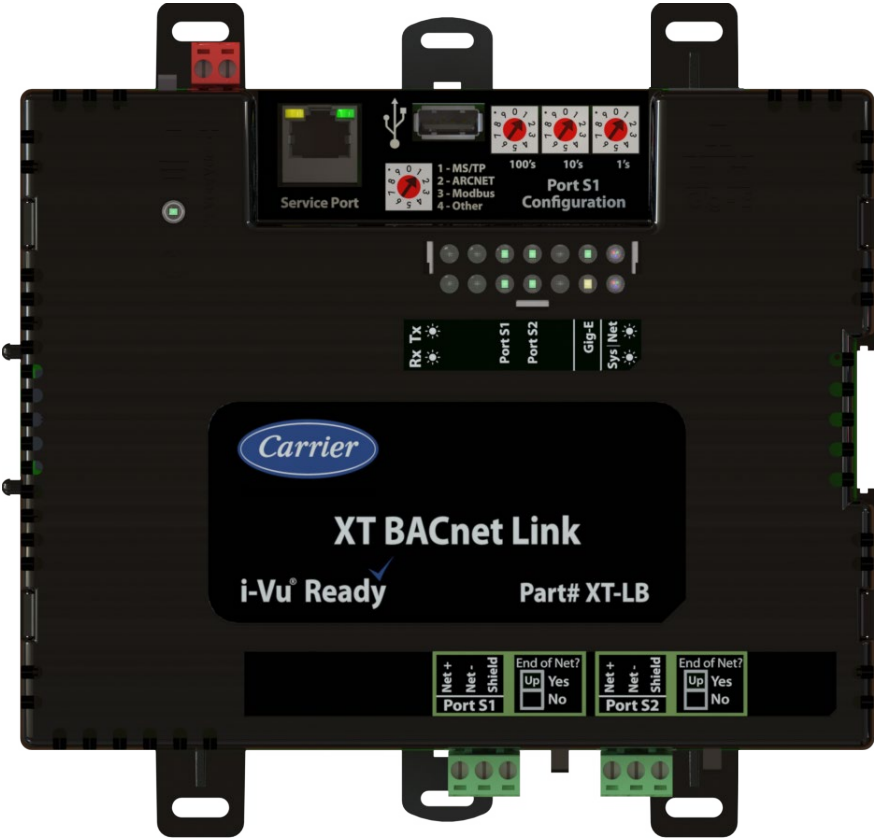


Modbus Integration Guide

For TruVu™ controllers (drv_gen5)





Verify that you have the most current version of this document from www.hvacpartners.com, the **Carrier Partner Community** website, or your local Carrier office.

Important changes are listed in **Document revision history** at the end of this document.

©2026 Carrier. All rights reserved.



Contents

| | |
|----------------------------------------------------------------------------------------------------------------------------|-----------|
| Overview | 1 |
| Controller as a client device on EIA-485 | 2 |
| Before-you-begin checklist..... | 2 |
| The integration process | 3 |
| 1 Create a control program in the Snap application | 3 |
| Formatting a Modbus address | 3 |
| Editing a microblock address | 4 |
| 2 Download the driver and control programs | 4 |
| 3 Connect the Carrier controller to the third-party device | 5 |
| Wiring specifications | 5 |
| To wire a third-party device with rotary switches..... | 5 |
| 4 Set up the driver properties | 6 |
| Control programs tab..... | 6 |
| Protocols > Modbus tab..... | 6 |
| Connections tab | 7 |
| Synchronize properties with the BAS | 8 |
| 5 Verify the integration is set up correctly | 9 |
| Controller as a client device on Ethernet | 11 |
| Before-you-begin checklist..... | 11 |
| The integration process | 12 |
| 1 Create a control program in the Snap® application | 12 |
| Formatting a Modbus address | 12 |
| IP address..... | 13 |
| Unit ID | 13 |
| Editing a microblock address | 13 |
| 2 Download the driver and control programs | 14 |
| 3 Connect the Carrier controller to the third-party device | 14 |
| 4 Set up the driver properties | 15 |
| Protocols > Modbus tab..... | 15 |
| Connections tab | 15 |
| Synchronize properties with the BAS | 16 |
| 5 Verify the integration is set up correctly | 16 |
| To capture communication using Wireshark..... | 18 |
| Register type/ Register addresses | 19 |
| Modbus Troubleshooting | 23 |
| Register addresses..... | 23 |
| Appendix A - Error codes and messages | 24 |
| PPD error codes..... | 24 |
| General error codes..... | 29 |
| Appendix B - Appendix B - Migrating a control program used with a Modbus driver for the Open Link or XT-LB | 30 |
| Appendix C - Modbus Protocol Conformance Statement | 31 |
| Appendix D - Configuring the driver parameters by using the Service Port | 33 |
| Appendix E - Module Status field descriptions | 34 |
| Document revision history | 35 |

Overview

You can use a Carrier i-Vu® TruVu™ controller to integrate Modbus device(s) with your i-Vu® system. The controller can serve as a client on a serial or TCP/IP network.

This document is divided into two sections that provide integration instructions for the two scenarios listed in the first two columns below. Determine how your Carrier controller will be used, then go to the appropriate page for the instructions

| If the Carrier® controller is to be a Modbus... | ...on this media type... | ...you can use this controller port... | see... |
|--------------------------------------------------------|---------------------------------|-----------------------------------------------|------------------------------------------------------------|
| Client | EIA-485, 2 wire | S1 or S2 | <i>Controller as a client device on EIA-485 (page 2)</i> |
| Client | Ethernet | Gig-E | <i>Controller as a client device on Ethernet (page 11)</i> |

NOTES

- The third-party Modbus point count for the controller equals the total number of points communicating on all ports that are using Modbus.
- If multiple ports are used for Modbus, the ports can have different functionalities. For example, Port S1 and/or S2 can be a serial client and the Gig-E port can be an Ethernet client.



CAUTION If you want to upgrade a system from an older Modbus application, carefully review *Appendix A*.

Controller as a client device on EIA-485

NOTE Modbus supports a single master on each network.

| Carrier® | |
|----------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------|
| Controller | TruVu™ gateways |
| Driver | drv_gen5 108-04-20088.driverx or later |
| Read/write capability | Can read from and write to the third-party equipment |
| Ports | S1 and/or S2 |
| Third party | |
| Supported equipment | Any server device that supports the Modbus (Modicon) protocol |
| Network media type | EIA-485 (2-wire) |
| Quantity of Modbus server devices you can physically connect to Port S1 or Port S2 on the Carrier® client controller | Up to 247 Modbus slaves |

Before-you-begin checklist

You need the following items, information, and skills for the integration process:

- A points list for each Modbus device that includes register addresses and read/write capabilities. Points lists are usually available from the third-party manufacturer's representative or website.
- Modbus device addresses
- A port pinout/configuration of the Modbus device. Pinouts are usually available from the third-party manufacturer's representative or website.
- The Modbus network's transmission mode, baud rate, data bits, parity, and stop bits
- Verification that all communication properties have been set on the Modbus devices
- Verification of communications through the port to which the Carrier controller will connect
- Experience creating control programs in the Snap® application
- Experience installing, wiring, setting up, and downloading to the Carrier controller

The integration process

Follow the steps in this section to integrate one or more third-party Modbus devices into the i-Vu® system using the Carrier controller. To install and network the Carrier controller, see the controller's *Installation and Start-up Guide*.

1 Create a control program in the Snap® application

When you create your control program, use a Network I/O microblock for each third-party point.



TIP Verify the third-party manufacturer's register addressing pattern by establishing communication with a few points whose values you can physically manipulate before you spend time addressing the remaining integration points.

Formatting a Modbus address

Use the information below to format a valid address in each microblock that you use to read or write to a third-party point.



CAUTION

When integrating third-party devices into the i-Vu® system, most communication problems are caused by incorrect data or typing errors in the microblock's **Address** field.

Address format:

protocol://register type/modbus register address/device address

| | | | |
|----------------------------------|----------------------------------------------------------------|----------------------------------------------------------------|-------------------------------|
| | | | |
| See <i>Protocol</i> below. | See <i>Register type/</i> <i>Register address</i> below. | See <i>Register type/</i> <i>Register address</i> below. | Third-party device (1-254) |

Example: modbus2://UINT/400128/26

For more information about the supported register types and how to address them, see *Register type/Register addresses* (page 19).

NOTE To send a broadcast to all Modbus devices on the port specified by the protocol signature, use 0 for the device address in a network output microblock.

Protocol

| If the point is in a Modbus device connected to the Carrier® controller's port... | Use the following protocol signature * |
|-----------------------------------------------------------------------------------|----------------------------------------|
| S1 | modbus1:// |
| S2 | modbus2:// |

* You can also use `modbus://` for points on one of the ports, but not both ports. For instructions on how to do this, see *Protocols > Modbus tab*.


Editing a microblock address

You can edit a microblock address in the following places:

- In the Snap Property Editor
- In the i-Vu® interface, on the microblock's **Properties** page > **Details** tab
- In the i-Vu® interface, on the control program's **Properties** page > **Network Points** tab

2 Download the driver and control programs

The Modbus PPD is available with driver `drv_gen5_XXX-XX-XXX.driverx` or later. To get and download the latest driver, see the controller's *Installation and Start-up Guide*.

- 1 In SiteBuilder's **Geographic** tree, add equipment for each of your control programs.
- 2 On the **Network** tree, assign the equipment to the controller by dragging each equipment from the **Geographic** tree and dropping it on the controller in the **Network** tree.
- 3 Click .
- 4 In the i-Vu® interface, download the driver and control programs to the Carrier controller.

See the "Managing third-party points and feature licenses" section of the controller's *Technical Instructions* for instructions on how to ensure you have adequate FlexPoints licensed for your integration.

3 Connect the Carrier controller to the third-party device


Wiring specifications

Use:

- 24 AWG twisted, shielded pair cable for up to 200 feet (60.96 meters)
or
- 22 AWG twisted, shielded pair cable for up to 2000 feet (609.6 meters)

To wire a third-party device with rotary switches

- 1 Turn off the Carrier controller's power.
- 2 Check the communications wiring for shorts and grounds.
- 3 Set switches on the Carrier controller. See table below.
- 4 Wire the Carrier controller to the third-party device. See table and notes below.
- 5 Turn on the Carrier controller's power.

| For... | Use Carrier controller port... | Wire Carrier terminal... | ...to third-party device terminal | Set switches on Carrier controller |
|-----------------|--------------------------------|--------------------------|-----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| EIA-485, 2-wire | S1 | Net+ Net- | + - | <ol style="list-style-type: none"> 1 Set the rotary protocol selector switch to 3 (Modbus). NOTE The rotary address switches are not used for Modbus in client mode. 2 If this controller is on the end of the EIA-485 network segment, set the Port S1 End of Net? switch to YES. If not, set to NO. 3 Set client mode, baud rate, and other relevant Modbus parameters in the driver properties, as detailed in the next section. |
| EIA-485, 2-wire | S2 | Net+ Net- | + - | <ol style="list-style-type: none"> 1 If this controller is on the end of the EIA-485 network segment, set the Port S2 End of Net? switch to YES. If not, set to NO. 2 To enable and configure Modbus on Port S2, set the parameters in the driver properties, as detailed in the next section. <p> CAUTION If Port S2 is enabled for BACnet and Port S2 is also set to be the Home Network, you cannot enable Modbus on Port S2.</p> |

NOTES

- If you cannot determine the media type or connections of the third-party device, contact your third-party representative.
- Use the same polarity throughout the network segment.
- Repeaters are required for more than 32 devices on an EIA-485 network. See your third-party device manufacturer's recommendations.
- To reduce communication and data errors, terminate each end of an EIA-485 network with a resistor whose value equals the network's characteristic impedance. i-Vu® controllers provide this terminating resistance by setting the **End of Net?** switch to **YES**. Make sure that only devices at the end of a network have termination enabled.

NOTE For controllers without a physical **End of Net?** switch, **End of Net?** is configured:

- By accessing the controller's service port and setting the **End of Network Command** to "Yes"
- In i-Vu®, under **Device > BACnet Router Properties**, set the **End of Network Command** to "Yes"
- A solid receive light on the Carrier controller indicates a wiring or polarity problem.
- If more than one controller is using the same server address on the same Modbus segment, you may see unexplained data errors in the control program. Each Modbus node on a Modbus serial network must be assigned a unique node address.

4 Set up the driver properties

The driver properties can be configured in either the:

- Controller's Service Port setup pages - See *Appendix C*.

OR

- i-Vu® driver pages - Select the controller's driver on the i-Vu® **Installer tree**.

Control programs tab

If you are integrating more Modbus points than are provided with the controller, you must activate your Modbus point license on the **Control Programs** tab. See the **Control Programs** tab section of the controller's *Installation and Start-up Guide* for instructions on how to temporarily or permanently activate Modbus points.

Protocols > Modbus tab

Select the **Enabled** checkbox.

If your control programs use the modbus:// prefix, select the port to which the modules associated with those programs will be connected (**Port S1** or **Port S2**) from the **Port Association for Modbus://** drop-down.

Connections tab

Select the port the Modbus devices are connected to. **Port S1** and **Port S2** can be used to simultaneously connect to independent Modbus networks.

On the **Port S1** or **Port S2** tab:

1. Select **Modbus** from the **Protocol** drop-down.

NOTE To use Port S1, the **Port S1 configuration** rotary switch can be set to either 3 or 4 on the controller. There is no rotary switch for Port S2.

2. Set the protocol properties:

| | |
|-------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Baud Rate | Select the baud rate according to the third-party manufacturer's documentation. |
| Parity | Select the parity according to the third-party manufacturer's documentation. |
| Data Bits | Select the data bits according to the third-party manufacturer's documentation. |
| Stop Bits | Select the stop bits according to the third-party manufacturer's documentation. |
| Response Timeout | Number of milliseconds the device waits for a response from the third-party device after sending a request or command. Valid range: 100-10000 milliseconds (default: 200) |
| Maximum Retries | Number of times the device resends a request or command if the third-party device does not send a proper response or does not respond at all. Valid range: 0-10 (default: 3) |
| Network Encoding | Select the serial encoding according to the third-party manufacturer's documentation (RTU or ASCII). |
| Advanced: Client/Server Mode | Identifies whether this device is a Modbus client (requesting and writing information to server devices) or a Modbus server (receiving and processing request from another client) on the Modbus network. NOTE Only client mode is supported at this time. |
| Advanced: Back Off Period | How long (min:sec) the integrator will wait before trying to connect to a Modbus device after the last failed connection attempt. Valid range: 0:01-10:00 (default: 5:00) |
| Advanced: Interpacket Delay | How long, in milliseconds, the integrator will wait between sending request packets to connect to the Modbus device. Valid range: 0-10000 milliseconds (default: 20) |
| Advanced: Max Registers to Request | The maximum number of adjacent Modbus registers that the Modbus device will support in each request. Valid range: 1-125 (default: 125) |

| | |
|------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Advanced: Max Coils to Request | The maximum number of adjacent Modbus Coils/Discrete inputs that the Modbus will support in each request. Valid range: 1-2000 (default: 2000) |
| Advanced: Restrict Coil and Register Commands | Set to No to have all write requests grouped using either Force Multiple Coils (function code 15) or Preset Multiple Registers (function code 16). Set to Yes to write ungrouped requests using either Write Single Coil (function code 05) or Write Single Register (function code 06). |
| Advanced: Endianness | Select the Modbus register ordering to correctly interpret Modbus (16-bit Word) registers. Select Big Endian for most-significant Word first (R1, R2, R3) or Little Endian (R3, R2, R1) for the least significant Word first. (default: Big Endian) |
| Advanced: Register Address Adjustment | Select the addressing method (Addresses start at 1 or Addresses start at 0) according to the third-party manufacturer's documentation. |

Synchronize properties with the BAS

- If you used the controller's Service Port to configure the properties:
 - If the **Restart** button is displayed, click it to restart the controller.
 - On the i-Vu® **Installer tree**, select the controller and **Upload** the parameters from the controller.
- If you used the controller's driver pages on the i-Vu® **Installer tree**, select the controller and **Download** the parameters to the controller.

5 Verify the integration is set up correctly

- 1 On the i-Vu® **Installer tree**, select the control program for the Carrier controller.
- 2 Select the **Properties** page > **Network Points** tab.

| If... | Then... |
|------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| You see the point value you expect with no errors in the Error column | You have successfully established communication with the third-party device. |
| All points show question marks instead of values | The i-Vu® application is not communicating with the Carrier controller or the control program. Troubleshoot the controller's communications. See the controller's <i>Installation and Start-up Guide</i> . |
| Error message appears | <p>Do one of the following actions based on the code or description in the Error column.</p> <ul style="list-style-type: none"> • Communications Disabled for this Microblock On the microblock's Network Points tab (or Properties page > Details tab), enable the microblock's Comm Enabled field. • No protocol support Verify that the Address in the microblock has the correct prefix: For Port S1: modbus1:// For Port S2: modbus2:// • Unlicensed Point You have configured more integration points than are licensed for this controller. See the “Managing third-party points and feature licenses” section of the controller’s <i>Technical Instructions</i> for instructions on how to ensure you have adequate FlexPoints licensed for your integration. • All other errors: See Modbus Troubleshooting (page 21) for troubleshooting information for displayed error codes |
| A value is incorrect | <p>Verify that:</p> <ul style="list-style-type: none"> • The Address in the microblock is correct. • The retrieved value is scaled properly, if necessary. For example, scaled from Celsius to Fahrenheit. Refer to the third-party manufacturer’s documentation or the controller’s <i>Installation and Start-up Guide</i> for scaling information. • The Register Address Adjustment configuration setting, in the Advanced section of the Connections > Port S1 (or Port S2) page, correctly identifies the addressing method used in the Modbus device. • The Endianness configuration setting, in the Advanced section of the Connections > Port S1 (or Port S2) page, is set correctly when reading a Float value. |

- If the above solutions do not resolve the problem, gather the following information for Technical Support:
- Screenshots of the following driver configuration pages:
 - **Protocols > Modbus** tab
 - **Control Programs** tab
 - **Connections > Port S1/S2** tab
- Log files downloaded from the driver's **Advanced > Diagnostics** tab
- A screenshot of the **Properties** page > **Network Points** tab showing addresses and errors
- All information from a controller Modstat copied into a text file. Right-click the Modstat, then select **Select All**. Press Ctrl+C to copy the information, then open Notepad and paste the information into a text file.
- Technical instructions for the third-party device, if available.

Controller as a client device on Ethernet

You can integrate Modbus devices into the i-Vu® system with the Carrier controller acting as an IP client in either of the following scenarios:

- The client controller reads and writes to points in a Modbus server.
- The client controller communicates with a Modbus gateway server to read and write to points in Modbus server devices.

Carrier

| | |
|-----------------------|------------------------------------------------------|
| Controller | TruVu™ gateways |
| Driver | drv_gen5 108-04-20088.driverx or later |
| Read/write capability | Can read from and write to the third-party equipment |
| Ports | Gig-E |

Third party

| | |
|---------------------|-------------------------------------------------------------|
| Supported equipment | A server device that supports the Modbus (Modicon) protocol |
| Network media type | Ethernet |

Before-you-begin checklist

You need the following items, information, and skills for the integration process:

- The IP addresses of the Modbus servers
- The Unit ID of each Modbus device that contains points that the Carrier controller will read and write to, and a points list for each device that includes point register addresses and read/write capabilities. Points lists are usually available from the third-party manufacturer's representative or website.
- Verification that all communication properties have been set on the Modbus devices
- Verification of Ethernet communication with each Modbus device that the Carrier controller will connect to
- Experience creating control programs in the Snap application
- Experience installing, wiring, setting up, and downloading to the Carrier controller

The integration process

Follow the steps in this section to integrate one or more third-party Modbus devices into the i-Vu® system using the Carrier controller. To install and network the controller, see the controller's *Installation and Start-up Guide*.

1 Create a control program in the Snap® application

When you create your control program, use a Network I/O microblock for each third-party point.



TIP Verify the third-party manufacturer's register addressing pattern by establishing communication with a few points whose values you can physically manipulate before you spend time addressing the remaining integration points.

Formatting a Modbus address

Use the information below to format a valid address in each microblock that you use to read or write to a third-party point.



CAUTION

When integrating third-party devices into the i-Vu® system, most communication problems are caused by incorrect data or typing errors in the microblock's **Address** field.

Address format:

mtcpip://register type/modbus register address/unit ID/IP address

See *Register type/ Register address* below.

Provided in third-party points list. See *Register type/ Register address* below

Third-party device See *Unit ID* below. IP address of the Modbus server

EXAMPLES::

- mtcpip://UINT/400128/3/192.168.168.17 (IP address specified directly)
- mtcpip://UINT/400128/3/17 (IP address specified indirectly, using index 17)

For more information about the supported register types and how to address them, see *Register type/ Register addresses (page 19)*.

IP address

You can specify the IP address of the third-party Modbus server either directly or indirectly.

| | |
|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Direct IP address | Enter the IP address of the third-party server in dot-decimal notation. For example, 192.168.168.17 |
| Indirect IP address | Enter the index number of the IP address to use; for example, 17. When you use the indirect IP address scheme, you must specify the IP addresses of the third-party Modbus servers outside of the control program logic. You configure the actual IP address corresponding to each index number in the Protocols > Modbus > IP Index Table on the driver properties page. See <i>Set up the driver properties</i> (page 6) . |

Unit ID

| If the register address is in... | The unit ID is... |
|------------------------------------------------------------|-------------------|
| The third-party Modbus server | 0 or 255 |
| A device on the third-party Modbus server's serial network | 1-254 |


Editing a microblock address

You can edit a microblock address in the following places:

- In the Snap Property Editor
- In the i-Vu® interface, on the microblock's **Properties** page > **Details** tab
- In the i-Vu® interface, on the control program's **Properties** page > **Network Points** tab

2 Download the driver and control programs

The Modbus PPD is available with driver drv_gen5_108-04-20088.driverx or later. To get and download the latest driver, see the controller's *Installation and Start-up Guide*.

1. In SiteBuilder's **Geographic** tree, add equipment for each of your control programs.
2. On the **Network** tree, assign the equipment to the controller by dragging each equipment from the **Geographic** tree and dropping it on the controller in the **Network** tree.
3. Click .
4. In the i-Vu® interface, download the driver and control programs to the Carrier controller.

See the “Managing third-party points and feature licenses” section of the controller's *Technical Instructions* for instructions on how to ensure you have adequate FlexPoints licensed for your integration.

3 Connect the Carrier controller to the third-party device

Use one of the following CAT5 or higher Ethernet cables:

- A cross-over cable to connect the Carrier controller directly to the third-party Modbus device
- A straight-through cable to connect the controller to a hub or switch, and a second straight-through cable to connect the hub or switch to the third-party Modbus device

Maximum cable length: 328 feet (100 meters)

- 1 Turn off the Carrier controller's power.
- 2 Check the communications wiring for shorts and grounds.
- 3 Wire the Carrier controller's **Gig-E** port to the network.
NOTE The Gig-E port remains capable of BACnet communication.
- 4 Turn on Carrier controller's power.
- 5 See the Modbus device's *Installation and Start-up Guide* to connect it to the network.

4 Set up the driver properties

The driver properties can be configured in either the:

- Controller's Service Port setup page – See *Appendix C*.
- i-Vu® driver pages – Select the controller's driver on the i-Vu® **Installer tree**.

Protocols > Modbus tab

Select the **Enabled** checkbox.

If you are using the Indirect IP Address method in your microblock addresses, enter an **Index** value and the associated **Address** into the **IP Addresses** table. Click **Add** to add a new row to the table. Click on a row and then click **Delete** to delete the selected row from the table.



TIP You can copy the table information from a spreadsheet by clicking **Add** enough times to create the number of rows you need, then copy/paste the information from the spreadsheet to the table.

Connections tab

Select the **Gig-E Port** tab, then open the **Modbus (mtcpi://)** section.

Set the protocol properties:

| | |
|------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Network: Modbus TCP Port | The port that the Modbus device will use for TCP/IP communications Valid range: 1-65535 (default: 502) |
| Advanced: Max Simultaneous Transactions | The maximum number of unique requests that the integrator can send to the Modbus device at one time. Valid range: 1-16 (default: 16) |
| Advanced: Client/Server Mode | Identifies whether this device is a Modbus Client (requesting and writing information to server devices) or a Modbus server (receiving and processing requests from another client) on the Modbus network. NOTE Only client mode is supported at this time. |
| Advanced: Back Off Period | How long (min:sec) the integrator will wait before trying to connect to a Modbus device after the last failed connection attempt. Valid range: 0:01-10:00 (default: 5:00) |
| Advanced: Interpacket Delay | How long in milliseconds the integrator will wait between sending request packets to connect to the Modbus device. Valid range: 0-10000 milliseconds (default: 20) |

| | |
|------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Advanced: Max Registers to Request | The maximum number of adjacent Modbus registers that the Modbus device will support in each request. Valid range: 1-125 (default: 125) |
| Advanced: Max Coils to Request | The maximum number of adjacent Modbus Coils/Discrete inputs that the Modbus device will support in each request. Valid range: 1-2000 (default: 2000) |
| Advanced: Restrict Coil and Register Commands | Set to No to have all write requests grouped using either Force Multiple Coils (function code 15) or Preset Multiple Registers (function code 16). Set to Yes to write ungrouped requests using either Write Single Coil (function code 05) or Write Single Register (function code 05). |
| Advanced: Endianness | Select the Modbus register ordering according to the third-party manufacturer's documentation: Big Endian (R1, R2, R3) or Little Endian (R3 R2 R1). |
| Advanced: Register Address Adjustment | Select the addressing method (Addresses start at 1 or Addresses start at 0) according to the third-party manufacturer's documentation. |

Synchronize properties with the BAS

- If you used the controller's Service Port to configure the properties:
 - If the **Restart** button is displayed, click it to restart the controller.
 - On the i-Vu@ **Installer tree**, select the controller and **Upload** parameters from the controller.
- If you used the controller's driver pages on the i-Vu@ **Installer tree**, select the controller and **Download** the parameters to the controller.

5 Verify the integration is set up correctly

- 1 On the i-Vu@ **Installer tree**, select the control program for the Carrier controller.
- 2 Select the **Properties** page > **Network Points** tab

| If... | Then... |
|------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| You see the point value you expect with no errors in the Error column | You have successfully established communication with the third-party device. |
| All points show question marks instead of values | The i-Vu@ application is not communicating with the Carrier controller or the control program. Troubleshoot the controller's communications. See the controller's <i>Installation and Start-up Guide</i> . |

| If... | Then... |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Error message appears | <p>Do one of the following actions based on the code or description in the Error column.</p> <ul style="list-style-type: none"> • Communications Disabled for this Microblock On the microblock's Network Points tab (or Properties page > Details tab), enable the microblock's Comm Enabled field. • No protocol support Verify that the Address in the microblock has the correct prefix: mtcip:// • Unlicensed Point You have configured more integration points than are licensed for this controller. See the "Managing third-party points and feature licenses" section of the controller's <i>Technical Instructions</i> for instructions on how to ensure you have adequate FlexPoints licensed for your integration. • All other errors: See Modbus Troubleshooting (page 21) for troubleshooting information for displayed error codes. |
| A value is incorrect | <p>Verify that:</p> <ul style="list-style-type: none"> • The Address in the microblock is correct. • The retrieved value is scaled properly, if necessary. For example, scaled from Celsius to Fahrenheit. Refer to the third-party manufacturer's documentation or the controller's Installation and Start-up Guide for scaling information. • The Register Address Adjustment configuration setting, in the Advanced section of the Connections > Port S1 (or Port S2) page, correctly identifies the addressing method used in the Modbus device. • The Endianness configuration setting, in the Advanced section of the Connections > Port S1 (or Port S2) page, is set correctly when reading a Float value. |

If the above solutions do not resolve the problem, gather the following information for Technical Support:

- A diagnostic capture using Wireshark. See the following section for Wireshark instructions.
- Screenshots of the driver configuration pages:
 - **Control Programs** tab
 - **Protocols > Modbus** tab
 - **Connections > Gig-E Port** tab, Modbus section
- Log files downloaded from the driver's **Advanced > Diagnostics** tab.
- A screenshot of the **Properties** page > **Network Points** tab showing addresses and errors.
- All information from a Modstat copied into a text file. Right-click the Modstat, then select **Select All**. Press Ctrl+C to copy the information, then open Notepad and paste the information into a text file.
- Technical instructions for the third-party device, if available.

To capture communication using Wireshark

Use Wireshark, a network analysis tool, to capture the Ethernet communication between the Carrier controller and the Modbus device.

PREREQUISITES

To use Wireshark to capture all Ethernet communication, provide one of the following devices:

- Ethernet hub (not a common switch)
- Port mirror on mirroring-capable switch
- Network sniffer/Test Access Port (TAP) such as SharkTap

- 1 Download the latest version of Wireshark from the Wireshark website (<http://www.wireshark.org>).
- 2 Run the Wireshark install program, accepting all defaults. Include WinPcap in the installation.
- 3 Place your capture device between the Carrier controller and the Modbus device by either:
 - Disconnecting the Carrier controller from the network and plugging its cable into the hub/TAP.
 - or
 - Disconnecting the Modbus device from the network and plugging its cable into the hub/TAP.
 - or
 - Configuring the mirroring port on your switch to mirror the port the Carrier controller is connected to.
- 4 Connect the Ethernet port of the computer running Wireshark to the hub/TAP/port mirror.
- 5 Identify the IP addresses of the controller and the Modbus device(s). These will be needed to decipher the capture.
- 6 On the computer, go to **Start > All Programs > Wireshark**.
- 7 From the menu bar, select **Capture > Interfaces**.
- 8 Click **Start** next to the interface that is connected to the network. This starts the IP capture.



TIP Choose the interface that shows activity.

- 9 Allow the capture to run long enough to ensure that there is sufficient data to allow a technician to review the problem.
- 10 On the menu bar, select **Capture > Stop** to stop the data capture.
- 11 Apply the "Modbus" filter to Wireshark and verify that you have actually captured Modbus traffic.
- 12 Select **File > Save** and save the capture to a convenient location. Leave the **Save as type** default set to **Wireshark/tcpdump/... - libpcap (*.pcap, *.cap)**.
- 13 Send the file to Carrier Technical Support for analysis.



TIP You can color code the information in the Wireshark capture file based on user-defined criteria. See *Wireshark's Help* for instructions on setting up Coloring Rules.

Register type/ Register addresses

| To... | this kind of value... | use this microblock... | with this register type... | and a Modbus register address in this range... ¹ |
|-------------|----------------------------------------------------------------------------------------------------------|------------------------|----------------------------|-------------------------------------------------------------|
| Read | 0 to 65,535 Unsigned 16-bit integer Input register (3XXXXX) | ANI/ANI2 | Uint (not Unit) | 300001-365535 |
| | 0 to 65,535 Unsigned 16-bit integer Holding register (4XXXXX) | ANI/ANI2 | Uint (not Unit) | 400001-465535 |
| | 0 to 4,294,967,296 Unsigned, 32-bit (long) integer Input register (3XXXXX) | ANI/ANI2 | Uint32 | 300001-365535 ² |
| | 0 to 4,294,967,296 Unsigned, 32-bit (long) integer Holding register (4XXXXX) | ANI/ANI2 | Uint32 | 400001-465535 ² |
| | 0 to 99,999,999 Unsigned, 32-bit (long) value Mod10 Input register (3XXXXX) | ANI/ANI2 | Uint32Mod10 | 300001-365535 ^{2, 6} |
| | 0 to 99,999,999 Unsigned, 32-bit (long) value Mod10 Holding register (4XXXXX) | ANI/ANI2 | Uint32Mod10 | 400001-465535 ^{2, 6} |
| | 0 to 999,999,999,999 Unsigned, 48-bit value Mod10 Input register (3XXXXX) | ANI/ANI2 | Uint48Mod10 | 300001-365535 ^{6, 7} |
| | 0 to 999,999,999,999 Unsigned, 48-bit value Mod10 Holding register (4XXXXX) | ANI/ANI2 | Uint48Mod10 | 400001-465535 ^{6, 7} |
| | 0 to 9,223,372,036,854,775,807 Unsigned, 64-bit (long) integer Input register (3XXXXX) | ANI/ANI2 | Uint64 | 300001-365535 ^{4, 8} |
| | 0 to 9,223,372,036,854,775,807 Unsigned, 64-bit (long) integer Holding register (4XXXXX) | ANI/ANI2 | Uint64 | 400001-465535 ^{4, 8} |
| | 0 to 10,000,000,000,000,000 Unsigned, 64-bit (long) value Mod10 Input register (3XXXXX) | ANI/ANI2 | Uint64Mod10 | 300001-365535 ^{4, 6, 8} |
| | 0 to 10,000,000,000,000,000 Unsigned, 64-bit (long) value Mod10 Holding register (4XXXXX) | ANI/ANI2 | Uint64Mod10 | 400001-465535 ^{4, 6, 8} |

| To... | this kind of value... | use this microblock... | with this register type... | and a Modbus register address in this range... ¹ |
|-------|-----------------------------------------------------------------------------------------------------------------------------------|------------------------|----------------------------|-------------------------------------------------------------|
| | -32,768 to +32,767 Signed 16-bit integer Input register (3XXXXX) | ANI/ANI2 | Sint | 300001-365535 |
| | -32,768 to +32,767 Signed 16-bit integer Holding register (4XXXXX) | ANI/ANI2 | Sint | 400001-465535 |
| | -2,147,483,648 to 2,147,483,647 Signed, 32-bit (long) integer Input register (3XXXXX) | ANI/ANI2 | Sint32 | 300001-365535 ² |
| | -2,147,483,648 to 2,147,483,647 Signed, 32-bit (long) integer Holding register (4XXXXX) | ANI/ANI2 | Sint32 | 400001-465535 ² |
| | -99,999,999 to 99,999,999 Signed, 32-bit (long) value Mod10 Input register (3XXXXX) | ANI/ANI2 | Sint32Mod10 | 300001-365535 ^{2,6} |
| | -99,999,999 to 99,999,999 Signed, 32-bit (long) value Mod10 Holding register (4XXXXX) | ANI/ANI2 | Sint32Mod10 | 400001-465535 ^{2,6} |
| | -999,999,999,999 to 999,999,999,999 Signed, 48-bit value Mod10 Input register (3XXXXX) | ANI/ANI2 | Sint48Mod10 | 300001-365535 ^{6,7} |
| | -999,999,999,999 to 999,999,999,999 Signed, 48-bit value Mod10 Holding register (3XXXXX) | ANI/ANI2 | Sint48Mod10 | 400001-465535 ^{6,7} |
| | -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 Signed, 64-bit (long) integer Input register (3XXXXX) | ANI/ANI2 | Sint64 | 300001-365535 ⁸ |
| | -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 Signed, 64-bit (long) integer Holding register (4XXXXX) | ANI/ANI2 | Sint64 | 400001-465535 ⁸ |
| | -10,000,000,000,000,000 to 10,000,000,000,000,000 Signed, 64-bit (long) value Mod10 Input register (3XXXXX) | ANI/ANI2 | Sint64Mod10 | 300001-365535 ^{6,8} |
| | -10,000,000,000,000,000 to 10,000,000,000,000,000 Signed, 64-bit (long) value Mod10 Holding register (4XXXXX) | ANI/ANI2 | Sint64Mod10 | 400001-465535 ^{6,8} |
| | Value with decimal point Input register (3XXXXX) | ANI/ANI2 | Float | 300001-365535 ² |
| | Value with decimal point Holding register (4XXXXX) | ANI/ANI2 | Float | 400001-465535 ² |

| To... | this kind of value... | use this microblock... | with this register type... | and a Modbus register address in this range... ¹ |
|-----------------------------|--------------------------------------------------------------------------------------------------------------------------|------------------------|------------------------------------------------------------|-------------------------------------------------------------|
| Write | Value with decimal point Input register (3XXXXX) | ANI/ANI2 | Float64 | 300001-365535 ^{5, 8} |
| | Value with decimal point Holding register (4XXXXX) | ANI/ANI2 | Float64 | 400001-465535 ^{5, 8} |
| | 0 or 1 Coil / Discrete (binary) output | BNI/BNI2 (or ANI/ANI2) | Do | 1-65535 ³ |
| | 0 or 1 Discrete (binary) input (1XXXXX) | BNI/BNI2 (or ANI/ANI2) | Di | 100001-165535 |
| | 0 or 1 Input register (3XXXXX) | BNI/BNI2 | BIT n (where n is a value 0-15 defined in points list) | 300001-365535 |
| | 0 or 1 Holding register (4XXXXX) | BNI/BNI2 | BIT n (where n is a value 0-15 defined in points list) | 400001-465535 |
| | 0 to 65,535 Unsigned 16-bit integer Holding register (4XXXXX) | ANO/ANO2 | Uint (not Unit) | 400001-465535 |
| | 0 to 4,294,967,296 Unsigned, 32-bit (long) integer Holding register (4XXXXX) | ANO/ANO2 | Uint32 | 400001-465535 ² |
| | 0 to 9,223,372,036,854,775,807 Unsigned, 64-bit (long) integer Holding register (4XXXXX) | ANO/ANO2 | Uint64 | 400001-465535 ^{4, 8} |
| | -32,768 to 32,767 Signed 16-bit integer Holding register (4XXXXX) | ANO/ANO2 | Sint | 400001-465535 |
| | -2,147,483,648 to 2,147,483,647 Signed, 32-bit (long) integer Holding register (4XXXXX) | ANO/ANO2 | Sint32 | 400001-465535 ² |
| | -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807 Signed, 64-bit (long) integer Holding register (4XXXXX) | ANO/ANO2 | Sint64 | 400001-465535 ⁸ |
| | Value with decimal point Holding register (4XXXXX) | ANO/ANO2 | Float | 400001-465535 ² |
| | Value with decimal point Holding register (4XXXXX) | ANO/ANO2 | Float64 | 400001-465535 ^{5, 8} |
| | 0 or 1 Coil Discrete (binary) output | BNO/BNO2 (or ANO/ANO2) | Do | 1-65535 ³ |
| | 0 or 1 Holding register (4XXXXX) | BNO/BNO2 (or ANO/ANO2) | BIT n (where n is a value 0-15 defined in points list) | 400001-465535 |
| 0x0000 command ⁹ | BNO/BNO2 | DIAG (serial only) | 1 ¹⁰ | |

- 1 The Modbus register address (sometimes called register, address, code, or parameter) must be a decimal value. If the register addresses in the points list include the letters A-F, use a scientific calculator to convert these hexadecimal values to decimal values. If the register addresses in the points list are 5-digit numbers in the range #0001-#9999 (where # = the prefix 1, 3, or 4), you may use the addresses as they are, or change them to fit the formats in the table above by shifting the prefix one digit to the left. For example, if you see the address 38129, you can also use 308129.
- 2 These types use 2 consecutive Modbus register addresses. Use the lower number in the microblock address.
- 3 Enter the number without any leading zeros. For example, enter 125, not 000125.
- 4 Unsigned 64 is limited to the maximum value of a Signed 64 value. Values above 9223372036854775807 cause an error.
- 5 Floating Point 64 values are rounded to the precision of a 32-bit Floating Point when transmitted or trended via BACnet. Positive and negative values larger than 3.40282E+38 are clipped to ±3.40282E+38.
- 6 Mod10 is a specific way of splitting up registers such that the value contained in any one register is limited to a decimal range of -9999 to +9999. This method uses four consecutive 16-bit registers. In an example where R4 is the last register and R1 is the first register, the formula would be: $R4 * 10,000^3 + R3 * 10,000^2 + R2 * 10,000 + R1$
- 7 These types use 3 consecutive Modbus register addresses. Use the lowest number in the microblock address.
- 8 These types use 4 consecutive Modbus register addresses. Use the lowest number in the microblock address.
- 9 The 0x0000 command is sent only when value is non-zero. The 0xFF00 command that also clears the port's communications event log is not supported.
- 10 This represents the sub-function of function code 8. Although function code 8's subfunctions range from 1-20, this driver only supports sub-function code 1 (Restart Communications Option)

Modbus Troubleshooting

Register addresses

Because Modbus register address formats differ from one manufacturer to another, the most common problems in a Modbus integration are caused by a microblock address having an incorrect register address.

If the Carrier controller is a serial master or Ethernet client, the register address that you include in a microblock address is the register address that you want to read from/write to in the slave/server. Microblocks support Modbus register addresses that consist of a base register address (1–65535) and a one digit prefix (except coils) that defines the type of value.

| This type of value... | Has the prefix... | Example of register address |
|-----------------------|-------------------|---------------------------------|
| Discrete input | 1 | 1 03459 or 1 3459 |
| Input register | 3 | 3 04564 or 3 4564 |
| Holding register | 4 | 4 01563 or 4 1563 |
| Coil | No prefix | 1563 |

Base address

Most manufacturers begin counting register addresses at 1; some begin at 0. Other than contacting the manufacturer to get clarification, you can look at the register address list to see if the addressing in any table starts at 0. If any table's addressing begins at 0, go to **Connections > Gig-E, Port S1, or Port S2** tab, then under **Modbus > Advanced**, set the **Register Address Adjustment** property to **Addresses start at 0**.

Prefix

Some manufacturers include the prefix in their register address. If you add another prefix, you will have an invalid register address. For example, if the third-party manufacturer shows a holding register address is **40123** and you add the prefix **4**, your register address will be **440123**, which is an invalid address.

Appendix A - Error codes and messages

The following list displays Modbus error codes with a brief description and possible solutions. The errors may appear addressed using one of the Modbus protocol signatures in the **Error** field of a Network I/O microblock.

Examples: mtcip://
modbus://

PPD error codes

| Error Code/Message | Possible Causes/Solutions |
|---------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 – Incomplete Address | One or more required parameters (register type, register address, etc.) are missing from this point's address field. See Formatting a Modbus address for required formatting and examples. |
| 2 – Missing Register Type | The register type field is missing from this point's address. Expected examples are Uint, Sint, Float, etc.. For more information about the supported register types, see <i>Register type/Register addresses</i> (page 19). |
| 3 – Unable to Determine Register Type | This driver is unable to interpret the value in the register type field of this point's address. Expected examples are Uint, Sint, Float, etc.. For more information about the supported register types, see <i>Register type/Register addresses</i> (page 19). |
| 4 – Invalid Register Type | The register type of this address is not valid for this type of read or write request. For more information about the supported register types for read and write requests, see <i>Register type/Register addresses</i> (page 19). |
| 5 – Missing Register Address | The Modbus register address field of this point's address is missing. Examples: 400123, 300456, 100789 For more information about the supported register addresses and valid address ranges, see <i>Register type/Register addresses</i> (page 19). |
| 6 - Address Error - Invalid URL Syntax | This driver is unable to interpret the value in the register address field of this point's address. Examples: 400123, 300456, 100789 For more information about the supported register addresses, and valid address ranges, see <i>Register type/Register addresses</i> (page 19). |
| 7 –Register Type and/or Register Type is Invalid | The register type and Modbus register address in this point's address are incompatible with each other. For example, register type Di requires a Modbus register address in the range of (1XXXXX). For more information about the supported register types and their valid Modbus register address ranges, see <i>Register type/Register addresses</i> (page 19). |

| Error Code/Message | Possible Causes/Solutions |
|---------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 8 – Unable to Determine Read Register Address and Type | The register type and Modbus register address in this point's address are not valid for use in a network input microblock. For example, register type Diag (Diagnostics) is only supported in a network output microblock. For more information about the supported register types and their valid Modbus register address ranges, see <i>Register type/Register addresses</i> (page 19). |
| 9 – Unable to Determine Read Register Address and Type | The register type and Modbus register address in this point's address are not valid for use in a network output microblock. For example, register type Di (Discrete input) with a Modbus register address in the (1XXXXX) range, can not be written to and is not supported in a network output microblock. For more information about the supported register types and their valid Modbus register address ranges, see <i>Register type/Register addresses</i> (page 19). |
| 10 – Missing Unit ID | The unit ID field is missing from this point's address. The expected value should be in the range of 0, 255, or 1-254. For more information about formatting the Modbus address with a unit ID, see <i>Formatting a Modbus address</i> (page 3) and for valid unit ID ranges, see <i>Unit ID</i> (page 13). |
| 11 – Unit ID Must Be A Number | This driver is unable to interpret the value in the unit ID field of this point's address. The expected value should be a number in the range of 0-255. For more information about formatting the Modbus address with a unit ID, see <i>Formatting a Modbus address</i> (page 3) and for valid unit ID ranges, see <i>Unit ID</i> (page 13). |
| 12 – Invalid Unit ID | The unit ID value in this point's address is invalid. The expected value should be a number in the range of 0-255. For more information about the supported unit ID value ranges, see <i>Unit ID</i> (page 13). |
| 13 – Missing Device Address | The device address field is missing from this point's address. The expected value should be a number in the range of 1-254. For more information about formatting the Modbus address with a device address, see <i>Formatting a Modbus address</i> (page 3). |
| 14 – Device Address Must be a Number | This driver is unable to interpret the value in the device address field of this point's address. The expected value should be a number in the range of 1-254. For more information about formatting the Modbus address with a device address, see <i>Formatting a Modbus address</i> (page 3). |
| 15 – Invalid Device Address | The device address value in this point's address is invalid. The expected value should be a number in the range of 1-254. For more information about the supported unit ID value ranges, see <i>Unit ID</i> (page 13). |
| 16 – Missing IP Address | The IP address field is missing from this point's address. The expected value should be an IPV4 address in dotted-decimal format. Example: 192.168.123.132. For more information about formatting the Modbus address with an IP address, see <i>Formatting a Modbus address</i> 3 and <i>IP Address</i> (page 13). |
| 17 – Invalid IP Address | The IP address field in this point's address is invalid. The expected value should be an IPV4 address in dotted-decimal format. For example, 192.168.123.132. For more information about formatting the Modbus address with an IP address, see <i>Formatting a Modbus address</i> (page 3) and <i>IP Address</i> (page 13). |

| Error Code/Message | Possible Causes/Solutions |
|-------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 18 – Value Out of Bounds | The requested input value to be written overloads the selected register type defined in this point's address field. For example, if the register type is "Sint" (signed integer), then only -32,678 to +32,767 values can be written. For more information about the supported register types and their supported value ranges, see <i>Register type/Register addresses</i> (page 19). |
| 19– Read/Write Request Failure | A read or write error occurred while attempting to send or receive a packet with the third party Modbus device. There are several reasons that could cause this error such as failed to open the port, TCP/IP socket errors, packet response unit ID mismatch, etc.. If you continue to get this error, contact our Technical Support group for assistance in capturing a trace of debugging information that will assist in understanding the exact I/O exception being thrown. |
| 20– No Connection Available | This error is one of two things: <ol style="list-style-type: none"> 1 There are no ports assigned to the addressed protocol signature (e.g. Modbus, modbus1, modbus2, etc.). See <i>Connections tab</i> (page 7) for more information regarding configuring ports for communication. 2 The driver is in the process of disconnecting the addressed device and all its associated points. |
| 21– Data Size Exceeds Max Registers to Request | The register type defined in this point's address field requires more Modbus registers than the Max Registers to Request configuration setting is set to, in order to read the value. For example, Float register types require two Modbus registers to read the value. Therefore, if the Max Registers to Request configuration is set to 1, you will get this error. For information on where to adjust the Max Registers to Request, see <i>Connections tab</i> (page 7). |
| 22– Write Restricted to Single Register | The register type defined in this point's address field requires more than one Modbus register to write the value. However, the Restrict Coil and Register Commands configuration is set to Yes which forces the driver to use the Modbus single register/coil function codes when writing a value. For example, Float register types require two Modbus registers to write the value. Therefore, the Restrict Coil and Register Commands configuration will need to be set to No so the driver can use the Preset Multiple Registers Modbus function code to send the Float value. For information on where to adjust the Restrict Coil and Register Commands, see <i>Connections tab</i> (page 7). |
| 23– Back Off | After all retries and timeouts have occurred trying to communicate with the third party Modbus device, all points addressed to that device will back off from any communications attempts for the configured Back Off Period. After the configured period has expired, communication attempts will resume. This cycle will repeat as long as this device is unable to communicate with the third party Modbus device. For information on where to adjust the Back Off Period, Maximum Retries, and Response Timeout, see <i>Connections tab</i> (page 7). |

| Error Code/Message | Possible Causes/Solutions |
|-----------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 24– Connection Failure | <p>This device failed to connect to the third party Modbus device.</p> <p>Things to verify:</p> <p>Verify the device address (for serial), unit ID and IP address (for TCP/IP) defined in this point's address matches the assigned address of the third party Modbus device;</p> <p>Verify the connection properties defined on the <i>Connection tab</i> (page 7) match the required communication configurations of the third party Modbus device;</p> <p>Verify the communication wiring connections (see Connect the Carrier® controller to the third-party device).</p> <p>For TCP/IP communications, use a network capturing tool such as Wireshark to analyze the Modbus communications (see <i>To capture communication using Wireshark</i> (page 18)).</p> |
| 25– Queued for Reconnect | <p>This point was in Back off, the back off period has expired, and now being queued up for retrying communications with the third party Modbus device. If successful and there are no other errors associated with this point, it will go into No error. If all retry attempts fail while trying to communicate with the third party Modbus device, all points addressed to that device will return to the back off error state (for more information on Back off, see error code 23 Back off above).</p> |
| 26– Reconnecting | <p>This point was in Back off, the back off period has expired, and is now retrying communications with the third party Modbus device. If successful and there are no other errors associated with this point, it will go into No error. If all retry attempts fail while trying to communicate with the third party Modbus device, all points addressed to that device will return to the back off error state (for more information on Back off, see error code 23 Back off above).</p> |
| 40–Modbus Exception 01 -Illegal Function | <p>The third party device has responded with Modbus Exception Code 01, which indicates that the third party Modbus device does not support the function code in the request. If receiving this error message on an output/write point, try setting the "Restrict Coil and Register Commands" parameter to "Yes".</p> |
| 41–Modbus Exception 02 -Illegal Data Address | <p>The third party device has responded with Modbus Exception Code 02, which indicates that the requested register address is not an allowable address in the third party Modbus device. Possible solutions: 1) Verify that the Modbus register address field, in this microblock's address, is configured for a Modbus register/coil that exists in the third party device; 2) In the Advance section of the port configuration page for this connection (e.g. Gig-E Port, S1 Port, S2 Port, etc.), try changing the Register Address Adjustment.</p> |
| 42–Modbus Exception 03 -Illegal Data Value | <p>The third party device has responded with Modbus Exception Code 03, which indicates that the third party Modbus device does not support one or all of the values received in the request. Verify that the Address field in the microblock is properly configured. See <i>Formatting a Modbus address</i> (page 3) for required formatting and examples.</p> |
| 43–Modbus Exception 04 -Slave Device Failure | <p>The third party device has responded with Modbus Exception Code 04, which indicates that the third party Modbus device had an unrecoverable error while attempting to read or write to a point. Contact the third party device vendor for further information.</p> |
| 44–Modbus Exception 05 -Acknowledge | <p>The third party device has responded with Modbus Exception Code 05, which indicates that the third party Modbus device needs more time to complete the request. In the Advance section of the port configuration page for this connection (e.g. Gig-E Port, S1 Port, S2 Port, etc.), try increasing the Interpacket Delay parameter.</p> |

| Error Code/Message | Possible Causes/Solutions |
|------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 45—Modbus Exception 06 -Slave Device Busy | The third party device has responded with Modbus Exception Code 06, which indicates that the third party Modbus device is busy processing the last request from this device. In the Advance section of the port configuration page for this connection (e.g. Gig-E Port, S1 Port, S2 Port, etc.), try increasing the Interpacket Delay parameter, and/or for TCP/IP connections, try reducing the Max Simultaneous Transactions. |
| 46—Modbus Exception 07 -Negative Acknowledge | The third party device has responded with Modbus Exception Code 07, which indicates that the third party Modbus device is unable to process the request. Contact the third party device vendor for further information. |
| 47—Modbus Exception 08 -Memory Parity Error | The third party device has responded with Modbus Exception Code 08, which indicates that the third party Modbus device experienced a parity error in its extended memory. Contact the third party device vendor for further information. |
| 48—Modbus Exception 0A -Gateway Path Unavailable | The third party device has responded with Modbus Exception Code 0A, which indicates that the third party Modbus device was unable to allocate an internal communication path from the input port to the output port for processing the request. This usually indicates that the third party gateway is misconfigured or overloaded. Contact the third party device vendor for further information. |
| 49—Modbus Exception 0B -Gateway Target Device Failed to Respond | The third party device has responded with Modbus Exception Code 0B, which indicates that the third party Modbus device had no response from the target device. For a Modbus TCP/IP server, this usually means that the slave device on a downstream Modbus serial network may be offline or not present. Verify that the target device is online and/or contact the third party device vendor for further information. Also verify that the Address field in this Microblock is referencing the correct device. |
| 1000—System Error: Missing read/write property | This is a program error as the system is unable to locate this point for processing. Contact Technical Support for further assistance regarding this error. |
| 1001—System Error: Invalid read/write property | This is a program error as the system is unable to determine if this point is a read or write point. Contact Technical Support for further assistance regarding this error. |
| 1002—System Error: Missing COV Increment | This is a program error as the system is unable to locate the point's COV increment. Contact Technical Support for further assistance regarding this error. |
| 1003—System Error: Invalid COV Increment | This is a program error as the system is unable to determine the point's COV increment's value. Contact Technical Support for further assistance regarding this error. |

General error codes

| Error Code/Message | Possible Causes/Solutions |
|----------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| No Protocol Support | The protocol defined in the signature of the address is disabled. On the Installer tree , click on the controller's driver, then select the Protocols tab, and then select the desired protocol tab (e.g. BACnet, Modbus, etc.) to enable. NOTE Enabling protocols requires a controller restart. |
| Initializing | This point is either: <ul style="list-style-type: none"> • In the process of being validated • Queued up for the initial read or write attempt to the third party device, • In the process of its initial read or write attempt to the third party device • Waiting for the initial response from the third party device. <p>Once the startup process has completed, this error should switch to No Error or a different error that will identify any problems that may have occurred.</p> |
| No Error | The microblock is not in error. No solution needed. |
| Communications Disabled for this Microblock | The microblock's communications are not currently enabled. Enable the microblock's communications by checking the box under Com Enable in i-Vu. |
| Not Linked | The microblock was not successfully linked to the object to which it is addressed. Ensure that the address is entered correctly and that the object the microblock is addressed to is functioning properly. |
| Programmer Error – Invalid MB State | The data integrity of the microblock was compromised. This is the default error code if none of the other errors apply. If this error is persistent, contact Technical Support to let them know there is a defect to address. |
| Undefined Client Microblock Error | An error occurred while the microblock was attempting to write a value. This is the default error code when something goes wrong trying to write a value over the network. If this error is persistent, contact Technical Support to let them know there is a defect to address. |
| Device Offline – Temporary Backoff | The device hosting the object that the microblock is attempting to interact with is not powered on. Ensure that the device hosting the object that the microblock is addressed to is powered on and functioning properly. |

Appendix B - Appendix B - Migrating a control program used with a Modbus driver for the Open Link or XT-LB

The gen5 driver for the i-Vu® XT controller is based on the Modbus Combo driver for the LGR line of controllers. If you are using a control program with the Modbus Combo driver, that control program is fully compatible with the i-Vu® XT controllers.



CAUTION If your control program was written for a different (older) Modbus driver, you must carefully follow the section below to make necessary changes to your control program.

| If your existing integration control program... | You must... |
|--------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------|
| Uses abbreviated register addresses | Change the abbreviated (invalid) addresses to complete addresses. For example, change modbus://UINT/123/1 to modbus://UINT/40123/1. |
| Has multiple network output microblocks in the same controller writing to the same address | Reconfigure the control program. The gen5 driver does not support this. |
| Has hundreds of microblocks with the same refresh time in sequentially addressed registers | Increase the refresh time for non-critical points. |

Appendix C - Modbus Protocol Conformance Statement

The following Modbus features and commands are supported by the Carrier® Modbus drivers. See the *Modbus website* (<http://www.modbus.org>) for complete Modbus protocol information.

Serial connection:

| | |
|-------------------|----------------------------------------------------------------------------------|
| Modes | RTU ^{1,2} ASCII |
| Media type | EIA-485, 2-wire |
| Baud rate | 9600 ¹ 57600 19200 ¹ 76800 38400 ¹ 115200 |
| Data bits | 7 ³ 8 ¹ |
| Parity | None ¹ Odd Even |
| Stop bits | 1 ¹ 1.5 2 |

¹ Most commonly used value(s)

² Requires 8 Data Bits for communications in this mode

³ Only supported in ASCII mode

Ethernet connection:

| | |
|---------------------------|----------------------------|
| Communication Type | TCP/IP |
| Media Type | Ethernet |
| Baud rate | 10 Mbps/100 Mbps/1000 Mbps |

Serial or Ethernet:

| Function codes - command | Register address range |
|---------------------------------|--------------------------------|
| 01 - Read coils | 1-65535 |
| 02 - Read discrete inputs | 1 00001- 1 65535 |
| 03 - Read holding registers | 4 00001- 4 65535 |
| 04 - Read input registers | 3 00001- 3 65535 |
| 05 - Write single coil | 1-65535 |
| 06 - Write single register | 4 00001- 4 65535 |
| 08 - Diagnostics | 1* |
| 15 - Write multiple coils | 1-65535 |
| 16 - Write multiple registers | 4 00001- 4 65535 |

*This represents the sub-function of function code 8. Although function code 8's subfunctions range from 1-20, this driver only supports sub-function code 1 (Restart Communications Option).

Appendix D - Configuring the driver parameters by using the Service Port

You can set many driver parameters locally from the controller by using the **Service Port's** web-based controller setup interface. You can set operational parameters, such as port and communications' protocol settings, without the need to connect the i-Vu® application to the Carrier controller. Any parameters set locally through this interface take effect immediately. To connect to the controller setup pages, some Carrier controllers have an Ethernet Service Port and some have a USB Service Port.



WARNING After setting parameters locally through the Service Port interface and then connecting the controller to the i-Vu® application, proceed carefully, as follows:

In the i-Vu® application, you must **upload** the parameters that you set locally **BEFORE** you **download** memory or parameters. Downloading, without uploading first, overwrites all the settings you made through the **Service Port**. Uploading first preserves those parameters.

NOTE There are a few parameters that can **only** be set through the Service Port, such as the controller's IP address, and these are not overwritten by a memory or parameter download from the i-Vu® application.

For more information on connecting to the Service Port, see the Connecting to the router through the Service Port and the Connecting to the router through the Gig-E Port sections of the controller's *Installation and Start-up Guide*.

Appendix E - Module Status field descriptions

| Field | Description |
|--------------------------|-------------------------------------------------------------------------------------------|
| Modbus Field Description | Communications - Number of data packets transmitted and received by the integrator |

Document revision history

Important changes to this document are listed below. Minor changes such as typographical or formatting errors are not listed.

| Date | Topic | Change description | Code* |
|-------------|----------------------|---------------------------|--------------|
| 6/22/25 | Advanced Client mode | Added note | |

