Trademarks

CompactLogix, ControlLogix, & PLC-5 are registered trademarks of Rockwell Automation, Inc. EtherNet/IP is a trademark of the ODVA. MicroLogix, RSLogix 500, and SLC are trademarks of Rockwell Automation, Inc. Microsoft, Windows, and Internet Explorer are registered trademarks of Microsoft Corporation. BACnet® is a registered trademark of American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE). All other trademarks and registered trademarks are the property of their holders.

Limited Warranty

Real Time Automation, Inc. warrants that this product is free from defects and functions properly.

EXCEPT AS SPECIFICALLY SET FORTH ABOVE, REAL TIME AUTOMATION, INC. DISCLAIMS ALL OTHER WARRANTIES, BOTH EXPRESSED AND IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR APPLICATION. THIS LIMITED WARRANTY GIVES YOU SPECIFIC LEGAL RIGHTS. YOU MAY ALSO HAVE OTHER RIGHTS, WHICH VARY FROM STATE TO STATE.

The examples and diagrams in this manual are included solely for illustrative purposes. Because of the many variables and requirements associated with any particular application, Real Time Automation, Inc. cannot assume responsibility or liability for actual use based on the examples and diagrams. Except as specifically set forth above, Real Time Automation and its distributors and dealers will in no event be liable for any damages whatsoever, either direct or indirect, including but not limited to loss of business profits, income, or use of data. Some states do not allow exclusion or limitation of incidental or consequential damages; therefore, the limitations set forth in this agreement may not apply to you.

No patent liability is assumed by Real Time Automation with respect to use of information, circuits, equipment, or software described in this manual.

Government End-Users

If this software is acquired by or on behalf of a unit or agency of the United States Government, this provision applies: The software (a) was developed at private expense, is existing computer software, and was not developed with government funds; (b) is a trade secret of Real Time Automation, Inc. for all purposes of the Freedom of Information Act; (c) is “restricted computer software” submitted with restricted rights in accordance with subparagraphs (a) through (d) of the Commercial “Computer Software-Restricted Rights” clause at 52.227-19 and its successors; (d) in all respects is proprietary data belonging solely to Real Time Automation, Inc.; (e) is unpublished and all rights are reserved under copyright laws of the United States. For units of the Department of Defense (DoD), this software is licensed only with “Restricted Rights”: as that term is defined in the DoD Supplement of the Federal Acquisition Regulation 52.227-7013 (c) (1) (ii), rights in Technical Data and Computer Software and its successors, and: Use, duplication, or disclosures is subject to restrictions as set forth in subdivision (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at 52.227-7013. If this software was acquired under GSA schedule, the U.S. Government has agreed to refrain from changing or removing any insignia or lettering from the Software or documentation that is provided or from producing copies of the manual or media. Real Time Automation, Inc.

© 2019 Real Time Automation, Inc. All rights reserved.
# Revision History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>6.3.4</strong></td>
<td>11/13/18</td>
<td>Features Added</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1. Released BC Protocol on the N2E Hardware</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. Released PS Protocol on NNA1 Hardware with updated GSDML file</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3. Released Base Triggering with ETC Protocol Optimization</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bug Fixes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1. WI Protocol handling of STRING data types</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Allow Origins in GET requests, and CR / LF on HTTP 200 responses</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. WI Protocol now has the Allow Origins as part of the HTTP Header</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3. WI Protocol receives a second CR / LF at the end of the HTTP 200 response</td>
</tr>
<tr>
<td><strong>6.4.7</strong></td>
<td>03/05/19</td>
<td>Features Added</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1. New Header Image</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. Released ASCII Protocol on N2E Hardware</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3. Released ES Protocol on N2E Hardware</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4. Updated Profinet Code and new GSDML V2.3 on NNA1 Hardware</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bug Fixes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1. Cleaned up ASCII parser and concatenation functionality</td>
</tr>
<tr>
<td><strong>8.3.9</strong></td>
<td>10/21/2019</td>
<td>Features Added</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1. Released BACnet MS/TP Slave Protocol on N2E Hardware</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. Released Profinet on N2E Hardware</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3. Updated GSMDL file with new icons</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4. Released S7 protocol</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5. Released Wi-Fi Hardware</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bug Fixes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1. Improved functionality of the Network configuration page when going from Independent to Switch Mode.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. Import/Export issues</td>
</tr>
<tr>
<td><strong>8.4.5</strong></td>
<td>11/18/2019</td>
<td>Features Added</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1. Released OPC UA Server (US) Protocol</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. Ability to now Import/Export Template Files with out an FTP session.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bug Fixes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1. Updated Profinet Server (PS) on N34 hardware Platform</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. Updated Wi-Fi software</td>
</tr>
</tbody>
</table>
Overview

The 460MSMM-NNA1 gateway connects a Modbus TCP Client with as many as 32 Modbus RTU Slaves. By following this guide, you will be able to configure the 460MSMM-NNA1 gateway.

For further customization and advanced use, please reference the appendices located on the CD or online at: [http://www.rtautomation.com/product/460-gateway-support/](http://www.rtautomation.com/product/460-gateway-support/).

If at any time you need further assistance do not hesitate to call Real Time Automation support.

Support Hours are Monday-Friday 8am-5pm CST

Toll free: 1-800-249-1612
Email: support@rtautomation.com
Hardware Platforms

The 460 Product Line supports a number of different hardware platforms. There are differences in how they are powered, what serial settings are supported, and some diagnostic features supported (such as LEDs). For these sections, be sure to identify the hardware platform you are using.

To find which hardware platform you are using:

1) Look on the front or back label of the unit for the part number.
2) On the webpage inside the gateway, navigate to the dropdown menu under Other and select Utilities. Click the Listing of Revisions button. The full part number is displayed here.

Once you have the full part number, the platform will be the number following the “-N”:

![Diagram of part number 460 P1P2 -NXXX with arrows pointing down to Product and Platform]
Hardware – NNA1

Powering the Gateway
- An 8-24 VDC power source to the gateway, Red Wire = (+) Black Wire = (-).
  a. The unit draws 175mA @ 12V.
Port Configuration

The Port Configuration page is where you set port specific parameters. These settings must match the settings of the device(s) that you are connecting to.

Only 1 mode can be configured for this hardware. Below are the wiring pinouts for each mode.

When you have completed your port configuration, click the Save Parameters button.

RS232 pinouts:

RS485 pinouts:
Mounting with a DIN Rail

Installing
Follow these steps to install your interface converter.

1) Mount your DIN Rail.
2) Hook the bottom mounting flange under the DIN Rail.
3) While pressing the 460MSMM-NNA1 against the rail, press up to engage the spring loaded lower clip and rotate the unit parallel to the DIN Rail.
4) Release upward pressure.

Removing
Follow these steps to remove your interface converter.

1) Press up on unit to engage the spring loaded lower clip.
2) Swing top of the unit away from DIN Rail.
Accessing the Main Page

The following steps will help you access the browser based configuration of the gateway. By default, DHCP is enabled. If the gateway fails to obtain an IP address over DHCP it will Auto IP with 169.254.X.Y. For more information on your Operating system network setting refer to the Access Browser Configuration Doc on the CD or download from our support web site.

1) Insert the provided CD-ROM into a computer also on the network.

2) Run the IPSsetup.exe program from the CD-ROM.

3) Find unit under “Select a Unit”.
   a. Change Gateway’s IP address to match that of your PC if DHCP has failed.
      i. You will know DHCP has failed if the gateway’s IP address is AutoIP at 169.254.X.Y.
      ii. If successful, it will say DHCP’d at ex: 192.168.0.100 or however your DHCP Client is set up.
   b. If you do not see the gateway in this tool, then your PC is most likely set up as a static IP.
      i. Change your PC’s network settings to be DHCP. If DHCP fails, then it will change to be on the 169.254.x.y network.
      ii. Relaunch the IP Setup tool to see if gateway can be discovered now.

4) Click Launch Webpage. The Main page should appear.

Default setting is set to DHCP. If DHCP fails, default IP Address is 169.254.x.y
Error: Main Page Does Not Launch
If the Main Page does not launch, please verify the following:

1) Check that the PC is set for a valid IP Address
   a. Open a MS-DOS Command Prompt
   b. Type “ipconfig” and press enter
   c. Note the PC’s IP Address, Subnet, and Default Gateway
2) The gateway must be on the same Network/Subnet as the PC whether it’s setup for DHCP or Static.

Once you have both devices on the same network, you should be able to ping the gateway using a MS-DOS Command Prompt.

The Screenshot above shows a gateway that is currently set to a static IP Address of 192.168.0.100.

If you are able to successfully ping your gateway, open a browser and try to view the main page of the gateway by entering the IP Address of the gateway as the URL.
Comitting Changes to the Settings

- All changes made to the settings of the gateway in Configuration Mode will not take effect until the gateway is restarted via the webpage. Changes will not be stored if the gateway’s power is removed prior to a reboot.

- **NOTE:** The gateway does not need to be restarted after every change. Multiple changes can be made before a restart, but they will not be committed until the gateway is restarted.

- When all desired changes have been made, press the **Restart Now** button.

- The webpage will redirect to our rebooting page shown below:

![Rebooting Page](image)

- The reboot can take up to 20 seconds.
  - If the IP address has not been modified, the gateway will automatically redirect to the main page.
  - If the IP address was modified, a message will appear at the top of the page to instruct the user to manually open a new webpage at that new IP.
Main Page

The main page is where important information about your gateway and its connections are displayed.

Mode (orange box below):

Running Mode:
- Protocol communications are enabled
- Configuration cannot be changed during Running Mode. If changes are needed, click the Configuration Mode button shown in the green box below

Configuring Mode:
- Protocol communication is stopped and no data is transmitted
- Configuration is allowed

Navigation (green box below):

You can easily switch between modes and navigate between pages (Configuration, Diagnostics, and Other pages) using the buttons on the left hand side.
Device Configuration

The device configuration area is where you assign the device description parameter. Changes can only be made when the gateway is in Configuration Mode.

Once you are done configuring the Description, click the Save Parameters button.
Network Configuration

The network configuration area is where you assign the IP address and other network parameters. Changes can only be made when the gateway is in Configuration Mode.

Once you are done configuring the Network Settings, click the Save Parameters button.

If you are changing the IP Address of the gateway, the change will not take effect until the unit has been rebooted. After reboot, you must enter the new IP Address into the URL.

It is recommended to leave the DNS Gateway set to 0.0.0.0 and the Ethernet Link as Auto-Negotiate. If configuring the gateway to use E-mail, the DNS Gateway must be set.
Modbus TCP/IP Server Configuration

Click the Modbus TCP/IP Server button to access the configuration page.

1) Select which Network Interface to use for this Modbus TCP/IP connection.

2) Enter a Device Label to identify the device within the gateway.

3) TCP Port: Enter a TCP Port for the Modbus TCP Server to listen on. Default for Modbus TCP/IP is 502.

4) Inactivity Timeout: Amount of time the gateway will wait for a read/write request before issuing a timeout.

5) To enable data swapping, select the required Swap Indicator. If the bytes appear in the wrong order, enable swapping to change the data. This swapping does NOT change Coils and their ordering inside the Bit Pack.

6) Bit Pack: Select the formatting of the Coil Status/Input Status. Automap will use this packing size to map coils to/from the other protocol. The bit pack selection here should match that of the other protocol. The starting address is considered Bit 0 and is the low-order bit.
Modbus TCP/IP Server Configuration - Data Groups

The bottom area of the Modbus TCP/IP Server Configuration page lets you configure up to 100 Data Groups for both the Read/Write.

There are three ways to configure this protocol:

- Auto-Configure Group by Device (Default)
- Auto-Configure Group by Data Type
- Manual Mode

**NOTE:** You may go back and forth between modes, but when reverting from Manual Mode to either of the two Auto-Configure Modes, all changes made in Manual Mode will be discarded.
Modbus TCP/IP Server Data Group Configuration: Auto-Configure

While in either of the two Auto-Configure Modes, the # of Data Groups and the actual Data Groups themselves cannot be edited. Auto-Configure Mode looks at the other protocol and then configures the Data Groups to match. The Data Formats will be defined after the other protocol is configured.

The data will be configured according to the following rules:

- Any Coils, 8 Bit Signed/Unsigned, or 1/8/16/32 Bit Binary Packs data will be mapped as **0x Coil Status**.

- Any 16 Bit Signed/Unsigned data will be mapped as **4x Hold Reg 16 Bit Int or 16 Bit Uint**, matching signs whenever possible.

- Any 32 Bit Signed/Unsigned data will be mapped as **4x Hold Reg 32 Bit Int or 32 Bit Uint**, matching signs whenever possible.

- Any 64 Bit Signed/Unsigned data will be mapped as **4x Hold Reg 64 Bit Int or 64 Bit Uint**, matching signs whenever possible.

- Any 32 Bit Float will be mapped as **4x Hold Reg 32 Bit Float**.

- Any 64 Bit Float will be mapped as **4x Hold Reg 64 Bit Float**.

- Any String Data Types will be mapped as **4x Hold Reg String**.

- The Read or Write direction depends on whether it is configured as a Read or Write on the other protocol.

- If the other protocol exceeds the number of Data Groups supported, then nothing will be mapped. You will see the # of Data Groups remain at 0 and the main page will display the following error:

  ![Error Message]

  - To fix this error, simply decrease the amount of data you configured on the other protocol so that the max number of Data Groups is not exceeded or call customer support to increase the limits.

To add additional or edit existing Data Groups you will need to go into Manual Configure Mode.
Auto-Configure Group by Device vs. Auto-Configure Group by Data Type

There are two different methods for Auto-Configure: Group by Device or Group by Data Type.

There are a couple of rules to keep in mind when using Auto-Configure Mode:

1) If the other protocol inside the gateway is a Server, Slave, or Adapter protocol, then there are no differences between the Auto-Configure Modes.
2) Group by Data Type is how previous versions of the 460 gateway Auto-Configured. Versions with build dates prior to May 21 2016 had the default set to this grouping method.
3) Both modes use the rules outlined on the previous page for setting up data.

Group by Device (Default Method)

Group by Device goes through the other protocol on the gateway and auto-configures the data groups on the Modbus TCP Server for all the data points on the other protocol’s first device. After it finishes with the first device, it will auto-configure all the points for the second device (if one is configured), and so on.

The data in this method is not optimized- there could potentially be a lot of wasted/unused data space, but it will be organized more logically from the Master/Client’s point of view.

Group by Data Type

Group by Data Type goes through the other protocol on the gateway and auto-configures the data groups on the Modbus TCP Server for all the data points within the other protocol.

Another way to view this option is to say that the data points allocated are packed together so there is very little wasted data space. The data is packed or optimized.

Example: Protocol A is a Master/Client protocol that has 2 devices with the same setup:

Device_1 has 1 integer scan line, 1 float scan line, 1 integer scan line- each for 1 point of data
Device_2 has 1 integer scan line, 1 float scan line, 1 integer scan line- each for 1 point of data

Protocol B is a Server/Slave/Adapter protocol that can be mapped as follows:

Group by Device - Protocol B will have 4 scan lines that will look like the following:

- Scan Line 1 => Type Integer, length of 2
- Scan Line 2 => Type Float, length of 1
- Scan Line 3 => Type Integer, length of 2
- Scan Line 4 => Type Float, length of 1

Group by Data Type - Protocol B will have 2 scan lines that will look like the following:

- Scan Line 1 => Type Integer, length of 4
- Scan Line 2 => Type Float, length of 2
Modbus TCP/IP Server Data Group Configuration: Manual Mode

1) To transition from either of the two Auto-Configure Modes to Manual Configure Mode, click the dropdown at the top of the Modbus TCP/IP Server Configuration page and select Manual Configure.
   a. When prompted, click **OK** to confirm mode change or **Cancel** to remain in Auto-Configure Mode.

![Message from webpage]

b. Once OK is clicked, there are two options on how to proceed.

![Message from webpage]

i. To keep the data groups that are already configured, press **OK**.
   - You would want this option if you are adding additional data groups or you want to modify the data group(s) that already exist.

ii. To delete the data groups that are already there and start over, press **Cancel**.

2) Enter the number of Read Data Groups and Write Data groups.

3) Click the **Generate Data Groups** button to have the read and write data groups auto-generate for you. You may manually configure the read and write data groups after they have been generated.

4) Follow these steps to manually configure Read Data Groups.
   a. Select **View Read Data Groups** if not already selected.
   b. Select a **Point Type** for each Scan Line. Options include: Coil Status, Input Status, Input Registers, and Holding Registers.

   **Note:** Input/Holding Registers have a data type associated with them.

   i. String Point Type- If the mating protocol supports strings, you may select string as a point type in Modbus. With this point type, 2 characters will be packed into a single register and the first register will be set aside for the length.

   EX: 4x Hold Reg (String) with a Starting Address of 1 for a length of 5 Registers
This means that Register 1 will hold the length of the string and Registers 2-5 will hold the string contents. This string can contain a max of 8 characters.

c. Enter a **Starting Address** (1-based).

d. Enter the **# of Points** to read. This will allocate the number of the data type selected.

5) Follow these steps to manually configure Write Data Groups.

a. Select **View Write Data Groups** if not already selected.

b. Select a **Point Type** to configure. Options include: Coil Status and Holding Registers. **Note:** Holding Registers have a data type associated with them.

i. **String Point Type**- If the mating protocol supports strings, you may select string as a point type in Modbus. With this point type, 2 characters will be packed into a single register and the first register will be set aside for the length.

**EX:** 4x Hold Reg (String) with a Starting Address of 1 for a length of 5 Registers

This means that Register 1 will hold the length of the string and Registers 2-5 will hold the string contents. This string can contain a max of 8 characters.

c. Enter a **Starting Address** (1-based).

d. Enter the **# of Points** to write. This will allocate the number of the data type selected.
Modbus RTU Master Configuration

Click the Modbus RTU Master button to access the configuration page.

1) **Serial Port**: Select which serial port is being used for communication. This port must be configured on the Port Configuration page. If it has not yet been configured, it will display *Disabled* after the Port descriptions in this dropdown.

   ![Serial Port Dropdown]

2) **Delay Between Messages**: Enter the length of time to delay between read and write scan line requests (ms).

3) **Response Timeout**: Enter the amount of time the gateway should wait before a timeout is issued for a read/write request (ms).

4) **Dependency Protocol**: If enabled, Modbus RTU Master communication will stop if communication to the selected protocol is lost.

![Modbus RTU Master Configuration Table]
Modbus RTU Master Device Configuration

The bottom area of the Modbus RTU Master Configuration page lets you configure up to 32 external Modbus RTU Slave devices.

1) To add additional Slave connections, click the -Select- dropdown under Modbus RTU Master Device List and select Add Generic Slave option.

Modbus RTU Master Device List

- To remove a device, navigate to the slave to delete using the << and >> buttons and click the Delete Slave button.

- To create a new slave with the same parameters already configured from another slave, click the -Select- dropdown and select the Add from Modbus RTU X option (where X represents the slave you wish to copy parameters from). Once created, you can make any additional changes needed to that new slave.

2) The Enable check box should be selected for the device.

3) Enter a Device Label to identify the device within the gateway.

4) Enter a unique Modbus RTU Slave Address for the device on the network.


6) Enable 0-Based Addressing: Check ONLY if the Slave you are connecting to begins their register numbering at 0 OR they specify that their device addresses are 0-based.
7) **Bit Pack**: Select the formatting of the Coil Status/Input Status. Automap will use this packing size to map coils to/from the other protocol. The bit pack selection here should match that of the other protocol. The starting address is considered Bit 0 and is the low-order bit.

8) **Enable Modbus ASCII**: Only select this if your Modbus device is also using ASCII messaging. By default, the device will use Modbus RTU.

9) To enable data swapping, select the required **Swap Indicator**. If the bytes appear in the wrong order, enable swapping to change the data. This swapping does *NOT* change Coils and their ordering inside the Bit Pack.

10) Enter the number of Read Scan Lines and Write Scan Lines.

11) Click the **Generate Scan Lines** button to have the read and write scan lines auto-generate for you. You may manually configure the read and write scan lines after they have been generated.
Configuring Read Scan Lines

Follow these steps to manually configure Read Scan Lines.

1) Select View Read Scan Lines if not already selected.

2) Select a Point Type for each Scan Line. Options include: Coil Status, Input Status, Input Registers, and Holding Registers. **Note**: Input/Holding Registers also have a data type associated with them.
   
   a. String Point Type- If the mating protocol supports strings, you may select string as a point type in Modbus. With this point type, 2 characters will be packed into a single register and the first register will be set aside for the length.

   **EX**: 4x Hold Reg (String) with a Starting Address of 1 for a length of 5 Registers

   This means that Register 1 will hold the length of the string and Registers 2-5 will hold the string contents. This string can contain a max of 8 characters.

3) Enter a Starting Address (This will be 1 based if Enable 0-Base Addressing is unchecked, or 0 based if checked).

4) Enter the # of consecutive points to read for that point/data type. See the Scan Line Data Limit section at the bottom of the webpage for max values in a scan line.

<table>
<thead>
<tr>
<th>Line #</th>
<th>Point Type</th>
<th>Starting Address</th>
<th># of Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0x coil status</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

*View Read Scan Lines*  
*View Write Scan Lines*  

---

Real Time Automation, Inc.  
26  
1-800-249-1612
Configuring Write Scan Lines
Follow these steps to manually configure Write Scan Lines.

1) Select View Write Scan Lines if not already selected.

2) Select a Point Type to configure. Options include: Coil Status and Holding Registers. **Note:** Holding Registers also have a data type associated with them.
   a. String Point Type- If the mating protocol supports strings, you may select string as a point type in Modbus. With this point type, 2 characters will be packed into a single register and the first register will be set aside for the length.

   **EX:** 4x Hold Reg (String) with a Starting Address of 1 for a length of 5 Registers
   This means that Register 1 will hold the length of the string and Registers 2-5 will hold the string contents. This string can contain a max of 8 characters.

3) Enter a Starting Address (This will be 1 based if **Enable 0-Base Addressing** is unchecked, or 0 based if checked).

4) Enter the # of consecutive points to write for that point/data type. See the **Scan Line Data Limit** section at the bottom of the webpage for max values in a scan line.
Configuring Read and Write Scan Lines (cont.)
If you are configuring more than 25 scan lines click << or >> to navigate to the next group of 25. When finished, click the Save Parameters button.

Below is the Scan Line Data Limit for each Point Type and the max Length Range associated with them.

**Note:** If the first address of the Modbus RTU Slave device starts at 0 (Register/Coil starting addresses can be found in the Modbus RTU Slave’s documentation), be sure to check the Enable 0-Base Addressing box in the gateway to ensure proper communication. If improperly configured, expected addressing may be off by +/- 1.

<table>
<thead>
<tr>
<th>Point Type</th>
<th>Length Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coil Status</td>
<td>512</td>
</tr>
<tr>
<td>Input Status</td>
<td>512</td>
</tr>
<tr>
<td>Input Register (16 Bit Int/Uint)</td>
<td>125</td>
</tr>
<tr>
<td>Input Register (32 Bit Int/Uint/Float)</td>
<td>62</td>
</tr>
<tr>
<td>Input Register (64 Bit Int/Uint/Float)</td>
<td>31</td>
</tr>
<tr>
<td>Input Register (String - 2 char/reg)</td>
<td>125</td>
</tr>
<tr>
<td>Holding Register (16 Bit Int/Uint)</td>
<td>125</td>
</tr>
<tr>
<td>Holding Register (32 Bit Int/Uint/Float)</td>
<td>62</td>
</tr>
<tr>
<td>Holding Register (64 Bit Int/Uint/Float)</td>
<td>31</td>
</tr>
<tr>
<td>Holding Register (String - 2 char/reg)</td>
<td>125</td>
</tr>
</tbody>
</table>
Mapping - Transferring Data Between Devices

There are 5 ways to move data from one protocol to the other. You can combine any of the following options to customize your gateway as needed.

Option 1 – Data Auto-Configure Mappings: The gateway will automatically take the data type (excluding strings) from one protocol and look for the same data type defined in the other protocol. If there isn’t a matching data type, the gateway will map the data to the largest available data type. See Data Auto-Configure section for more details.

Option 2 – String Auto-Configure: The gateway will automatically take the string data type from one protocol and map it into the other. See String Auto-Configure section for more details.

Option 3 – Manual Configure Mappings: If you don’t want to use the Auto-Configure Mappings function, you must use the manual mapping feature to configure translations.

Option 4 – Manipulation/Scaling: You can customize your data by using math operations, scaling, or bit manipulation. See Data Mapping-Explanation section for more details.

Option 5 – Move Diagnostic Information: You can manually move diagnostic information from the gateway to either protocol. Diagnostic information is not mapped in Auto-Configure Mappings Mode. See Diagnostic Info section for more details.
Display Mapping and Values

The Display Data and Display String pages are where you can view the actual data for each mapping that is set up.

Display Data

Click the Display Data button to view how the data is mapped and what the values of each mapping are. Here you will see how each data point (excluding strings) is mapped. To view, select the device from the dropdown menu and click View to generate the information regarding that device. Then select either the Protocol 1 to Protocol 2 or Protocol 2 to Protocol 1 button, correlating to the direction you wish to see the data.

This page is very useful when verifying that all data is mapped somehow from one protocol to another. If a data point is not mapped, it will display on this page in a yellow highlighted box.

In the above example, we see the following:

- Modbus 400001 from Slave 1 is being mapped to AI1 on BACnet
- Nothing is being moved from Modbus to AI2 on BACnet
- Modbus 400030 from Slave 1 is being mapped to AI3 on BACnet

**NOTE:** If a data point is mapped twice, only the first instance of it will show here. EX: If Modbus 400001 & 400040 from Slave 1 are both mapped to AI1, only 400001 will show as being mapped to AI1.
If there are values of “-” on this page, it indicates that the source has not yet been validated and no data is being sent to the destination.

The example below reflects the Modbus to PLC flow of data. The Modbus (right side) is the source and the PLC (left side) is the destination.

- The 460 gateway has received valid responses from Modbus registers 400001-400005 and therefore can pass the data on to the PLC tag called MC2PLC_INT.

- The 460 gateway has NOT received valid responses from Modbus register 400011 & 400012. As a result, the data cannot be passed to the PLC tag ETC01_GNO_INT2 and indicates so by using “-” in the value column of the table.

To view the actual data mappings, click the Edit Mapping button. For more details, see the Data Mapping-Explanation section.

To view the data mappings purely as text, click the View as Text button. For more details, see the View Data Mapping as Text section.
Display String

Click the **Display String** button to view how the string data types are mapped and what the values of each string are. Here you will see how each string from each protocol is mapped to the other. To view, select the source or destination group and the String from the dropdown menu to generate the information regarding that device. The string data will be displayed in both hex and ASCII.

<table>
<thead>
<tr>
<th>Display String</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Select a Group</strong></td>
<td><strong>Src: MC02 400001</strong> and a <strong>String</strong> <strong>400001</strong> (0 bytes)</td>
</tr>
</tbody>
</table>

If there are values of “Data Not Valid” on this page, it indicates that the source has not been validated yet and no data is being sent to the destination.

In the example below, this page reflects the Modbus to PLC flow of data. Since the Destination “Dst: ETC01 ETC01_G2N0_STRING” displays “Data Not Valid”, it can be assumed that the source field has not yet been validated.

<table>
<thead>
<tr>
<th>Display String</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Select a Group</strong></td>
<td><strong>Dst: ETC01 ETC01_G2N0_STRING</strong> and a <strong>String</strong> <strong>ETC01_G2N0_STRING</strong> (0 bytes)</td>
</tr>
</tbody>
</table>

Data Not Valid!

To view the string mappings, click the **Edit Mapping** button. For more details see the String Mapping-Explanation section.

To view the string mappings purely as text, click the **View as Text** button. For more details see the View String Mapping as Text section.
Data and String Mapping – Auto-Configure

The Auto-Configure function looks at both of the protocols and will map the data between the two protocols as best as it can so that all data is mapped. Inputs of like data types will map to outputs of the other protocols like data types first. If a matching data type cannot be found, then the largest available data type will be used. Only when there is no other option is data truncated and mapped into a smaller data type.

If the Auto-Configure function does not map the data as you want or you want to add/modify the mappings, you may do so by going into Manual Configure mode.

The following are examples of the Auto-Configure function.

1) This example shows a common valid setup.

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>8-bit Sint</td>
<td>8-bit Sint</td>
</tr>
<tr>
<td>16-bit Int</td>
<td>16-bit Int</td>
</tr>
</tbody>
</table>

   a. Both Source values were able to be mapped to a corresponding Destination value.

2) This example shows how Auto-Configure will make its best guess.

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>8-bit Sint</td>
<td>8-bit Sint</td>
</tr>
<tr>
<td>16-bit Int</td>
<td>16-bit Int</td>
</tr>
<tr>
<td>32-bit Uint</td>
<td>32-bit Uint</td>
</tr>
<tr>
<td>32-bit Float</td>
<td>32-bit Uint</td>
</tr>
</tbody>
</table>

   a. The 32-bit Float from the Source location could not find a matching Destination data-type. After all other like data types were mapped, the only data type available was the 2\textsuperscript{nd} 32-bit Uint data type. Auto-Configure was completed even though the data in the Float will be truncated.
Data Mapping – Explanation

Below are the different parts that can be modified to make up a data mapping.

1) **Enable (red box above):** Check to enable mapping. If not checked, this mapping is skipped.

2) **Source Field (yellow box above):**
   a. Group - Select the data group you set up in the protocol config to use for this mapping.
   b. Start - This is the starting point for this mapping.
   c. End - This is the final point to be included for this mapping.

3) **Manipulation Area (green box above):**
   a. Enable the Data Manipulation. This can be enabled for any mapping.
   b. Click **Add Math Operation** for each operation needed. Up to 3 are allowed unless you are using the Scale, Set Bit, or Invert Bit functions. If using Scale, Set Bit, or Invert Bit, then only 1 operation is allowed.
   c. Select the Operation(s) to perform.
      i. Math Operations are performed in the order they are selected.
      ii. If more than one point is selected on the source, the Math Operations will be performed on every point.
   d. Enter the value(s) for the operation.

   **Example of Add (similar for Subtract, Multiple, Divide, and MOD).** This will add a value of 10 to the source field before it is written to the destination field.

   **Example of Scale.** This will scale the source values from 1-10 into 1-100 for the destination.

   **Example of Set Bit (similar to Invert Bit).** This will take the value of the 0th source bit and copy it into the value of the 5th destination bit.

4) **Destination Field (blue box above):**
   a. Group - Select the data group you set up in the protocol config to use for this mapping.
   b. Start - This is the starting point for where the data is being stored.
   c. End - The End point is derived from the length of the source and cannot be modified.
Data Mapping – Adding Diagnostic Information

Data Mapping offers 5 different types of information in addition to any scan lines specified for each protocol.

**IMPORTANT NOTE:** Only add Diagnostic Information **AFTER** both sides of the gateway have been configured. If changes to either protocol are made after diagnostic information has been added to the mapping table, it is necessary to verify all mappings. Remapping may be necessary.

1) Temporary Ram (Int64)
   a. This offers five levels of 64bit Integer space to assist in multiple stages of math operations. For example, you may wish to scale and then add 5. You can set up a single translation to scale with the destination as the temporary ram. Then another translation to add 5 with the source as the temporary ram.
   b. The gateway will automatically convert the Source to fit the Destination, so there is no need for Int 8, 16, 32 since the 64 may be used for any case.

   ![Mapping 1](image)

   In this example, _Ram0 is scaled into Ram1_. _Ram1 is then increased by 5 and stored into Ram2_. _Ram0 and Ram2 could be considered a source or destination group._

2) Temporary Ram (Double)
   a. This is similar to the Temporary Ram (Int 64), except manipulations will be conducted against the 64bit floating point to allow for large data.

3) Ticks Per Second
   a. The gateway operates at 200 ticks per second. This equates to one tick every 5ms. Thus, mapping this to a destination will give easy confirmation of data flow without involving one of the two protocols.

4) XY_NetBmpStat
   a. If a protocol is a Client/Master, there is a Network Bitmap Status that is provided. Since a Client/Master may be trying to communicate with multiple devices on the network, it may be beneficial to know if a Server/Slave device is down. By using this Network Bitmap Status you can expose the connection statuses of individual devices.
   b. 0x00000002 shows that only device 2 is connected
   c. 0x00000003 shows that only devices 1 and 2 are connected
   d. 0x00000004 shows that only device 3 is connected
5) Status_XY

   a. There are two Statuses provided, one for each protocol. This gives access to the overall
      status of that Protocol. Each Bit has its own meaning as follows:

   **Common Status:** 0x000000FF (bit 0-7) 1st byte

<table>
<thead>
<tr>
<th>Hex</th>
<th>Bit Position</th>
<th>Decimal</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>0</td>
<td>0</td>
<td>if we are a Slave/Server</td>
</tr>
<tr>
<td>0x01</td>
<td>0</td>
<td>1</td>
<td>if we are a Master/Client</td>
</tr>
<tr>
<td>0x02</td>
<td>1</td>
<td>2</td>
<td>connected (0 not connected)</td>
</tr>
<tr>
<td>0x04</td>
<td>2</td>
<td>4</td>
<td>first time scan</td>
</tr>
<tr>
<td>0x08</td>
<td>3</td>
<td>8</td>
<td>idle (usually added to connected)</td>
</tr>
<tr>
<td>0x10</td>
<td>4</td>
<td>16</td>
<td>running (usually added to connected)</td>
</tr>
<tr>
<td>0x20</td>
<td>5</td>
<td>32</td>
<td>bit not used</td>
</tr>
<tr>
<td>0x40</td>
<td>6</td>
<td>64</td>
<td>recoverable fault</td>
</tr>
<tr>
<td>0x80</td>
<td>7</td>
<td>128</td>
<td>nonrecoverable fault</td>
</tr>
</tbody>
</table>

   For this example the ETC Status is mapped to a PLC tag called PLC_Status

   PLC | 460ETCMC | Modbus TCP/IP
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Value (Hex)</td>
<td>Manipulation</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
<td>--------------</td>
</tr>
<tr>
<td>PLC_Status</td>
<td>0x00000013</td>
<td></td>
</tr>
</tbody>
</table>

   **Example:** ETC Status is 0x00000013 (19 decimal), here is the break down

<table>
<thead>
<tr>
<th>Hex</th>
<th>Bit</th>
<th>Decimal</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x01</td>
<td>0(on)</td>
<td>1</td>
<td>if we are a Master/Client</td>
</tr>
<tr>
<td>0x02</td>
<td>1(on)</td>
<td>2</td>
<td>connected (0 not connected)</td>
</tr>
<tr>
<td>0x10</td>
<td>4(on)</td>
<td>16</td>
<td>running (usually added to connected)</td>
</tr>
<tr>
<td>Total:</td>
<td>19</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

   **External Faults:** 0x000FF00 (bit 8-15) 2nd byte

<table>
<thead>
<tr>
<th>Hex</th>
<th>Bit Position</th>
<th>Decimal</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>8</td>
<td>0</td>
<td>local control</td>
</tr>
<tr>
<td>0x01</td>
<td>8</td>
<td>256</td>
<td>remotely idle</td>
</tr>
<tr>
<td>0x02</td>
<td>9</td>
<td>512</td>
<td>remotely faulted</td>
</tr>
<tr>
<td>0x04</td>
<td>10</td>
<td>1,024</td>
<td>idle due to dependency</td>
</tr>
<tr>
<td>0x08</td>
<td>11</td>
<td>2,048</td>
<td>faulted due to dependency</td>
</tr>
</tbody>
</table>

   **Recoverable Faults:** 0x00FF0000 (bit 16-23) 3rd byte

<table>
<thead>
<tr>
<th>Hex</th>
<th>Bit Position</th>
<th>Decimal</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x01</td>
<td>16</td>
<td>65,536</td>
<td>recoverable fault - timed out</td>
</tr>
<tr>
<td>0x02</td>
<td>17</td>
<td>131,072</td>
<td>recoverable fault - Slave err</td>
</tr>
</tbody>
</table>
Non-Recoverable Faults  0xFF000000 (bit 24-31) 4th byte

<table>
<thead>
<tr>
<th>Hex</th>
<th>Bit Position</th>
<th>Decimal</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x01</td>
<td>24</td>
<td>16,777,216</td>
<td>nonrecoverable fault – task fatal err</td>
</tr>
<tr>
<td>0x02</td>
<td>25</td>
<td>33,554,432</td>
<td>nonrecoverable fault – config missing</td>
</tr>
<tr>
<td>0x04</td>
<td>26</td>
<td>67,108,864</td>
<td>nonrecoverable fault – bad hardware port</td>
</tr>
<tr>
<td>0x08</td>
<td>27</td>
<td>134,217,728</td>
<td>nonrecoverable fault – config err</td>
</tr>
<tr>
<td>0x10</td>
<td>28</td>
<td>268,435,456</td>
<td>Configuration Mode</td>
</tr>
<tr>
<td>0x20</td>
<td>29</td>
<td>536,870,912</td>
<td>No Ethernet Cable Plugged In</td>
</tr>
</tbody>
</table>

For this example the MC Status is mapped to a PLC tag called MC_Status

Example: MC Status is 0x00010041 (65601 decimal), here is the break down, we know that bytes 1 and 3 are being used, so here is the break down,

**Common Status:**

<table>
<thead>
<tr>
<th>Hex</th>
<th>Bit</th>
<th>Decimal</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x01</td>
<td>0(on)</td>
<td>1</td>
<td>if we are a Master/Client</td>
</tr>
<tr>
<td>0x40</td>
<td>6(on)</td>
<td>64</td>
<td>recoverable fault</td>
</tr>
</tbody>
</table>

**Recoverable Faults:**

<table>
<thead>
<tr>
<th>Hex</th>
<th>Bit</th>
<th>Decimal</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x01</td>
<td>16</td>
<td>65,536</td>
<td>recoverable fault – timed</td>
</tr>
</tbody>
</table>

Total: 0x010041  65,601
String Mapping – Explanation

Below are the different parts that can be modified to make up a string mapping.

String data types can only be mapped to other string data types. There is no manipulation that can be done on the string.

1) Enable (red box above): Check to enable mapping. If not checked, this mapping is skipped.
2) Source Field (yellow box above):
   a. Group - Select the string data group you set up in the protocol config to use for this mapping.
   b. String - This is the string used for this mapping.
3) Destination Field (green box above):
   a. Group - Select the string data group you set up in the protocol config to use for this mapping.
   b. String - This is the string where the data is being stored.
Mapping – Auto-Configure Mode to Manual Configure Mode

To transition from Auto-Configure Mapping Mode to Manual Configure Mode, click the dropdown at the top of the Mapping Configuration page and select Manual Configure.

After you click this button, you will be prompted to confirm if this is really what you want to do.

Click **OK** to proceed to Manual Configure Mode or click **Cancel** to remain in Auto-Configure Mappings Mode.

Once OK is clicked, there are 2 options on how to proceed from here.

1) To keep the mappings that are already configured press **OK**.
   a. You would want this option if you are adding additional mappings or you want to modify the mapping(s) that already exist.

2) To delete the mappings that are already there and start over press **Cancel**.

To modify the number of mappings, enter a number next to **# of Mappings to Configure** and click the **Set Max # of Mappings** button. You can always add more mappings if needed.
Mapping – Manual Configure Mode to Auto-Configure Mode

To transition from Manual Configure Mode to Auto-Configure Mapping Mode, click the dropdown menu at the top of the Mapping Configuration page and select Auto-Configure Mappings.

Click **OK** to proceed to delete all current mappings and go back to Auto-Configure Mappings Mode. Click **Cancel** to keep all mappings and remain in Manual Configure Mode.

**NOTE:** Once you revert back to Auto-Configure Mapping Mode there is no way to recover the mappings you lost. Any mappings you previously have added will be deleted as well.
View as Text

Data Mapping
The View as Text page displays the point to point mapping(s) you set up in the Data Mapping section. This will also display any manipulation(s) that are configured.

Each line on this page will read as follows:

```
Mapping number: source point  Len: Number of points mapped -> manipulation (if blank then no manipulation) -> destination point
```

If you are looking for a specific point to see if it is mapped, you can do a find in this text box for your point in question. Example: you defined 20 Registers starting at register 1 and want to see if 400011 is mapped. If it is not in this text box then it is not mapped and no data will be transferred.

This is the text display for the example shown under the Data Mapping - Adding Diagnostic Information section.

| Mapping 1: | Temporary R00 Len: 1 -> 1:10 Scale to 1:100 -> | Temporary R01 |
| Mapping 2: | Temporary R01 Len: 1 -> Add 5 -> | Temporary R02 |

String Mapping
The View as Text page displays the string mapping(s) you set up in the String Mapping section.

Each line on this page will read as follows:

```
Mapping number: source point  -> Copy -> destination point
```

If you are looking for a specific point to see if it is mapped, you can do a find in this text box for your point in question. Example: you defined 20 String Tags in the PLC and want to see if “Test_String” in the Logix PLC is mapped. If it is not in this text box, then it is not mapped, and no data will be transferred.

<table>
<thead>
<tr>
<th>String Mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mapping 1: Logix Test_String -&gt; Copy -&gt; MC02 40001</td>
</tr>
</tbody>
</table>

Real Time Automation, Inc. 41 1-800-249-1612
Security Configuration

To setup security on the 460 gateway, navigate to Other->Security Configuration. You can configure Security for 3 administrators, 5 users, and 1 guest.

**THIS IS NOT A TOTAL SECURITY FEATURE**

The security feature offers a way to password protect access to diagnostics and configuration on the network. The security feature does not protect against “Air Gap” threats. If the gateway can be physically accessed, security can be reset. All security can be disabled if physical contact can be made. From the login page, click the Reset Password button twice. You will be forced to do a hard reboot (power down) on the gateway within 15 minutes of clicking the button. This process should be used in the event a password is forgotten.

**Note:** Only Admins have configuration access to all web pages.

1) Log Out Timer: The system will automatically log inactive users off after this period of time. **NOTE:** A time of 0 means that the user will not be automatically logged off. Instead, they must manually click the Logout button.
2) Username: Enter a username, max of 32 characters.
3) Password: Enter a password for the username, max of 32 characters, case sensitive.
   a. Re-enter the Password
4) E-mail: In case the password was forgotten, a user can have their password e-mailed to them if e-mail was configured.
5) Hint: A helpful reminder of what the password is.
Security Configuration-Security Levels

Each webpage in the gateway can have a separate security level associated with it for each user.

Security Levels:

1) **Full Access**: Capability to view and configure a web page.
2) **View Access**: Capability to view a web page, but cannot configure parameters.
3) **No Access**: No capability of viewing the web page and page will be removed from Navigation.
### Security - Log In

**Username**: Name of the user to login.

**Password**: Password of the user to login.

**Log In**: If login is successful, the user will be redirected to the Main Page.

**Send Password to Email**: Sends the specified User’s Password to the email configured for that user.

**Display Hint**: Displays the hint specified for the User if one was set up.
**Reset Password:** This is used to reset security settings. Confirm reset password must be selected to confirm this action. Once confirmed, there is a 15 minute window to do a hard reset of the gateway by physically removing and restoring power from the gateway. Once power is restored, you may navigate to the IP address of the gateway as normal.

**Security - Log Out**

Once a user is done with a session they may click **logout** at the top of any page. The user may also be logged out for inactivity based off of the Log Out Timer specified during the configuration.

**Closing the browser is not sufficient to log out.**
Email Configuration

To setup e-mails on the 460 gateway, navigate to Other->Email Configuration.

You can configure up to 10 email addresses.

1) SMTP Mail Username: The email address that the SMTP server has set up to use.
2) SMTP Mail Password: If authentication is required, enter the SMTP Server’s password (Optional).
3) SMTP Server: Enter the Name of the SMTP Server or the IP Address of the Server.
4) From E-mail: Enter the e-mail that will show up as the sender.
5) To E-mail: Enter the e-mail that is to receive the e-mail.
6) E-mail Group: Choose a group for the user. This is used in other web pages.

Click the Save Parameters button to commit the changes and reboot the gateway.
Alarm Configuration

To setup alarms on the 460 gateway, navigate to Other->Alarm Configuration.

1) Alarm Delay upon Powerup: At Powerup, the gateway will have values of ‘0’ stored for all data. This may cause alarms to trigger before these values are updated by the mating protocols. Set this field to provide needed time to update fields before considering values for alarms.

2) Enter the number of alarms to configure and click Set Max # Alarms to generate those lines.

3) In the Data Point Section:
   a. Top dropdown: select the Data Group. This dropdown menu will contain all groups that go from the gateway to the network.
   b. Lower dropdown: select the Data Point’s Specific Point. This is used to select which point in the group will be monitored for alarms.

4) In the Set Error Section:
   a. Select the Set Error Operation in the top dropdown menu. Available options are <, >, <=, >=, !=, ==, and Change of State (COS). This is the operation that will be used to compare the Data Point value against the Error Value to determine if the alarm needs to be set.
   b. Select the Set Error Value. This value is used as: ‘Data Point’s Value’ ‘Operation’ ‘Value.’ Ex: Ticks Since Powerup >= 1000. This will set the alarm after 1000 ticks have elapsed since the unit powered up.
5) In the Clear Error Section:
   a. Select the Clear Error Operation. Available options are <, >, <=, >=, !=, ==, and Change of
      State (COS). This is the operation that will be used to compare the Data Point value against
      the Error Value to determine if the alarm needs to be cleared.
   b. Select the Clear Error Value.
      - Ex: Ticks Since Powerup >= 5000. This will clear the alarm after 5000 ticks have elapsed
      since the unit powered up.
6) Enter an Alarm Name. This will make the alarm unique and will be available in the Alarm Status page
   as well as in the email generated by the alarm.
7) Select an email to associate this alarm with. When an alarm is set, it sends an email. When an alarm
   is cleared, it will also send an email.

Click the **Save Parameters** button to commit the changes to memory and reboot the gateway.
Diagnostics – Alarm Status

Alarm Status will only display under the Diagnostic menu tab if at least 1 Alarm is enabled.

1) # Alarms Enabled: This is a count of enabled alarms.
2) # Alarms Active: This is how many alarms are presently active (set).
3) Last Active Alarm: This is the last alarm that the gateway detected.
4) Clear # of Times Active: This will reset all alarms ‘# of Times Active’ to 0.
5) Alarm #: The reference number to the given alarm on the alarm setup page.
6) Name: The name of the alarm.
7) Status: The current status of the alarm, either OK or ALARM.
8) # of Times Active: This count represents the number of times this alarm has become active. If an alarm is triggered, this count will increment.

![Alarm Status Table]

Alarms – Active

While one or more alarms are active, every page will display ‘Alarms Active’ at the top of the page. This will no longer be displayed if all active alarms have been cleared.

![Alarms Active]

When an alarm is activated, the following will occur:

1) A one-time notification will be sent out to the email associated with the alarm.
2) For duplicate emails to occur, the alarm must be cleared and then become active again.
3) # Alarms Active and # of Times Active will be incremented.
4) Status of the Individual Alarm will be set to Alarm.
5) Last Active Alarm field will be populated with details on what triggered the alarm.
Alarms – Clear

When an alarm is cleared, the following will occur:

1) A one-time notification will be sent to the email associated with the alarm.
   a. For duplicate emails to occur, the alarm must become active and then be cleared again.

2) Total # Alarms Active will decrement. Last Active Alarm will not be changed.

3) Status of the Individual Alarm will be reset to OK.
Change of State (COS) Configuration

To access the configuration files in the 460 gateway, navigate to dropdown Other->COS Configuration. The gateway, by default only writes when data has changed. The gateway also waits to write any data to the destination until the source protocol is successfully connected.

Default values should fit most applications. Change these values with caution as they affect performance.

1) Stale Data Timer: If the data has not changed within the time allocated in this Stale Data Timer, the data will be marked as stale within the gateway and will force a write request to occur. This timer is to be used to force cyclic updates in the gateway, since data will only be written if it has changed by default. There is a separate timer per data mapping.

   Gateway behavior:
   • If time = 0s => (DEFAULT) The gateway will write out new values on a Change of State basis.
   • If time > 0s => The gateway will write out new values whenever the timer expires to force cyclic updates (write every x seconds).

2) Production Inhibit Timer: Amount of time after a Change of State write request has occurred before allowing a new Change of State to be written. This is to be used to prevent jitter. Default value is 0ms. This timer takes priority over the Stale Data Timer. There is a separate timer per data mapping. This timer is active only after the first write goes out and the first COS event occurs.

3) Writes Before Reads: If multiple writes are queued, execute # of Writes Before Reads before the next read occurs. Default is 10 and should fit most applications.

   Warning: A value of 0 here may starve reads if a lot of writes are queued. This may be useful in applications where a burst of writes may occur and you want to guarantee they all go out before the next set of reads begin.

4) Reads Before Writes: If multiple writes are queued, the # of Writes Before Reads will occur before starting the # of Reads Before Writes. Once the # of Reads Before Writes has occurred, the counter for both reads and write will be reset. Default is 1 and should fit most applications.

5) Enable Data Integrity: If enabled, do not execute any write requests to the destination until the source data point is connected and communicating. This prevents writes of 0 upon power up.

Click the Save Parameters button to commit the changes to memory and reboot the gateway.
Diagnostics Info

The Diagnostic page is where you can view the gateway’s translations and protocol specific status information.

For protocol specific diagnostic information, refer to the next three pages.

Diagnostics – Data and String Mapping

The Diagnostics->Diagnostic Info->System section displays the number of translations that are enabled, for both Data and String data types, the number of mappings that have an error, and the first mapping that has an error.

# Enabled: Number of mappings that are enabled.

# Error: The number of mappings that are enabled that have an error.

First Error: This is a detailed description of the first mapping that has an error.

Common Errors:

1) Destination or Source Point does not exist
   - Solution: Re-map the mapping
2) Source or Destination Pointer too small
   - There is not enough space on either the Source, or the Destination for the data you want to copy. This is typically seen when the Destination is smaller than the amount of data being transferred to it.
3) Range Discard, Min or Max Value
   - The actual data value is outside of the defined range
4) Math Error
   - Operation value cannot be 0
5) Scaling Error
   - Source Min must be smaller than Source Max
   - Destination Min must be smaller than Destination Max
Diagnostics – Modbus TCP/IP Server

Select the Modbus TCP/IP Server in the dropdown menu on the Diagnostics Page to view a breakdown of the diagnostics and common strings that are displayed on the page. Additional diagnostic information can be found by clicking the Help button.

**NOTE:** This page will auto-refresh every five seconds with the latest data.

**Clear All Values** - This will only affect displayed values.

1) This will reset all displayed values back to zero.
2) If viewing Modbus TCP/IP Server, this will only clear the values for the Modbus TCP/IP Server section of the gateway.

**Device Status**

1) Connected - A Modbus TCP/IP Client has a connection for the gateway.

2) Not Connected:
   a. Ethernet Cable not plugged in.
   b. The Modbus TCP/IP Client has not initiated communication to the gateway.
   c. The Modbus TCP/IP Client has not communicated to the gateway in “x” milliseconds, where “x” is the inactivity timeout specified in the Modbus TCP/IP Server Configuration.

---

### Diagnostics (MAC: 00:1:

<table>
<thead>
<tr>
<th>Modbus TCP/IP Server</th>
<th>View</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clear All Values</td>
<td></td>
</tr>
</tbody>
</table>

### Device Status

Connected and Running

### LED Status

Connection Status: Connected

### Variables

- FC01 Read Coil Status: 6
- FC02 Read Input Status: 0
- FC03 Read Holding Registers: 0
- FC04 Read Input Registers: 0
- FC05 Force Single Coil: 0
- FC06 Preset Single Register: 0
- FC15 Force Multiple Coils: 0
- FC18 Preset Multiple Registers: 0
- FC23 Read/Write 4X Registers: 0
- Successful Responses Sent: 12
- Error Responses Sent: 0
- Timeout, TCP Closed: 0
- TCP Connections Open: 1
LED Status:

1) Good Communication (Solid Green) – The gateway is connected to a Modbus TCP/IP Client and communicating as expected.
2) Not Connected (Flashing Green) – The Modbus TCP/IP Client has never attempted to connect to the gateway.
3) Connection Timeout (Flashing Red) - The gateway has lost a connection to the Modbus TCP/IP Client.
4) Off:
   a. No power.
   b. No Ethernet cable plugged in.

Variables:

1) FC01 Read Coil Status – Modbus Function Code 1: Number of Read Coil Status requests received.
2) FC02 Read Input Status – Modbus Function Code 2: Number of Read Input Status requests received.
3) FC03 Read Holding Registers – Modbus Function Code 3: Number of Read Holding Registers requests received.
4) FC04 Read Input Registers – Modbus Function Code 4: Number of Read Input Registers requests received.
5) FC05 Force Single Coil – Modbus Function Code 5: Number of Write Coil Status requests received.
6) FC06 Preset Single Register – Modbus Function Code 6: Number of Write Holding Register requests received.
7) FC15 Force Multiple Coils – Modbus Function Code 15: Number of Write Multiple Coil Status requests received.
8) FC16 Preset Multiple Registers – Modbus Function Code 16: Number of Write Multiple Holding Register requests received.
9) FC23 Read/Write 4X Register – Modbus Function Code 23: Number of Read/Write Holding Registers requests received.
10) Successful Responses Sent – Total number of Read/Write messages sent by the gateway.
11) Error Responses Sent - Total number of Read/Write errors sent by the gateway.
12) Timeouts, TCP Closed – Total number of Read/Write timeouts that cause the Modbus TCP/IP connection to close.
13) TCP Connections Open – Number of Modbus TCP/IP connections that have been opened to the gateway.
Diagnostics – Modbus RTU Master

Select the Modbus RTU Master in the top dropdown menu on the Diagnostics Page to view a breakdown of the diagnostics and common strings that are displayed on the page. You may also view individual Slave counters by selecting the device in the All Slaves dropdown and clicking View. Additional diagnostic information can be found by clicking the Help button.

NOTE: This page will auto-refresh every five seconds with the latest data.

Clear All Values - This will only affect displayed values.
1) This will return all values displayed to zero and clear the Status Strings.
   Example: If viewing Modbus RTU Master – Slave Address 1, this will only clear the values for Slave Address 1. This will reduce the All Slaves values indirectly.

Device Status - This will only display when viewing All Slaves.
1) Connected and Running – the gateway is connected to all the Modbus RTU Slaves.
2) Error: Timeout – No Modbus RTU scan lines are configured under an enabled Slave. Or, one or more enabled Modbus RTU Slaves are missing.
   a. Verify Modbus RTU device for Slave Address, 0 or 1 Base Addressing, ASCII Messaging, and Starting Addresses
   b. Verify that Port Settings used match the Modbus Slave that the gateway is communicating with.
   c. Verify wires for specific port settings.
3) Dependency Protocol Faulted – The dependent protocol is missing causing the communication to go inactive.
4) Unknown: First Scan Not Complete – Multiple scan lines are set up for the device and the gateway has not completed all the scan lines for the first time.
5) Fatal Error: Couldn’t Open Hardware Port – The serial port selected on the Modbus RTU Master Configuration page is not configured.
6) Fatal Error: No Configuration – No Modbus RTU Slaves are enabled though a Serial Port is enabled.
**LED Status** - This is the Status for *All Slaves* or the specific slave selected.

1) Solid Green (Connected) – The gateway is connected to all the Modbus RTU Slaves that are configured and enabled.
2) Flashing Green (Not Connected) – No Modbus RTU Slaves are enabled/configured.
   a. Verify Modbus RTU settings and ensure that the *Enable* checkbox is checked for the appropriate Slave(s).
3) Flashing Red (Connection Timeout) - The gateway cannot open a connection to one or more of the enabled Modbus RTU devices.
   a. Verify Modbus RTU device for Slave Address, 0 or 1 Base Addressing, ASCII Messaging, and Starting Addresses
   b. Verify port settings used match the Modbus Slave that the gateway is communicating with.
   c. Verify wires for specific port settings.
4) Flashing Red (Empty Scan List) - One or more enabled Modbus Slaves have no scan lines configured.
5) Flashing Red (Communication not attempted yet) – (Specific slave only) No reads are configured and data needed for writes isn't valid yet.
6) Flashing Red (Dependency Error) - The dependent protocol is missing causing the communication to go to inactive.
   a. The other protocol must be *Connected*.
7) Solid Red (Fatal Error) – The serial port selected on the Modbus RTU Master Configuration page is not configured.
   a. Verify that Modbus RTU has an enabled Port selected. If needed, configure port settings.

**Variables** - These are the values for *All Slaves* or the specific slave selected.

Network Bitmap Status (Displayed in Hex):
- Each bit corresponds to a slave. If the bit is set, the slave is connected, otherwise the bit is 0.
- Bit 0 corresponds to Slave 1 and Bit 4 is for Slave 5 and so on.

FC01 Read Coil Status:
- Function Code 1: Number of read Coil Status requests sent
- Point Type Used: 0x Coil Status
- # of Points: Any

FC02 Read Input Status:
- Function Code 2: Number of read Input Status requests sent
- Point Type Used: 1x Input Status
- # of Points: Any

FC03 Read Holding Registers:
- Function Code 3: Number of read Holding Register requests sent
- Point Type Used: 4x Hold Reg
- # of Points: Any

FC04 Read Input Registers:
- Function Code 4: Number of read Input Register requests sent
- Point Type Used: 3x Input Reg
- # of Points: Any

FC05 Force Single Coil:
- Function Code 5: Number of write Coil Status requests sent
- Point Type Used: 0x Coil Status
- # of Points: 1
FC06 Preset Holding Register:
- Function Code 6: Number of write Holding Register requests sent
- Point Type Used: 4x Holding Reg
- # of Points: 1

FC15 Force Multiple Coils:
- Function Code 15: Number of write Multiple Coil Status requests sent
- Point Type Used: 0x Coil Status
- # of Points: 2 or More OR Force Function Code 15/16 Enabled for # of Points of 1

FC16 Preset Multiple Registers:
- Function Code 16: Number of write Multiple Holding Register requests sent
- Point Type Used: 4x Holding Reg
- # of Points: 2 or More OR Force Function Code 15/16 Enabled for # of Points of 1

Successful Responses Received:
- Total number of Read and Write response messages received
  - Note: Add up all the Function Code Variables and it should equal the number of Successful Responses Received

Error Responses Received:
- Total number of read and write error messages received

Timeouts:
- Total number of read and write response messages not received

Status Strings - These are the values for All Slaves or the specific slave selected.

Last Error Code:
- Last read request error that the gateway received

Error Code Breakdown:
1) Error Code "code" - "Function" (N:"SlaveAddr" A:"StartAddr" L:"Length")
   a. Note: The Slave Address will inform you of the device that had the error. The Starting Address and Length will inform you the specific scan line that had the error

2) Error Codes:
   a. Error Code 1: Function Code received by the slave is not valid
   b. Error Code 2: The Register/Status received by the slave is not valid
   c. Error Code 3: The value received by the slave is not allowable
   d. Error Code 4: An unrecoverable error occurred while the slave was attempting to reply
   e. Error Code 5: The slave has accepted the request and is processing it, but a long duration of time will be required to reply
   f. Error Code 6: The slave is processing another message. The gateway will skip this message.
   g. Error Code 7: The slave has replied with a NAK. The server cannot perform the program function received in the query

3) Functions:
   - Specific to the Function Code being used for the scan line

4) N (Slave Address):
   - Slave address of the slave that the error was received from

5) A (Starting Address):
   - Starting address of the register/status that the error was received from

6) L (Length):
   - Number of points of the register/status that the error was received from
Example:

<table>
<thead>
<tr>
<th>Successful Responses Received</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error Responses Received</td>
<td>29</td>
</tr>
<tr>
<td>Timeouts</td>
<td>0</td>
</tr>
</tbody>
</table>

Status Strings

<table>
<thead>
<tr>
<th>Last Error Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error Code 2 - FC15_WrMCls (N:1 A:101 L:32)</td>
</tr>
</tbody>
</table>

This Error Code indicates Code 2, the register was not valid. Other details are:

- Received the error with FC 15, trying to Force Multiple Coils (WrMCls – Write Multiple Coils)
- N:1, from device 1, the first configured device
- A:101, Starting address of 101; aka: 000101 or 00101
- L:32, Multiple registers were trying to write 32 coils.

The Error Code indicates *not valid*, so the starting address was not found, or there were not 32 sequential coils to be written (101 through 132). To solve this, change the starting address, or reduce the *# of Points* configured.
LED Configuration

To modify the behavior of the LEDs on the 460 gateway, navigate to Other->Setup LEDs.

The LED Configuration page lets you configure the LEDs on the gateway.

Each LED may be set to Disabled, Protocol 1, or Protocol 2. If either Protocol is a Master/Client, you may set the LED to represent either all Slaves/Servers configured in the gateway or a particular Slave/Server device.

To select a particular Slave/Server device:

1) Select the protocol in the left dropdown menu.

2) Click Save Parameters to generate the second dropdown menu.

3) Select the individual Slave/Server in the right dropdown menu.

Click the Save Parameters button to commit the changes and reboot the gateway.
Configuration Files

To access the configuration files in the 460 gateway, navigate to dropdown **Other->Export/Import Config.**

Export Configuration

The export tool allows you to save your gateway configuration for backup or to be exported into another gateway. This file is named *rta_cfg.rtax* by default. Network settings will not be saved in this configuration since they must be unique for each gateway.

Upon clicking the **Save Configuration to File** button, you will be prompted to select a location to save the file.

![File Download dialog box](image)

Import Configuration

You can import a previously exported configuration file or a configuration file from another device into the 460 gateway whenever it is in Configuration Mode.

Upon clicking the **Choose File** button, you will be prompted to select a location from which to load the saved file. Once the location is selected, click the **Load Configuration** button.

If it has successfully loaded, the gateway will indicate that it was successful and a message will appear under the Load Configuration button indicating you should reboot the gateway.

If it encountered an error while trying to load the saved configuration, the gateway will indicate the first error it found and a brief description about it under the Load Configuration button. The configuration file is xml and can be modified with any text editor. Once that error is fixed, try loading again until it is successful.
Save and Replace Configuration Using SD Card

Saving Configuration Using SD Card
This function saves the gateway’s configuration automatically to an SD Card each time the gateway is rebooted via the Restart Now button on the webpage. If this unit should fail in the future, the last configuration the gateway used is stored on the SD card and can be used for a new gateway to get the application back up and running quickly.

This SD Card replaces every configurable field in the gateway, EXCEPT for IP Address, Subnet Mask, and Default Gateway.

Replacing Configuration Using SD Card
To replace a configuration in a gateway using the SD Card, a specific sequence of events must be followed for the replacement to happen correctly:

1) Extract SD Card from gateway you wish to copy the configuration from.
2) Power up the gateway you wish to copy the configuration to. DO NOT INSERT SD CARD YET.
3) Navigate to the webpage inside the unit.
4) Navigate to the dropdown Other->Utilities.
5) If you are not currently in Mode: Configuring, go into Configuration Mode by clicking on the Configuration Mode button at the top left-hand side of the screen.
6) Press the Revert to Manufacturing Defaults button on the Utilities Page. The Configuration will ONLY be replaced by the SD Card if the gateway does not have a configuration already in it.
7) When the unit comes back in Mode: Running, now insert the SD Card.
8) Do a hard Power Cycle to the unit by unplugging power. DO NOT RESET POWER VIA WEBPAGES.
   a. It will take an additional 30 seconds for the unit to power up while it is transferring the configuration. During this time, the gateway cannot be accessed via the webpage.
9) When the unit comes back up, the configuration should be exactly what was on the SD Card.
Intelligent Reset Button

If the IP Address of the gateway is forgotten or is unknown, there is an easy way to recover the IP Address using a reset button on the hardware.

1) On the side of the gateway with the SD card slot, there is a small pinhole. Using a paperclip, press the button through this pinhole and hold the button for at least 5 seconds.

2) After 5 seconds, the unit will acknowledge the command and LED 1 and LED 2 will start an alternate Blink Green quickly pattern.

3) Release the button and the gateway will reset to default IP settings (DHCP).
Utilities

To access the Utilities page in the 460 gateway, navigate to Other->Utilities. The Utilities screen displays information about the gateway including Operation Time, File System Usage, Memory Usage, and Memory Block Usage.

Here you can also:

- View the full revision of the software.
- View all the files stored in the Flash File System within the gateway.
- Identify your device by clicking the Start Flashing LED’s button. By clicking this button, the two diagnostic LED’s will flash red and green. Once you have identified which device you are working with, click the button again to put the LED’s back into running mode.
- Configure the size of the Log through the Log Configuration.
- Bring the device back to its last power up settings.
- Bring the device back to its original manufacturing defaults.
- Remove the Configuration File and Flash Files within the gateway.

![Utilities Page]

- Operation Time Since Last Reboot
  - 0 days, 1 hours, 16 mins, 42 secs
- File System Usage
  - Bytes Used: 344064 (43.75%)
  - Bytes Free: 442388 (56.25%)
  - Bytes Bad: 0 (0.00%)
- Memory Usage
  - Bytes Used: 341004 (32.52%)
  - Bytes Free: 707572 (67.48%)
- Used Memory Blocks
  - Memory Blocks Used: 13 out of 2000
- Revisions:
  - Listing of Revisions
- File List
  - File List
- Identify Device
  - Start Flashing LED’s
- Set Up Log
  - Log Configuration
- Revert To Last Powerup
  - Revert to Last Powerup
- Revert All
  - Revert to Manufacturing Defaults
- Reformat Flash
  - Reformat Flash