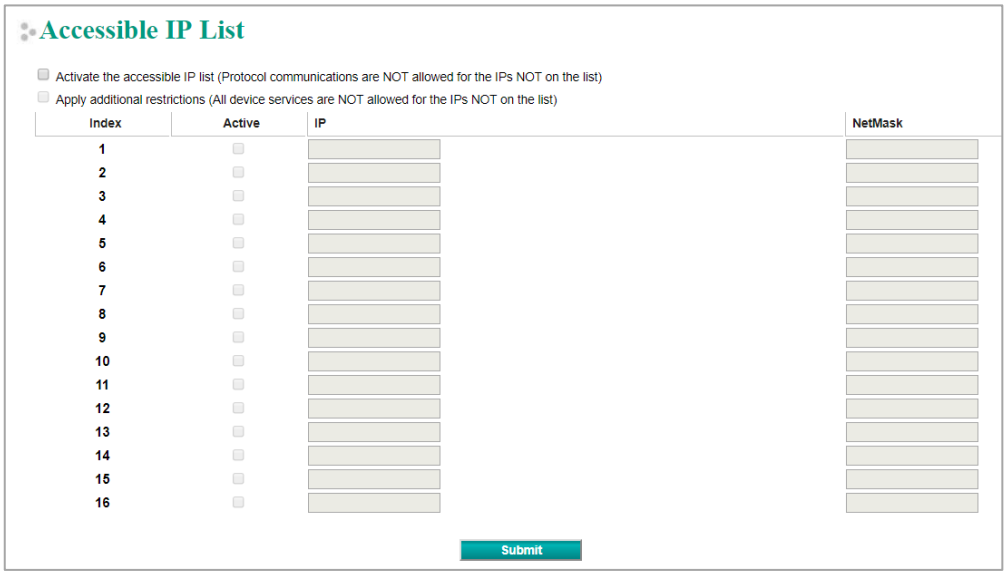
Once the change has been made, each of the two commands will be allocated to unique address ranges in the gateway's memory. That is, the address for command3 will be in the range 1 to 20, whereas command2 will be in the range 21 to 40.

System Management

This configuration tab includes several system level settings. Most of these settings are optional.

Accessible IP List

The Accessible IP List function allows you to add or block remote host IP addresses to prevent unauthorized access. Access to the MGate MB3660 is controlled by the IP address. That is, if a host's IP address is in the accessible IP table, then the host will be allowed to access the MGate MB3660. The different restrictions are listed in the table below (the checkbox **Apply additional restrictions** only can be activated if **Activate the accessible IP list** is activated).



Activate the accessible IP list	Apply additional restrictions	IPs on the list (Active checked)	IPs NOT on the list (Active NOT checked)
✓	–	All protocol communication and services* are allowed.	Protocol communication is not allowed, but services* are still allowed.
✓	✓	All protocol communication and services* are allowed.	All services* are not allowed.

*Services indicate HTTP, HTTPS, TELNET, SNMP, SMTP, DNS, NTP, DSU

These settings are used to restrict access to the module by the IP address. Only IP addresses on the list will be allowed access to the device. You may add a specific address or range of addresses by using a combination of IP address and netmask, as follows:

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").

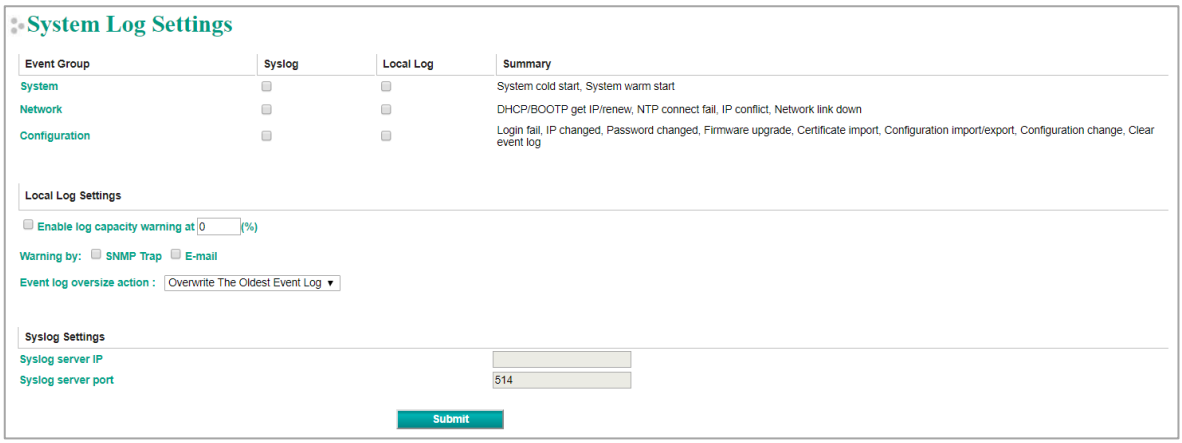
To allow access to all IP addresses: Make sure that **Enable** the accessible IP list is not checked.

Additional configuration examples are shown in the following table:

Desired IP Range	IP Address Field	Netmask Field
Any host	Disable	Enable
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

System Log Settings

These settings enable the MGate firmware to record important events for future verification. The recorded information can only be shown on the page of "**System Log**".



The available information that can be recorded includes the following events:

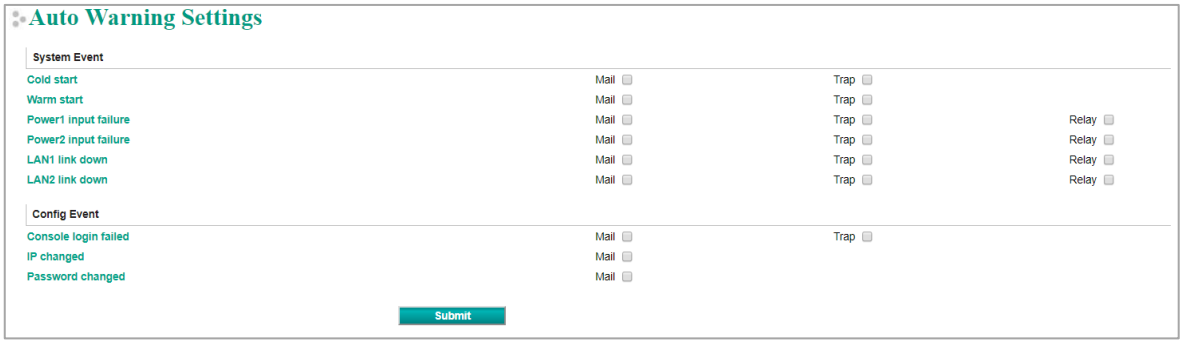
Event Group	Description
System	System Cold Start, System Warm Start
Network	DHCP/BOOTP Get IP/Renew, NTP Connect Fail, IP Conflict, Network Link Down
Configuration	Login Fail, IP Changed, Password Changed, Firmware Upgrade, SSL Certificate Import, Configuration Import/Export , Clear Event Log
Local Log Settings	Description
Enable Log Capacity Warning (%)	When the log amount exceeds the warning percentage, it will trigger an event to SNMP Trap or Email
Warning by	SNMP Trap Email
Event log oversize action	Overwrites the oldest event log Stops recording event log

Syslog Settings	Description
Syslog server IP	IP address of the server that will record the log data
Syslog server Port	514

Users can view the recorded information from the web console or the text mode console.

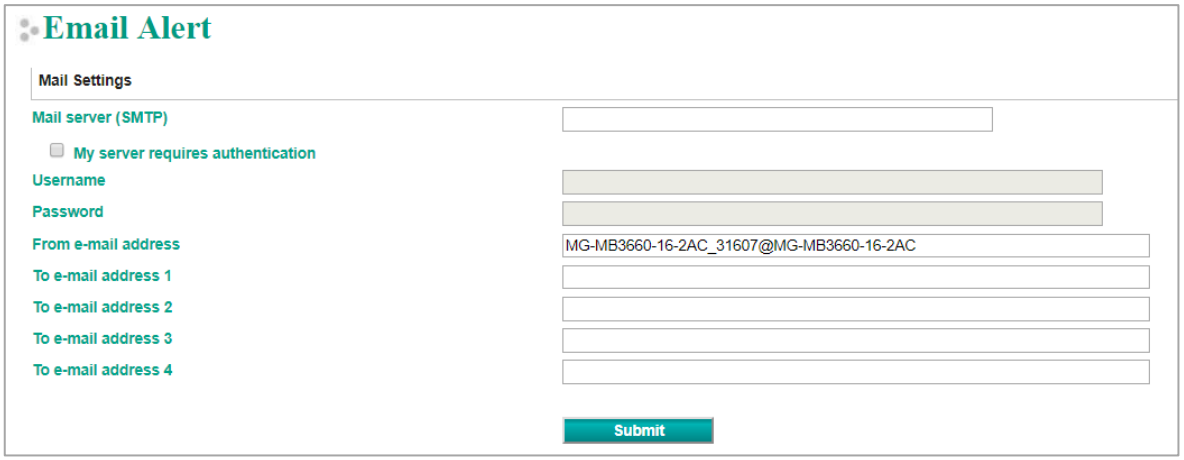
Auto Warning Settings

Auto Warning is triggered by different events. When a checked trigger condition occurs, the MGate can send e-mail alerts, SNMP Trap messages, or open/close the circuit of the relay output and trigger the Fault LED to start blinking. To enable an e-mail alert, configure the e-mail address on the **E-mail Alert** page. Likewise, to enable SNMP Trap alerts, configure the SNMP trap server on the SNMP Trap page.



Email Alert

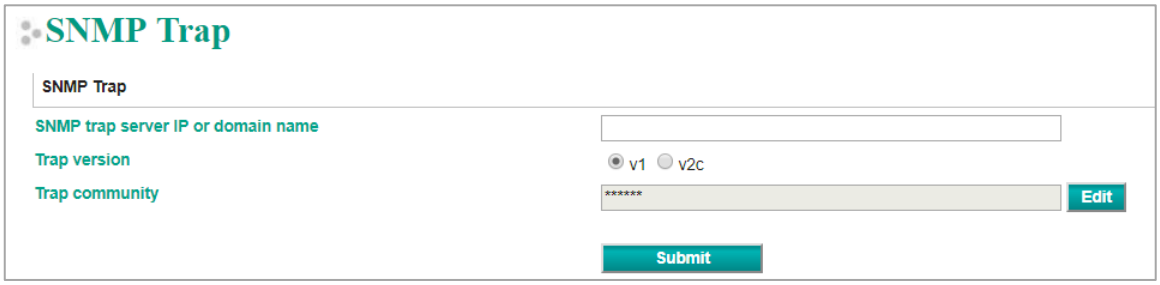
Along with activating the Mail function from events on **Auto Warning Settings** page, the E-mail alert should be set up then.



Parameters	Description
Mail server	The mail server's domain name or IP address.
Username	This field is for your mail server's username, if required.
Password	This field is for your mail server's password, if required.
From e-mail address	This is the e-mail address from which automatic e-mail warnings will be sent.
To e-mail address 1 to 4	This is the e-mail address or addresses

SNMP Trap

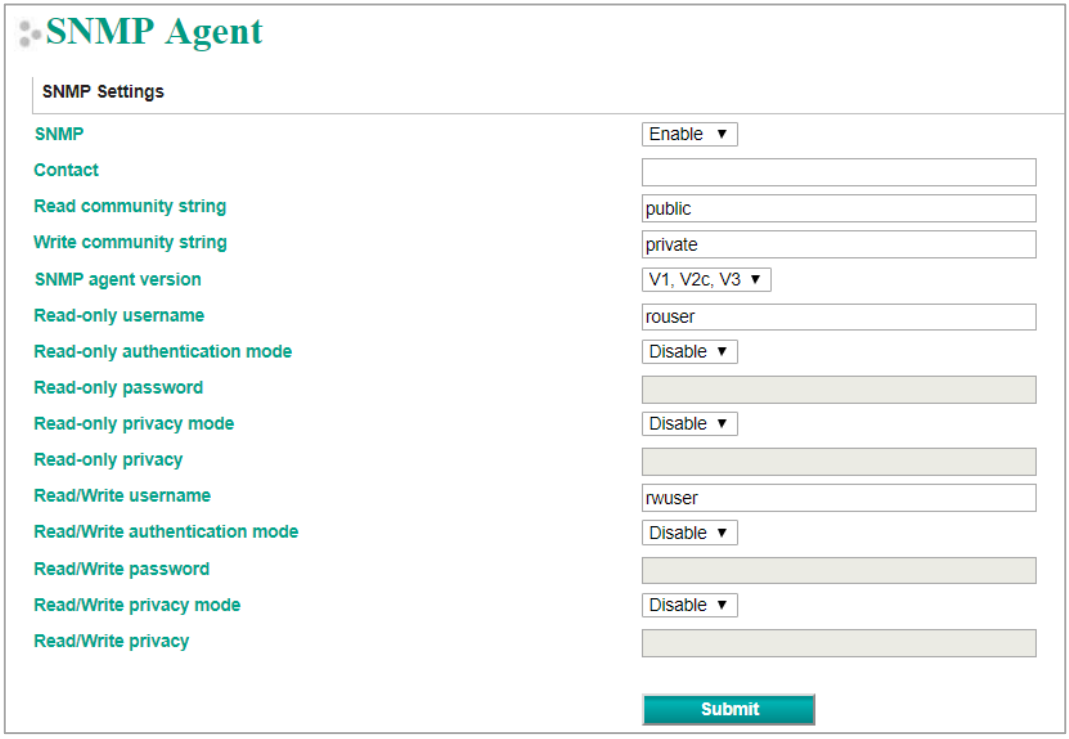
Along with activating "Trap" function from events on **Auto Warning Settings** page, the SNMP Trap should be set up then.



Parameters	Description
SNMP trap server IP	Use this field to show the IP address to use for receiving SNMP traps.
Trap version	Use this field to select the SNMP trap version.
Trap community	Use this field to designate the SNMP trap community.

SNMP Agent

The SNMP Agent is a network-management for collecting and organizing information about managed devices on IP network and for modifying the information on the device.



Parameters	Description
SNMP	To enable the SNMP Agent function, select the **Enable** option, and enter a community name (e.g., public).
Contact name	The optional SNMP contact information usually includes an emergency contact name and telephone number.
Read community string	This is a text password mechanism that is used to weakly authenticate queries to agents of managed network devices.
Write community string	This is a text password mechanism that is used to weakly authenticate changes to agents of managed network devices.
SNMP agent version	The MGate MB3660 Series supports SNMP V1, V2c, and V3.

Read-only and Read/write access control

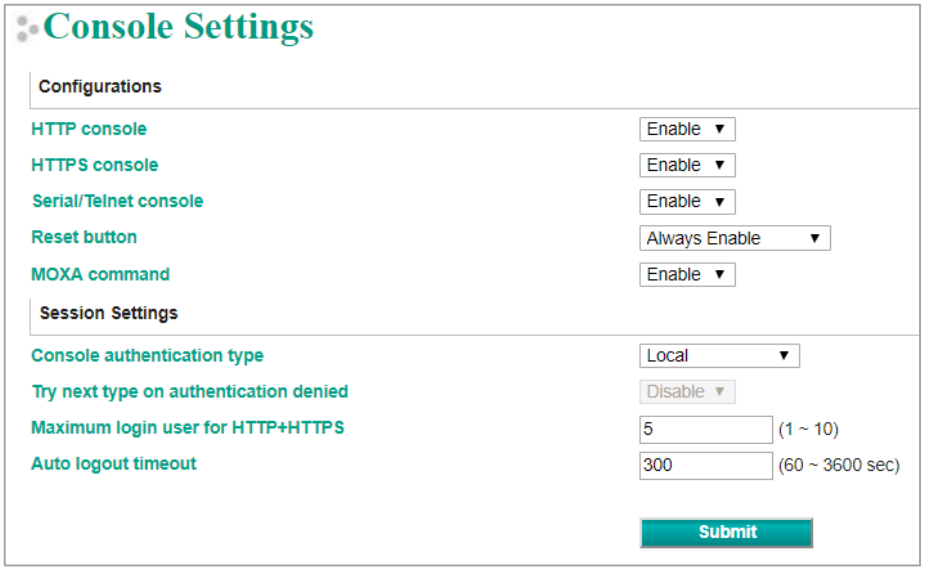
While selecting SNMP agent V3, it is needed to configure read-only and read/ write access control parameters. The following fields allow you to define usernames, passwords, and authentication parameters for two levels of access: read-only and read/write. The name of the field will indicate which level of access it refers to. For example, **Read-only** authentication mode allows you to configure the authentication mode for read-only access, whereas **Read/write** authentication mode allows you to configure the authentication mode for read/write access. For each level of access, you may configure the following:

Parameters	Description
Username	Use this optional field to identify the username for the specified level of access.
Authentication mode	Use this field to select MD5 or SHA as the method of password encryption for the specified level of access, or to disable authentication.
Privacy mode	Use this field to enable or disable DES_CBC data encryption for the specified level of access.
Password	Use this field to set the password for the specified level of access.
Privacy	Use this field to define the encryption key for the specified level of access.

Misc. Settings

Console Settings

To support various security levels, the console and session can be further configured.

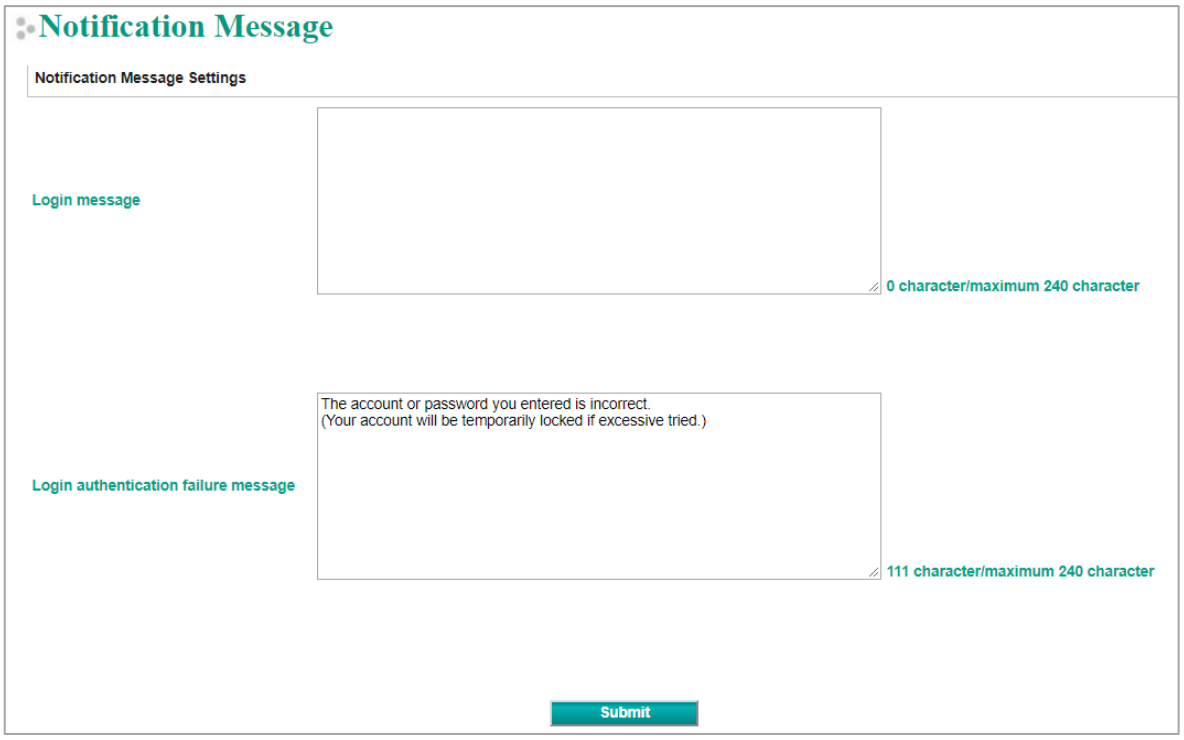


Configuration	Value	Description
HTTP/HTTPS	Enable/Disable	This setting is to enable/disable the web console. For security issues, users can only enable the HTTPS or just disable all settings.
Serial/Telnet	Enable/Disable	The MGate Telnet function can be enabled or disabled.
Reset button protect	Disable after 60 sec, Always enable	The MGate provides the reset button to clear password or load factory default settings. But for security issues, users can disable this function. In disabled mode, the MGate will still enable this function within 60 seconds after boot-up just in case users really need to reset this function.
Moxa command	Enable/Disable	The MGate can be searched by the Device Search Utility (DSU). If you have any security concerns, you can choose Disable to deny the DSU right to access.

Session Settings	Value	Description
Console Authentication type	Local/ Radius/ Radius-Local/ Local-Radius	The MGate MB3660 offers not only local authentication network but also Remote Authentication Dial-In User Service (RADIUS) network service.
Try next types on authentication denied	Enable/Disable	While selecting combination types of console authentication, users may further decide to enable or disable to try the second type of authentication method under the first one being denied
Maximum Login User for HTTP + HTTPS	1 to 10	The number of users that can access the MGate at the same time.
Auto Logout Setting	60 to 3600 sec	Sets the auto logout time

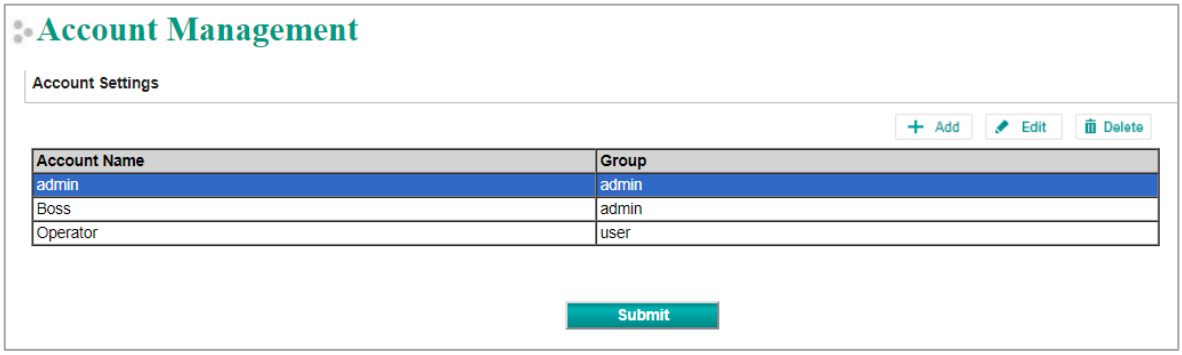
Notification Message

The login message and login authentication failure message can be configured.



Account Management

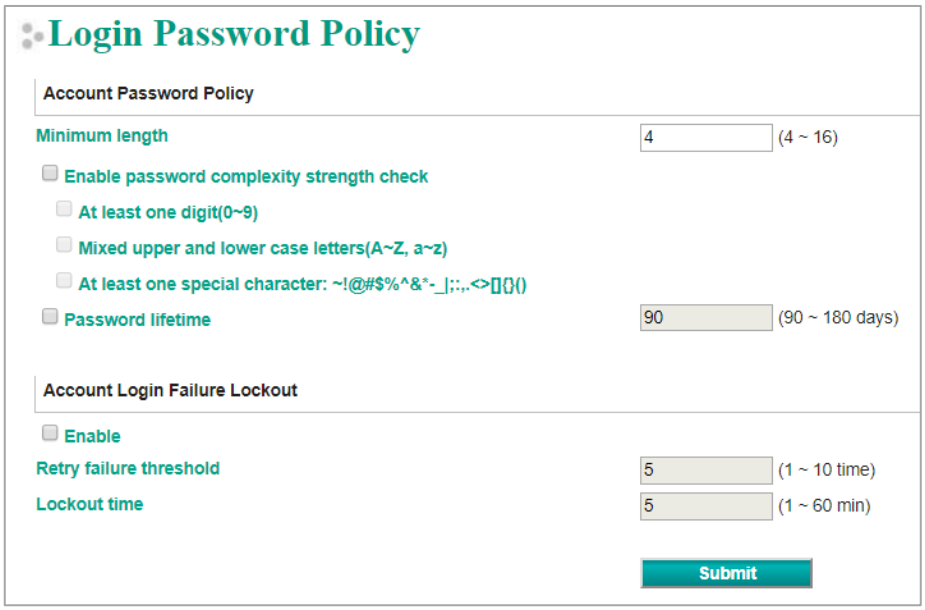
A group of accessible accounts with two different user levels can be managed.



Parameters	Value	Description
Account Name		Users can set up the account name for login purpose.
Group	Admin, user	Users can change the password for different accounts. The MGate provides two different user levels: admin and user with a maximum of 16 accounts. The admin account can access and change all the settings through the web console. The user account can only view the settings and cannot change anything.

Login Password Policy

Considering security level, the login password policy and failure lockout can be configured.

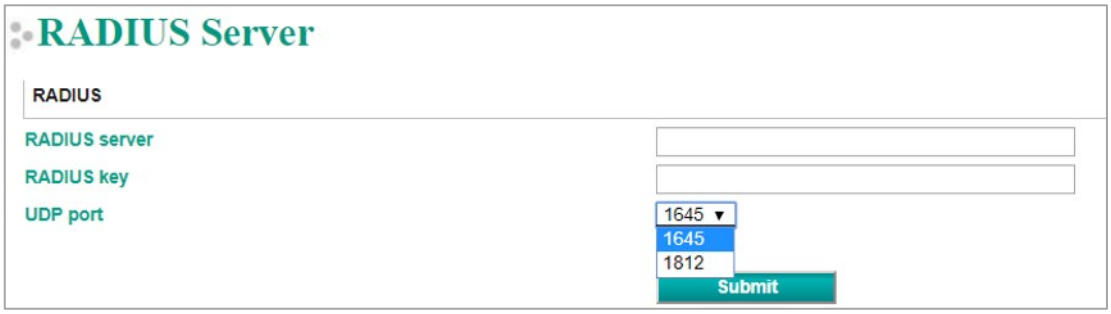


Account Password Policy	Value	Description
Minimum length	4 to16	
Enable password complexity strength check		Select how the MGate checks the password strength
Password lifetime	90 to 180 days	Set the password lifetime period

Account Login Failure Logout	Value	Description
Retry failure threshold	1 to 10 time(s)	
Lockout time	1 to 60 min	

RADIUS Server

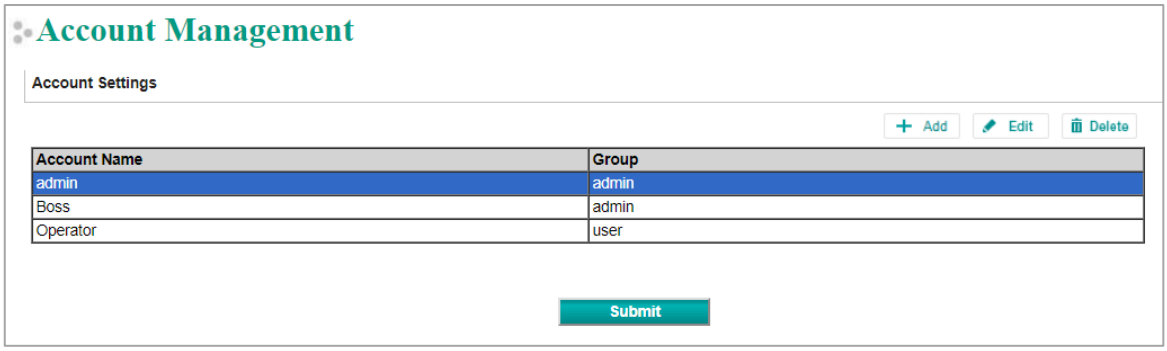
While selecting RADIUS as for console authentication, it's needed to configure the RAIDUS server in this session.



RADIUS Server	Value	Description
RADIUS Server		The RADIUS server's domain name or IP address.
RADIUS Key		The RADIUS key parameter is used to encrypt RADIUS packets before they are sent over the network.
UDP Port	1646/1812	The UDP port for the RADIUS server's domain name or IP address.

NOTE

If you are using a RADIUS server for user authentication, make sure the ID string on the RADIUS server matches the Group Name on the Account Management page. Also, the Service-Type must be set as "Login". To grant users access to the admin group, the filter ID of the RADIUS server should be set as "admin" and the Service-Type as "Login".



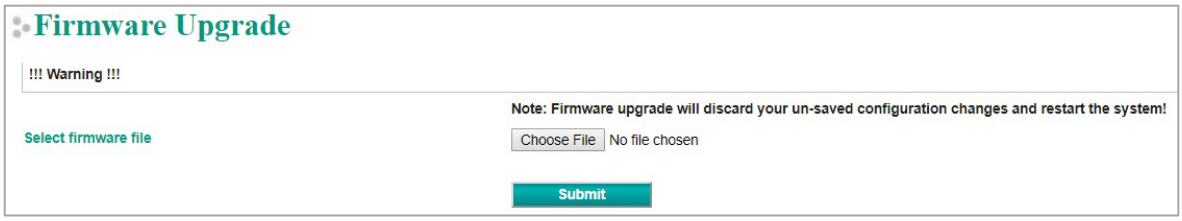
Maintenance

Ping

The MGate gateway will send an ICMP packet through the network to a specified host, and the result can be viewed on the web console immediately.

Firmware Upgrade

The updated MGate MB3000 Series firmware is at http://www.moxa.com. After you have downloaded the new firmware onto your PC, you can use MGate Manager to write it onto your MGate MB3000 gateway. Select the desired unit from the list in MGate Manager and click **Upgrade Firmware** to begin the process.



ATTENTION

DO NOT turn off the MGate power before the firmware upgrade process is completed. The MGate will erase the old firmware to make room for the new firmware to flash memory. If you power off the MGate and terminate the progress, the flash memory will contain corrupted firmware and the MGate will fail to boot. If this happens, call Moxa RMA services.

While executing firmware upgrade, it is recommended to back up the configuration file by using **Export** function. Once the latest firmware is successfully written onto the unit, it is recommended to reset the device to default by using **Load Default** function.

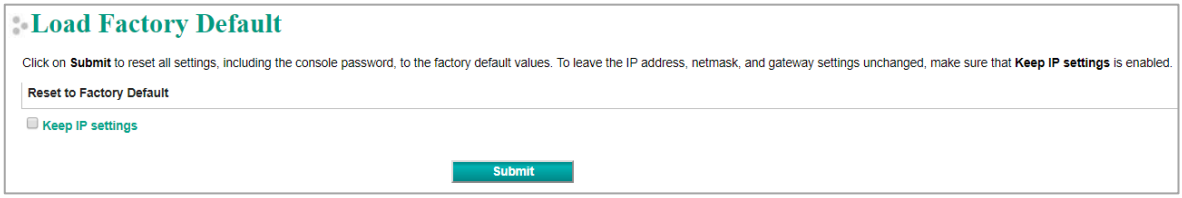
Configuration Import/Export

There are three main reasons for using the Import and Export functions:

- **Applying the same configuration to multiple units.**
  The Import/Export configuration function is a convenient way to apply the same settings to units in different sites. You can export the configuration as a file and then import the configuration file onto other units at any time.
- **Backing up configurations for system recovery.**
  The export function allows you to export configuration files that can be imported onto other gateways to restore malfunctioning systems within minutes.
- **Troubleshooting.**
  Exported configuration files can help administrators to identify system problems that provide useful information for Moxa's Technical Service Team when maintenance visits are requested.

Load Factory Default

To clear all the settings on the unit, use the Load Factory Default to reset the unit to its initial factory default values.



ATTENTION

Load Default will completely reset the configuration of the unit, and all the parameters you have saved will be discarded. Do not use this function unless you are sure you want to completely reset your unit.

Certificate

For the MGate self-signed certificate:

When we encounter the valid date of the certificate expired, we can regenerate the "MGate self-signed" certificate through the following steps.

Step 1: Users should delete the SSL certificate file originating from the MGate device.

Step 2: Then, enable the NTP server by setting up the time zone and local time.

Step 3: After restarting the device, the "MGate self-signed" certificate will be regenerated with the updated valid time.

For importing the third-party trusted SSL certificate:

By importing the third-party trusted SSL certificate, the security level can be enhanced. A snapshot of the GUI for the web console is shown below. To generate the SSL certificate through the third party, here are the steps:

Step 1: Create a certification authority (Root CA), such as Microsoft AD Certificate Service (https://mizitechinfo.wordpress.com/2014/07/19/step-by-step-installing-certificate-authority-on-windows-server-2012-r2/)

Step 2: Find a tool to issue a "Certificate Signing Requests" file, where you can find it from third-party CA companies, such as DigiCert (https://www.digicert.com/easy-csr/openssl.htm).

Step 3: Submit it to a public certification authority for signing the certificate.

Step 4: Import the certificate to the MGate Series. Please note that the MGate Series only accepts "xxxx.pem" format.

NOTE

The maximum key length of the MGate devices supports 2,048 bits.

Some well-known third-party CA (Certificate Authority) companies are listed below for your reference: (https://en.wikipedia.org/wiki/Certificate_authority):

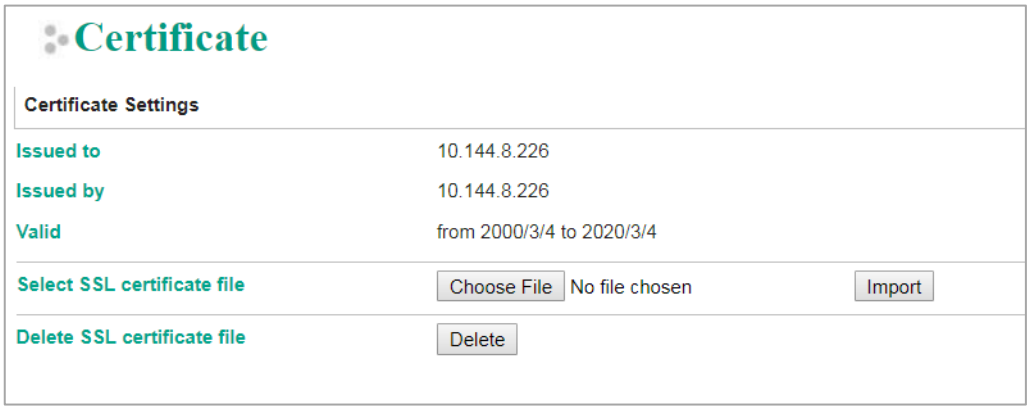
IdenTrust (https://www.identrust.com/)

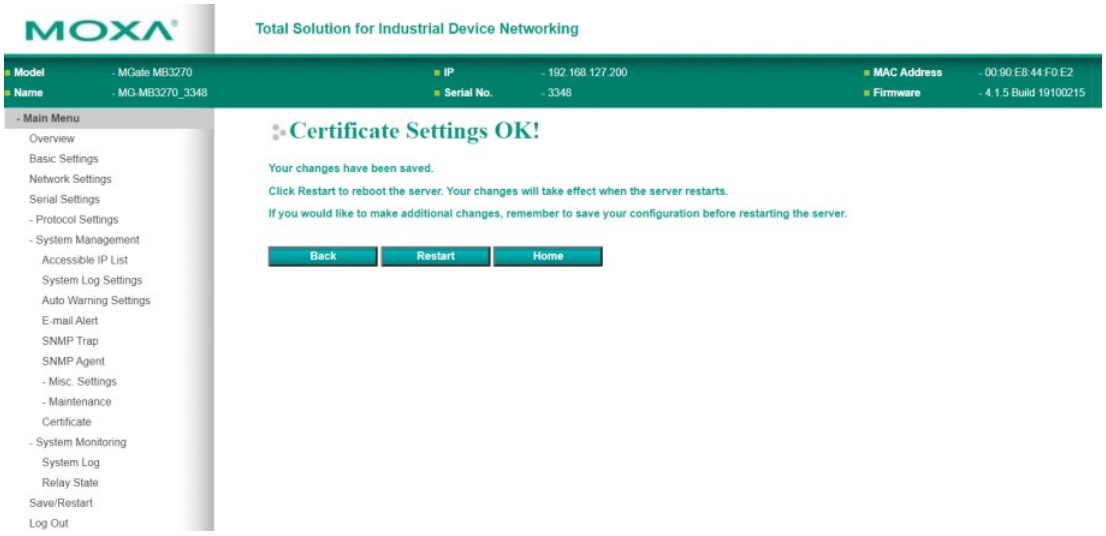
DigiCert (https://www.digicert.com/)

Comodo Cybersecurity (https://www.comodo.com/)

GoDaddy (https://www.godaddy.com/)

Verisign (https://www.verisign.com/)





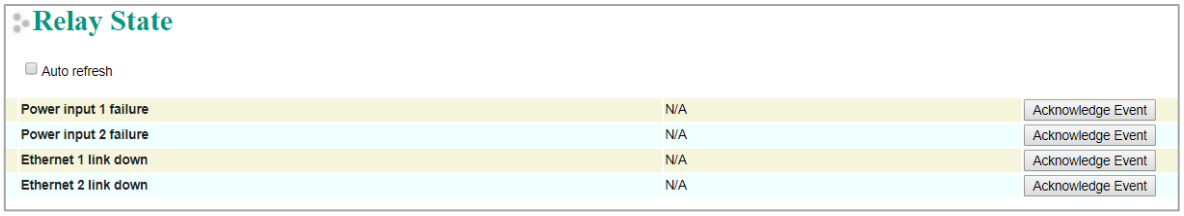
System Monitoring

System Log

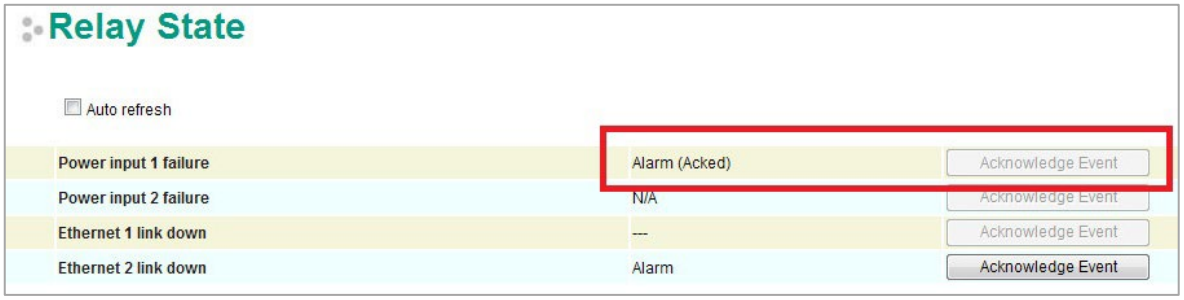
Go to **System Log** under **System Status** to view network connection information.

Relay State

The MGate MB3660 has a built-in 3-pin relay output. It can be triggered by power input failure and LAN link down. Enable the relay output functions by clicking the relay checkbox in the **Auto Warning Settings**.



When a warning event occurs, the relay circuit will activate to enable the warning device, such as a beeper. The field engineer can click the **Acknowledge Event** button to temporarily deactivate the relay circuit and then take some time to troubleshoot the problem.



Once the abnormality has been resolved, the relay will return to normal status.

Protocol Status

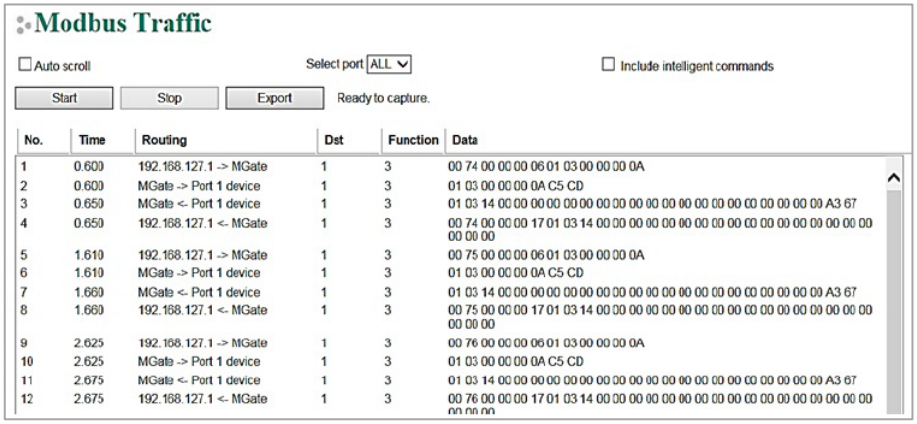
Depending on the protocol modes users set in **Protocol Settings**, the **Protocol Status** supported features would be various accordingly.

Protocol Status Under Transparent Mode

Modbus Traffic

The MGate MB3660 has a built-in Modbus diagnosis/traffic monitor function. For troubleshooting or management purposes, you can diagnose the Modbus protocol communication status and monitor Modbus RTU/ASCII/TCP data passing through the MGate MB3660.

For **transparent mode**, it presents the data in an intelligent, easy-to-understand format with clearly designated fields, including source, destination, function code, and data. Events can be filtered in different ways, and the complete log can be saved to a file for later analysis.

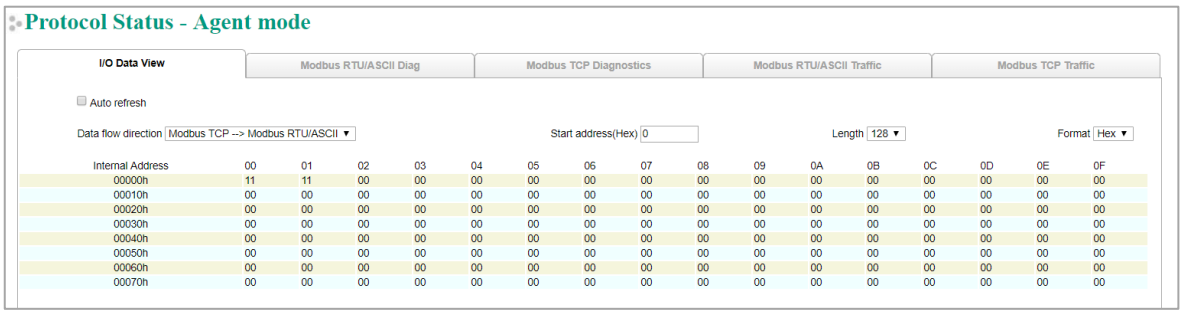


Protocol Status Under Agent Mode

For **agent mode**, it includes I/O Data View, Modbus RTU/ASCII/TCP Diagnose, and Modbus RTU/ASCII/TCP Traffic.

I/O Data View

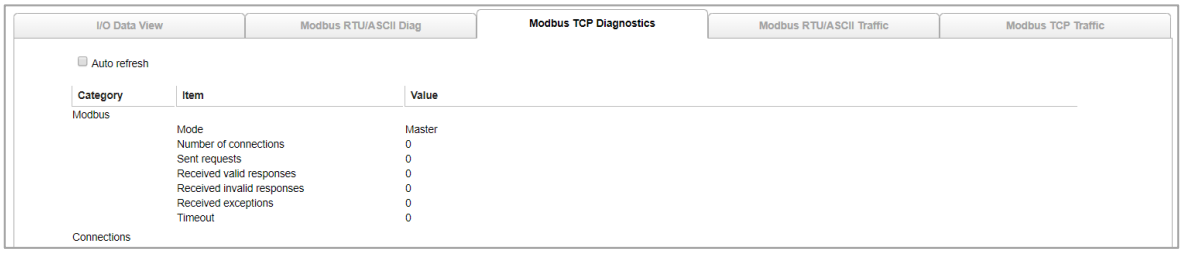
The I/O Data view page displays the internal memory information for input and output data transfers. This page displays the internal memory information for input and output data transfers. View updated values for communication verification here. This function is only available on the web console.



Modbus RTU/ASCII Diag

It includes the Diagnose function, which provides status information for Modbus RTU/ASCII/TCP troubleshooting. Verify the connection status or packet counters to make sure communications are running smoothly.

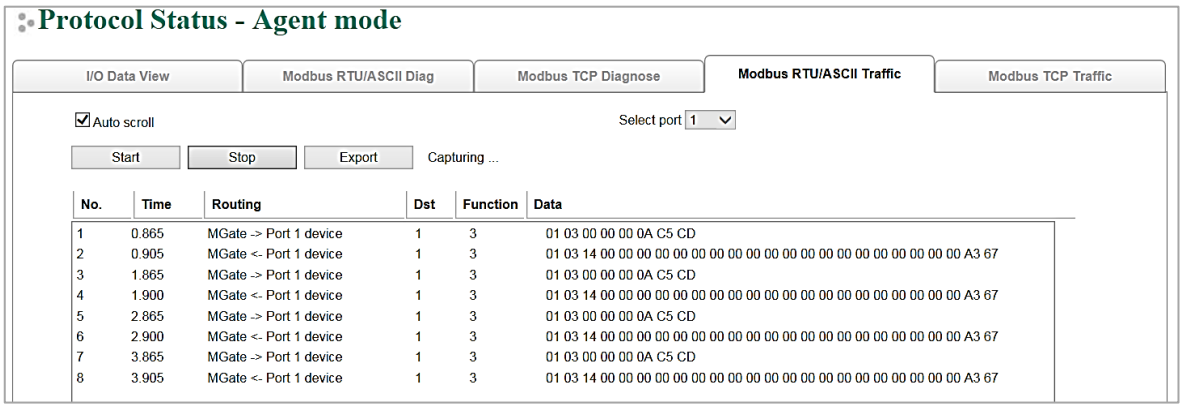
Modbus TCP Diagnostics



Modbus RTU/ASCII Traffic

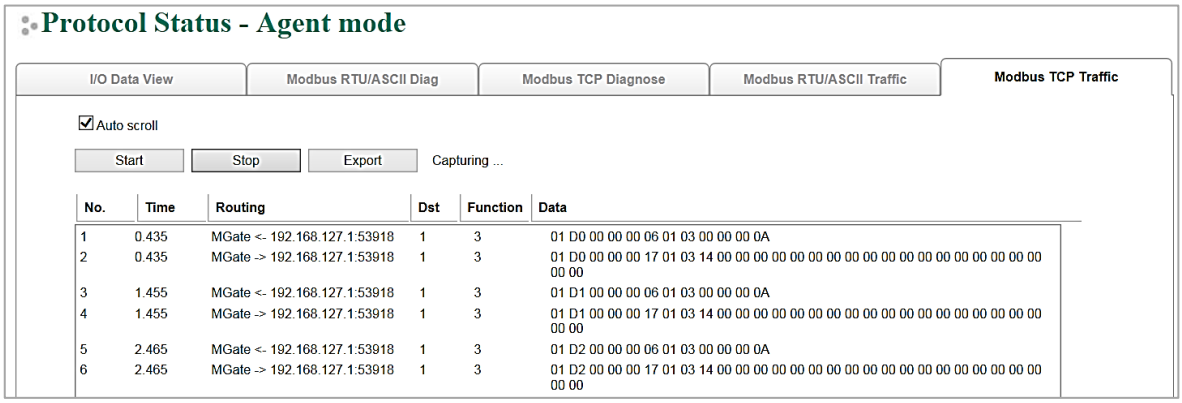
For troubleshooting or management purposes, you can monitor the Modbus RTU/ASCII data passing through the MGate MB3660 on the network. Rather than simply echoing the data, the web console presents the data in an intelligent, easy-to-understand format with clearly designated fields, including source, type, destination, contents, and more.

Events can be filtered in different ways, and the complete log can be saved to a file for later analysis.



Modbus TCP Traffic

For troubleshooting or management purposes, you can monitor the Modbus TCP data passing through the MGate MB3660 on the network.



Serial Redirector Settings

The MGate MB3660 supports the serial redirector function, which integrated Modbus RTU/ASCII and Modbus TCP devices at the same time. There are many serial control systems in the field and local control devices, such as HMI, connected to serial field devices. Using Ethernet-based equipment for remote access and monitoring has become a trend. By setting up the MGate, you will be able to keep the original serial control system and add Modbus TCP client/master (e.g., SCADA) or/and Modbus TCP server/slave (e.g., PLC) into the system. Both Modbus TCP and Modbus RTU/ASCII clients/masters can control Modbus TCP and Modbus RTU/ASCII servers/slaves. The MGate can act as a "Serial Redirector" by configuring the **Protocol Settings**.

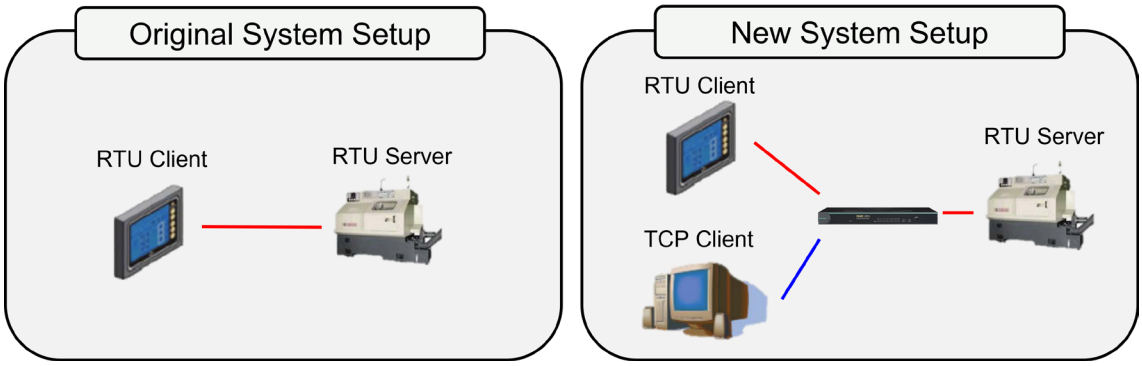
Configuring Serial Redirector

Serial Redirector can be set up by Protocol Settings – **Mode** and **Slave ID Map**. For detailed information, refer to the following scenario and steps.

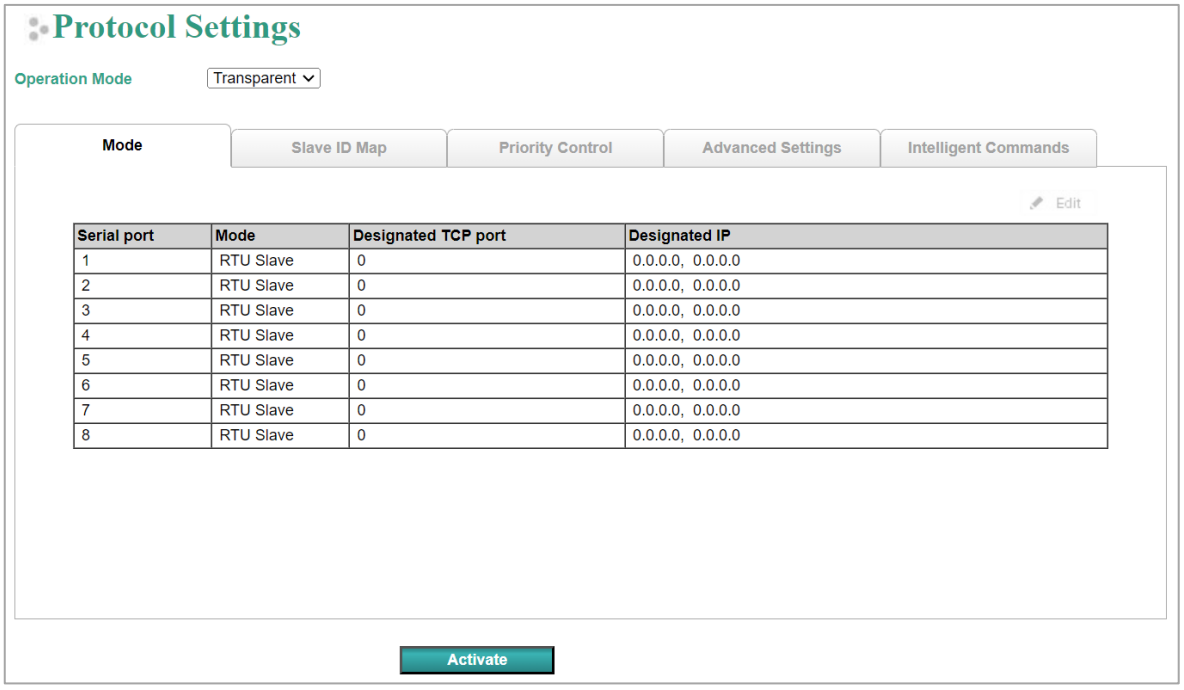
Scenario 1: Adding Modbus TCP Client/Master (SCADA) to an existing Modbus serial system

A Modbus RTU client/master, such as HMI, controls the Modbus RTU server/slave in the original system. Now, you want to add a Modbus TCP client/master such as SCADA into the system and keep the serial Modbus RTU system.

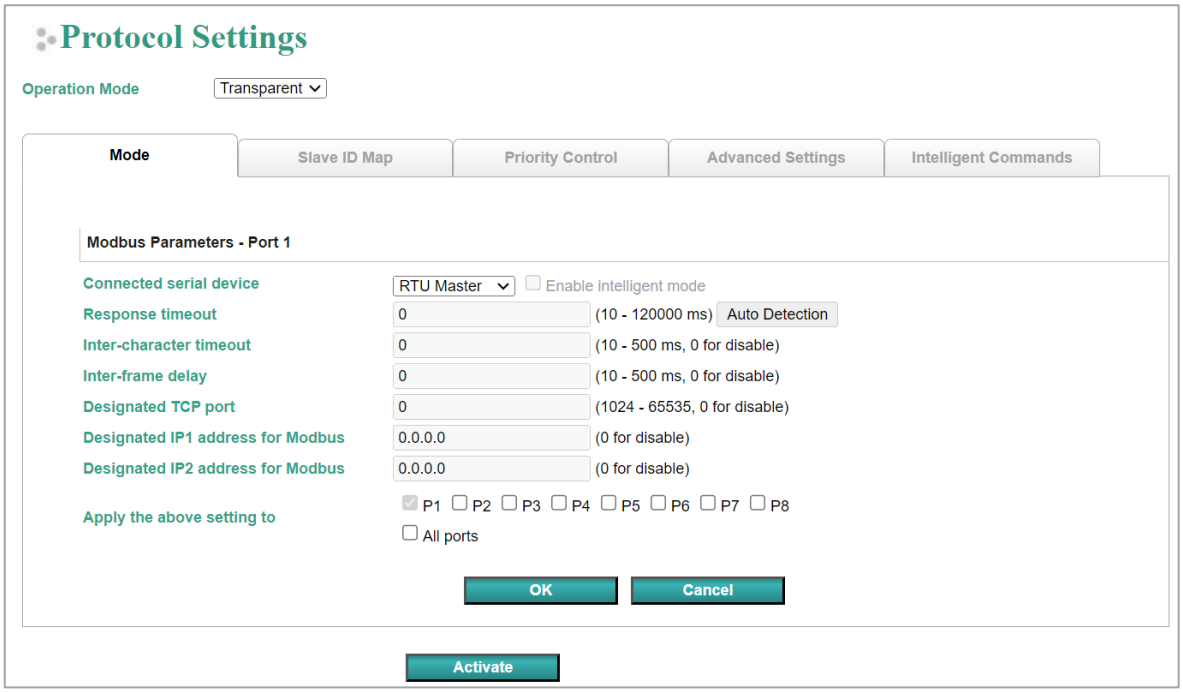
Set up the serial redirector using a MGate MB3660. Configuring with the following steps allows the serial client/master to continue controlling the serial servers/slaves even when implementing a new TCP network.

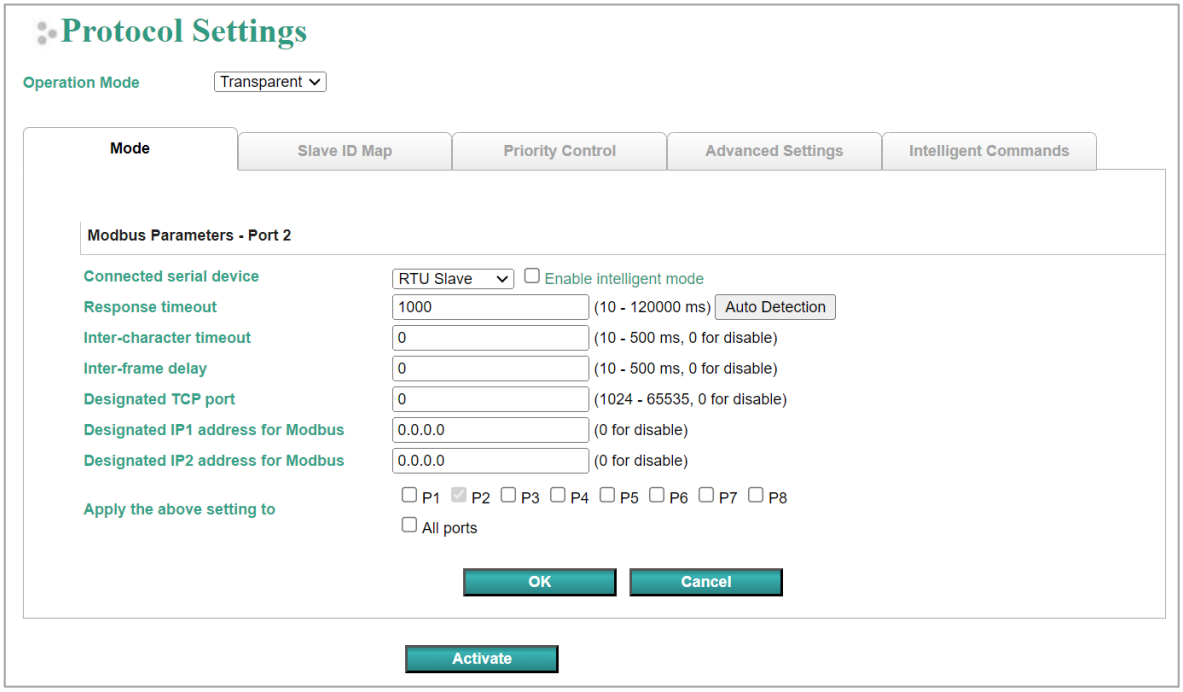


Under **Mode** settings, double-click the intended serial port column to set detailed Modbus parameters, or click Edit to enter the settings page. If the Modbus RTU client/master is connected to the serial port 1, select "RTU Master Mode" under port 1. If a Modbus RTU server/slave is connected to the serial port 2, select "RTU Slave Mode" under port 2.

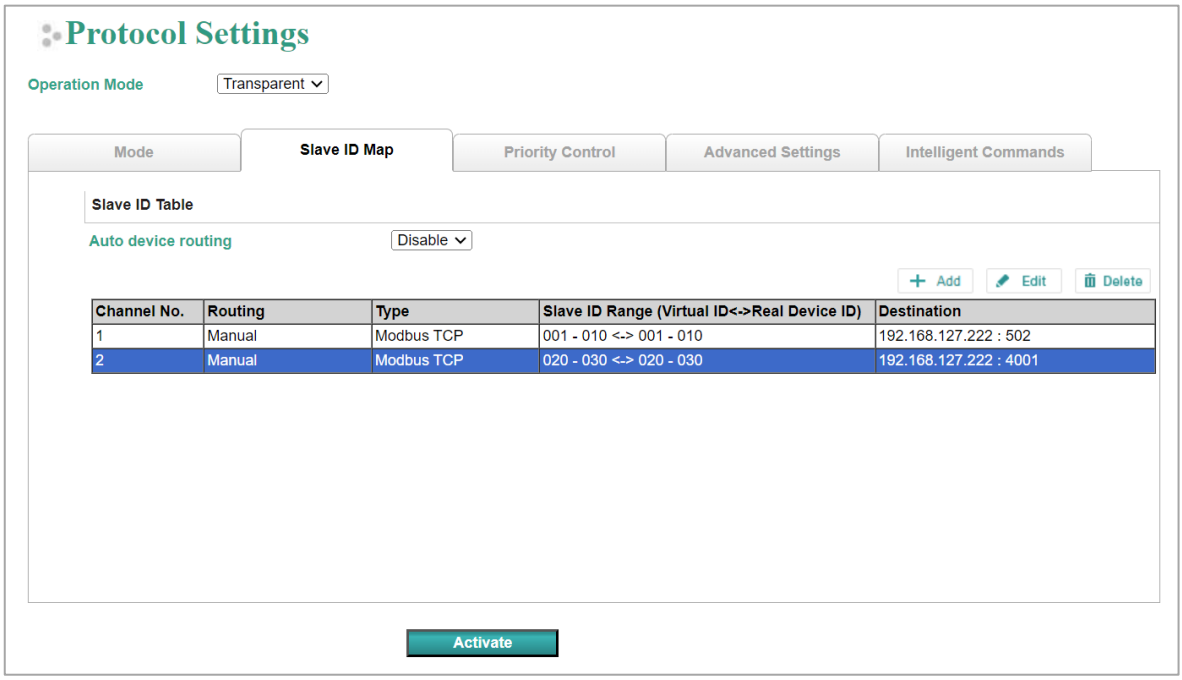


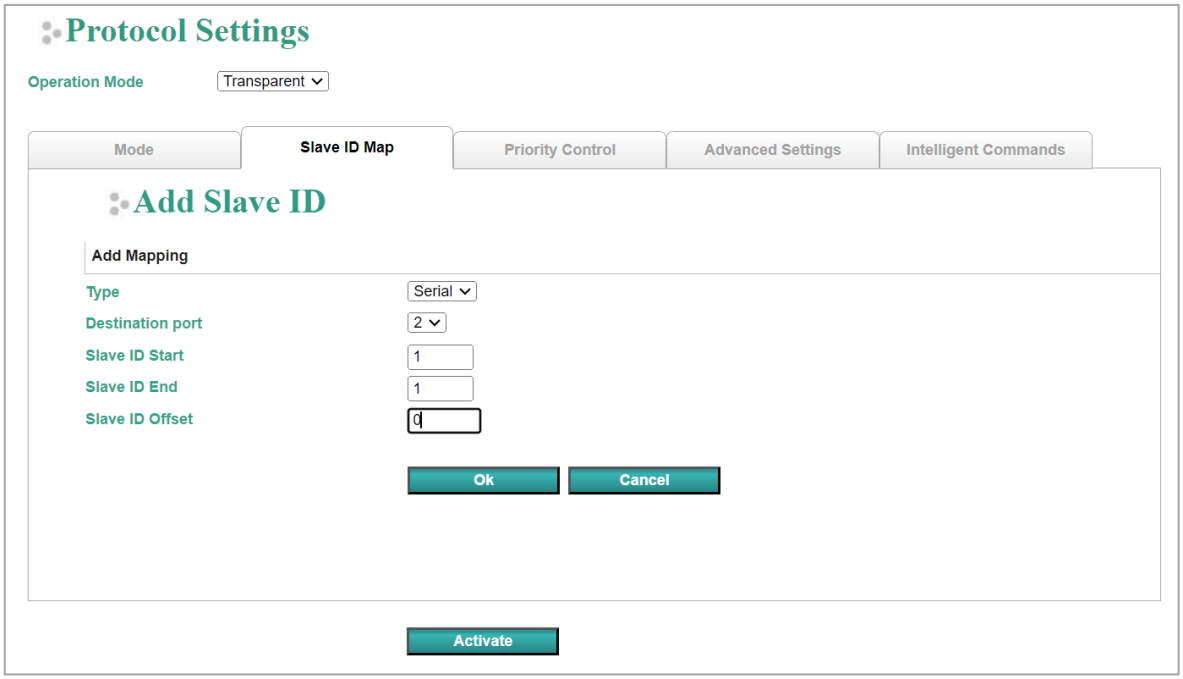
Serial port	Mode	Designated TCP port	Designated IP
1	RTU Slave	0	0.0.0.0, 0.0.0.0
2	RTU Slave	0	0.0.0.0, 0.0.0.0
3	RTU Slave	0	0.0.0.0, 0.0.0.0
4	RTU Slave	0	0.0.0.0, 0.0.0.0
5	RTU Slave	0	0.0.0.0, 0.0.0.0
6	RTU Slave	0	0.0.0.0, 0.0.0.0
7	RTU Slave	0	0.0.0.0, 0.0.0.0
8	RTU Slave	0	0.0.0.0, 0.0.0.0


Protocol Settings
Operation Mode
Transparent
Mode
Slave ID Map
Priority Control
Advanced Settings
Intelligent Commands
Modbus Parameters - Port 1
Connected serial device
RTU Master
Enable intelligent mode
Response timeout
0
(10 - 120000 ms)
Auto Detection
Inter-character timeout
0
(10 - 500 ms, 0 for disable)
Inter-frame delay
0
(10 - 500 ms, 0 for disable)
Designated TCP port
0
(1024 - 65535, 0 for disable)
Designated IP1 address for Modbus
0.0.0.0
(0 for disable)
Designated IP2 address for Modbus
0.0.0.0
(0 for disable)
Apply the above setting to
P1 P2 P3 P4 P5 P6 P7 P8
All ports
OK
Cancel
Activate


Protocol Settings
Operation Mode
Transparent
Mode
Slave ID Map
Priority Control
Advanced Settings
Intelligent Commands
Modbus Parameters - Port 2
Connected serial device
RTU Slave
Enable intelligent mode
Response timeout
1000
(10 - 120000 ms)
Auto Detection
Inter-character timeout
0
(10 - 500 ms, 0 for disable)
Inter-frame delay
0
(10 - 500 ms, 0 for disable)
Designated TCP port
0
(1024 - 65535, 0 for disable)
Designated IP1 address for Modbus
0.0.0.0
(0 for disable)
Designated IP2 address for Modbus
0.0.0.0
(0 for disable)
Apply the above setting to
P1 P2 P3 P4 P5 P6 P7 P8
All ports
OK
Cancel
Activate

Next, go to the **Slave ID Map** tab to configure how the gateway routes Modbus requests. Click Add, Edit, or double-click the intended column to set slave ID mapping. If the slave ID of the Modbus RTU server/slave is 1 and it is connected to serial port 2. Select Serial type and set the destination port 2 for slave ID 1, so the Modbus RTU requests sent to slave ID 1 will be routed to serial port 2.

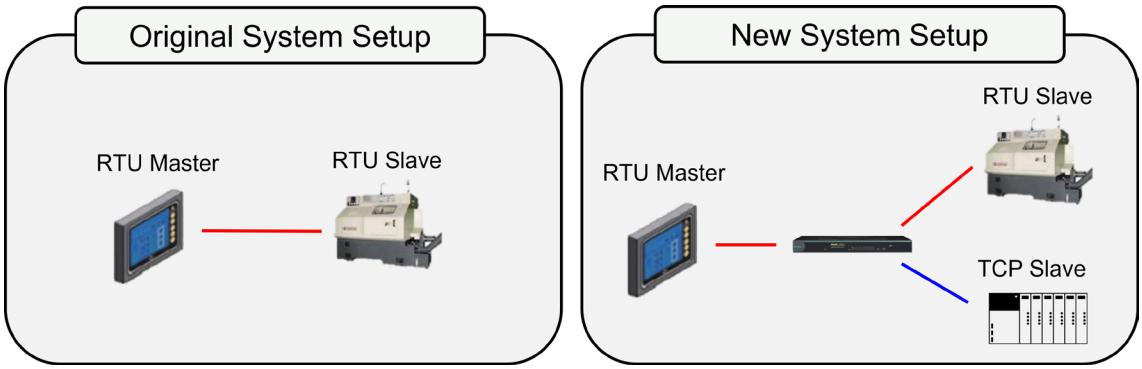




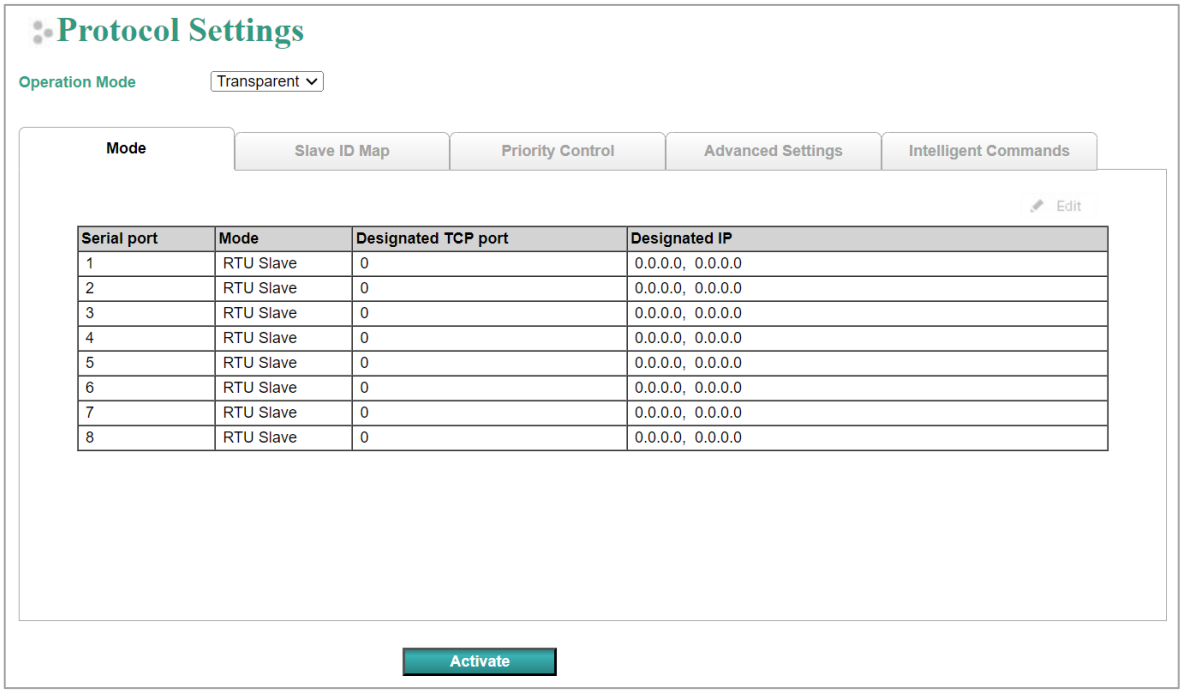
Scenario 2: Adding Modbus TCP Server/Slave to an existing Modbus serial system

A Modbus RTU client/master such as HMI controls the Modbus RTU server/slave in the original system. Now, you want to add the Modbus TCP server/slave such as PLC into the system and keep the serial Modbus RTU system.

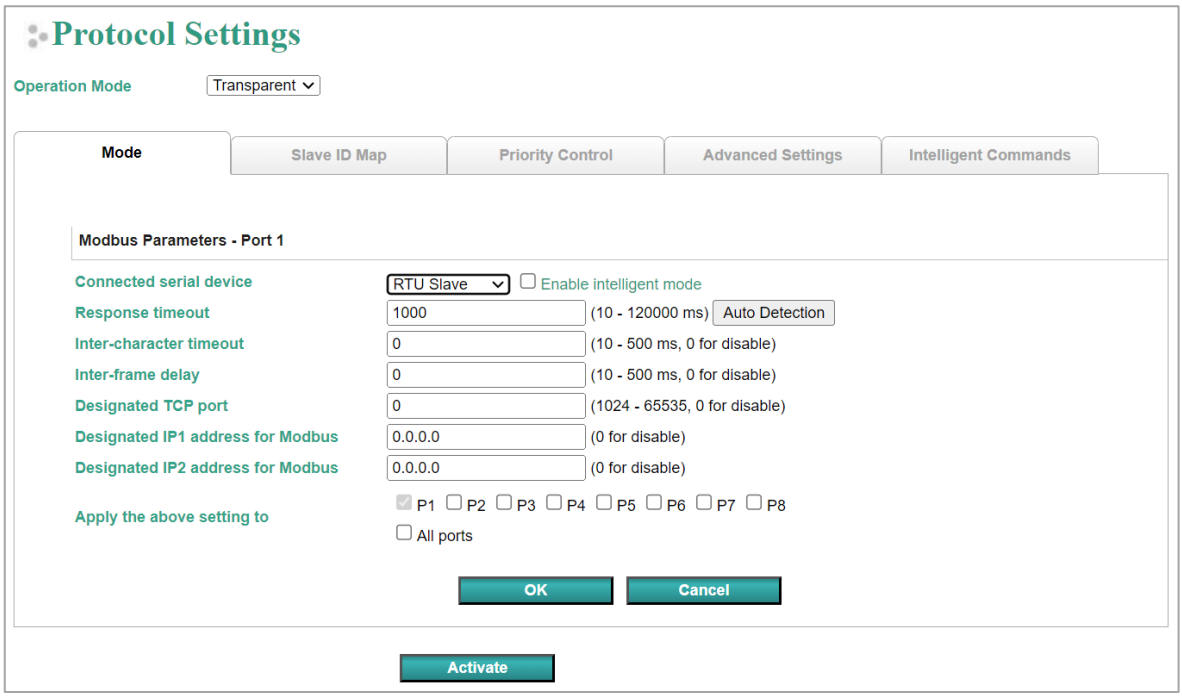
Set up the serial redirector using a MGate MB3660. Configuring with the following steps allows the serial client/master to continue controlling the serial servers/slaves even when implementing a new TCP network.

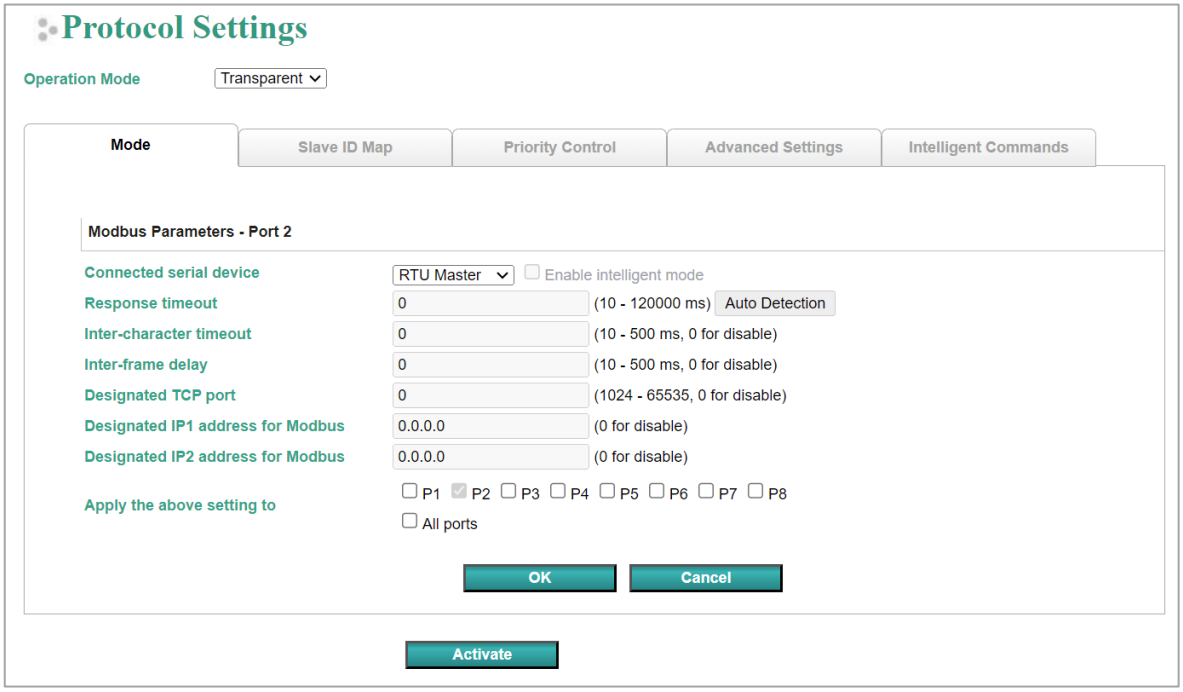


Under Mode settings, double-click the intended serial port column to set detailed Modbus parameters, or click Edit to enter the settings page. If the Modbus RTU server/slave is connected to serial port 1, select "RTU Slave Mode" under port 1. If a Modbus RTU client/master is connected to serial port 2, select "RTU Master Mode" under port 2.

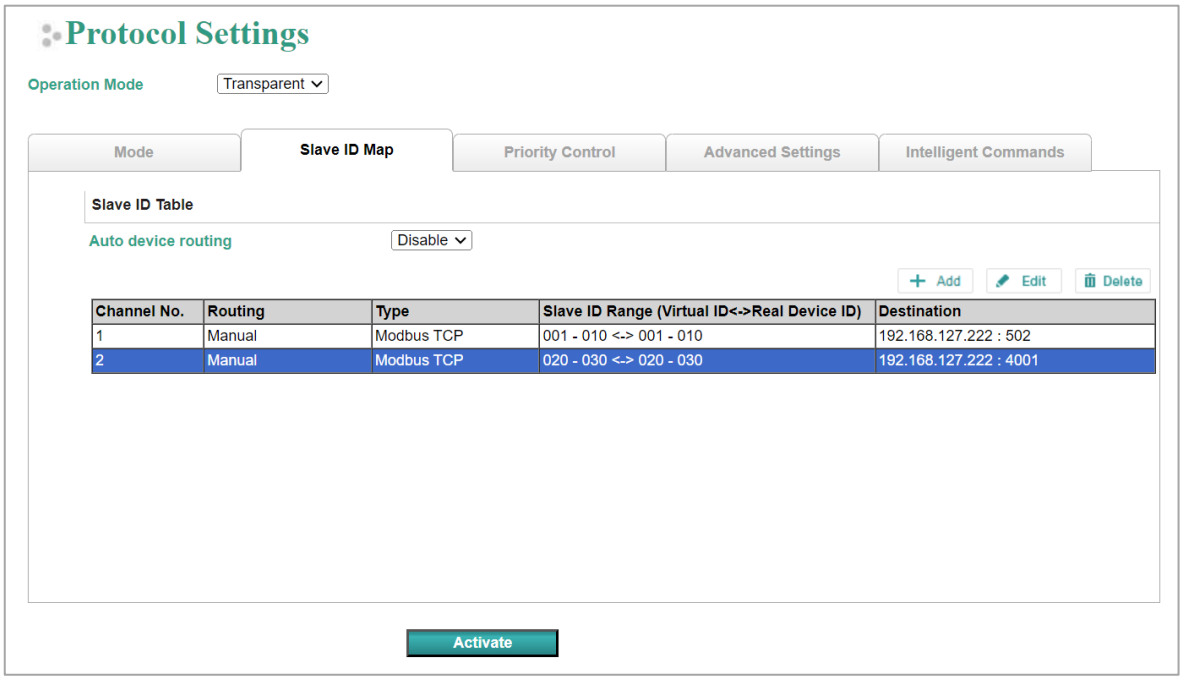


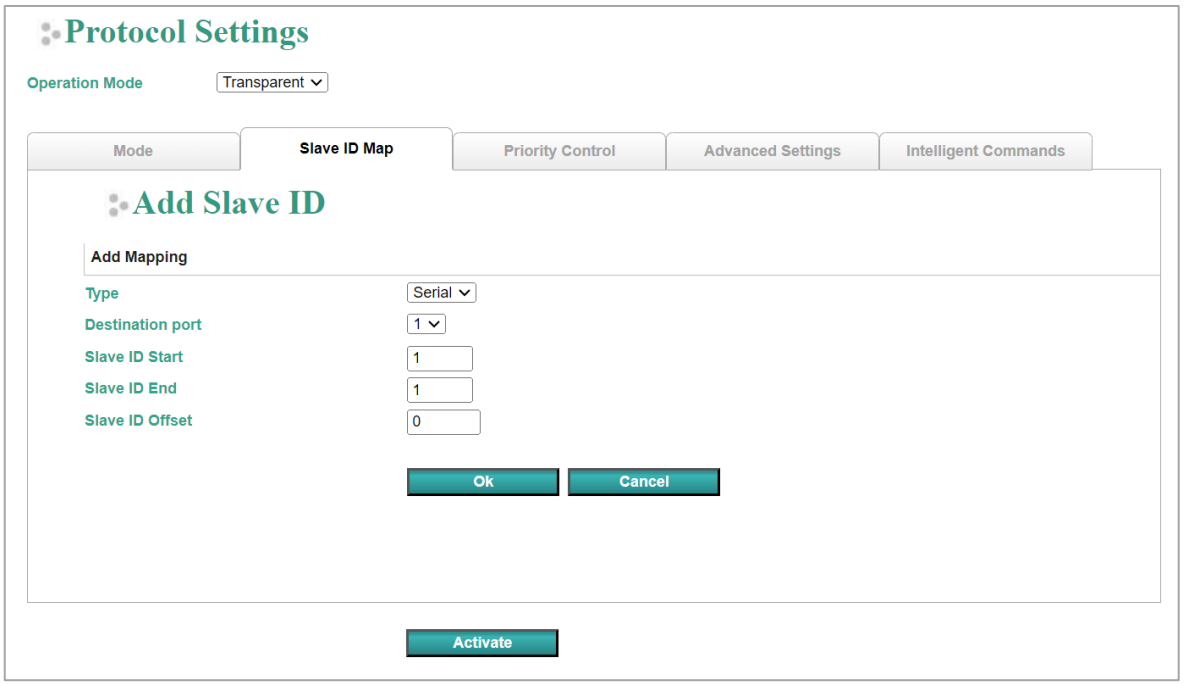
Serial port	Mode	Designated TCP port	Designated IP
1	RTU Slave	0	0.0.0.0, 0.0.0.0
2	RTU Slave	0	0.0.0.0, 0.0.0.0
3	RTU Slave	0	0.0.0.0, 0.0.0.0
4	RTU Slave	0	0.0.0.0, 0.0.0.0
5	RTU Slave	0	0.0.0.0, 0.0.0.0
6	RTU Slave	0	0.0.0.0, 0.0.0.0
7	RTU Slave	0	0.0.0.0, 0.0.0.0
8	RTU Slave	0	0.0.0.0, 0.0.0.0

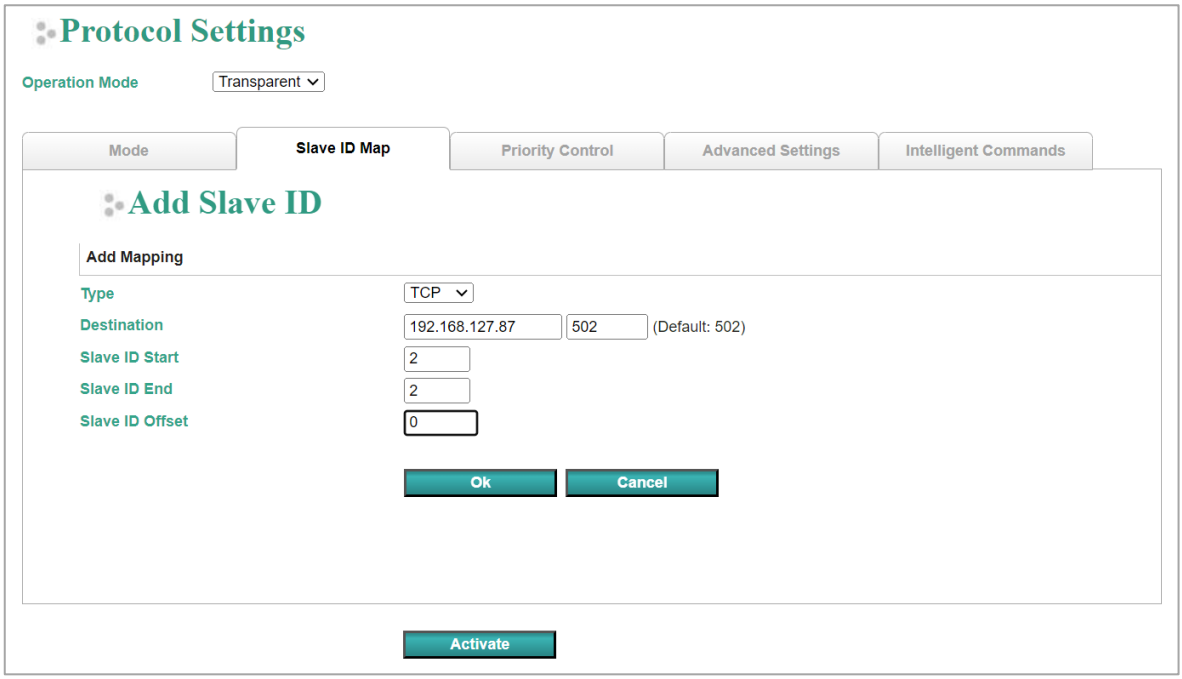

Protocol Settings
Operation Mode
Transparent
Mode
Slave ID Map
Priority Control
Advanced Settings
Intelligent Commands
Modbus Parameters - Port 1
Connected serial device
RTU Slave
Enable intelligent mode
Response timeout
1000
(10 - 120000 ms)
Auto Detection
Inter-character timeout
0
(10 - 500 ms, 0 for disable)
Inter-frame delay
0
(10 - 500 ms, 0 for disable)
Designated TCP port
0
(1024 - 65535, 0 for disable)
Designated IP1 address for Modbus
0.0.0.0
(0 for disable)
Designated IP2 address for Modbus
0.0.0.0
(0 for disable)
Apply the above setting to
P1 P2 P3 P4 P5 P6 P7 P8
All ports
OK
Cancel
Activate


Protocol Settings
Operation Mode
Transparent
Mode
Slave ID Map
Priority Control
Advanced Settings
Intelligent Commands
Modbus Parameters - Port 2
Connected serial device
RTU Master
Enable intelligent mode
Response timeout
0
(10 - 120000 ms)
Auto Detection
Inter-character timeout
0
(10 - 500 ms, 0 for disable)
Inter-frame delay
0
(10 - 500 ms, 0 for disable)
Designated TCP port
0
(1024 - 65535, 0 for disable)
Designated IP1 address for Modbus
0.0.0.0
(0 for disable)
Designated IP2 address for Modbus
0.0.0.0
(0 for disable)
Apply the above setting to
P1 P2 P3 P4 P5 P6 P7 P8
All ports
OK
Cancel
Activate

Next, go to the Slave ID Map tab to configure how the gateway routes Modbus requests. Click Add, Edit, or double-click the intended column to set slave ID mapping. If the slave ID of the Modbus RTU server/slave is 1 and connected to serial port 1, then the slave ID of the Modbus TCP server/slave is 2 with an IP address of 192.168.127.87:502. Set up two routing rules. Select Serial type and set the destination port 1 for slave ID 1, so the Modbus RTU requests sent to slave ID 1 will be routed to serial port 1. Select TCP type and set the IP address 192.168.127.87:502 to be the destination for slave ID 2, then Modbus RTU requests sent to slave ID 2 will be routed to 192.168.127.87:502.

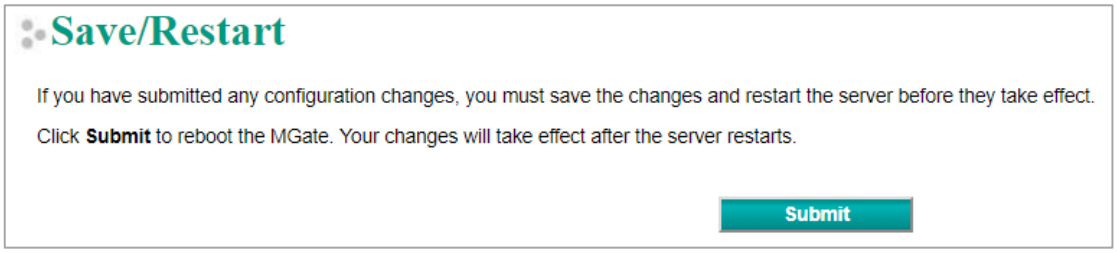






Save/Restart

All changes will be activated by clicking the **Submit** button first and then restarting the gateway. If a lot of settings need to be changed, you can click **Submit** for each setting and then click **Save/Restart** to activate all the changes.



Logout

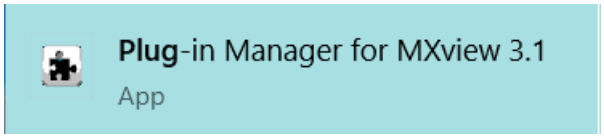
For safety reasons, remember to log out of the web utility to prevent people who do not have the proper authorization from accessing the gateway.

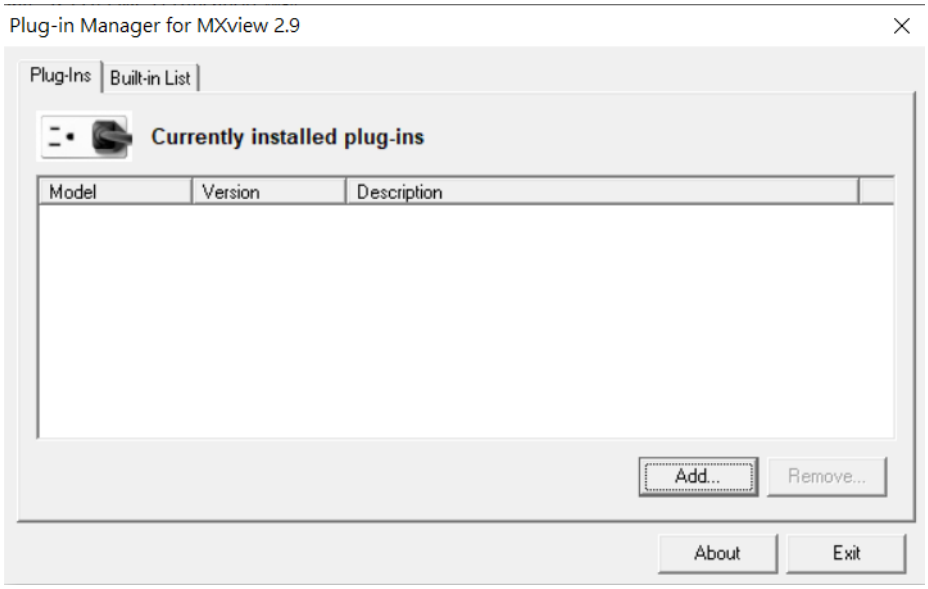
MXview

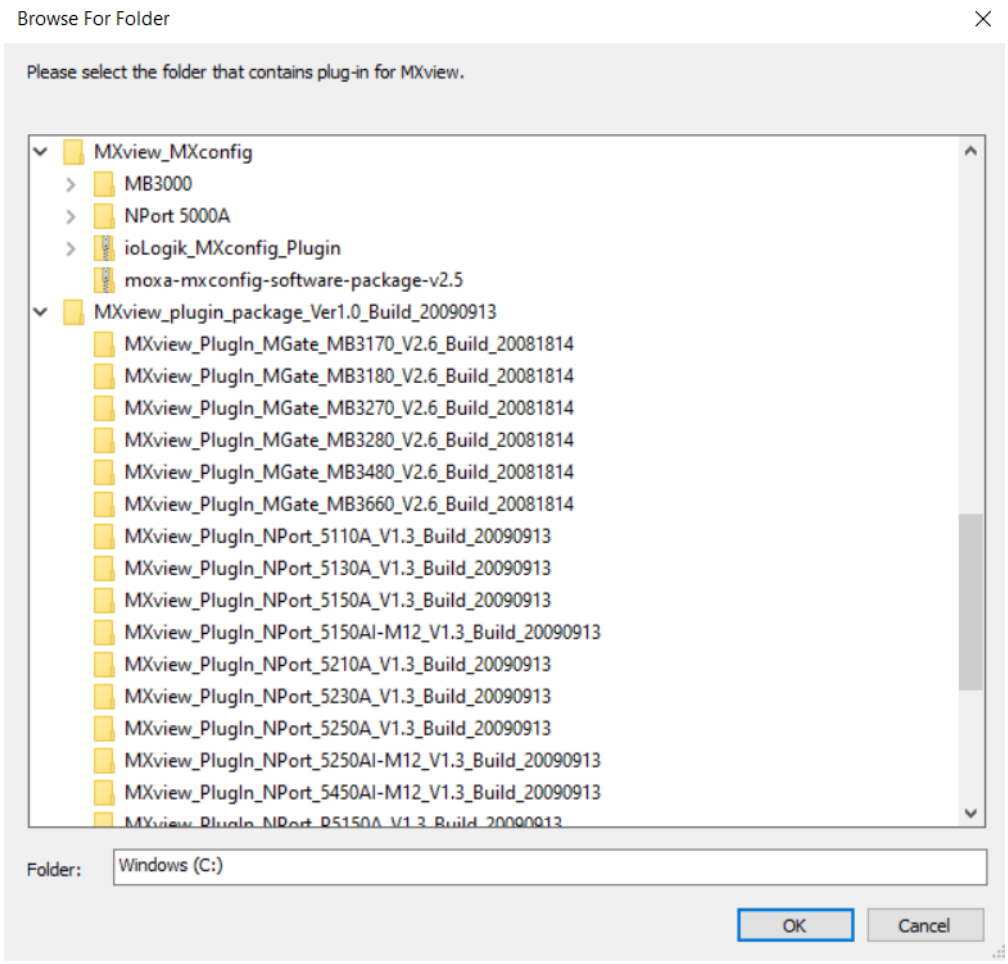
The Moxa MXview network management software gives you a convenient graphical representation of your Ethernet network and allows you to configure, monitor, and diagnose Moxa networking devices. MXview provides an integrated management platform that can manage the Moxa MGate series of products as well as Ethernet switches and wireless APs, and SNMP-enabled and ICMP-enabled devices installed on subnets. MXview includes an integrated MIB compiler that supports any third-party MIB. It also allows you to monitor third-party OIDs and Traps. Network and Trap components that have been located by MXview can be managed via web browsers from both local and remote sites—anytime, anywhere.

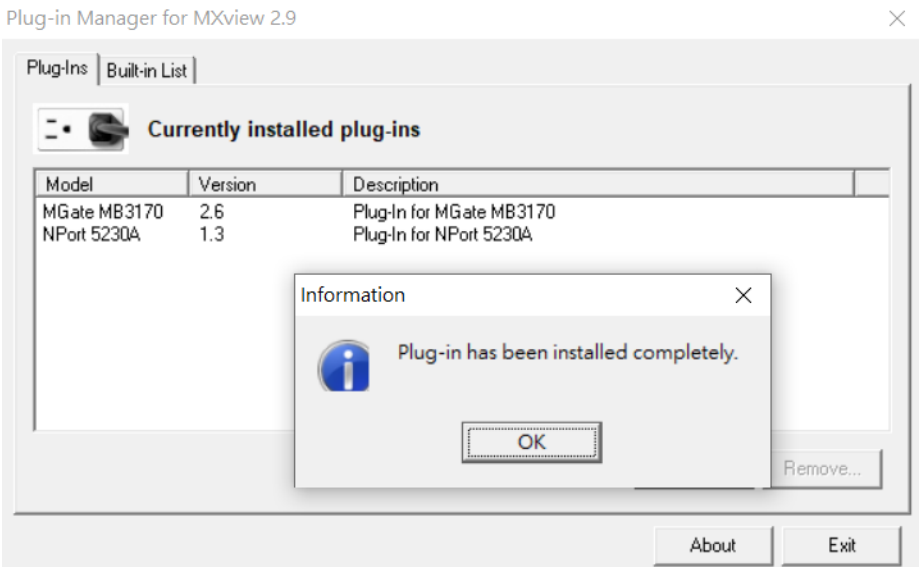
Additionally, the Moxa MXview supports Security View function to follow Moxa's security guidelines, which are based on current IEC 62243 component-level recommendations. Security View checks the security level of Moxa's network devices, including MGate MB3000 Series.

Before adding the MGate MB3000 devices to the MXview utility, you must add the MB3000 plug-in package to MXview via Plug-in Manager. The Plug-in Manager is automatically installed when setting up MXview. You can download the plug-in package on the product page. Please execute **Plug-in Manager** and **add** the plug-in package.







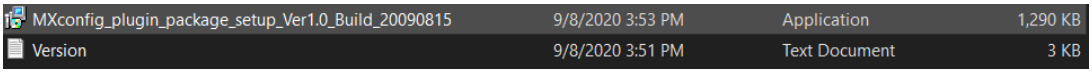


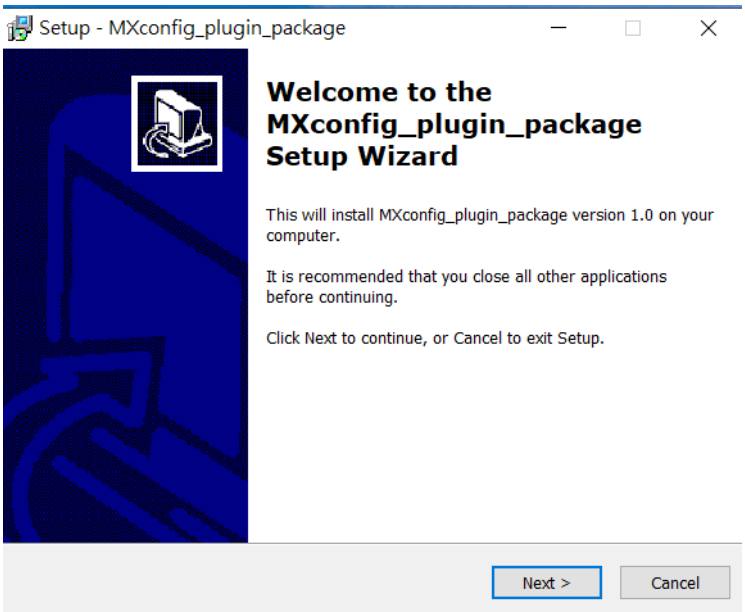
MXconfig

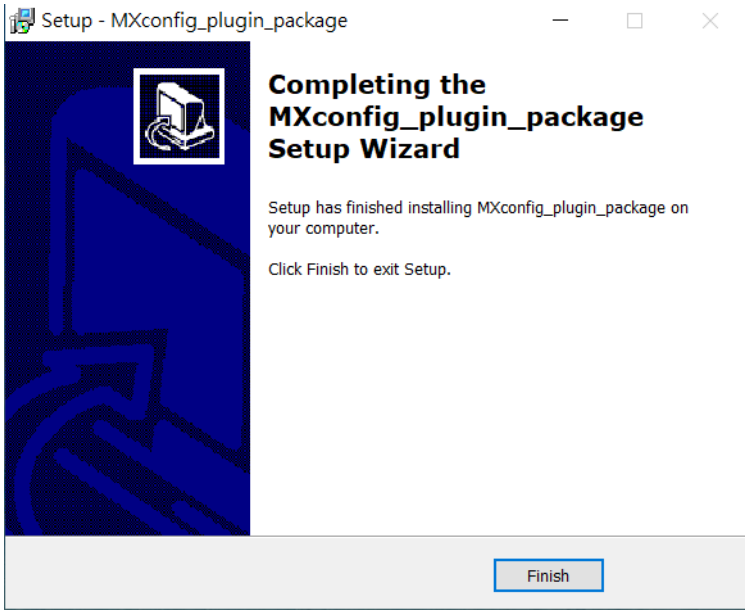
Moxa's MXconfig is a comprehensive Windows-based utility that is used to install, configure, and maintain multiple Moxa devices in industrial networks. This suite of useful tools helps users set the IP addresses of multiple devices with one click, configure the redundant protocols and VLAN settings, change multiple network configurations of multiple Moxa devices, upload firmware to multiple devices, export/import configuration files, copy configuration settings across devices, easily link to web and Telnet consoles, and test device connectivity. MXconfig gives device installers and control engineers a powerful and easy way to mass configure devices, and effectively reduces the setup and maintenance cost.

Through MXconfig, users can access the MGate MB3000 devices and take advantage of additional functions, such as searching for the MGate MB3000 devices, setting network configurations, upgrading firmware, and importing/exporting configurations.

Before configuring the MGate MB3000 devices via MXconfig utility, you must add the MB3000 plug-in package to MXconfig. You can download the plug-in package from the product page and execute plug-in package with just a few clicks.







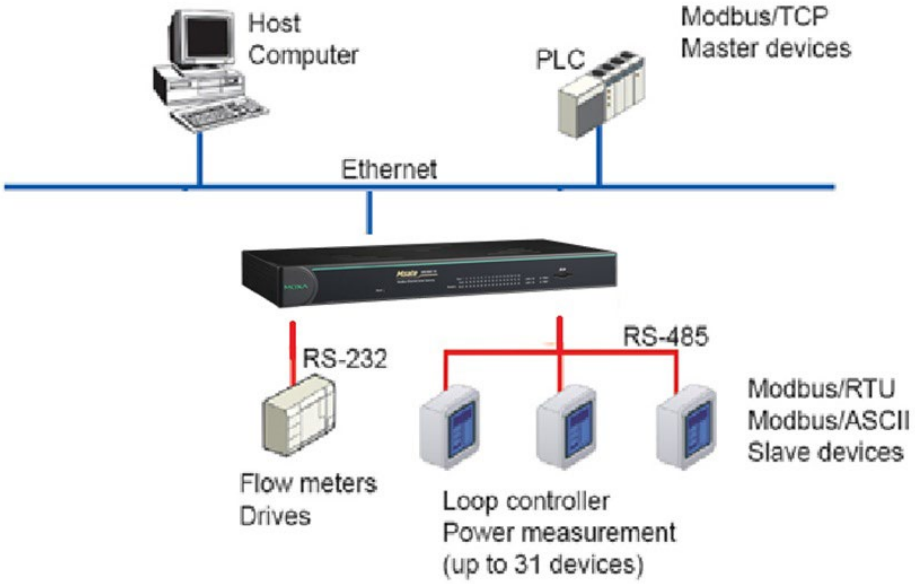
For more detailed information regarding MXview/MXconfig, download the user's manual from Moxa's website at http://www.moxa.com

Ethernet Clients/Masters With Multiple Serial Servers/Slaves

Connect all Modbus devices over an Ethernet network

Most modern PLCs and host computers support Modbus TCP over Ethernet. To access discrete Modbus RTU/ASCII devices for data collection and control, they can rely on the MGate MB3660 Modbus gateway.

The MGate MB3660 supports Modbus TCP with up to 256 simultaneous connections. The serial interface supports both RS-232 and RS-422/485, selectable through software. Each serial port can be connected to one RS-232 or RS-422 serial device, or to 31 RS-485 serial devices.

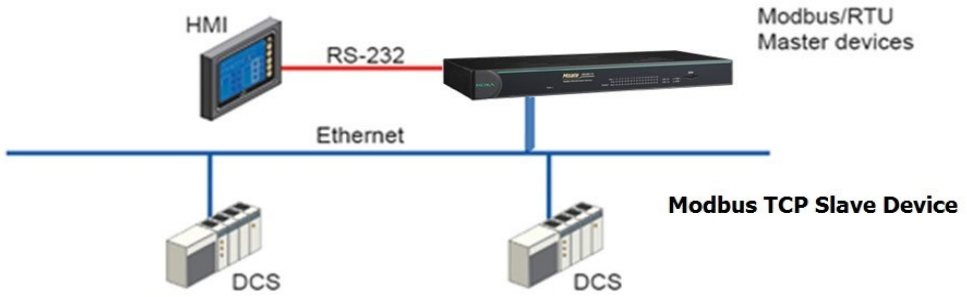


Serial Clients/Masters with Multiple Ethernet Servers/Slaves

Link a serial client/master device with Ethernet server/slave devices

Many HMI (Human Machine Interface) systems use a serial interface to connect to a discrete DCS (Data Control System). However, many DCSs are now Ethernet-based and operate as a Modbus TCP server/slave device.

The MGate MB3660 Modbus gateway can link a serial-based HMI to distributed DCSs over an Ethernet network. Up to 128 Modbus TCP server/slave devices are supported by each MGate MB3660.



Modbus TCP Clients/Masters with ASCII and RTU Servers/Slaves

Link TCP client/master devices with both ASCII and RTU serial devices simultaneously

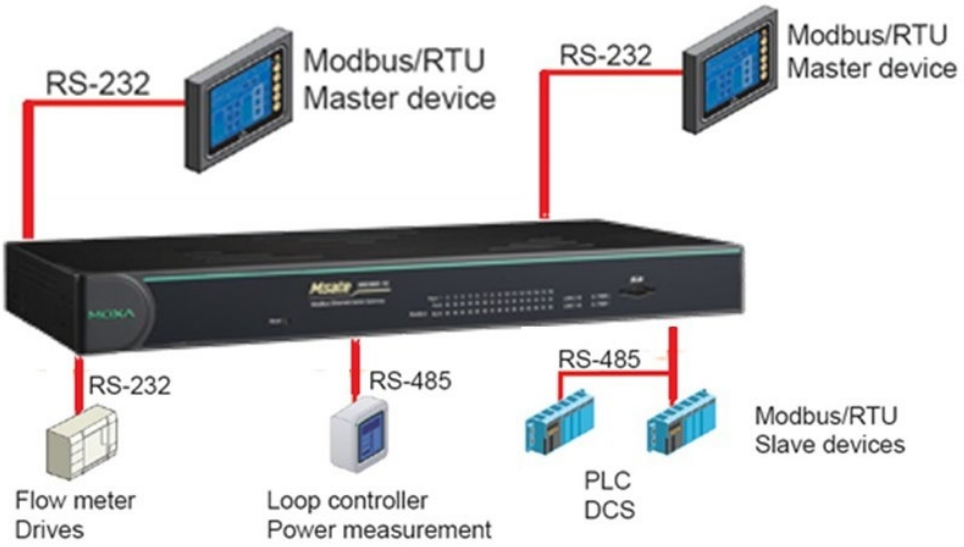
When integrating Modbus networks, you may encounter different Modbus serial networks that use different baudrates or a different protocol. Modbus ASCII might be used by some devices, while Modbus RTU is used by other devices.

The MGate models with two or more ports can integrate serial Modbus networks that use different parameters or protocols. You can configure each serial port to a specific Modbus serial environment to set up a slave ID map. After configuration, only the gateway will be visible to Modbus TCP clients/masters, and all serial devices will be integrated behind it.

Serial Client(s)/Master(s) with Serial Servers/Slaves

Let Modbus serial devices communicate

The MGate MB3660 provides a feature for connecting serial client(s)/master(s) with serial server/slave devices by using transparent mode. You only need to set up the slave ID routing mechanism.



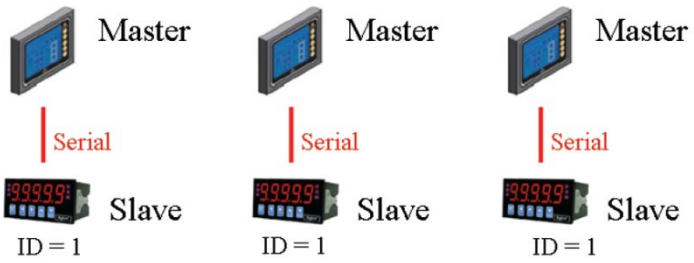
6. Case Studies

Introduction

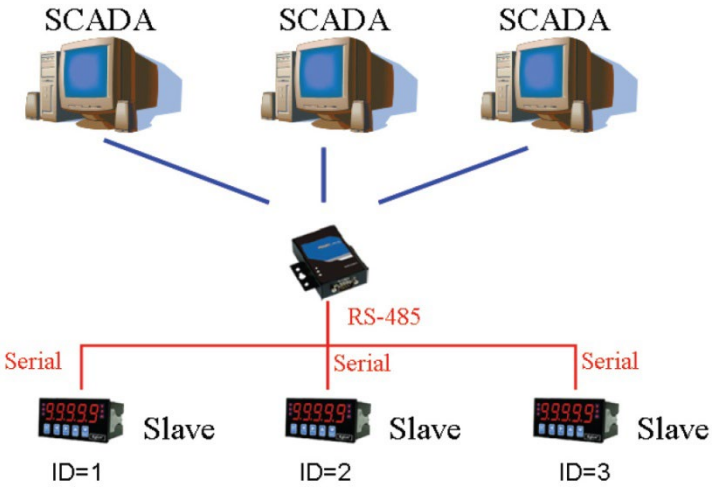
For many reasons, a Modbus gateway might be used to integrate Modbus networks. However, every situation has its own requirements and difficulties. Users may wonder how the gateway can help or even if the gateway is suitable for the system.

Replace Serial Clients/Masters with Ethernet Client(s)/Master(s), Configurable Slave IDs

In this scenario, the original control system comprises several serial-based systems. In each system, a serial client/master directly controls serial server/slave devices as follows:

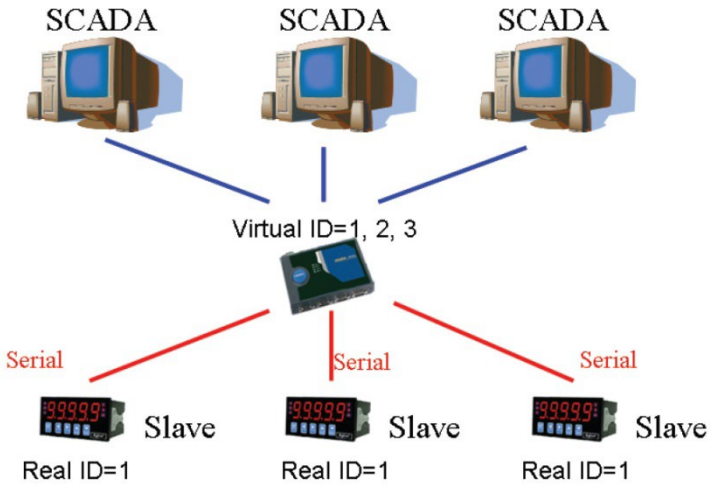


The MGate MB3660 can connect to each serial server/slave so Ethernet SCADA clients/masters will control them. However, since slave IDs cannot be repeated in a system, we will need to change the IDs of some of the slaves to integrate them into a single network, as follows:



Replace Serial Clients/Masters with Ethernet Client(s)/Master(s), Fixed Slave IDs

Some legacy Modbus slave devices have fixed IDs that cannot be changed. In order to integrate the devices into a Modbus TCP network, a multiport MGate model can be used to assign virtual slave IDs. For more information about virtual slave IDs, refer to Set Up Slave ID Mapping, chapter 4.

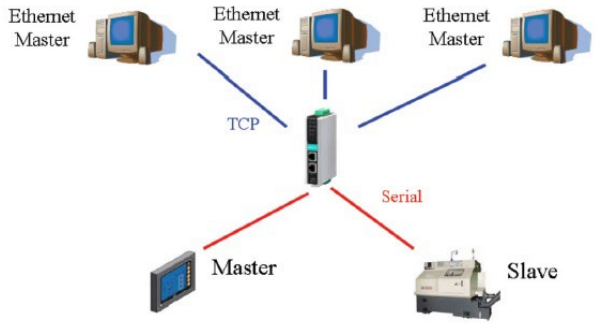


Keep Serial Client/Master and Add Ethernet Client(s)/Master(s)

In this scenario, the serial control system is a direct, low-latency system. The serial client/master must not be replaced, but Ethernet clients/masters will need to have access to the serial servers/slaves for monitoring or supervision.

Serial Redirector

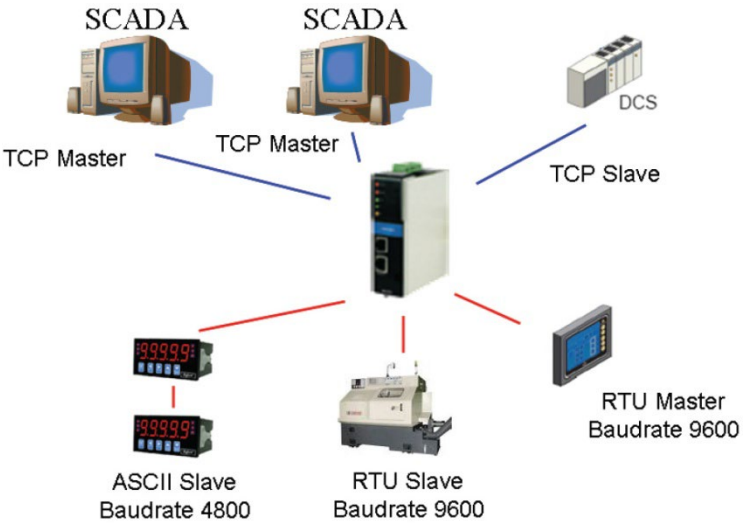
The MGate MB3660 supports the serial redirector function, which integrates Modbus RTU/ASCII and Modbus TCP devices at the same time. There are many serial control systems in the field and local control devices, such as HMI, connected to serial field devices. Using Ethernet-based equipment for remote access and monitoring has become a trend. By setting up the MGate, you will be able to keep the original serial control system and add Modbus TCP client/master (e.g., SCADA) or/and Modbus TCP server/slave (e.g., PLC) to the system. Both Modbus TCP and Modbus RTU/ASCII clients/masters can control Modbus TCP and Modbus RTU/ASCII servers/slaves. The MGate can act as a "Serial Redirector" by configuring the protocol settings. For more information about how to set up the serial redirector function, please refer to the Serial Redirector Settings in this manual.



Integrate Modbus RTU, ASCII, and TCP at the Same Time

There can be a wide range in number, type, and sophistication of devices on the factory floor. The most common devices are simple serial-based meters, which report certain information relating to factory environment or equipment. However, other Modbus serial servers/slaves may be as complex as a manufacturing machine or a PLC controller.

When integrating these devices, there may be issues if different serial environments are used for different devices. One system may use a different baudrate than another or may use Modbus ASCII instead of Modbus RTU. The MGate MB3660 allows the different Modbus systems to be integrated into one network, regardless of the protocol or communication parameters.



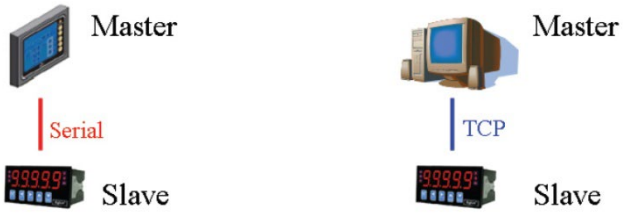
A. Modbus Overview

Introduction

Modbus is one of the most popular automation protocols in the world. It supports both serial and Ethernet devices. Many industrial devices, such as PLCs, DCSs, HMIs, instruments, meters, motors, and drivers, use Modbus as their communication standard.

Devices are Either Clients/Masters or Servers/Slaves

All Modbus devices are classified as either a client/master or a server/slave. Clients/Masters initiate all communication with servers/slaves and do not communicate to other clients/masters. Servers/Slaves are completely passive and communicate only by sending a response to a clients/master's request.



Slaves are Identified by ID

Each Modbus slave in a system is assigned a unique ID between 1 and 247. Whenever a client/master makes a request, the request must include the ID of the intended recipient. Client/Master devices themselves have no ID.

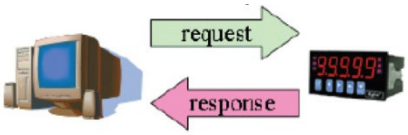
0	1~247	248~255
Broadcast address	Slave individual address	Reserved

Communication is by Request and Response

All Modbus communication is by request and response. A client/master sends a request, and a server/slave sends a response. The client/master will wait for the server/slave's response before sending the next request. For broadcast commands, no response is expected. This is illustrated by three scenarios as follows:

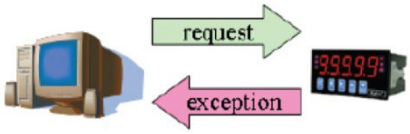
Normal

The client/master sends a request to the server/slave. The server/slave sends a response with the requested information.



Exception

The client/master sends a request to the server/slave. The server/slave may not support the command, or an error is detected, so it sends an exception to the client/master.



Broadcast

The client/master sends a broadcast command, such as a reset command. Every server/slave on the network complies with the command, and no response is sent to the client/master.

Requests Need a Time Limit

The original Modbus protocol was not designed for simultaneous requests or simultaneous clients/masters, so only one request on the network can be handled at a time. When a client/master sends a request to a server/slave, no other communication may be initiated until after the server/slave responds. The Modbus protocol specifies that clients/masters use a response timeout function to identify when a server/slave is nonresponsive due to device or line failure. This function allows a client/master to give up on a request if no response is received within a certain amount of time. This is illustrated as follows:

Response Timeout

The client/master sends a request. The server/slave is unresponsive for the amount of time specified by the response timeout function. The client/master gives up on the request and resumes operation, allowing another request to be initiated.

To allow for a wide range of devices, baudrates, and line conditions, actual response timeout values are left open for manufacturers to determine. This allows the Modbus protocol to accommodate a wide range of devices and systems. However, this also makes it difficult for system integrators to know what response timeout value to use during configuration, especially with older or proprietary devices.

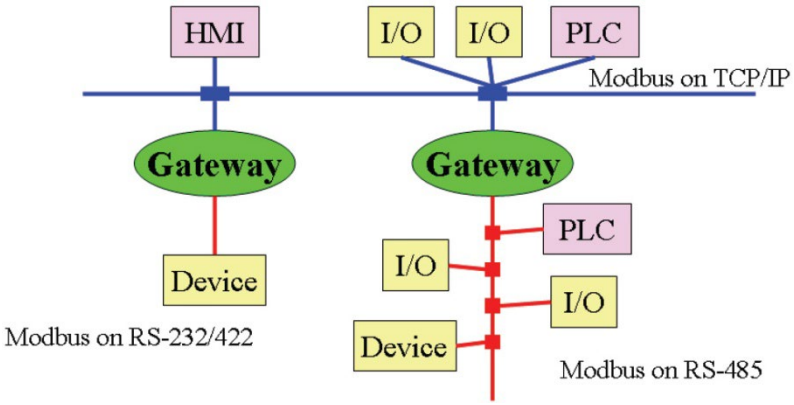
The MGate MB3660 provides a patent-pending function that tests all attached devices and recommends a response timeout value. This function saves considerable time and effort for system integrators and results in more accurate timeout settings.

Modbus Ethernet vs. Modbus Serial

Although Modbus is intended as an application-layer messaging protocol, the data format and communication rules for Ethernet-based Modbus TCP are different from serial-based Modbus ASCII and RTU.

The major difference between the Ethernet and serial Modbus protocols is the behavior of the communication model. Modbus ASCII and RTU allow only one request on the network at a time. Once a request is sent, no other communication on the bus is allowed until the server/slave sends a response, or until the request times out. However, Modbus TCP allows simultaneous requests on the network, from multiple clients/masters to multiple servers/slaves. TCP clients/masters cannot send more than one request at a time to a server/slave, but they can send requests to other servers/slaves before a response is received. The Modbus TCP standard recommends that servers/slaves be able to queue up to 16 requests at a time. The MGate MB3660 will queue up to 32 requests from each TCP client/master, for up to 16 TCP clients/masters.

Integrate Modbus Serial and Ethernet with Gateways



Ordinarily, Modbus TCP and Modbus ASCII/RTU cannot communicate with each other. However, with a Modbus gateway in between the Modbus serial network and the Modbus Ethernet network, TCP clients/masters can communicate with serial servers/slaves and serial clients/masters can communicate with TCP servers/slaves.

B. SNMP Agents with MIB II and RS-232-Like Groups

The MGate MB3660 has built-in Simple Network Management Protocol (SNMP) agent software that supports SNMP Trap, RFC1317 and RS-232-like groups, and RFC 1213 MIB-II.The following topics are covered in this appendix:

RFC1213 MIB-II Supported SNMP Variable

System MIB	Interfaces MIB	IP MIB	ICMP MIB
sysDescr	ifNumber	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	icmpInAddrMaskReps
	ifInDiscards	ipReasmReqds	icmpOutMsgs
	ifInErrors	ipReasmOKs	icmpOutErrors
	ifInUnknownProtos	ipReasmFails	icmpOutDestUnreachs
	ifOutOctets	ipFragOKs	icmpOutTimeExcds
	ifOutUcastPkts	ipFragFails	icmpOutParmProbs
	ifOutNUcastPkts	ipFragCreates	icmpOutSrcQuenchs
	ifOutDiscards	ipAdEntAddr	icmpOutRedirects
	ifOutErrors	ipAdEntIfIndex	icmpOutEchos
	ifOutQLen	ipAdEntNetMask	icmpOutEchoReps
	ifSpecific	ipAdEntBcastAddr	icmpOutTimestamps
		ipAdEntReasmMaxSize	icmpOutTimestampReps
		ipRouteDest	icmpOutAddrMasks
		ipRouteIfIndex	icmpOutAddrMaskReps
		ipRouteMetric1	
		ipRouteMetric2	
		ipRouteMetric3	
		ipRouteMetric4	
		ipRouteNextHop	
		ipRouteType	
		ipRouteProto	
		ipRouteAge	
		ipRouteMask	
		ipRouteMetric5	
		ipRouteInfo	
		ipNetToMediaIfIndex	
		ipNetToMediaPhysAddress	
		ipNetToMediaNetAddress	
		ipNetToMediaType	
		ipRoutingDiscards	

Address Translation MIB	TCP MIB	UDP MIB	SNMP MIB
atIfIndex	tcpRtoAlgorithm	udpInDatagrams	snmpInPkts
atPhysAddress	tcpRtoMin	udpNoPorts	snmpOutPkts
atNetAddress	tcpRtoMax	udpInErrors	snmpInBadVersions
	tcpMaxConn	udpOutDatagrams	snmpInBadCommunityNames
	tcpActiveOpens	udpLocalAddress	snmpInBadCommunityUses
	tcpPassiveOpens	udpLocalPort	snmpInASNParseErrs
	tcpAttemptFails		snmpInTooBigs
	tcpEstabResets		snmpInNoSuchNames
	tcpCurrEstab		snmpInBadValues
	tcpInSegs		snmpInReadOnlys
	tcpOutSegs		snmpInGenErrs
	tcpRetransSegs		snmpInTotalReqVars
	tcpConnState		snmpInTotalSetVars
	tcpConnLocalAddress		snmpInGetRequests
	tcpConnLocalPort		snmpInGetNexts
	tcpConnRemAddress		snmpInSetRequests
	tcpConnRemPort		snmpInGetResponses
	tcpInErrs		snmpInTraps
	tcpOutRsts		snmpOutTooBigs
			snmpOutNoSuchNames
			snmpOutBadValues
			snmpOutGenErrs
			snmpOutGetRequests
			snmpOutGetNexts
			snmpOutSetRequests
			snmpOutGetResponses
			snmpOutTraps
			snmpEnableAuthenTraps
			snmpSilentDrops
			snmpProxyDrops

RFC1317 RS-232-Like Groups

RS-232 MIB	Async Port MIB
rs232Number	rs232AsyncPortIndex
rs232PortIndex	rs232AsyncPortBits
rs232PortType	rs232AsyncPortStopBits
rs232PortInSigNumber	rs232AsyncPortParity
rs232PortOutSigNumber	
rs232PortInSpeed	
rs232PortOutSpeed	

Input Signal MIB	Output Signal MIB
rs232InSigPortIndex	rs232OutSigPortIndex
rs232InSigName	rs232OutSigName
rs232InSigState	rs232OutSigState