



Modbus[®]/TCP User Guide



Trademark Notices

Control, DeviceMaster, and PortVision are registered trademarks of Control Corporation.

Concept is a trademark of Schneider Electric.

Modbus is a registered trademark of Schneider Electric.

PLC is a registered trademark of Allen-Bradley Company, Inc.

Ethernet is a registered trademark of Digital Equipment Corporation, Intel, and Xerox Corporation.

Portions of SocketServer are copyrighted by GoAhead Software, Inc. Copyright © 2001. GoAhead Software, Inc. All Rights Reserved.

Windows is a registered trademark of Microsoft Corporation in the United States and/or other countries.

Other product names mentioned herein may be trademarks and/or registered trademarks of their respective owners.

Tenth Edition, January 23, 2019

Copyright © 2005-2018. Control Corporation.

All Rights Reserved.

Control Corporation makes no representations or warranties with regard to the contents of this document or to the suitability of the Control product for any particular purpose. Specifications subject to change without notice. Some software or features may not be available at the time of publication. Contact your reseller for current product information.

Table of Contents

Chapter 1. Introduction	7
1.1. Product Name Change Notification	7
1.2. Audience	8
1.3. Control Modbus Solutions	8
1.4. Product Overview	8
1.5. Modbus/TCP Multi-Mode Connectivity	9
1.5.1. PLC Master/DeviceMaster Slave Mode	9
1.5.2. PLC Slave/DeviceMaster Master Mode	9
1.5.3. Dual Master (Virtual Peer-to-Peer) - Write Mode	10
1.5.4. Dual Master (Virtual Peer-to-Peer) - Read Mode (Dual Polling)	10
1.5.5. Filtering and Data Extraction Functionality (Patent Pending)	11
1.6. Definitions and Terms	12
1.6.1. Data Type Definitions	12
1.6.2. Glossary	12
1.7. Locating the Latest Software and Documents	13
1.8. Modbus/TCP Application Setup	14
Chapter 2. Programming Interface	15
2.1. Overview	15
2.1.1. Modbus Master Requirements	15
2.1.2. What is Modbus/RTU?	16
2.1.3. What is Modbus/ASCII?	16
2.1.4. What is Modbus/TCP?	17
2.2. Raw Data Interface	17
2.2.1. Supported Modbus Messages	17
2.2.2. Serial Port Raw/ASCII Interface	18
2.2.3. Ethernet Device Raw/ASCII Interface	18
2.2.4. Raw/ASCII Transfer Modes	19
2.2.4.1. Data-Stream Transfer Mode	19
2.2.4.2. Command/Response Mode	20
2.2.5. Receive Data Message (Raw Data)	21
2.2.5.1. Format	21
2.2.5.2. Communication Methodology (Receive Raw Data in Slave Mode)	22
2.2.5.3. Communication Methodology (Receive Data Master Mode)	22
2.2.6. Transmit Data Message (Raw Data)	23
2.2.6.1. Format	23
2.2.6.2. Communication Methodology (Transmit Raw Data Slave Mode)	24
2.2.6.3. Communication Methodology (Transmit Data Master Mode)	24
2.2.7. Sequence Number Messages (Raw Data)	25
2.3. I/O Scanner (Raw Data)	25
2.4. Modbus/RTU and Modbus/ASCII To-Slaves Protocol Interface	27
2.4.1. Communication Methodology	27
2.4.2. Modbus Slave Device Search Methodology	28
2.5. Retrieve Statistics Message	29

Chapter 3. Configuration Overview.....	31
3.1. Home Page.....	31
3.2. Serial Port - Configuration Overview	33
3.3. Ethernet Device - Configuration Overview.....	37
Chapter 4. Serial Menus.....	41
4.1. Serial Port Overview Page	42
4.2. Port Serial Configuration Page	43
4.3. Port Raw/ASCII Configuration Page	45
4.4. Port Modbus Configuration (Raw/ASCII Only) Page.....	48
4.5. Port Filtering/Data Extraction Configuration (Raw/ASCII Only) Page.....	51
4.6. Application TCP Configuration (Raw/ASCII Only) Page.....	55
Chapter 5. Ethernet Menus.....	59
5.1. Ethernet Device Overview Page	60
5.2. Device Interface Configuration Page	61
5.3. Device Raw/ASCII Configuration Page.....	63
5.4. Device Modbus Configuration Page.....	66
5.5. Device Filtering/Data Extraction Configuration Page.....	69
5.6. Application TCP Configuration	70
Chapter 6. Modbus Menus.....	71
6.1. Known Modbus Slave Devices List Page	71
6.2. Alias Device ID Functionality Overview	72
6.3. Modbus Alias Statistics Page	74
6.4. Modbus Alias Configuration Page.....	75
Section 7. Network Menus.....	77
7.1. Network Configuration Page.....	77
7.2. Password Page	78
7.3. Security Page.....	80
7.4. Key and Certificate Management Page.....	81
Chapter 8. Diagnostic Menus.....	83
8.1. Serial Communication Statistics Page.....	83
8.2. Ethernet Device Statistics Page	86
8.3. Modbus/TCP and Serial Modbus Master Statistics	88
8.4. Serial Interface Logs	91
8.5. Ethernet Device Logs	92
8.6. System Log	93
Section 9. System Menus.....	95
9.1. Update Firmware	95
9.2. Configuration File Page.....	98
9.2.1. Saving a Configuration File	98
9.2.2. Loading a Configuration File	99
9.3. System Snapshot Page	100
9.4. Restore Defaults Page	100
9.5. Reboot	101

Chapter 10. Troubleshooting and Technical Support	103
10.1. Troubleshooting Checklist	103
10.2. General Troubleshooting.....	104
10.3. Technical Support.....	104
Appendix A. Programming the PLC via Concept	105
A.1. Overview	105
A.1.1. What is Concept?.....	105
A.1.2. Requirements	105
A.1.3. Example Program Considerations (Raw Data)	105
A.2. Concept Program Screens	106
A.2.1. Processor and Ethernet Setup.....	106
A.2.2. Message Screens.....	107
A.2.2.1. Read Serial Data via Read Holding Registers Message	108
A.2.2.2. Transmit Serial Data via Write Multiple Registers Message.....	109
A.2.2.3. Set Receive Sequence Number via Write Multiple Registers Message	110
A.2.2.4. Set Transmit Sequence Number via Write Multiple Registers Message	111
A.2.2.5. Read Serial Port Statistics via Read Holding Registers Message	112
A.2.2.6. Modbus/TCP Slot/Index and DeviceMaster IP Address Definition	113
A.2.3. Concept Example Programs	114
A.2.3.1. LPBKCNCP	114
A.2.3.2. SCANCNCP	114
A.2.3.3. Setting up and Running the Concept Example Programs	115
Appendix B. LPBKCNCP Example Program	123
Appendix C. SCANCNCP Example Program	129

Chapter 1. Introduction

This *User Guide* provides detailed information about the following topics:

- [Programming Interface](#) on Page 15
- [Configuration Overview](#) on Page 31
- [Serial Menus](#) on Page 41
- [Ethernet Menus](#) on Page 59
- [Modbus Menus](#) on Page 71
- [Network Menus](#) on Page 77
- [Diagnostic Menus](#) on Page 83
- [System Menus](#) on Page 95
- [Programming the PLC via Concept](#) on Page 105
- [LPBKCNC Example Program](#) on Page 123
- [SCANCNC Example Program](#) on Page 129

The [Modbus Hardware Installation and Configuration Guide](#) provides the following information:

- Connecting the hardware and devices
- Programming the DeviceMaster IP address,
- Uploading Modbus/TCP firmware

The [Modbus/TCP Interface Configuration Quick Start](#) provides embedded web page configuration procedures.

See [Locating the Latest Software and Documents](#) on Page 13 to locate the latest firmware, documentation, and tools.

1.1. Product Name Change Notification

Control has implemented a product name change for the DeviceMaster UP 2-port DIN rail models that align with our new 1-port and 4-port DIN rail model names.

Note: Please note that the form, fit, and function of the DeviceMaster has not changed - only the name.

Old Name/Description	New Model Name
DeviceMaster UP 2-Port 1E Modbus Router	DeviceMaster MOD-2202
DeviceMaster UP 2-Port DB9 1E Modbus Router	DeviceMaster MOD-2102
DeviceMaster UP 2-Port 2E Modbus Router	DeviceMaster MOD-2402
DeviceMaster UP 2-Port DB9 2E Modbus Router	DeviceMaster MOD-2302

In this manual, the DeviceMaster MOD or UP are simply referred to as the DeviceMaster.

Note: Unless you special-ordered the DeviceMaster, you will need to load the Modbus/TCP firmware.

1.2. Audience

The primary audience of this document is the person responsible for installing the DeviceMaster and programming the PLC. This guide assumes you are familiar with the following topics:

- Windows operating system
- Modbus/TCP, Modbus/RTU, and/or Modbus/ASCII
- A PLC, SCADA System, or OPC Server that communicates with Modbus/TCP, Modbus/RTU, or Modbus/ASCII
 - Raw/ASCII devices such as barcode scanners, weigh scales, and printers
 - Modbus/RTU and/or Modbus/ASCII slave devices.

1.3. Control Modbus Solutions

If you ordered the Modbus part number for your DeviceMaster, Modbus/TCP is loaded on the DeviceMaster by default. You may want to review our other Modbus solutions to make sure that the feature rich Modbus/TCP application is what you want to use. Optionally, Modbus Router or Modbus Server may be more effective for your particular environment.

The Control web site provides information about the differences between the [three Modbus solutions](#).

If Modbus Server or Modbus Router is a better solution, you can [DOWNLOAD](#) the appropriate firmware and corresponding documentation.

1.4. Product Overview

The DeviceMaster operates as a highly versatile Modbus gateway when the Modbus/TCP firmware is uploaded to the DeviceMaster. The DeviceMaster provides Modbus/TCP, Modbus/RTU, Modbus/ASCII, and Ethernet TCP/IP controller interfaces to both serial and Ethernet TCP/IP raw/ASCII devices, and both Modbus/RTU and Modbus/ASCII slave devices.

Your particular DeviceMaster model may or may not have the Modbus/TCP firmware loaded (depending on the model you purchased).

Note: *Models that have Modbus/TCP loaded on the DeviceMaster are identified in PortVision DX and the DeviceMaster is labeled accordingly.*

For example:

- The DeviceMaster 1-port provides Modbus/TCP support for one raw/ASCII or Modbus/RTU serial device and one raw/ASCII Ethernet device for a total of two devices.
- The DeviceMaster 2-port provides Modbus/TCP support for two raw/ASCII or Modbus/RTU serial device and two raw/ASCII Ethernet device for a total of four devices.
- The DeviceMaster 4-port provides Modbus/TCP support for four raw/ASCII or Modbus/RTU serial devices and four raw/ASCII Ethernet devices for a total of eight devices.

Modbus/TCP firmware 3.x provides an application interface for both serial and Ethernet raw/ASCII devices. You can connect any application, such as a configuration, database, or control application, via the application socket port to raw/ASCII serial and/or Ethernet devices while the device(s) are attached to the PLC via Modbus/TCP.

1.5. Modbus/TCP Multi-Mode Connectivity

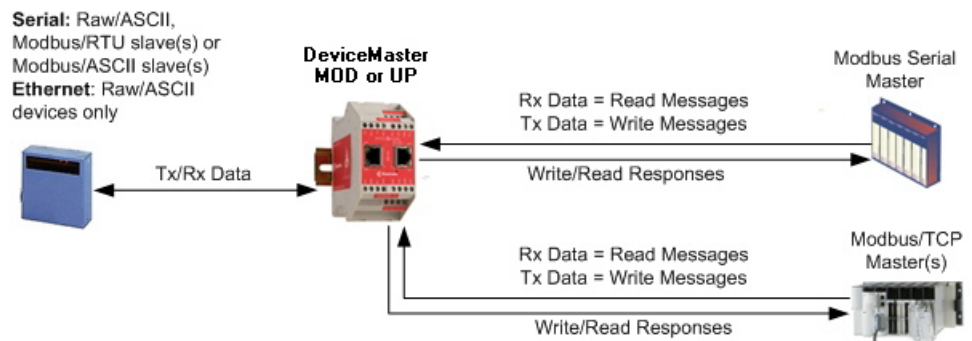
The Modbus/TCP firmware 7.x supports the following Modbus/TCP communication modes:

- [PLC Master/DeviceMaster Slave Mode](#) on Page 9
- [PLC Slave/DeviceMaster Master Mode](#) on Page 9
- [Dual Master \(Virtual Peer-to-Peer\) - Write Mode](#) on Page 10
- [Dual Master \(Virtual Peer-to-Peer\) - Read Mode \(Dual Polling\)](#) on Page 10
- [Filtering and Data Extraction Functionality \(Patent Pending\)](#) on Page 11

1.5.1. PLC Master/DeviceMaster Slave Mode

PLC Master/DeviceMaster Slave mode:

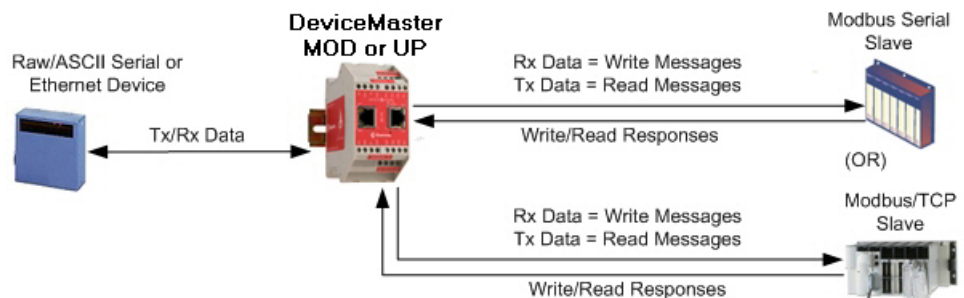
- Standard Modbus master to slave device method of communication. All read and write messages are initiated by the Modbus master.
- Raw/ASCII, Modbus/RTU slave, and Modbus/ASCII slave devices are supported in this mode.
- For raw/ASCII mode, the Receive Transfer mode and Transmit Transfer mode are both set to Slave (In Modbus/RTU-to-Slaves and Modbus/ASCII-to-Slaves mode, the DeviceMaster port only operates in To-Slave mode).



1.5.2. PLC Slave/DeviceMaster Master Mode

PLC slave/DeviceMaster master mode:

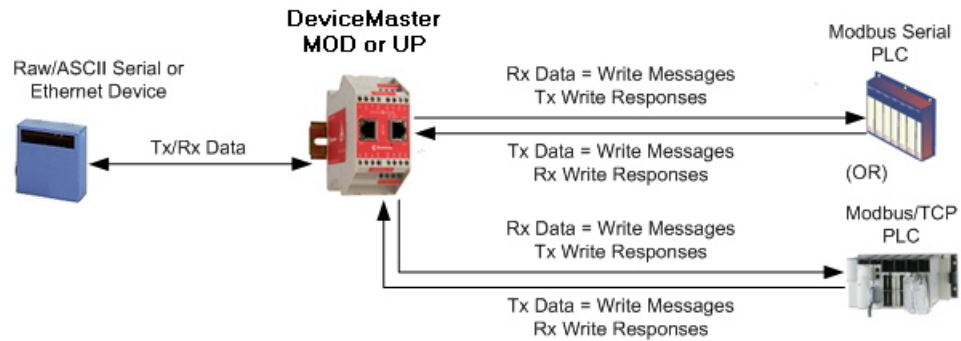
- The DeviceMaster initiates all read and write messages.
- The DeviceMaster writes received serial and/or Ethernet device data directly into PLC memory with minimal latency.
- The DeviceMaster polls the PLC for transmit data for serial and/or Ethernet devices.
- PLC programs can be simplified to eliminate both polling for received data and sending of write messages to transmit data.
- Only raw/ASCII devices are supported in this mode.
- The DeviceMaster *Receive Transfer* mode and *Transmit Transfer* mode are both set to *Master*.



1.5.3. Dual Master (Virtual Peer-to-Peer) - Write Mode

Dual master (virtual peer-to-peer) - write mode:

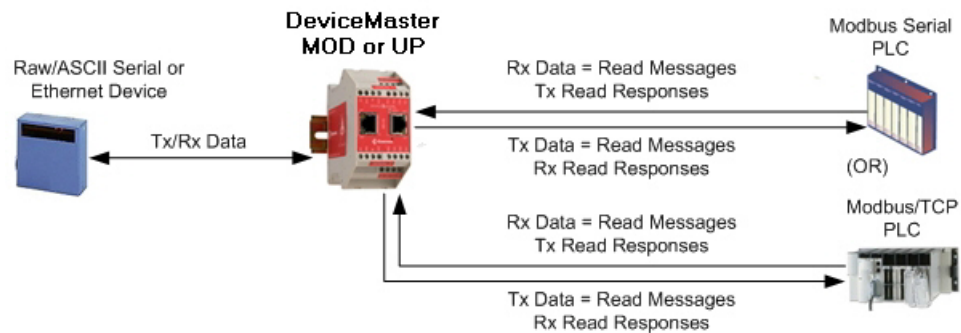
- The DeviceMaster and PLC initiate only write messages to each other.
- The DeviceMaster writes received serial and/or Ethernet device data directly into PLC memory with minimal latency.
- The PLC can write to serial and/or Ethernet devices through the DeviceMaster with minimal latency.
- This mode provides the lowest possible Ethernet bandwidth usage and most efficient usage of PLC and DeviceMaster processing power.
- Only raw/ASCII devices are supported in this mode.
- The *DeviceMaster Receive Transfer* mode is set to *Master* and *Transmit Transfer* mode is set to *Slave*.



1.5.4. Dual Master (Virtual Peer-to-Peer) - Read Mode (Dual Polling)

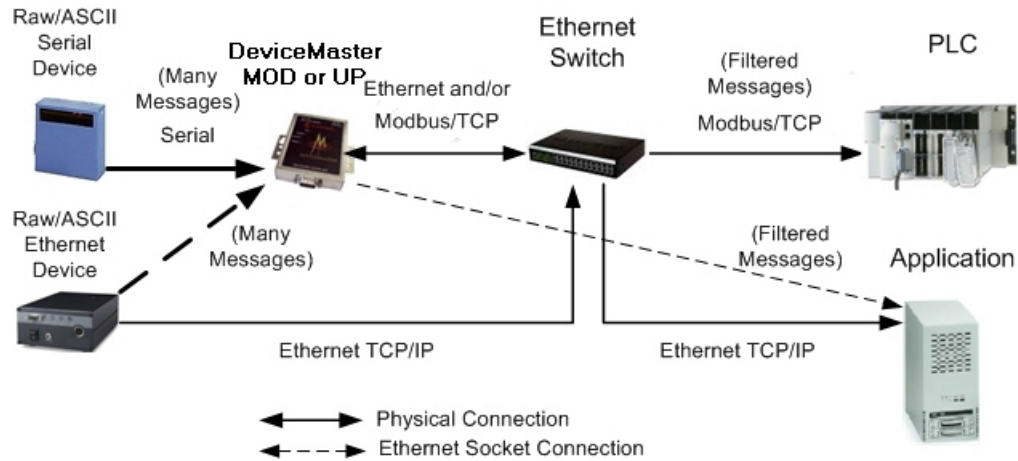
Dual master (virtual peer-to-peer) - read mode (dual polling):

- This is provided for programmers who strongly prefer polling.
- The DeviceMaster and PLC initiate only read messages to each other.
- The PLC polls for received serial and/or Ethernet device data.
- The DeviceMaster polls for transmit data to serial and/or Ethernet devices.
- This mode requires the highest possible Ethernet bandwidth usage and provides the least efficient usage of PLC and DeviceMaster processing power.
- Only raw/ASCII devices are supported in this mode.
- The DeviceMaster *Receive Transfer* mode is set to *Slave* and *Transmit Transfer* mode is set to *Master*.



1.5.5. Filtering and Data Extraction Functionality (Patent Pending)

The DeviceMaster provides the following filtering and data extraction functionality.



- Filtering:
 - String Filtering of up to 128 bytes of raw/ASCII data to both the PLC and/or application.
 - RFID filtering of EPCglobal formatted RFID tag data to both the PLC and/or application.
 - Barcode filtering of all UPC/EAN formatted barcodes data to both the PLC and/or application.
 - Simplifies PLC and application programming tasks.
- Data Extraction:
 - RFID data extraction extracts all parameters, such as company code, product code, and serial numbers, from any or all of the 43 EPCglobal tag formats. It then transfers the data to the PLC and/or application in a consistent and simple format.
 - Barcode data extraction extracts the company, product, and numbering codes from UPC/EAN formatted barcodes. It then transfers the data to the PLC and/or application in a consistent and simple format.
 - Simplifies PLC and application programming tasks.
- Environment specific support:
 - Support for multiple RFID reader tag formats.
 - RFID antenna grouping.
 - Aging of filtered string/RFID/barcode entries.
 - Discarding of unrecognized RFID and barcode messages.

For detailed information about filtering and data extraction, see the [DeviceMaster Filtering and Data Extraction Reference Guide](#).

1.6. Definitions and Terms

This section describes the Modbus/TCP definitions and terms included in the Modbus/TCP interface and supported by the DeviceMaster.

1.6.1. Data Type Definitions

The following list defines the available data types.

Data Type	Definition
BYTE	Bit String (8-bits)
DINT	Signed Double Integer (32-bits)
DWORD	Bit String (32-bits)
INT	Signed Integer (16-bits)
STRING	Character String (1-byte per character)
UDINT	Unsigned Double Integer (32-bits)
USINT	Unsigned Short Integer (8-bits)
WORD	Unsigned Integer (16-bits)

1.6.2. Glossary






The following list defines terms associated with Modbus/TCP.


Term	Definition
Alias Device ID	The device ID that the original received ID is changed to when an Alias Device ID is configured.
Device ID	The address of the slave device and the term is identical to <i>Unit Identifier</i> and <i>Slave Address</i> .
Ethernet Device	A device that communicates through an Ethernet TCP/IP connection.
Master Device	A device that transmits Modbus/TCP messages to slave devices and receives the corresponding responses.
Modbus	An application layer messaging protocol that provides client/server communications between devices connected on different types of buses.
Modbus Serial	The Modbus protocol over a serial connection.
Modbus/ASCII	Modbus Serial in ASCII format. This form of Modbus communication requires two characters for each byte.
Modbus/RTU	Modbus Serial in binary format.
Modbus/TCP	The Modbus protocol over an Ethernet TCP/IP connection, also known as Modbus over Ethernet.
Raw Serial Device	A common serial device that communicates over serial ports through plain byte or ASCII data messages.
Slave Address	The address of the slave device. This term is identical to Unit Identifier and Device ID.
Slave Device	A device that only responds to Modbus messages.
Socket Port	The Ethernet socket port that is used to communicate to an Ethernet device.

Term	Definition (Continued)
Unit Identifier	The address of the slave device and the term is identical to <i>Device ID</i> and <i>Slave Address</i> .

1.7. Locating the Latest Software and Documents

You can use the links in this table to check for updated software or documentation.

Software and Documentation		
<p>PortVision DX (Windows)</p>	<p>Use PortVision DX to manage Control Ethernet-attached devices to:</p> <ul style="list-style-type: none"> • Scan the network for attached devices • View networked devices in real-time • Access product-specific network settings configurations • Assign IP addresses and network settings to one or multiple devices • Upload the latest firmware or Bootloader • Save and load configuration files • Access DeviceMaster configuration web pages • Access Telnet/SSH sessions • Remotely reboot devices • Download technical documentation • Enable event logging to assist in monitoring and troubleshooting • Organize devices into folders and create multiple views • Enter notes about a folder or device 	
<p>Modbus/TCP Firmware</p>	<p>This is the application that may or may not have been loaded on the DeviceMaster depending on the model that was ordered.</p> <p>You may need to use PortVision DX or the web interface to load the firmware.</p>	
<p><i>Modbus Hardware Installation and Configuration Guide</i></p>	<p>This contains hardware installation, configuration information, and connector information.</p> <p>This includes using PortVision DX to configure the IP address and if necessary, how to update the firmware.</p>	
<p><i>Modbus/TCP Interface Configuration Quick Start</i></p>	<p>This document with web interface configuration procedures.</p>	
<p><i>Modbus/TCP User Guide</i> (This Guide)</p>	<p>The <i>User Guide</i> contains detailed information about the Modbus/TCP (application) firmware, which includes additional information about the web configuration interface for Modbus/TCP.</p> <p>It also discusses the example PLC programs that were installed on your system and provides a <i>Programming Interface</i> chapter.</p>	

Software and Documentation		
<i>DeviceMaster Filtering and Data Extraction Reference Guide</i>	<p>This <i>Guide</i> discusses the data extraction and filtering processes in the DeviceMaster are designed to off load as much work as possible from the PLC and/or application and provide a very simple and easy to use interface for standard RFID and barcode data.</p> <p>This functionality and interface is designed to save dozens, possibly hundreds of lines of ladder logic in a typical PLC program.</p>	

1.8. Modbus/TCP Application Setup

Before you can configure the Modbus/TCP firmware on the DeviceMaster, you must have previously performed the following steps:

- Install the hardware
- Install PortVision DX
- If necessary, upload the Modbus/TCP firmware using PortVision DX.
Note: Models that have Modbus/TCP loaded on the DeviceMaster are identified in PortVision DX and the DeviceMaster is labeled accordingly.
- Configure the DeviceMaster IP address using PortVision DX
Note: If necessary, refer to the [DeviceMaster MOD or UP Hardware Installation and Configuration Guide](#) for the above procedures.

Use the following steps to complete the DeviceMaster configuration for Modbus/TCP.

1. Program the Modbus/TCP PLC (refer to the information in [Programming Interface](#) on Page 15).
2. Configure the DeviceMaster serial and Ethernet device interface settings using the [Modbus/TCP Interface Configuration Quick Start](#). You can use the following chapters if you need additional information about fields on the web pages.
3. Connect your serial device or devices and make sure all Ethernet devices are attached to the same Ethernet subnet.

Chapter 2. Programming Interface

2.1. Overview

The DeviceMaster provides highly flexible Modbus connectivity.

- Modbus masters supported include Modbus/TCP and Modbus/RTU and Modbus/ASCII serial masters.
- Both serial Modbus/RTU and Modbus/ASCII slave devices are supported.
- All Modbus masters can communicate with all Modbus slave devices.
- The Modbus/RTU and Modbus/ASCII Protocol Interface is defined in ([2.4. Modbus/RTU and Modbus/ASCII To-Slaves Protocol Interface](#) on Page 27).

The DeviceMaster provides highly advanced raw/ASCII device functionality:

- Both serial and Ethernet TCP/IP devices are supported.
- Modbus interfaces include Modbus/TCP masters, Modbus/TCP slaves, and both Modbus/RTU and Modbus/ASCII serial masters.
- Up to six Ethernet TCP/IP Application connections per serial or Ethernet TCP/IP device.
- The raw/ASCII interface is defined in ([2.2. Raw Data Interface](#) on Page 17).

You must configure the DeviceMaster through its embedded web pages discussed in the following chapters.

The DeviceMaster uses normal Modbus addressing conventions and provides receive, transmit, and statistical data.

[Appendix A. Programming the PLC via Concept](#) on Page 105 describes the Concept™ PLC programming examples provided with the DeviceMaster. It describes how to configure the DeviceMaster for raw serial data and start running the example programs using the embedded web pages and the example PLC program code.

Note: *While the Concept PLC example programs directly apply only to the Schneider Electric Momentum, Quantum, and Compact PLCs, they can be used as a guide for programming other PLCs.*

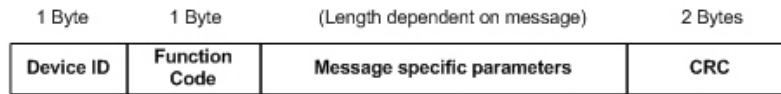
2.1.1. Modbus Master Requirements

Modbus Masters (Modbus/TCP, Modbus/RTU serial, and Modbus/ASCII) must meet these requirements:

- The Modbus Master must support the corresponding protocol.
- For raw/ASCII data, the Modbus Master must support the Read Holding Registers and Write Multiple Registers commands or, alternatively, the Read/Write Multiple Registers command.
- The Modbus Master must be able to write enough data in one message to handle the maximum sized messages required for the serial or Ethernet device.

2.1.2. What is Modbus/RTU?

Modbus/RTU is native Modbus in hexadecimal format. These are the base Modbus messages that contain simple read and write requests. The format is as follows:



Modbus/RTU Message Format

Where:

- The terms Master or Client are used to identify the sender of the message.
- The terms Slave or Server are used to identify the devices responding to the message.

Modbus/RTU is used primarily for:

- *Serial port connectivity.* RS-485 is the most common serial mode, but RS-232 and RS-422 are also widely used. Commonly used by both Master and Slave devices.
- *Ethernet TCP/IP socket connections.* This is not the same as Modbus/TCP (please see section on Modbus/TCP), but does provide a very simple method of interfacing to remote devices. It is used by many applications and some OPC servers.

Note: *This communication method typically is not supported by PLCs.*

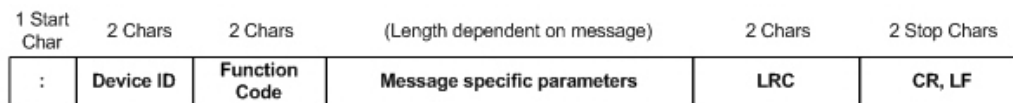
2.1.3. What is Modbus/ASCII?

Modbus/ASCII is native Modbus in ASCII format. This protocol is used primarily by legacy devices and is no longer supported as widely as Modbus/RTU.

Like Modbus/RTU, Modbus/ASCII contains the base Modbus messages that contain simple read and write requests. The differences between Modbus/ASCII and Modbus/RTU are:

1. The message data is sent in ASCII format, so the message length is twice as long. It requires two ASCII characters for each byte of data.
2. An 8 bit LRC is attached to verify the message instead of a 16 bit CRC. The LRC is also transmitted in ASCII format.
3. There are defined starting and ending characters to determine a Modbus/ASCII messages.

The format is as follows:



Modbus/ASCII Message Format

Where:

- The terms Master or Client are used to identify the sender of the message.
- The terms Slave or Server are used to identify the devices responding to the message.

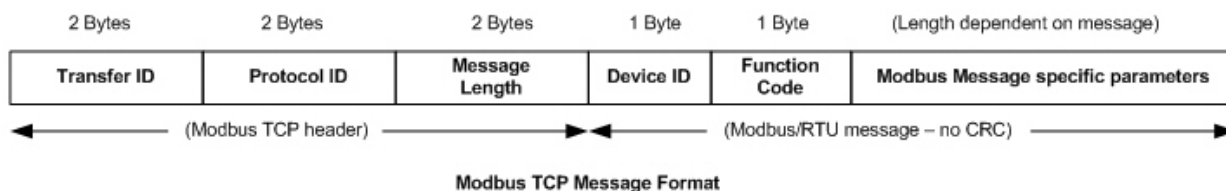
Modbus/ASCII is used primarily for:

- *Serial port connectivity.* RS-485 is the most common serial mode, but RS-232 and RS-422 are also used. Used primarily by legacy Slave devices.
- *Ethernet TCP/IP socket connections.* This is not the same as Modbus/TCP (please see section on Modbus/TCP), but does provide a very simple method of interfacing to remote devices. It is used by some applications and some OPC servers.

Note: *This communication method typically is not supported by PLCs.*

2.1.4. What is Modbus/TCP?

Modbus/TCP is an Ethernet network based protocol that contains a Modbus/RTU message, with the exception of the 2 byte CRC. The Modbus/TCP message contains a header with information designed to provide message identification and routing information. The format is as follows:



Where:

- The terms Master or Client are used to identify the sender of the message.
- The terms Slave or Server are used to identify the devices responding to the message.
- Modbus TCP messages are typically sent to and received on a defined Ethernet TCP/IP socket of 502.
- Modbus TCP implementations provide more capability, but also require more processing than simpler Modbus/RTU implementations.

Modbus TCP is used for connecting advanced Ethernet based devices, such as PLCs, HMIs, SCADA Systems, and most OPC Servers to:

- Other Ethernet devices supporting Modbus TCP.
- Serial Modbus/RTU and/or Modbus/ASCII devices through gateways (such as the DeviceMaster running the Modbus/TCP or Modbus Router applications).
- Serial or Ethernet TCP/IP raw/ASCII devices (barcode scanners, printers, RFID readers, visions systems, etc) through a gateway (such as the DeviceMaster running the Modbus/TCP application).

2.2. Raw Data Interface

This subsection contains the following topics:

- [Supported Modbus Messages](#) on Page 17
- [Serial Port Raw/ASCII Interface](#) on Page 18
- [Ethernet Device Raw/ASCII Interface](#) on Page 18
- [Receive Data Message \(Raw Data\)](#) on Page 21
- [Transmit Data Message \(Raw Data\)](#) on Page 23
- [Sequence Number Messages \(Raw Data\)](#) on Page 25

2.2.1. Supported Modbus Messages

DeviceMaster supports the following Modbus messages over Modbus/TCP for raw data transfer.

Message Type	Function Code	Maximum Message Size	Maximum Serial Packet Size
Read Holding Registers	3	250 BYTEs (125 WORDs)	246 BYTEs (123 WORDs)
Write Multiple Registers	16 (10 hex)	240 BYTEs (120 WORDs)	236 BYTEs (118 WORDs)
Read/Write Multiple Registers	23 (17 hex)	236 BYTEs (118 WORDs)	232 BYTEs (116 WORDs)

Note: Your PLC programming software may not allow maximum size serial packets.

2.2.2. Serial Port Raw/ASCII Interface

Serial Port Raw/ASCII Addressing	Serial Port 1	Serial Port 2	Serial Port 3	Serial Port 4	Access Rule
Unit ID	255 (FF hex)	255 (FF hex)	255 (FF hex)	255 (FF hex)	N/A
Receive Data Address	1000 (Base 0) 1001 (Base 1)	2000 (Base 0) 2001 (Base 1)	3000 (Base 0) 3001 (Base 1)	4000 (Base 0) 4001 (Base 1)	Read Only
Receive Data Sequence Number Address	1256 (Base 0) 1257 (Base 1)	2256 (Base 0) 2257 (Base 1)	3256 (Base 0) 3257 (Base 1)	4256 (Base 0) 4257 (Base 1)	Read/Write
Transmit Data Address	1300 (Base 0) 1301 (Base 1)	2300 (Base 0) 2301 (Base 1)	3300 (Base 0) 3301 (Base 1)	4300 (Base 0) 4301 (Base 1)	Read/Write
Transmit Data Sequence Number Address	1556 (Base 0) 1557 (Base 1)	2556 (Base 0) 2557 (Base 1)	3556 (Base 0) 3557 (Base 1)	4556 (Base 0) 4557 (Base 1)	Read/Write
Statistics Address	1600 (Base 0) 1601 (Base 1)	2600 (Base 0) 2601 (Base 1)	3600 (Base 0) 3601 (Base 1)	4600 (Base 0) 4601 (Base 1)	Read/Write

2.2.3. Ethernet Device Raw/ASCII Interface

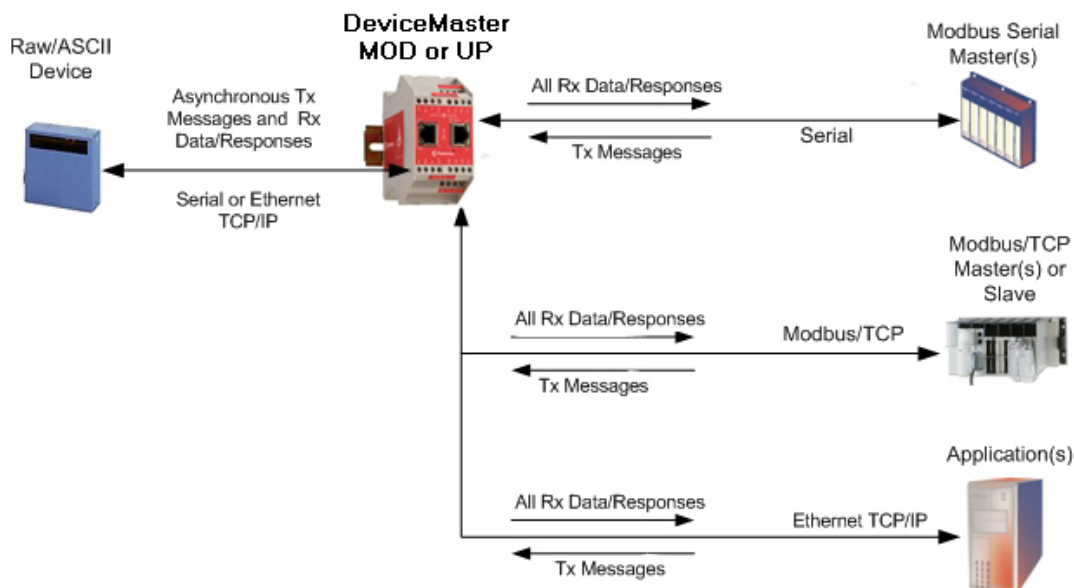
Socket Port Raw Data Addressing	Socket Port 1	Socket Port 2	Socket Port 3	Socket Port 4	Access Rule
Unit ID	254 (FE hex)	254 (FE hex)	254 (FE hex)	254 (FE hex)	N/A
Receive Data Address	1000 (Base 0) 1001 (Base 1)	2000 (Base 0) 2001 (Base 1)	3000 (Base 0) 3001 (Base 1)	4000 (Base 0) 4001 (Base 1)	Read Only
Receive Data Sequence Number Address	1256 (Base 0) 1257 (Base 1)	2256 (Base 0) 2257 (Base 1)	3256 (Base 0) 3257 (Base 1)	4256 (Base 0) 4257 (Base 1)	Read/Write
Transmit Data Address	1300 (Base 0) 1301 (Base 1)	2300 (Base 0) 2301 (Base 1)	3300 (Base 0) 3301 (Base 1)	4300 (Base 0) 4301 (Base 1)	Read/Write
Transmit Data Sequence Number Address	1556 (Base 0) 1557 (Base 1)	2556 (Base 0) 2557 (Base 1)	3556 (Base 0) 3557 (Base 1)	4556 (Base 0) 4557 (Base 1)	Read/Write

2.2.4. Raw/ASCII Transfer Modes

The DeviceMaster supports two different raw/ASCII message transfer modes. The default *Data-Stream* mode is the traditional transfer mode that asynchronously transmits messages and returns received data/responses. The *Command/Response* mode provides a *synchronous transfer* mode for sending and returning responses.

2.2.4.1. Data-Stream Transfer Mode

The *Data-Stream transfer* mode is the default transfer mode that asynchronously transmits messages from all Modbus and Application interfaces and returns received data/responses to all Modbus and Application interfaces. This mode is typically used in installations that utilize only one controller and for receive-only devices such as barcode scanners, RFID readers, weigh scales, and position encoders.

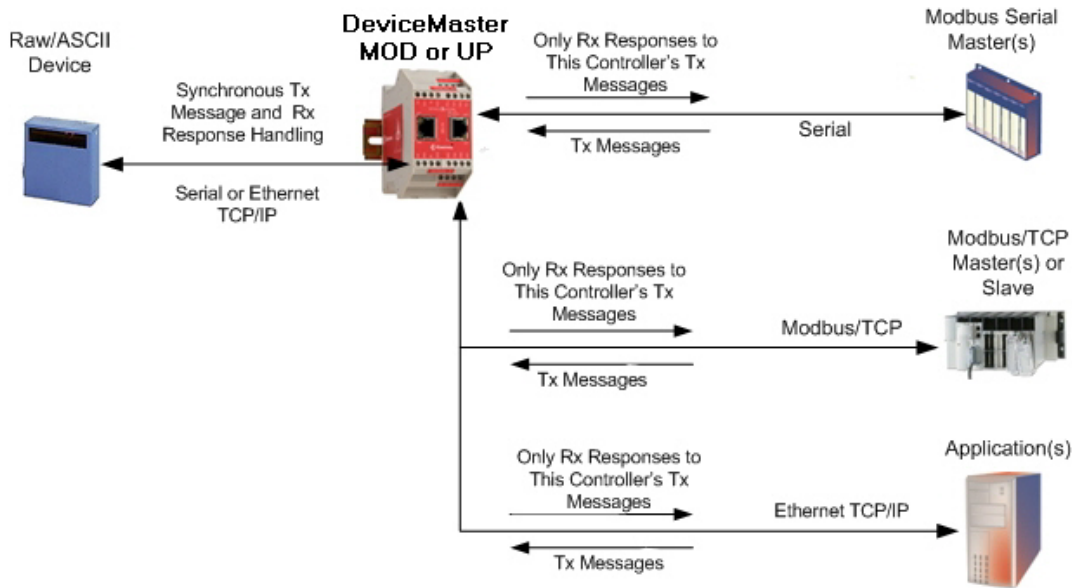


2.2.4.2. Command/Response Mode

The *Command/Response* mode provides the following functionality:

- A *synchronous transfer* mode for sending and returning responses from all Modbus and Application Ethernet TCP/IP interfaces to serial and Ethernet TCP/IP devices.
- Only one command message is transmitted at a time. Command messages are queued if a command message is active.
- Responses are routed only to the message sender.
- Responses are timed out and old responses, (ones not requested within a certain time frame), destined for the Modbus interface are discarded.
- The expected response count is configurable. While this is typically one, some devices may return multiple responses per message.

The *Command/Response transfer* mode is typically required in installations that require multiple controllers sending raw/ASCII messages with expected responses, and it is desired that each controller only receive its own responses.



2.2.5. Receive Data Message (Raw Data)

The following topics are discussed:

- [Format](#) on Page 21
- [Communication Methodology \(Receive Raw Data in Slave Mode\)](#) on Page 22
- [Communication Methodology \(Receive Data Master Mode\)](#) on Page 22

2.2.5.1. Format

The *Receive Data* message for raw data contains a simple protocol including a sequence number, length and serial data fields. The Modbus standard requires a WORD format.

The following table displays the format of the *Receive Data* message.

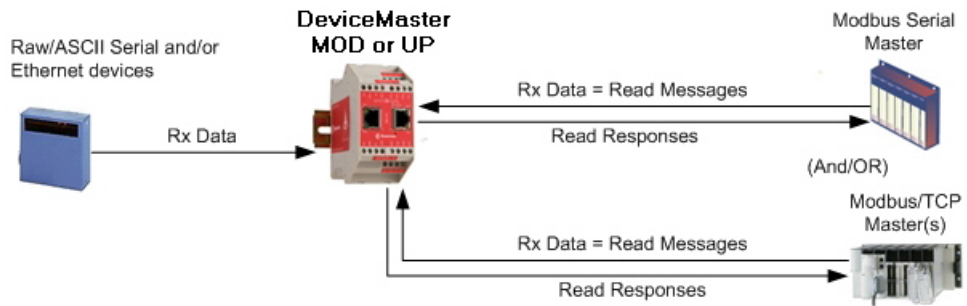
Name	Data Type	Data Value(s)	Access Rule
Receive (DeviceMaster to PLC) message data			Read only
Structure of:			
Produced data sequence	WORD	0-65535 (FFFF hex)	
Data length (in bytes)	WORD	0-246 (slave Rx mode)	
Data array	Array of WORD	1-1024 (serial Rx master) 1-2048 (Ethernet Rx master)	

Receive messages have the following characteristics:

- It returns all data in WORDs.
- The DeviceMaster increments the sequence number when it returns new data.
- The message received from the PLC determines the actual length of the Modbus message returned to the PLC. (This is often greater than the length of the actual number of valid bytes in the Receive Data Message.)
- All unused bytes in a Modbus message returned to the PLC are filled with zeros.
- The default order of the bytes is Least Significant Byte First. However, you can select the *Rx MS Byte First* option in the web page to return bytes by **Most Significant Byte First**. For more information, see *Rx MS Byte First* under [4.4. Port Modbus Configuration \(Raw/ASCII Only\) Page](#) on Page 48.

2.2.5.2. Communication Methodology (Receive Raw Data in Slave Mode)

Raw serial and/or EtherNet device data is returned in the response to the *Read Holding Registers* message or, optionally, the *Read/Write Multiple Register* message. The data is requested by accessing the corresponding receive data address for the desired port.

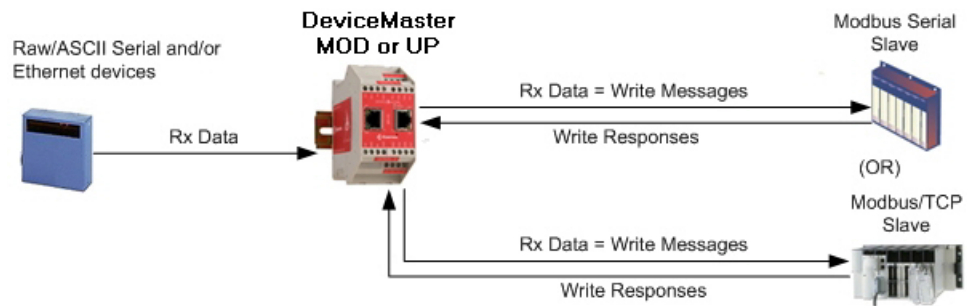


The following restrictions apply to this method:

- The *Device Index* must be 255 (FF hex) for raw/ASCII serial data and 254 (FE hex) for raw/ASCII Ethernet device data.
- The variable to receive the data on the PLC must be:
 - In the 40xxxx address range. (For Modicon type PLCs.)
 - An array of 16 bit words.
 - Of sufficient size to contain the sequence number, length, and data field associated with the received data structure. For more information, see the [2.2.5. Receive Data Message \(Raw Data\)](#) definition on Page 21.
- New data is indicated with an incremented sequence number.
The same data may be returned more than once. However, the same data packet is also return the same sequence number.
- No data is indicated with a length of zero.

2.2.5.3. Communication Methodology (Receive Data Master Mode)

Raw serial and/or Ethernet device data is written to the PLC at the configured address.



The following restrictions apply to this method:

- The *Device Index* must be configured for the target PLC.
- The variable to receive the data on the PLC must be:
 - In the 40xxxx. (For Modicon type PLCs.)
 - An array of 16 bit words.
 - Of sufficient size to contain the sequence number, length, and data field associated with the received data structure.
- New data is indicated with an incremented sequence number.

2.2.6. Transmit Data Message (Raw Data)

The following topics are discussed:

- [Format](#) on Page 23
- [Communication Methodology \(Transmit Raw Data Slave Mode\)](#) on Page 24
- [Communication Methodology \(Transmit Data Master Mode\)](#) on Page 24

2.2.6.1. Format

The *Transmit Data* message for raw data contains a simple protocol including a sequence number, length and serial data fields. The Modbus standard requires a WORD format.

The following table displays the format of the *Transmit Data* message.

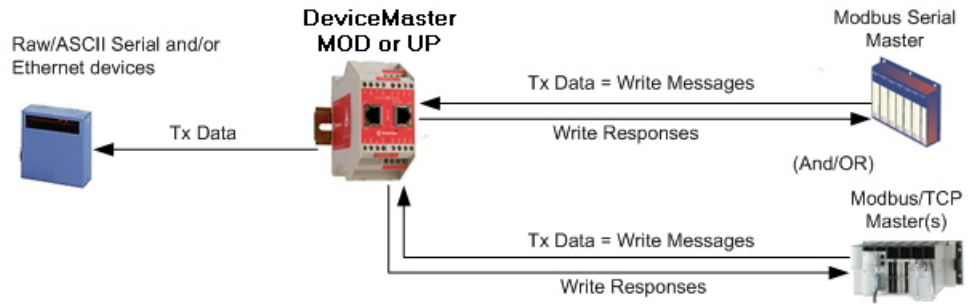
Name	Data Type	Data Value(s)	Access Rule
Transmit (PLC to DeviceMaster) message data			Read/Write
Structure of:			
Produced data sequence	WORD	0-65535 (FFFF hex)	
Data length (in bytes)	WORD	1-236 (Slave Tx Mode) 1-246 (Master Tx Mode)	
Data array	Array of WORD	0-65535	

Transmit messages have the following characteristics:

- It transfers all data in WORDs.
- If the **Disable Tx Sequence Number Check** option is not selected, the sequence number must be incremented when there is new data to transmit.
- The data length field indicates the number of valid bytes contained in the message.
- The actual length of a message received from the PLC may contain extra, unused data.
- It ignores all unused bytes in a Modbus message.
- The default order of the bytes is **Least Significant Byte First**. However, you can select the *Tx MS Byte First* option in the web page to transmit bytes by **Most Significant Byte First**. For more information, see *Tx MS Byte First* under [4.4. Port Modbus Configuration \(Raw/ASCII Only\) Page](#) on Page 48.
- A request for the *Transmit data* returns the last transmit data message.

2.2.6.2. Communication Methodology (Transmit Raw Data Slave Mode)

Raw serial and/or EtherNet device data is sent in the *Write Multiple Registers* message or, optionally, the *Read/Write Multiple Register* message. The data is requested by accessing the corresponding transmit data address for the desired port.

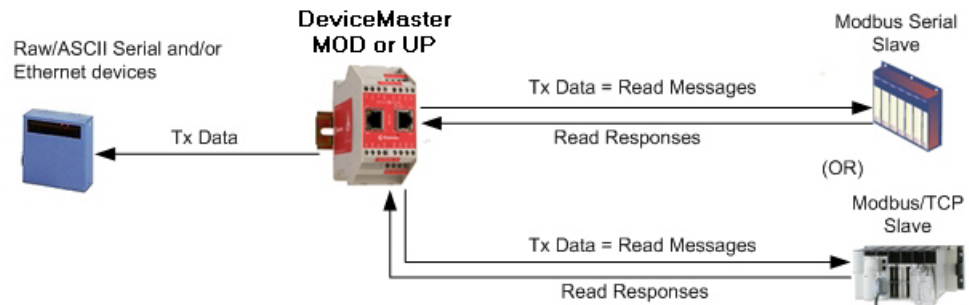


The following restrictions apply to this method:

- The *Device Index* must be 255 (FF hex) for raw/ASCII serial data and 254 (FE hex) for raw/ASCII Ethernet device data.
- The variable to transmit the data on the PLC must be:
 - In the 40xxxx address range. (For Modicon type PLCs.)
 - An array of words.
 - Of sufficient size to contain the sequence number, length, and data field associated with the transmit data structure, typically 128 words. See [2.2.6. Transmit Data Message \(Raw Data\)](#) on Page 23 for more information.
- If the **Disable Tx Sequence Number Check** option is not selected, the sequence number must be incremented when there is new data to transmit. The same transmit data message may be sent to the DeviceMaster more than once. However, the data packet is only transmitted when a new sequence number is received.

2.2.6.3. Communication Methodology (Transmit Data Master Mode)

Raw serial and/or Ethernet transmit data is polled from the PLC at the configured address and, when the DeviceMaster receives a transmit message with an updated sequence number, the data is transmitted to the serial or Ethernet device.



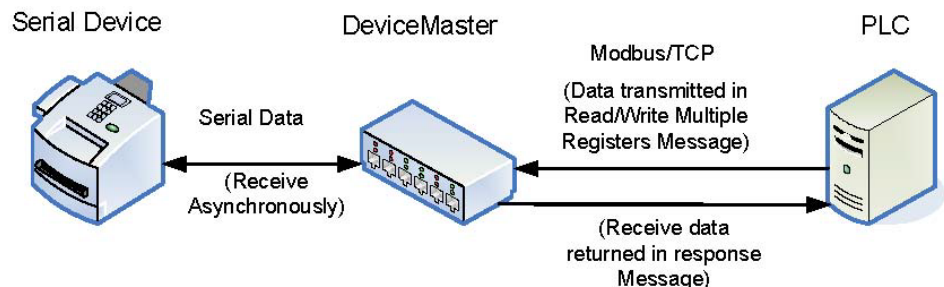
- The following restrictions apply to this method:
- The *Device Index* must be configured for the target PLC.
- The variable to receive the data on the PLC must be:
 - In the 40xxxx. (For Modicon type PLCs.)
 - An array of 16 bit words.
 - Of sufficient size to contain the sequence number, length, and data field associated with the transmit data structure.
- The PLC indicates new data to transmit with an incremented sequence number. (The **Disable Tx Sequence Number Check** option does not apply to *transmit data master* mode.)
- The length indicates the number of bytes to transmit.
- The DeviceMaster expects the length parameter and data to transmit to be updated before the transmit sequence number is incremented. Therefore, as soon as the DeviceMaster receives an incremented transmit number, it transmits the data to the serial or Ethernet device.

2.2.7. Sequence Number Messages (Raw Data)

Read Holding Registers and *Write Multiple Register* messages can read and modify both receive and transmit produced data sequence numbers. These are the same sequence numbers returned to the PLC in the *Receive Data Message* and sent to the DeviceMaster in the *Transmit Data* message. Access to these sequence numbers are provided primarily for initialization purposes at the start of the PLC program when you may want to initialize the sequence numbers on the PLC, DeviceMaster or both.

2.3. I/O Scanner (Raw Data)

The I/O Scanner is an optional PLC communications method that is implemented on some PLC programming software such as the Concept programming package for use with the Schneider Electric Modicon PLCs. The I/O Scanner provides a rather simple method that requires minimal programming effort. It automatically performs the polling and transmitting of data at set time intervals and typically utilizes the *Read / Write multiple Registers* message.



The following restrictions apply to this method:

- The *Receive* and *Transmit* mode for the serial and/or Ethernet device must both be set to *Slave* mode.
- The *Device Index* must be 255 (FF hex) for raw/ASCII serial data and 254 (FE hex) for raw/ASCII Ethernet device data.
- The variable to receive the data on the PLC must be:
 - In the 40xxxx address range. (For Modicon type PLCs.)
 - An array of words.
 - Of sufficient size to contain the sequence number, length, and data field associated with the received data structure, typically 128 words. For more information, see the [2.2.5. Receive Data Message \(Raw Data\)](#) definition on Page 21.
- New received data is indicated with an incremented sequence number.

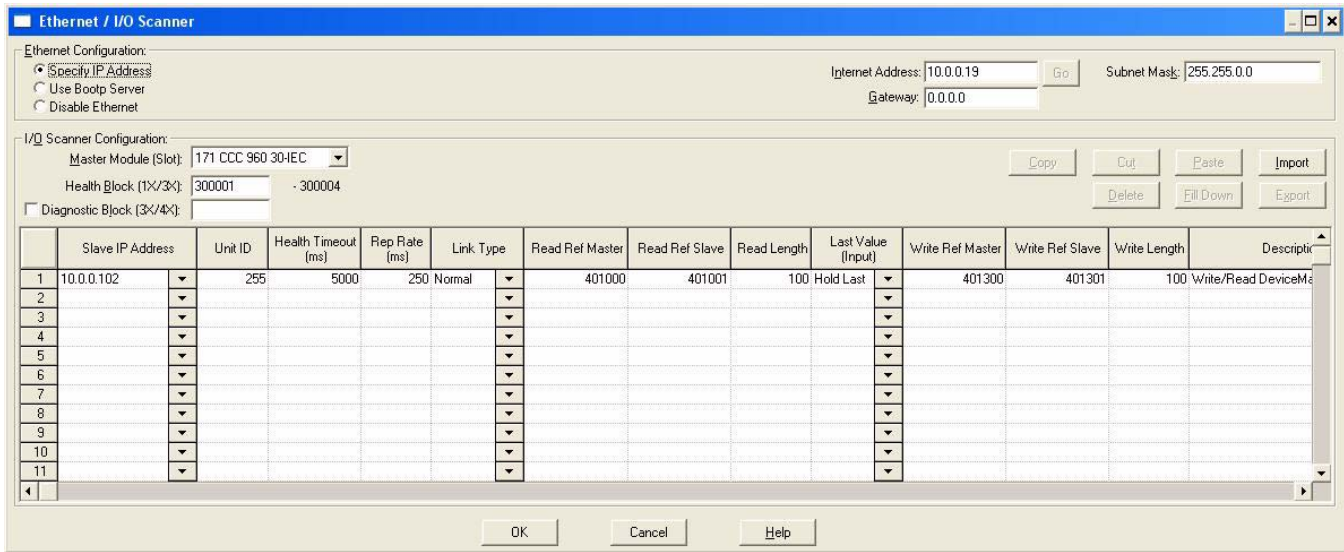
The same data may be returned more than once. However, the same data packet returns the same sequence number.

- No receive data is indicated with a length of zero.
- The variable to transmit the data on the PLC must be:
 - In the 40xxxx address range. (For Modicon PLCs.)
 - An array of words.
 - Of sufficient size to contain the sequence number, length, and data field associated with the transmit data structure, typically 128 words. See [2.2.6. Transmit Data Message \(Raw Data\)](#) on Page 23 for more information.
- If the **Disable Tx Sequence Number Check** option is not selected, the sequence number must be incremented when there is new data to transmit.

The same transmit data message may be sent to the DeviceMaster more than once. However, the data packet is only transmitted when a new sequence number is received.

- The DeviceMaster should be reset before starting a PLC program using the I/O Scanner due to PLC program execution scheduling. If the DeviceMaster is not reset, the sequence numbers may be out of sync. This may result in receiving outdated serial data as well as an unexpected transmission of serial data. A **Transmit Unexpected Sequence Number** error may also occur.

The following depicts a typical I/O Scanner screen.



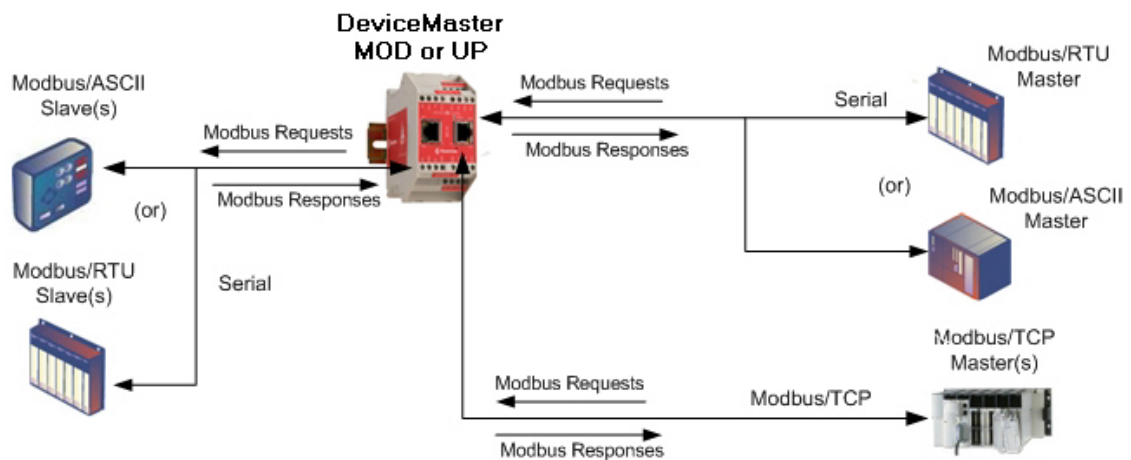
2.4. Modbus/RTU and Modbus/ASCII To-Slaves Protocol Interface

The DeviceMaster provides access to serial Modbus/RTU and Modbus/ASCII slave devices via Modbus/TCP, serial Modbus/RTU masters, and serial Modbus/ASCII masters. Modbus master messages are translated to Modbus/RTU or Modbus/ASCII messages, devices are automatically located, and appropriate Modbus responses are returned to the Modbus masters.

2.4.1. Communication Methodology

The DeviceMaster translates Modbus master messages into Modbus/RTU or Modbus/ASCII messages and forwards them to slave devices attached to the Modbus/RTU or Modbus/ASCII slave serial ports. Each Modbus message is transmitted and a response is expected. The DeviceMaster times out the Modbus/RTU or Modbus/ASCII messages if there is no response returned within the configured timeout period.

The following diagram displays the Modbus message transfer.



The following apply to Modbus slaves serial ports.

- All valid Modbus master messages are translated to Modbus slave messages for serial port transmission.
- Modbus slave devices are automatically located on a DeviceMaster 2-port or 4-port.
- Messages are timed out if no response is returned within the configured timeout period.
- Appropriate Modbus responses are returned to the Modbus master.
- Broadcast Modbus messages, those with a unit identifier of zero, are transmitted out all Modbus slave serial ports on the DeviceMaster.

The following restrictions apply to the Modbus slave interface:

- The DeviceMaster serves as a *slave* Modbus/TCP device, a *master* on Modbus To-Slaves serial ports, and a *slave* on Modbus To-master serial ports.
- All Modbus slave devices attached to a DeviceMaster gateway (1, 2, or 4-port) must have unique *Unit Identifiers*. Valid Unit Identifiers are 1 to 247 and the Broadcast Identifier is zero.

To communicate to Modbus slave device(s) through a DeviceMaster, perform the following steps.

1. Using the embedded web page, select the appropriate **Port**.
2. Under *Serial Configuration*, configure the serial port parameters such as the Mode, Baud rate, Data Bits, and so forth.
3. Under *General Protocol Settings*, set the *Select Serial Port Protocol* to **Modbus slave**.
4. Under *Modbus Slave Protocol Settings*, set the **Device Response Timeout** to the desired value.

Note: 2- and 4-Port only: set the *Lost Device Search Enable* setting. For a discussion on this setting, see [2.4.2. Modbus Slave Device Search Methodology](#) on Page 28.

5. In the PLC program, address messages to the Modbus slave device using the IP Address of the DeviceMaster and the Unit Identifier of the slave device(s).

2.4.2. Modbus Slave Device Search Methodology

Locating a Modbus slave device on a DeviceMaster 1-port is relatively simple. Either the Modbus slave device is connected to the port or it is not. However, if more than one port is configured for Modbus slave on a DeviceMaster 2- or 4-port, the device must be found. The following is an explanation of how the search algorithm works on a DeviceMaster 2- or 4-port.

Locating a Modbus slave device after a reboot or port reset:

When the DeviceMaster receives a message for a Modbus slave device for the first time since reboot or port initialization, it transmits the Modbus slave message out all Modbus slave serial ports and wait for a response to be returned. Once the response is returned, the device port is known and all messages sent to the device are routed through the serial port.

Lost Devices:

Lost devices, or devices that time out, are a special case. The DeviceMaster provides two methods for handling lost devices via the **Lost Device Search Enable** option on the web page.

- Disabling this option on a Modbus slave port:
 - Prevents the DeviceMaster from searching for a lost device on other Modbus slave ports.
 - Prevents lost devices known to have been on other ports from being searched for on this port.

Note: *This is the recommended setting whenever it is desired to prevent timeout delays on other Modbus slave ports in the event that a device times out.*
- Enabling this option on a Modbus slave port:
 - Allows the DeviceMaster to search for lost devices on all Modbus slave ports with the **Lost Device Search Enable** option turned on.

Note: *This can be useful for locating devices if a device has been moved onto another port by moving the serial cable or, perhaps, by moving the device onto a different Modbus slave serial loop.*

 - This causes timeout delays on all Modbus slave ports with the **Device Search Enable** option turned on until the device is found.

2.5. Retrieve Statistics Message

The data returned from the *Retrieve Statistics* message contains various counters. The *Retrieve Statistics* message formats the data into 32-bit integers and returns data in an array of WORDs. The first WORD contains the most significant word and the second WORD contains the least significant word.

DeviceMaster Addressing for the Statistics Modbus/TCP Messages	Serial Port 1	Serial Port 2	Serial Port 3	Serial Port 4	Access Rule
Unit ID	255 (FF hex)	255 (FF hex)	255 (FF hex)	255 (FF hex)	N/A
Statistics Address	1600 (Base 1) 1599 (Base 0)	2600 (Base 1) 2599 (Base 0)	3600 (Base 1) 3599 (Base 0)	4600 (Base 1) 4599 (Base 0)	Read Only

Note: Some Modicon PLC programming software, such as Concept, requires one to be added to the address offset. This is because their address range begins at 40001, while the address range on the DeviceMaster begins at zero.

The following table displays the format of the *Retrieve Statistics* message.

Index	Name	Data Type	Data Value(s)	Access Rule
1	Receive Byte Count	UDINT	0=default	Read only
2	Receive Packet Count	UDINT	0=default	Read only
3	Transmit Byte Count	UDINT	0=default	Read only
4	Transmit Packet Count	UDINT	0=default	Read only
5	Dropped Packet Count to PLC	UDINT	0=default	Read only
6	Parity Error Count	UDINT	0=default	Read only
7	Framing Error Count	UDINT	0=default	Read only
8	Overrun Error Count	UDINT	0=default	Read only
9	Unexpected Transmit Sequence Number errors	UDINT	0=default	Read only
10	Invalid Modbus/RTU Device Responses	UDINT	0=default	Read only
11	Modbus/RTU Device Timeouts	UDINT	0=default	Read only
12	Reserved	UDINT	0	Read only

The *Retrieve Statistics* messages have the following characteristics.

Retrieve Statistics Message Description	
Receive Byte Count	This attribute counts the number of bytes received on the serial port.
Receive Packet Count	This attribute counts the number of packets received on the serial port.
Transmit Byte Count	This attribute counts the number of bytes transmitted on the serial port.
Transmit Packet Count	This attribute counts the number of packets transmitted on the serial port.
Dropped Packet Count to PLC	This attribute counts the number of dropped receive packets on the serial port intended for the PLC due to: <ul style="list-style-type: none"> No STX byte(s) found No ETX byte(s) found Time-outs Too large of packet Receive buffer queue overflows

Retrieve Statistics Message Description (Continued)	
Parity Error Count	This attribute counts the number of packets with parity errors received on the serial port.
Framing Error Count	This attribute counts the number of packets with framing errors received on the serial port.
Overrun Error Count	This attribute counts the number of packets with overrun type errors received on the serial port.
Unexpected Transmit Sequence Number Error Count	This attribute counts the number of Unexpected Transmit Sequence Number errors. The DeviceMaster increments this number when it receives a raw data transmit message with a sequence number that is not equal to either the previous sequence number or the previous sequence number plus one. (The DeviceMaster expects this sequence number to be incremented by one with each new transmit message.)
Invalid Modbus/RTU Device Responses	The number of invalid messages returned from Modbus/RTU devices on this port. Such invalid responses could be the result of any or all of the following: <ul style="list-style-type: none"> • Invalid CRC • Invalid returned function code • Invalid Unit Identifier • Duplicate Unit Identifier
Modbus/RTU Device Timeouts	The number of messages that timed out waiting for a response from a Modbus/RTU device on this port.

Chapter 3. Configuration Overview

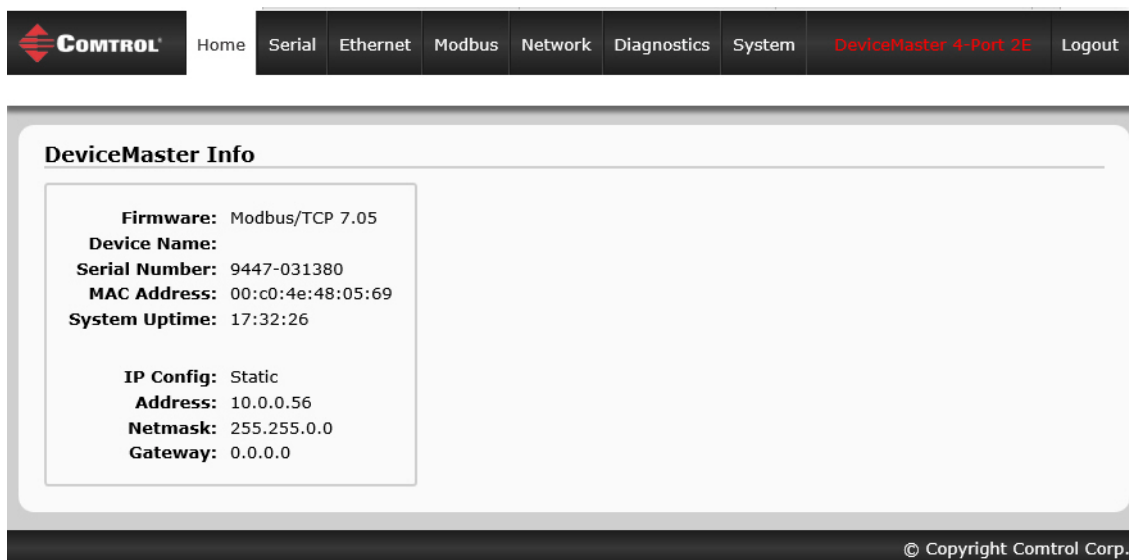
The latest Modbus/TCP firmware must be installed before you can configure network or serial/socket port characteristics. For firmware installation and setup information, see the [Modbus/TCP Hardware Installation and Configuration Guide](#) or the PortVision DX help system.

Use the [Modbus/TCP Interface Configuration Quick Start](#) to locate detailed configuration procedures for your site. Use the following chapters as a reference if you need information about specific fields. The *Interface Configuration Quick Start* is intended to provide you with a way to quickly configure DeviceMaster for your devices.

3.1. Home Page

If you have not configured the network information into the DeviceMaster during initial setup, you must configure the network information before configuring serial/socket port characteristics. See the [Modbus/TCP Hardware Installation and Configuration Guide](#) or the PortVision DX help system for help configuring the network settings.

From PortVision DX, highlight the DeviceMaster that you want to configure and click **Webpage** or enter the IP address of the DeviceMaster in the **Address** box of your web browser. The **Home** page displays.



Modbus/TCP Home Page	
Firmware	Modbus/TCP firmware version currently running on the DeviceMaster.
Device Name	You can enter a Device Name in the <i>Network Configuration</i> page, which will display in this field.
Serial Number	The DeviceMaster serial number.
MAC Address	This is the MAC address of this DeviceMaster, which is located on the compliance label on the DeviceMaster.

Modbus/TCP Home Page (Continued)	
System Uptime	Displays how long this DeviceMaster has been on-line since powered on or rebooted.
IP Config	Type of IP configuration currently in use (static or DHCP).
IP Address, IP Netmask, IP Gateway	IP address, netmask, and gateway configured in the DeviceMaster.

Use the appropriate chapter as a reference for the configuration options.

- [Chapter 4. Serial Menus](#) on Page 41
- [Chapter 5. Ethernet Menus](#) on Page 59
- [Chapter 6. Modbus Menus](#) on Page 71

You can review the configuration pages and basic procedures in the following subsections.

- [3.2. Serial Port - Configuration Overview](#)
- [3.3. Ethernet Device - Configuration Overview](#) on Page 37

3.2. Serial Port - Configuration Overview

Use the following steps to access the serial port configuration pages.

1. Open the DeviceMaster web page using the IP address in your browser or PortVision DX.
2. Click the **Serial** menu.
3. Click the port number that you want to configure and the *Serial Settings | Port Configuration* page appears.

The screenshot displays the 'Port 1 Serial Configuration' page in the DeviceMaster web interface. The navigation bar at the top includes 'Home', 'Serial', 'Ethernet', 'Modbus', 'Network', 'Diagnostics', 'System', 'DeviceMaster 4-Port 2E', and 'Logout'. Below the navigation bar, there are tabs for 'Overview', 'Port 1', 'Port 2', 'Port 3', and 'Port 4'. The 'Serial Settings' tab is active, showing sub-tabs for 'Raw/ASCII Settings', 'Modbus Settings', 'Filtering', and 'Application Interface'. The main content area is titled 'Port 1 Serial Configuration' and is divided into two columns. The left column, 'Serial Configuration', contains the following settings: Port Name (empty text box), Port Mode (RS-232 dropdown), Baud Rate (9600 dropdown), Parity (none dropdown), Data Bits (8 dropdown), Stop Bits (1 dropdown), Flow Control (none dropdown), DTR Mode (off dropdown), Rx Timeout Between Packets (ms) (200 text box), and Discard Rx Pkts With Errors (checked checkbox). The right column, 'General Protocol Settings', contains: Serial Port Protocol (Raw-Data dropdown), Modbus Slave and Raw/ASCII Data Device Settings (Response Timeout (ms) 250 text box), and Modbus Slave Settings (Lost Device Search Enable unchecked checkbox). At the bottom left, there is a checkbox for 'Clone settings to all serial ports'. At the bottom right, there is a 'Save' button. The footer of the page reads '© Copyright Control Corp.'

4. Change the serial port configuration properties (Page 41) as required for your site.
5. Click the **Save** button.

6. If desired, click the **Raw/ASCII Settings** menu (Page 45) and configure the settings for your site.

The screenshot shows the CONTROL web interface for configuring a serial port. The top navigation bar includes 'Home', 'Serial', 'Ethernet', 'Modbus', 'Network', 'Diagnostics', and 'System'. The 'Serial' menu is active, and the 'Raw/ASCII Settings' sub-menu is selected. The main content area is titled 'Port 1 Raw/ASCII Configuration' and contains the following sections:

- Raw/ASCII Data Control:** Includes 'Raw/ASCII Data Message Transfer Mode' (set to Data-Stream), 'Cmd/Resp Age Time, Discard Responses After (sec):' (10), 'Cmd/Resp Expected Responses Per Command:' (1), and 'Cmd/Resp Mode Response To Modbus/TCP Based On:' (IP-Address).
- Serial Packet Identification:** Includes 'STX (Start of Transmission) Rx Detect:' (none, Byte 1: 0, Byte 2: 0) and 'ETX (End of Transmission) Rx Detect:' (none, Byte 1: 0, Byte 2: 0).
- PLC Specific Settings:** Includes 'STX (Start of Transmission) Tx Append:' (none, Byte 1: 0, Byte 2: 0), 'ETX (End of Transmission) Tx Append:' (none, Byte 1: 0, Byte 2: 0), and 'Strip Rx STX/ETX:' (checked).
- Application Specific Settings:** Includes 'STX (Start of Transmission) Tx Append:' (none, Byte 1: 0, Byte 2: 0), 'ETX (End of Transmission) Tx Append:' (none, Byte 1: 0, Byte 2: 0), and 'Strip Rx STX/ETX:' (unchecked).

At the bottom left, there is a checkbox for 'Clone settings to all serial ports'. At the bottom right, there is a 'Save' button. The footer of the interface reads '© Copyright Control Corp.'

7. Click the **Save** button.

- Click **Modbus Settings** (Page 48) to configure Modbus for this DeviceMaster.

The screenshot displays the DeviceMaster configuration interface. At the top, there is a navigation bar with the following tabs: Home, Serial, Ethernet, Modbus, Network, Diagnostics, System, DeviceMaster 4-Port 2E, and Logout. Below this, there are sub-tabs for Serial Settings, Raw/ASCII Settings, Modbus Settings (which is highlighted), Filtering, and Application Interface. The main content area is titled "Port 1 Modbus Configuration (Raw/ASCII Only)".

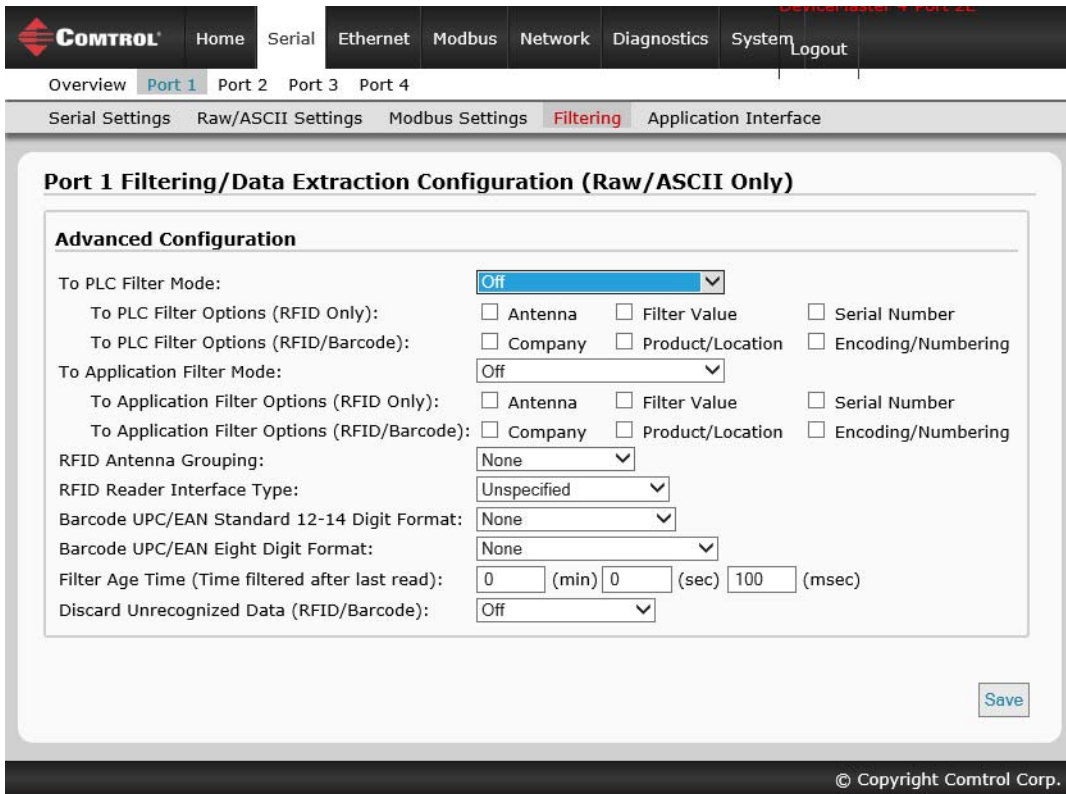
The configuration is divided into two main sections:

- Serial Modbus Master and Modbus/TCP Interface Settings:**
 - Rx (To PLC) Transfer Mode: Slave (PLC Polls) (dropdown)
 - Tx (From PLC) Transfer Mode: Slave (PLC Writes) (dropdown)
 - Maximum Rx Data Packet Size: 246 (input field)
 - Oversized Rx Packet Handling: Truncate (dropdown)
 - Rx MS Byte First:
 - Tx MS Byte First:
 - Disable Non-Filtered To PLC Rx Queue:
 - Disable Tx Sequence Number Check:
- Modbus/TCP Master Rx/Tx Settings:**
 - Modbus/TCP Master Rx/Tx Settings:**
 - PLC IP Address: 0.0.0.0 (input field)
 - PLC Device ID: 1 (input field)
 - Disable Sequence Number/Length Header:
 - Master Rx Mode Only:**
 - PLC Rx Data Address (Base 1): 1 (input field)
 - Maximum PLC Update Rate (msec): 40 (input field)
 - Use Maximum Sized Modbus Messages:
 - Master Tx Mode Only:**
 - PLC Tx Data Address (Base 1): 1 (input field)
 - PLC Tx Poll Rate (msec): 100 (input field)
 - PLC Tx Poll Message Length (bytes): 0 (input field)
 - Tx Sequence Number Syncing Enable:
 - PLC Tx Consumed Seq Nbr Address (Base 1): 1 (input field)

A "Save" button is located at the bottom right of the configuration area. The footer of the page reads "© Copyright Control Corp."

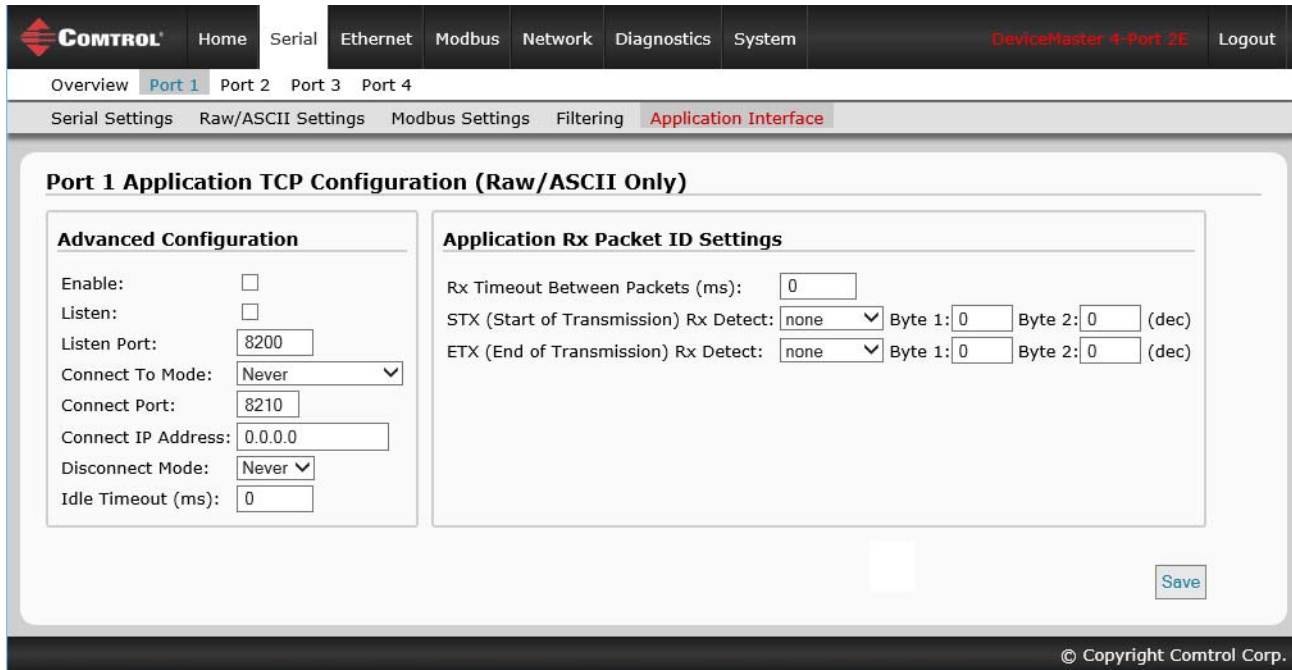
- Make the changes for your site and click the **Save** button.

10. If required for your site, click **Filtering** and change the filtering configuration (Page 51) to meet your needs.



11. Click the **Save** button when you have completed the changes.

12. If required for your site, click **Application Interface** and change the configuration (Page 55) to meet your needs.



13. Click the **Save** button when you have completed the changes.

3.3. Ethernet Device - Configuration Overview

Use the following overview to configure an Ethernet device.

1. Click the **Ethernet** menu.
2. Click the appropriate **Device** number to access the *Device Interface Configuration* page for that port.
3. Change the socket configuration properties (Page 61) as required for your site.

The screenshot displays the web interface for configuring an Ethernet device. The top navigation bar includes the 'COMTROL' logo and menu items: Home, Serial, Ethernet, Modbus, Network, Diagnostics, System, DeviceMaster 4-Port 2E, and Logout. Below this, a secondary navigation bar shows 'Overview', 'Device 1', 'Device 2', 'Device 3', and 'Device 4'. The main content area is titled 'Device 1 Interface Configuration' and contains a 'Socket Configuration' section with the following fields:

Device Name:	<input type="text"/>
Enable:	<input type="checkbox"/>
Listen:	<input type="checkbox"/>
Listen Port:	<input type="text" value="8000"/>
Connect To Mode:	<input type="text" value="Never"/>
Connect Port:	<input type="text" value="8010"/>
Connect IP Address:	<input type="text" value="0.0.0.0"/>
Disconnect Mode:	<input type="text" value="Never"/>
Idle Timer (ms):	<input type="text" value="0"/>
Rx Timeout Between Packets (ms):	<input type="text" value="0"/>

A 'Save' button is located at the bottom right of the configuration area. The footer of the page reads '© Copyright Comtrol Corp.'

4. Click the **Save** button after completing the changes.

5. If necessary, click **Raw/ASCII Settings**.

The screenshot shows the 'Device 1 Raw/ASCII Configuration' page. The navigation bar includes 'CONTROL', 'Home', 'Serial', 'Ethernet', 'Modbus', 'Network', 'Diagnostics', 'System', 'DeviceMaster 4-Port 2E', and 'Logout'. The breadcrumb trail is 'Overview > Device 1 > Device 2 > Device 3 > Device 4'. The sub-menu is 'Socket Connection > Raw/ASCII Settings > Modbus/TCP Settings > Filtering > Application Interface'. The main content area is titled 'Device 1 Raw/ASCII Configuration' and contains two columns of settings:

- Raw/ASCII Data Control:**
 - Raw/ASCII Data Message Transfer Mode: Data-Stream (dropdown)
 - Cmd/Resp Response Timeout (ms): 200 (input)
 - Cmd/Resp Age Time, Discard Responses After (sec): 10 (input)
 - Cmd/Resp Expected Responses Per Command: 1 (input)
 - Cmd/Resp Mode Response To Modbus/TCP Based On: IP-Address (dropdown)
- Device Socket Packet ID Settings:**
 - STX (Start of Transmission) Rx Detect: one byte (dropdown), Byte 1: 2 (input), Byte 2: 0 (input) (dec)
 - ETX (End of Transmission) Rx Detect: one byte (dropdown), Byte 1: 3 (input), Byte 2: 0 (input) (dec)
- PLC Specific Settings:**
 - STX (Start of Transmission) Tx Append: none (dropdown), Byte 1: 0 (input), Byte 2: 0 (input) (dec)
 - ETX (End of Transmission) Tx Append: none (dropdown), Byte 1: 0 (input), Byte 2: 0 (input) (dec)
 - Strip Rx STX/ETX:
- Application Specific Settings:**
 - STX (Start of Transmission) Tx Append: none (dropdown), Byte 1: 0 (input), Byte 2: 0 (input) (dec)
 - ETX (End of Transmission) Tx Append: none (dropdown), Byte 1: 0 (input), Byte 2: 0 (input) (dec)
 - Strip Rx STX/ETX:

At the bottom left, there is a checkbox 'Clone settings to all Ethernet devices'. At the bottom right, there is a 'Save' button. The footer of the page reads '© Copyright Control Corp.'

- Update the settings (Page 63) to meet your site requirements.
- Click the **Save** button after completing the changes.
- Click **Modbus/TCP Settings** to configure Modbus/TCP for this DeviceMaster.

The screenshot shows the 'Device 1 Modbus Configuration' page. The navigation bar is the same as in the previous screenshot. The breadcrumb trail is 'Overview > Device 1 > Device 2 > Device 3 > Device 4'. The sub-menu is 'Socket Connection > Raw/ASCII Settings > Modbus/TCP Settings > Filtering > Application Interface'. The main content area is titled 'Device 1 Modbus Configuration' and contains two columns of settings:

- Serial Modbus Master and Modbus/TCP Interface Settings:**
 - Rx (To PLC) Transfer Mode: Slave (PLC Polls) (dropdown)
 - Tx (From PLC) Transfer Mode: Slave (PLC Writes) (dropdown)
 - Maximum Rx Data Packet Size: 246 (input)
 - Oversized Rx Packet Handling: Truncate (dropdown)
 - Rx MS Byte First:
 - Tx MS Byte First:
 - Disable Non-Filtered To PLC Rx Queue:
 - Disable Tx Sequence Number Check:
- Modbus/TCP Master Rx/Tx Settings:**
 - PLC IP Address: 0.0.0.0 (input)
 - PLC Device ID: 1 (input)
 - Disable Sequence Number/Length Header:
- Master Rx Mode Only:**
 - PLC Rx Data Address (Base 1): 1 (input)
 - Maximum PLC Update Rate (msec): 40 (input)
 - Use Maximum Sized Modbus Messages:
- Master Tx Mode Only:**
 - PLC Tx Data Address (Base 1): 1 (input)
 - PLC Tx Poll Rate (msec): 100 (input)
 - PLC Tx Poll Message Length (bytes): 0 (input)
 - Tx Sequence Number Syncing Enable:
 - PLC Tx Consumed Seq Nbr Address (Base 1): 1 (input)

At the bottom right, there is a 'Save' button. The footer of the page reads '© Copyright Control Corp.'

9. Make the necessary changes (Page 66) for your environment and then click **Save**.
10. If required for your site, click the **Filtering** menu.
11. Change the parameters (Page 69) to meet your needs.

The screenshot displays the web-based configuration interface for the DeviceMaster 4-Port 2E. The top navigation bar includes the Control logo and menu items: Home, Serial, Ethernet, Modbus, Network, Diagnostics, System, DeviceMaster 4-Port 2E, and Logout. Below this, a secondary navigation bar shows Overview, Device 1 (selected), Device 2, Device 3, and Device 4. A third navigation bar contains Socket Connection, Raw/ASCII Settings, Modbus/TCP Settings, Filtering (selected), and Application Interface.

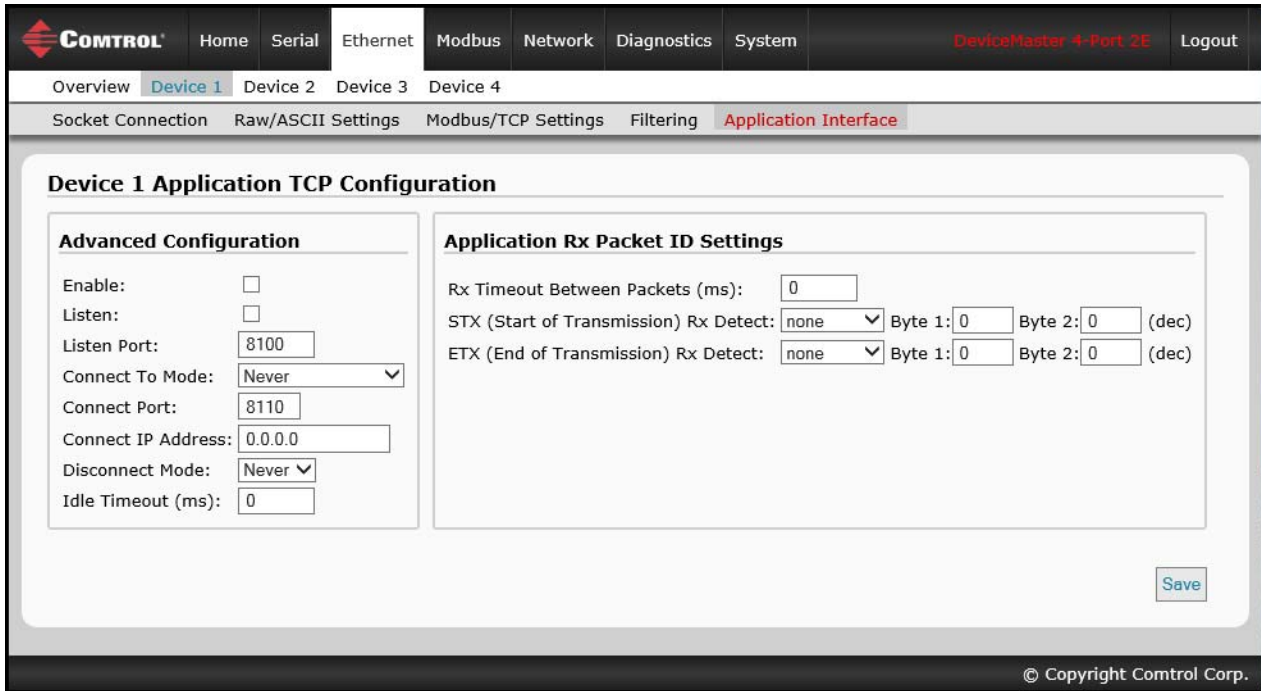
The main content area is titled "Device 1 Filtering/Data Extraction Configuration" and contains an "Advanced Configuration" section with the following settings:

- To PLC Filter Mode: Off (dropdown)
- To PLC Filter Options (RFID Only):
 - Antenna
 - Filter Value
 - Serial Number
- To PLC Filter Options (RFID/Barcode):
 - Company
 - Product/Location
 - Encoding/Numbering
- To Application Filter Mode: Off (dropdown)
- To Application Filter Options (RFID Only):
 - Antenna
 - Filter Value
 - Serial Number
- To Application Filter Options (RFID/Barcode):
 - Company
 - Product/Location
 - Encoding/Numbering
- RFID Antenna Grouping: None (dropdown)
- RFID Reader Interface Type: Unspecified (dropdown)
- Barcode UPC/EAN Standard 12-14 Digit Format: None (dropdown)
- Barcode UPC/EAN Eight Digit Format: None (dropdown)
- Filter Age Time (Time filtered after last read): 0 (min) 0 (sec) 100 (msec)
- Discard Unrecognized Data (RFID/Barcode): Off (dropdown)

A "Save" button is located at the bottom right of the configuration area. The footer of the interface reads "© Copyright Control Corp."

12. Click the **Save** button.

- 13. If required, click the **Application Interface** option.
- 14. Make the necessary changes (Page 70) for your site.



- 15. Click the **Save** button.
- 16. Repeat the previous steps for each serial port and Ethernet device.
- 17. Go to [Appendix A. Programming the PLC via Concept](#) on Page 105 to complete the DeviceMaster installation.

Chapter 4. Serial Menus

This chapter provides detailed information about the following web serial configuration pages:

- [4.1. Serial Port Overview Page](#) on Page 42
- [4.2. Port Serial Configuration Page](#) on Page 43
- [4.3. Port Raw/ASCII Configuration Page](#) on Page 45
- [4.4. Port Modbus Configuration \(Raw/ASCII Only\) Page](#) on Page 48
- [4.5. Port Filtering/Data Extraction Configuration \(Raw/ASCII Only\) Page](#) on Page 51
- [4.6. Application TCP Configuration \(Raw/ASCII Only\) Page](#) on Page 55

Note: *The Diagnostics menu is discussed in [Chapter 8. Diagnostic Menus](#) on Page 83.*

The latest Modbus/TCP firmware must be installed before you can configure serial port characteristics.

Use the [Modbus/TCP Interface Configuration Quick Start](#) to locate configuration procedures for your site and use this chapter as a reference if you need information about specific fields. The *Interface Configuration Quick Start* is intended to provide you with a way to quickly configure DeviceMaster for your devices.

4.1. Serial Port Overview Page

You can access the *Serial Port Overview* page by clicking the **Serial** menu. This page provides an overview of all of the serial settings on the serial configuration pages.

This is the current configuration of the serial port(s). Select a serial port from the menu above to change the configuration.

Serial Port Settings	Port 1	Port 2	Port 3	Port 4
Port Name:	RS-232	RS-232	RS-232	RS-232
Port Mode:	RS-232	RS-232	RS-232	RS-232
Baud Rate:	38400	38400	38400	38400
Parity:	none	none	none	none
Data Bits:	8	8	8	8
Stop Bits:	1	1	1	1
Flow Control:	none	none	none	none
DTR Mode:	off	off	off	off
Rx Timeout Between Packets (ms):	200	200	200	200
Discard Messages With Errors:	yes	yes	yes	yes
General Protocol Settings				
Serial Port Protocol:	Raw-Data	Modbus/ASCII-to-Slaves	Modbus/RTU-to-Slaves	Modbus/RTU-to-Master
Modbus Slave and Raw/ASCII Data Device Settings				
Response Timeout (ms):	N/A	250	250	N/A
Modbus Slave Settings				
Last Device Search Enable:	N/A	no	no	N/A
Raw/ASCII Data Control				
Raw/ASCII Data Message Transfer Mode:	Data-Stream	N/A	N/A	N/A
Cmd/Resp Age Time, Discard Responses After (sec):	N/A	N/A	N/A	N/A
Cmd/Resp Expected Responses Per Command:	N/A	N/A	N/A	N/A
Cmd/Resp Mode Response To Modbus/TCP Based On:	N/A	N/A	N/A	N/A
Serial Packet ID Settings				
STX Rx Detect:	none	N/A	N/A	N/A
STX Rx (dec):		N/A	N/A	N/A
ETX Rx Detect:	none	N/A	N/A	N/A
ETX Rx (dec):		N/A	N/A	N/A
PLC Specific Settings				
STX Tx Append:	none	N/A	N/A	N/A
STX Tx (dec):		N/A	N/A	N/A
ETX Tx Append:	none	N/A	N/A	N/A
ETX Tx (dec):		N/A	N/A	N/A
Strip Rx STX/ETX chars:	yes	N/A	N/A	N/A
Serial Modbus Master and Modbus/TCP Interface Settings				
Rx (To PLC) Transfer Mode:	Slave (PLC Polls)	N/A	N/A	N/A
Tx (From PLC) Transfer Mode:	Slave (PLC Writes)	N/A	N/A	N/A
Maximum Rx Data Packet Size:	246	N/A	N/A	N/A
Oversized Rx Packet Handling:	Truncate	N/A	N/A	N/A
Rx MS Byte First:	no	N/A	N/A	N/A
Tx MS Byte First:	no	N/A	N/A	N/A
Disable Non-Filtered To PLC Rx Queue:	no	N/A	N/A	N/A
Disable Tx Sequence Number Check:	no	N/A	N/A	N/A
Modbus/TCP Master Rx/Tx Settings				
PLC IP Address:	N/A	N/A	N/A	N/A
PLC Device ID:	N/A	N/A	N/A	N/A
Disable Sequence Number/Length Header:	N/A	N/A	N/A	N/A
Master Rx Mode Only				
PLC Rx Data Address (Base 1):	N/A	N/A	N/A	N/A
Maximum PLC Update Rate (msec):	N/A	N/A	N/A	N/A
Use Maximum Sized Modbus Messages:	N/A	N/A	N/A	N/A
Master Tx Mode Only				
PLC Tx Data Address (Base 1):	N/A	N/A	N/A	N/A
PLC Tx Poll Rate (msec):	N/A	N/A	N/A	N/A
PLC Tx Poll Message Length (bytes):	N/A	N/A	N/A	N/A
Tx Sequence Number Syncing Enable:	N/A	N/A	N/A	N/A
PLC Tx Consumed Seq Nbr Address (Base 1):	N/A	N/A	N/A	N/A

© Copyright Comtrol Corp.

4.2. Port Serial Configuration Page

Access the *Port Serial Configuration* page by clicking **Serial | Port | Serial Settings** to configure serial port characteristics for the device that you plan on connecting to the port.

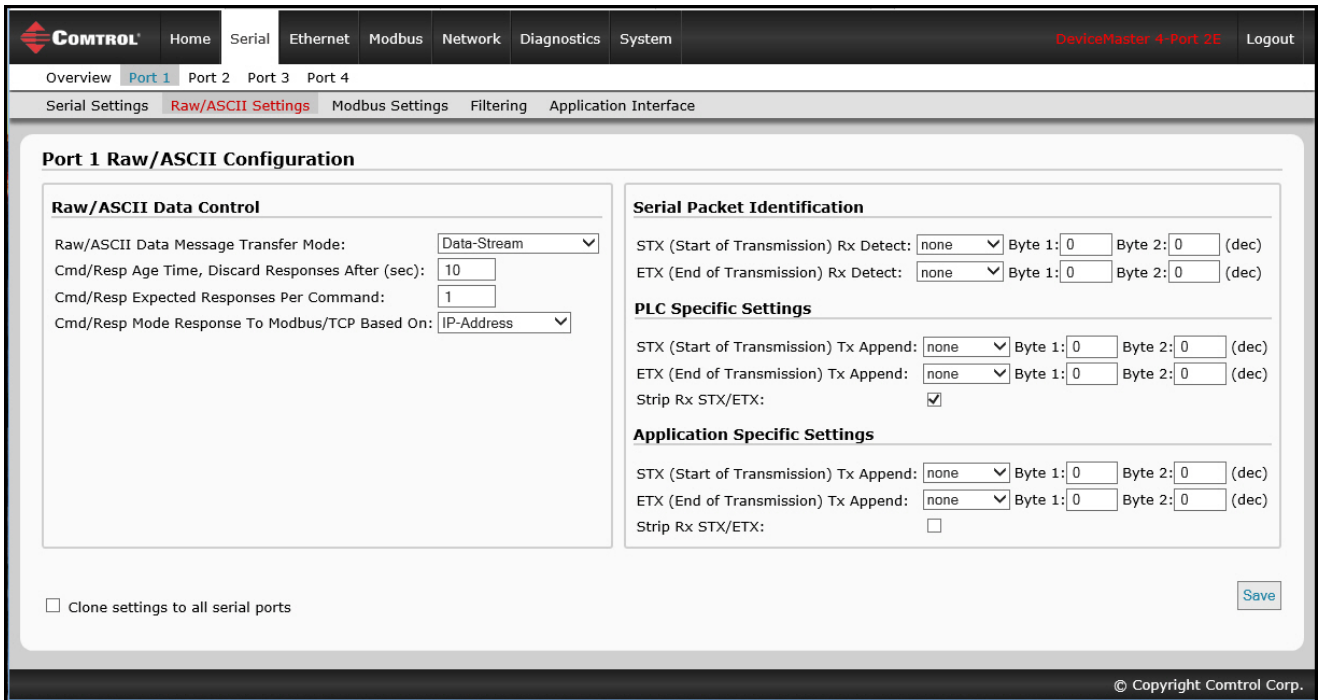
The screenshot shows the 'Port 1 Serial Configuration' page. The navigation bar includes 'Home', 'Serial', 'Ethernet', 'Modbus', 'Network', 'Diagnostics', 'System', 'DeviceMaster 4-Port 2E', and 'Logout'. Below the navigation bar, there are tabs for 'Overview', 'Port 1', 'Port 2', 'Port 3', and 'Port 4'. The 'Serial Settings' tab is active, with sub-tabs for 'Raw/ASCII Settings', 'Modbus Settings', 'Filtering', and 'Application Interface'. The main content area is titled 'Port 1 Serial Configuration' and is divided into two columns. The left column, 'Serial Configuration', contains fields for Port Name, Port Mode (RS-232), Baud Rate (9600), Parity (none), Data Bits (8), Stop Bits (1), Flow Control (none), DTR Mode (off), Rx Timeout Between Packets (ms) (200), and Discard Rx Pkts With Errors (checked). The right column, 'General Protocol Settings', contains a Serial Port Protocol dropdown (Raw-Data), 'Modbus Slave and Raw/ASCII Data Device Settings' with a Response Timeout (ms) field (250), 'Modbus Slave Settings' with a Lost Device Search Enable checkbox (unchecked), and a 'Save' button. At the bottom left, there is a checkbox for 'Clone settings to all serial ports' and a copyright notice '© Copyright Control Corp.' at the bottom right.

Port Serial Configuration Page	
Port Name	Up to 80 character ASCII string. A user definable string used to describe the serial interface. Valid characters include a-z, A-Z, 0-9, underscores, spaces and dashes. All other characters are discarded. The default name is blank.
Port Mode	Select the communications mode for the serial device that you are connecting to the port. The available modes are RS-232, RS-422, and RS-485.
Baud Rate	Select a baud rate from the list. The baud rate that you select determines how fast information is transferred through a port.
Parity	Select a method for error checking. <ul style="list-style-type: none"> • None - When the parity is set to none, there is no parity bit, and DeviceMaster does not perform parity checking. • Odd - Indicates that the sum of all the 1-bits in the byte plus the parity bit must be odd. When the total is odd, the parity bit is set to zero, when it is even, the parity bit is set to one. • Even - When the sum of all the 1-bits is even, the parity bit must be set to zero; when it is odd, the parity bit must be set to one.
Data Bits	Select the number of bits that make up the data. Choose from 5, 6, 7 or 8-bits.
Stop Bits	Select the number of bits to mark the end of data transmission.

Port Serial Configuration Page (Continued)	
Flow Control	<p>Specifies the ability to start and stop the flow of data without the loss of bytes. Select a method for controlling the flow of data from the following list:</p> <ul style="list-style-type: none"> • None - Indicates flow control is not in affect. • RTS/CTS - Request To Send (RTS) tells the receiving device that the sending device has data that is ready to send and Clear To Send (CTS) indicates the device is ready to accept data. • XON/XOFF - When selected, applies the standard method of controlling data flow between two modems. • Half Duplex - Transmits data in half-duplex mode.
DTR Mode	<p>Select the state of Data Terminal Ready (DTR).</p> <ul style="list-style-type: none"> • on - Enables DTR. • off - Disables DTR. • WhenEnabled - Select this option when enabling the serial port through the PLC.
Rx Timeout Between Packets (ms)	<p>Specifies the following information, once the start of a packet is received:</p> <ul style="list-style-type: none"> • How long the DeviceMaster should wait (in milliseconds) before timing-out, if the ETX Rx Detect length is one byte or two bytes and the ETX byte(s) are not received. • The time to wait in milliseconds between serial packets if the ETX Rx Detect length is set to none.
Discard Rx Pkts with Errors	<p>By default, this box is checked and the DeviceMaster discards serial packets with errors. Clear the check box when you need to receive a serial packet with errors to troubleshoot an issue.</p>
Clone settings to all ports	<p>Selecting this option before selecting Save applies these settings to all of the serial ports.</p>
General Protocol Settings	
Serial Port Protocol	<p>This is the serial port protocol.</p> <ul style="list-style-type: none"> • If you select Raw-Data, the port receives raw/ASCII type data. • If you select Modbus/RTU-to-Slaves, the serial port operates in Modbus/RTU slave(s) mode. • If you select Modbus/ASCII-to-Slaves, the serial port operates in to Modbus/ASCII slave(s) mode. • If you select Modbus/RTU-to-Master, the serial port operates in to Modbus/RTU master mode. • If you select Modbus/ASCII-to-Master, the serial port operates in to Modbus/ASCII master mode.
Modbus Slave and Raw/ASCII Data Device Settings	
Response Timeout (ms)	<p>Modbus Slave and Raw-Data Command/Response mode timeout setting. The DeviceMaster waits for the response(s) until this time has elapsed before transmitting another message. The default is 250 ms.</p>
Modbus Slave Settings	
Lost Device Search Enable (2- and 4-port, only)	<p>If this is set, lost devices that were on this Modbus to-slave port is searched for on other Modbus to-slave ports that also have this option set. For a discussion of this setting, see 2.4.2. Modbus Slave Device Search Methodology on Page 28.</p>

4.3. Port Raw/ASCII Configuration Page

Access the *Port Raw/ASCII Configuration* page by clicking **Serial | Port | Raw/ASCII Settings**.



Port Raw/ASCII Configuration Page	
Raw/ASCII Data Control	
Raw-Data Message Transfer Mode	If you select Data-Stream (default), the serial port operates in Data-Stream mode, see 2.2.4.1. Data-Stream Transfer Mode on Page 19. If you select Command/Response , the serial port operates in Command/Response mode, see 2.2.4.2. Command/Response Mode on Page 20.
Cmd/Resp Age Time, Discard Responses After (seconds)	The Age Time, or elapsed time, when responses destined for Modbus interface(s) are discarded. This typically occurs when a Modbus controller stops communicating and leaves a response in the response queues. The default is 10 seconds.
Cmd/Resp Expected Responses Per Command	The number of responses expected for each transmitted command. The default is 1.
Cmd/Resp Mode Response To Modbus/TCP Based On	Method or returning responses to Modbus/TCP interfaces. <ul style="list-style-type: none"> If you select IP-Address (default), all responses are returned to the IP-Address of the original command was received from. This may the same or different TCP connection. If you select TCP-Connection, all responses are returned to the same TCP connection the original command was received from. <p>Note: <i>TCP-Connection is typically required when multiple controllers are communicating from the same PLC or computer.</i></p>

Port Raw/ASCII Configuration Page (Continued)

Serial Packet Identification

<p>STX (Start of Transmission) Rx Detect</p>	<p>When enabled, the DeviceMaster detects an STX (start of transmission) byte sequence which is configured as one byte or two bytes when it receives a serial packet.</p> <p>The length indicates the number of STX bytes, valid values for length are:</p> <ul style="list-style-type: none"> • none - Disables this function and the DeviceMaster accepts the first byte received after the last ETX byte(s) as the start of the next data packet. • one byte - Scans serial data for one STX byte and when the DeviceMaster finds an STX byte it collects the data. If the first byte is not the STX byte, it discards the byte. The DeviceMaster continues to discard the bytes until it finds an STX byte. • two bytes - Scans serial data for two STX bytes and when the DeviceMaster finds two STX bytes it collects the data. If the STX bytes cannot be found, it discards the bytes. The DeviceMaster continues to discard the bytes until it finds the two STX bytes. <p>Byte 1 - Specifies the character that represents the first STX byte. The DeviceMaster looks for this character in the first STX byte, if the length is one byte or two bytes. You can specify a value between 0 and 255 in decimal format.</p> <p>Byte 2 - Specifies the character that represents the second STX byte. The DeviceMaster looks for this character in the second STX byte, only if the length is two bytes. You can specify a value between 0 and 255 in decimal format.</p>
<p>ETX (End of Transmission) Rx Detect</p>	<p>When enabled, the DeviceMaster detects an ETX (end of transmission) byte sequence that is configured as one byte or two bytes marking the end of the serial packet.</p> <p>The length indicates the number of ETX bytes, valid values for length are:</p> <ul style="list-style-type: none"> • none - Disables this function and the DeviceMaster uses the <i>Rx Timeout Between Packets</i> to indicate the end of data packet. • one byte - Scans serial data for one ETX byte and when the DeviceMaster finds the ETX byte, it identifies the data as a serial packet. • two bytes - Scans serial data for two ETX bytes and when the DeviceMaster finds the ETX bytes, it identifies the data as a serial packet. <p>Byte 1 - Specifies the character to scan for in the first ETX byte, if the length is one byte or two bytes. You can specify a value between 0 and 255 in decimal format.</p> <p>Byte 2 - Specifies the character to scan for in the second ETX byte, if the length is two bytes. You can specify a value between 0 and 255 in decimal format.</p>
<p>Discard Rx Pkts with Errors</p>	<p>By default, this box is checked and the DeviceMaster discards serial packets with errors. Clear the check box when you need to receive a serial packet with errors to troubleshoot an issue.</p>

PLC Specific Settings and Application Specific Settings Options

<p>STX Tx Append</p>	<p>When enabled, the DeviceMaster appends an STX (start of transmission) byte sequence which is configured as one byte or two bytes to the beginning of the serial packet before it is sent.</p> <p>The length indicates the number of STX bytes, values for length are:</p> <ul style="list-style-type: none"> • none - Disables this function. • one byte - Inserts one STX byte before the data. • two bytes - Inserts two STX bytes before the data. <p>Byte 1 - Specifies the transmit character associated with the first STX byte, if the length is one byte or two bytes. You can specify a value between 0 and 255 in decimal format.</p> <p>Byte 2 - Specifies the transmit character associated with the second STX byte, if the length is two bytes. You can specify a value between 0 and 255 in decimal format.</p>
----------------------	--

Port Raw/ASCII Configuration Page (Continued)	
ETX Tx Append	<p>When enabled, the DeviceMaster appends an ETX (end of transmission) byte sequence which is configured as one byte or two bytes to the end of the serial packet before it is sent.</p> <p>The length indicates the number of ETX bytes, valid values for length are:</p> <ul style="list-style-type: none"> • none - Disables this function. • one byte - Inserts one ETX byte at the end of the data. • two bytes - Inserts two ETX bytes at the end of the data. <p>Byte 1 - Specifies the transmit character associated with the first ETX byte, if the length is set to one byte or two bytes. You can specify a value between 0 and 255 in decimal format.</p> <p>Byte 2 - Specifies the transmit character associated with the second ETX byte, if the length is two bytes. You can specify a value between 0 and 255 in decimal format.</p>
Strip Rx STX/ETX	<p>When you select this check box, the DeviceMaster strips STX/ETX characters from received serial packets. Clear the check box when you do not want the DeviceMaster to strip STX/ETX characters from received serial packets.</p> <p>Serial Packets sent from the PLC or application to the DeviceMaster (over Ethernet), and then sent out the serial port, are not checked for STX/ETX.</p> <p>No STX/ETX character stripping occurs in these serial packets, and framing/parity/overrun error checking does not apply.</p>

4.4. Port Modbus Configuration (Raw/ASCII Only) Page

Access the *Port Modbus Configuration (Raw/ASCII Only)* page by clicking **Serial | Port | Modbus Settings**.

The following table also provides supporting information for the *Device Modbus Configuration* page that is found under **Ethernet | Device | Modbus/TCP Settings**.

Note: (Raw-Data Only) displays on the serial version of the Modbus configuration pages.

Port Modbus Configuration (Raw/ASCII Only) Page	
Serial Modbus Master and Modbus/TCP Interface Settings	
Rx (To PLC) Transfer Mode	<p>Specifies the Modbus/TCP raw-data receive data transfer mode to the PLC. There are three possible settings.</p> <ul style="list-style-type: none"> • Slave (PLC Polls) – The PLC polls the DeviceMaster for received data by sending read requests on a continual basis. • Master (Write to PLC) – The DeviceMaster writes received data into the specified PLC address using write messages. • Off – Received data is not sent to the PLC. <p>The default is Slave Mode.</p> <p>Note: <i>Slave (PLC Polls) must be selected to interface to serial Modbus masters.</i></p>

Port Modbus Configuration (Raw/ASCII Only) Page (Continued)	
Tx (From PLC) Transfer Mode	<p>Specifies the Modbus/TCP raw-data transmit data transfer mode to the PLC. There are three possible settings.</p> <ul style="list-style-type: none"> • Slave (PLC Writes) – The PLC sends write messages to transmit data. A message received with an incremented sequence number indicates new data to send, unless the Disable Tx Sequence Number Check option is selected. • Master (Poll the PLC) – The DeviceMaster sends read messages to poll the PLC at the specified address, rate, and message length. A message received with an incremented sequence number indicates new data to send, unless the Disable Tx Sequence Number Check option is selected. • Off – Transmit data is not accepted from the PLC. <p>The default is Slave Mode.</p> <p>Note: <i>Slave (PLC Writes) must be selected to interface to serial Modbus masters.</i></p>
Maximum Rx Data Packet Size	<p>Specifies the maximum acceptable size of a received serial or Ethernet packet. Default is 246 bytes. Maximums, <i>Slave Receive mode = 246, Serial receive master mode=1518, Ethernet receive master mode = 2048.</i></p>
Oversized Rx Packet Handling	<p>Specifies how to process oversized received packets.</p> <ul style="list-style-type: none"> • Truncate – truncate the packet to the <i>Maximum Rx Data Packet Size</i>. • Drop – drop the packet. <p>Default = Truncate</p>
Rx MS Byte First	<p>When you select this check box, the DeviceMaster receives the Most Significant (MS) byte of a WORD first. This check box is clear by default. Clear the check box when you need to receive the Least Significant (LS) byte of a WORD first.</p>
Tx MS Byte First	<p>This check box is clear by default. When you select this check box, DeviceMaster transmits the Most Significant (MS) byte of a WORD first. Clear the check box when you need to transmit the Least Significant (LS) byte of a WORD first.</p>
Disable Non-Filtered To PLC Rx Queue	<p>If filtering is disabled, only the last message received is sent to the PLC. This box is clear by default.</p>
Disable Tx Sequence Number Check	<p>Controls the transmit sequence number checking.</p> <ul style="list-style-type: none"> • If selected, the transmit sequence number checking is disabled. All transmit messages are transmitted if the sequence number has been incremented or not. • If not selected, the sequence number is checked and the message only is transmitted if the sequence number has been updated. <p>Default is <i>Not selected</i>.</p>
Modbus/TCP Master Rx/Tx Settings	
PLC IP Address	<p>Specifies the PLC IP Address of the PLC for operating in either <i>Receive</i> or <i>Transmit Master</i> mode.</p>
PLC Device ID	<p>Specifies the PLC Device ID of the PLC for operating in either <i>Receive</i> or <i>Transmit Master</i> mode. The default is 1.</p>
Disable Sequence Number/Length Header	<p>The data is transmitted without the 4 byte header (sequence number and data length).</p>

Port Modbus Configuration (Raw/ASCII Only) Page (Continued)	
Master Rx Mode Only	
PLC Rx Data Address	<p>Specifies the PLC address to write the received data at while operating in the <i>Master Receive Transfer</i> mode. The data area must be comprised of 16 bit words and large enough to contain the largest possible received message plus two 16 bit words for the sequence and length parameters.</p> <p>The address is base 1. Therefore, if your address scheme starts at zero, you need to add one to your address.</p>
Maximum PLC Update Rate	<p>The maximum rate (or minimum time interval) in milliseconds that messages are sent to the PLC in the <i>Master Receive Transfer</i> mode. This setting configures the DeviceMaster to space the messages to the PLC in order to prevent overrunning of data before the PLC can process it.</p>
Use Maximum Sized Modbus Messages	<p>Controls the size of Modbus/TCP messages used to write to the Modbus/TCP slave device.</p> <ul style="list-style-type: none"> • If selected, maximum sized Modbus messages of 242 bytes per message. • If not selected, Modbus messages of no more than 200 bytes are used. <p>The default is <i>Not selected</i>.</p> <p>Note: <i>This option only takes affect when large messages are received. Selecting this option may decrease the number of messages sent to the Modbus/TCP slave, thereby reducing network traffic and latency. However, not all Modbus/TCP slaves support maximum sized Modbus messages. So, this setting must be tested with the Modbus/TCP slave to ensure operability</i></p>
Master Tx Mode Only	
PLC Tx Data Address	<p>Specifies the PLC address to request transmit data messages while operating in the <i>Master Transmit Transfer</i> mode. The data area must be comprised of 16 bit words and contain a sequence number, length, and data to transmit. An updated sequence number indicates new data to transmit. Therefore, the length and data must be written to the transmit data area before updating the sequence number.</p> <p>The address is base 1. Therefore, if your address scheme starts at zero, you need to add one to your address.</p>
PLC Tx Poll Rate	<p>Specifies the rate, in milliseconds, that the DeviceMaster polls the PLC for transmit data. The default is 100ms</p>
PLC TX Poll Message Length (bytes)	<p>Specifies the length, in bytes, of the transmit message the DeviceMaster requests from the PLC. This must be large enough to contain the largest size data packet plus four bytes for the sequence number and length fields. Any extra bytes received from the PLC are ignored. The default is 250 bytes</p>
Tx Sequence Number Syncing Enable	<p>If the Tx (From PLC) Transfer Mode is operating in <i>Master (Poll the PLC)</i>; this specifies whether or not to enable synchronizing transmit data messages between the PLC and the DeviceMaster. This setting is clear by default.</p>
PLC Tx Consumed Sequence Number Address	<p>Specifies the PLC memory address at which the DeviceMaster writes the transmit consumed sequence number. This memory address must point to a 16-bit word and, like the other address definitions, is base 1. When the Tx Produced Sequence Number (at the PLC Tx Data Address) and this consumed sequence number are equal, the DeviceMaster has transmitted the last message and is ready for the next transmit message.</p>

4.5. Port Filtering/Data Extraction Configuration (Raw/ASCII Only) Page

Access the *Port Filtering/Data Extraction Configuration (Raw/ASCII Only)* page by clicking **Serial | Port | Filter** or **Ethernet | Device | Filter**.

The options on this page are the same as the *Device Filtering/Data Extraction Configuration* page that is accessed by clicking **Ethernet | Device | Filtering**.

The screenshot shows the 'Port 1 Filtering/Data Extraction Configuration (Raw/ASCII Only)' page. The 'Advanced Configuration' section includes the following settings:

- To PLC Filter Mode: Off (dropdown)
- To PLC Filter Options (RFID Only):
 - Antenna
 - Filter Value
 - Serial Number
- To PLC Filter Options (RFID/Barcode):
 - Company
 - Product/Location
 - Encoding/Numbering
- To Application Filter Mode: Off (dropdown)
- To Application Filter Options (RFID Only):
 - Antenna
 - Filter Value
 - Serial Number
- To Application Filter Options (RFID/Barcode):
 - Company
 - Product/Location
 - Encoding/Numbering
- RFID Antenna Grouping: None (dropdown)
- RFID Reader Interface Type: Unspecified (dropdown)
- Barcode UPC/EAN Standard 12-14 Digit Format: None (dropdown)
- Barcode UPC/EAN Eight Digit Format: None (dropdown)
- Filter Age Time (Time filtered after last read): 0 (min) 0 (sec) 100 (msec)
- Discard Unrecognized Data (RFID/Barcode): Off (dropdown)

A 'Save' button is located at the bottom right of the configuration area. The footer of the page reads '© Copyright Control Corp.'

Port Filtering/Data Extraction Configuration (Raw/ASCII Only)	
To PLC Filter Mode	<p>Defines the filter/data extraction mode to be employed on data to be sent to the PLC.</p> <ul style="list-style-type: none"> • Off • String (128 char max) - Raw/ASCII data is filtered up to 128 characters (or bytes) in length. • RFID (EPCglobal formats) - RFID data in any of the EPCglobal formats is filtered, the associated parameters are extracted, and the extracted data and RFID tag are sent to the PLC in a specified format. • Barcode (UPC/EAN formats) - Barcode data in specified UPC/EAN formats is filtered, the associated parameters are extracted, and the extracted data and barcode are sent to the PLC in a specified format.

Port Filtering/Data Extraction Configuration (Raw/ASCII Only) (Continued)	
To PLC Filtering Options (RFID Only)	<p>Defines the RFID filtering criteria to the PLC. If an option is enabled, it is used to decide when an RFID tag can be filtered or sent to the PLC.</p> <ul style="list-style-type: none"> • Antenna - Include the antenna number in the filtering criteria. This is data from the RFID reader and not from the RFID tag itself. • Filter Value - Include the filter value in the filtering criteria, which is part of the RFID tag data. • Serial Number - Include the serial number in the filtering criteria, which is part of the RFID tag data.
To PLC Filtering Options (RFID/Barcode)	<p>Defines the RFID filtering criteria and the barcode filtering criteria to the application. If an option is enabled, it is used to decide when a valid RFID tag or barcode can be filtered or sent to the PLC.</p> <ul style="list-style-type: none"> • Company - Include the company code in the filtering criteria, which is part of the RFID tag or barcode data. • Product/Location - Include the product/location code in the filtering criteria, which is part of the RFID tag or barcode data. • Encoding/Numbering - Include the encoding/numbering code in the filtering criteria, which is part of the RFID tag or barcode data.
To Application Filter Mode	<p>The filter/data extraction mode to be employed on data to be sent to the application.</p> <ul style="list-style-type: none"> • Off • String (128 char max) - Raw/ASCII data is filtered up to 128 characters (or bytes) in length. • RFID (EPCglobal formats) - RFID data in any of the EPCglobal formats are filtered, the associated parameters are extracted, and the extracted data and RFID tag are sent to the application in a specified format. • Barcode (UPC/EAN formats) - Barcode data in specified UPC/EAN formats is filtered, the associated parameters are extracted, and the extracted data and barcode are sent to the application in a specified format. <p><i>Note: The application filter mode can be set independently of the PLC filtering mode. The only exceptions are:</i></p> <ul style="list-style-type: none"> • <i>If the PLC filter mode is set to RFID, the application filter mode cannot be set to Barcode.</i> • <i>If the PLC filter mode is set to Barcode, the application filter mode cannot be set to RFID.</i>
To Application Filtering Options (RFID Only)	<p>Defines the RFID filtering criteria to the application. If an option is enabled, it is used to decide when an RFID tag can be filtered or sent to the PLC.</p> <ul style="list-style-type: none"> • Antenna - Include the antenna number in the filtering criteria. This is data from the RFID reader and not part of the RFID tag. • Filter Value - Include the filter value in the filtering criteria, which is part of the RFID tag data. • Serial Number - Include the serial number in the filtering criteria, which is part of the RFID tag data.

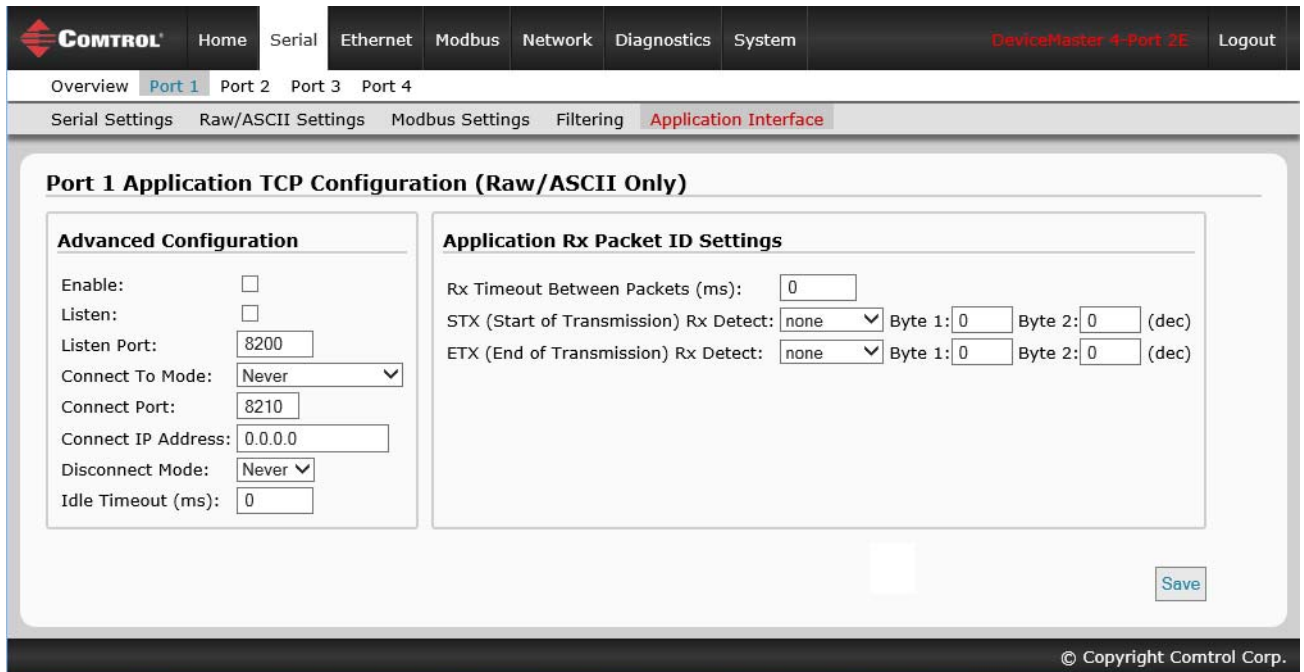
Port Filtering/Data Extraction Configuration (Raw/ASCII Only) (Continued)																																																													
To Application Filtering Options (RFID/Barcode)	<p>Defines the barcode filtering criteria and part of the RFID filtering criteria to the application. If an option is enabled, it is used to decide when a valid RFID tag or barcode can be filtered or sent to the application.</p> <ul style="list-style-type: none"> • Company - Include the company code in the filtering criteria, which is part of the RFID tag or barcode data. • Product/Location - Include the product/location code in the filtering criteria, which is part of the RFID tag or barcode data. • Encoding/Numbering - Include the encoding/numbering code in the filtering criteria, which is part of the RFID tag or barcode data. 																																																												
RFID Antenna Grouping	<p>This setting is applicable only to RFID filtering and only if the antenna filtering option is enabled. It allows the DeviceMaster to filter RFID tags based on antenna groupings. The possible groupings are:</p> <table border="1"> <thead> <tr> <th>Setting</th> <th>Group 1 Antennas</th> <th>Group 2 Antennas</th> <th>Group 3 Antennas</th> <th>Group N Antennas</th> </tr> </thead> <tbody> <tr> <td>None</td> <td>1</td> <td>2</td> <td>3</td> <td>4</td> </tr> <tr> <td>Groups of Twos</td> <td>1,2</td> <td>3,4</td> <td>5,6</td> <td>Etc.</td> </tr> <tr> <td>Groups of Threes</td> <td>1,2,3</td> <td>4,5,6</td> <td>7,8,9</td> <td>Etc.</td> </tr> <tr> <td>Groups of Fours</td> <td>1,2,3,4</td> <td>5,6,7,8</td> <td>9,10,11,12</td> <td>Etc.</td> </tr> <tr> <td>First Two Only</td> <td>1,2</td> <td>3</td> <td>4</td> <td>N+1</td> </tr> <tr> <td>First Three Only</td> <td>1,2,3</td> <td>4</td> <td>5</td> <td>N+2</td> </tr> </tbody> </table>	Setting	Group 1 Antennas	Group 2 Antennas	Group 3 Antennas	Group N Antennas	None	1	2	3	4	Groups of Twos	1,2	3,4	5,6	Etc.	Groups of Threes	1,2,3	4,5,6	7,8,9	Etc.	Groups of Fours	1,2,3,4	5,6,7,8	9,10,11,12	Etc.	First Two Only	1,2	3	4	N+1	First Three Only	1,2,3	4	5	N+2																									
Setting	Group 1 Antennas	Group 2 Antennas	Group 3 Antennas	Group N Antennas																																																									
None	1	2	3	4																																																									
Groups of Twos	1,2	3,4	5,6	Etc.																																																									
Groups of Threes	1,2,3	4,5,6	7,8,9	Etc.																																																									
Groups of Fours	1,2,3,4	5,6,7,8	9,10,11,12	Etc.																																																									
First Two Only	1,2	3	4	N+1																																																									
First Three Only	1,2,3	4	5	N+2																																																									
RFID Reader Interface Type	<p>Defines the expected RFID data format to be used while operating in the RFID filtering mode. Each Reader Interface Type is unique and pertains to the RFID reader manufacturer. If a different RFID reader is to be used and it provides a similar format to any of the RFID readers listed below, it can also be used in the RFID filtering mode.</p> <ul style="list-style-type: none"> • Unspecified - The DeviceMaster assumes a HEX ASCII format and attempts to locate the antenna number. • Alien (Text Mode) - Specifies the Alien RFID reader Text Mode. • Alien (Terse Mode) - Specifies the Alien RFID reader Terse Mode. • Intermec (Hex ASCII Mode) - Specifies the Intermec reader returning data in the Hex ASCII Mode. <p>See the DeviceMaster Filtering and Data Extraction Reference Guide for further details.</p>																																																												
Barcode Formats:	<p>Defines barcode format to be used for both standard and eight digit UPC labels. The term <i>standard</i> refers to UPC-A, EAN-13, JAN, and EAN-14 barcodes which all have ten company/product digits.</p> <p>The standard and eight digit formats are selected independently and each operates independently. Barcode filtering/data extraction does not function if no format is selected.</p> <table border="1"> <thead> <tr> <th>Format</th> <th>Numbering Digits</th> <th>Company Digits</th> <th>Product Digits</th> <th>Check Digit</th> </tr> </thead> <tbody> <tr> <td colspan="5">Standard Formats</td> </tr> <tr> <td>None</td> <td>N/A</td> <td>N/A</td> <td>N/A</td> <td>N/A</td> </tr> <tr> <td>Company-5/ Product-5</td> <td>1-3</td> <td>5</td> <td>5</td> <td>1</td> </tr> <tr> <td>Company-6/ Product-4</td> <td>1-3</td> <td>6</td> <td>4</td> <td>1</td> </tr> <tr> <td>Company-7/ Product-3</td> <td>1-3</td> <td>7</td> <td>3</td> <td>1</td> </tr> <tr> <td>Company-8/ Product-2</td> <td>1-3</td> <td>8</td> <td>2</td> <td>1</td> </tr> <tr> <td>Company-9/ Product-1</td> <td>1-3</td> <td>9</td> <td>1</td> <td>1</td> </tr> <tr> <td colspan="5">Eight Digit Formats</td> </tr> <tr> <td>EAN-8 Number-2/Product 5</td> <td>2</td> <td>0</td> <td>5</td> <td>1</td> </tr> <tr> <td>EAN-8 Number-3/Product 4</td> <td>3</td> <td>0</td> <td>4</td> <td>1</td> </tr> <tr> <td>UPC-E</td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table> <p>See the DeviceMaster Filtering and Data Extraction Reference Guide for further details.</p>	Format	Numbering Digits	Company Digits	Product Digits	Check Digit	Standard Formats					None	N/A	N/A	N/A	N/A	Company-5/ Product-5	1-3	5	5	1	Company-6/ Product-4	1-3	6	4	1	Company-7/ Product-3	1-3	7	3	1	Company-8/ Product-2	1-3	8	2	1	Company-9/ Product-1	1-3	9	1	1	Eight Digit Formats					EAN-8 Number-2/Product 5	2	0	5	1	EAN-8 Number-3/Product 4	3	0	4	1	UPC-E				
Format	Numbering Digits	Company Digits	Product Digits	Check Digit																																																									
Standard Formats																																																													
None	N/A	N/A	N/A	N/A																																																									
Company-5/ Product-5	1-3	5	5	1																																																									
Company-6/ Product-4	1-3	6	4	1																																																									
Company-7/ Product-3	1-3	7	3	1																																																									
Company-8/ Product-2	1-3	8	2	1																																																									
Company-9/ Product-1	1-3	9	1	1																																																									
Eight Digit Formats																																																													
EAN-8 Number-2/Product 5	2	0	5	1																																																									
EAN-8 Number-3/Product 4	3	0	4	1																																																									
UPC-E																																																													
UPC/EAN Standard 12-14 Digit																																																													
UPD/EAN Eight Digit																																																													

Port Filtering/Data Extraction Configuration (Raw/ASCII Only) (Continued)	
Filter Age Time (Time filtered after last read)	<p>Defines the time a filter string, RFID tag, or barcode continues to be filtered after the last time it was received.</p> <p>If an entry is received before the Filter Age Time has passed, the entry is filtered and the data is not sent to the PLC and/or application. However, if the Filter Age Time has passed, it passes filtering and be sent to the PLC and/or application.</p>
Discard Unrecognized Data (RFID/Barcode)	<p>Specifies what to do with unrecognized RFID or barcode data.</p> <ul style="list-style-type: none"> • Off - Sends unrecognized data to the PLC and/or application. • To-PLC - Discards unrecognized data to the PLC. Allows sending of unrecognized data to the application. • To-Application - Discards unrecognized data to the application. Allows sending of unrecognized data to the PLC. • To-PLC/Application - Discards unrecognized data to both the PLC and application.

4.6. Application TCP Configuration (Raw/ASCII Only) Page

Access the *Port Application TCP Configuration (Raw/ASCII Only)* page by clicking **Serial | Port | Application Interface**.

The options on the *Device Application TCP Configuration* page that is accessed by clicking **Ethernet | Device | Application Interface** are the same.



Note: The following image shows both Serial device and Ethernet device connections to an application.

Port Application TCP Configuration (Raw/ASCII Only) Page	
Enable	<p>Enables/disables the <i>Application Socket Interface</i>. Enabling this function allows an application to be connected to the device serial/socket port.</p> <p>If both the PLC and application are connected to the device serial/socket port, both can transmit to and receive data from the serial/socket port. However, the PLC and application cannot communicate directly to each other.</p>
Listen	<p>Enabling this setting allows the application to connect to the DeviceMaster via an Ethernet TCP/IP socket.</p> <ul style="list-style-type: none"> • Not selected - Disables listening and the DeviceMaster does not accept connection attempts. • Selected - Enables listening and the DeviceMaster accepts connection attempts from the specified Listen Port.

Port Application TCP Configuration (Raw/ASCII Only) Page(Continued)	
Listen Port	The socket port number on the DeviceMaster the application connects to if the Application Listen Enable is selected.
Connect To Mode	Specifies if and how the DeviceMaster attempts to connect to the application at the specified Connect IP Address and Connect Port . <ul style="list-style-type: none"> • Never - The DeviceMaster does not attempt to connect to the application. • Connect-Always - The DeviceMaster attempts to connect to the application until a connection is made. • Connect-On-Data – The DeviceMaster does not attempt to connect to the application until there is data to send to the application. Once data is received from the serial/socket device, the DeviceMaster attempts to connect to the application until a connection is made.
Connect Port	The application socket port number the DeviceMaster connects to if the Application Connect To Mode is set to either Connect-Always or Connect-On-Data .
Connect IP Address	The application IP address the DeviceMaster connects to if the Application Connect To Mode is set to either Connect-Always or Connect-On-Data .
Disconnect Mode	Controls if and how the DeviceMaster disconnects from an application. <ul style="list-style-type: none"> • Never – The DeviceMaster does not disconnect from the application. • Idle - The DeviceMaster disconnects when there has been no transmit or received data between the serial/socket device and application for a specified Idle Timer period.
Idle Timer	The idle timeout period in milliseconds that is used if the application Disconnect Mode is set to Idle .
Application Rx Packet ID Settings	
Rx Timeout Between Packets	Specifies the following information, once the start of the packet is received: <ul style="list-style-type: none"> • How long the DeviceMaster should wait (in milliseconds) before timing-out, if the ETX Rx Detect length is one byte or two bytes and the ETX byte(s) are not received. • The time to wait in milliseconds between Ethernet packets if the ETX Rx Detect length is set to none.
STX (Start of Transmission) Rx Detect	When enabled, the DeviceMaster detects an STX (start of transmission) byte sequence which is configured as one byte or two bytes when it receives an Ethernet packet. The length indicates the number of STX bytes, valid values for length are: <ul style="list-style-type: none"> • none - Disables this function and the DeviceMaster accepts the first byte received after the last ETX byte(s) as the start of the next Ethernet packet. • one byte - Scans Ethernet data for one STX byte and when the DeviceMaster finds an STX byte it collects the data. If the first byte is not the STX byte, it discards the byte. The DeviceMaster continues to discard the bytes until it finds an STX byte. • two bytes - Scans Ethernet data for two STX bytes and when the DeviceMaster finds two STX bytes it collects the data. If the STX bytes cannot be found, it discards the bytes. The DeviceMaster continues to discard the bytes until it finds the two STX bytes. <p>Byte 1 - Specifies the character that represents the first STX byte. The DeviceMaster looks for this character in the first STX byte, if the length is one byte or two bytes. You can specify a value between 0 and 255 in decimal format.</p> <p>Byte 2 - Specifies the character that represents the second STX byte. The DeviceMaster looks for this character in the second STX byte, only if the length is two bytes. You can specify a value between 0 and 255 in decimal format.</p>

Port Application TCP Configuration (Raw/ASCII Only) Page(Continued)

ETX (End of Transmission) Rx Detect	<p>When enabled, the DeviceMaster detects an ETX (end of transmission) byte sequence that is configured as one byte or two bytes marking the end of the Ethernet packet. The length indicates the number of ETX bytes, valid values for length are:</p> <ul style="list-style-type: none"> • none - Disables this function and the DeviceMaster uses the Rx Timeout Between Packets to indicate the end of data packet. • one byte - Scans Ethernet data for one ETX byte and when the DeviceMaster finds the ETX byte, it identifies the data as an Ethernet packet. • two bytes - Scans Ethernet data for two ETX bytes and when the DeviceMaster finds the ETX bytes, it identifies the data as an Ethernet packet. <p>Byte 1 - Specifies the character to scan for in the first ETX byte, if the length is one byte or two bytes. You can specify a value between 0 and 255 in decimal format.</p> <p>Byte 2 - Specifies the character to scan for in the second ETX byte, if the length is two bytes. You can specify a value between 0 and 255 in decimal format.</p>
-------------------------------------	---

Chapter 5. Ethernet Menus

The following Ethernet menus are discussed in this chapter.

- [5.1. Ethernet Device Overview Page](#) on Page 60
- [5.2. Device Interface Configuration Page](#) on Page 61
- [5.3. Device Raw/ASCII Configuration Page](#) on Page 63
- [5.4. Device Modbus Configuration Page](#) on Page 66
- [5.5. Device Filtering/Data Extraction Configuration Page](#) on Page 69
- [5.6. Application TCP Configuration](#) on Page 70

5.1. Ethernet Device Overview Page

When you select the Ethernet menu, the Ethernet Device Overview page displays, which displays the configured Ethernet settings.

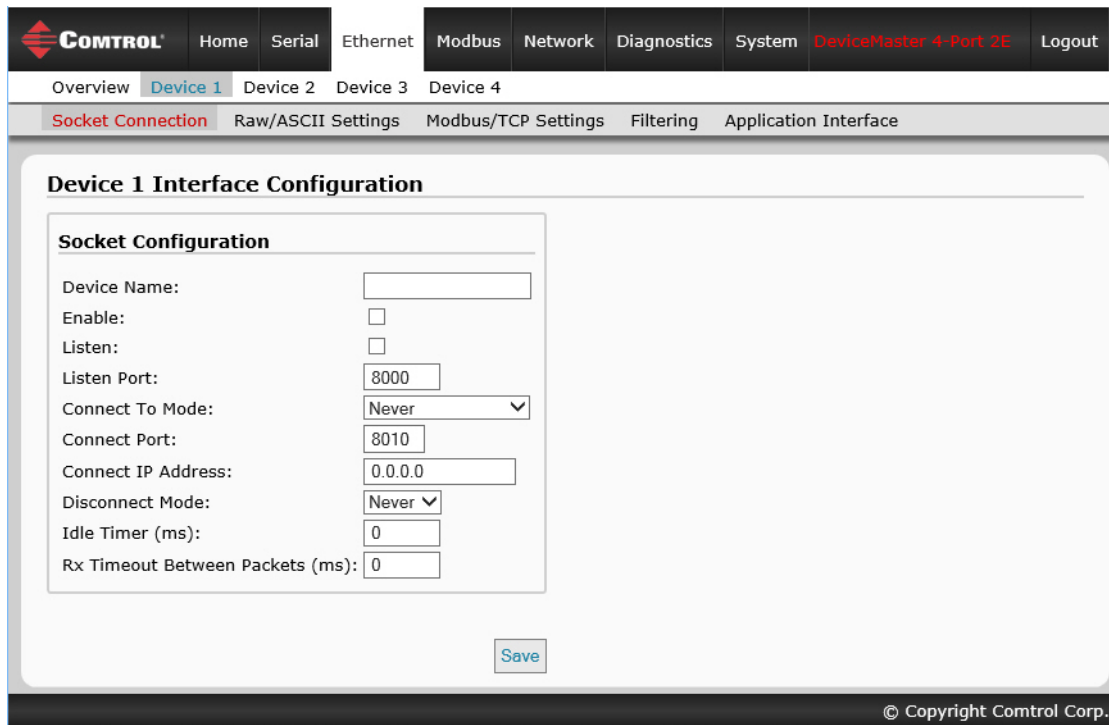
This is the current configuration of the Ethernet device socket ports. Select an Ethernet device from the menu above to change the configuration.

Device TCP Connection Configuration	Device 1	Device 2	Device 3	Device 4
Device Name:				
Enabled:	no	no	no	no
Listen:	no	no	no	no
Listen Port:	8000	8001	8002	8003
Connect To Mode:	Never	Never	Never	Never
Connect Port:	8010	8011	8012	8013
Connect to IP address:	0.0.0.0	0.0.0.0	0.0.0.0	0.0.0.0
Disconnect Mode:	Never	Never	Never	Never
Idle timeout (ms):	0	0	0	0
RX timeout between packets (ms):	0	0	0	0
Raw/ASCII Data Control				
Raw/ASCII Data Message Transfer Mode:	Data-Stream	Data-Stream	Data-Stream	Data-Stream
Cmd/Resp Age Time, Discard Responses After (sec):	N/A	N/A	N/A	N/A
Cmd/Resp Expected Responses Per Command:	N/A	N/A	N/A	N/A
Cmd/Resp Mode Response To Modbus/TCP Based On:	N/A	N/A	N/A	N/A
Socket Packet ID Settings				
STX Rx Detect:	none	none	none	none
STX Rx (dec):				
ETX Rx Detect:	none	none	none	none
ETX Rx (dec):				
PLC Specific Settings				
STX Tx Append:	none	none	none	none
STX Tx (dec):				
ETX Tx Append:	none	none	none	none
ETX Tx (dec):				
Strip Rx STX/ETX chars:	no	no	no	no
Serial Modbus Master and Modbus/TCP Interface Settings				
Rx (To PLC) Transfer Mode:	Slave (PLC Polls)	Slave (PLC Polls)	Slave (PLC Polls)	Slave (PLC Polls)
Tx (From PLC) Transfer Mode:	Slave (PLC Writes)	Slave (PLC Writes)	Slave (PLC Writes)	Slave (PLC Writes)
Maximum Rx Data Packet Size:	246	246	246	246
Oversized Rx Packet Handling:	Truncate	Truncate	Truncate	Truncate
Rx MS Byte First:	no	no	no	no
Tx MS Byte First:	no	no	no	no
Disable Non-Filtered To PLC Rx Queue:	no	no	no	no
Disable Tx Sequence Number Check:	no	no	no	no
Modbus/TCP Master Rx/Tx Settings				
PLC IP Address:	N/A	N/A	N/A	N/A
PLC Device ID:	N/A	N/A	N/A	N/A
Disable Sequence Number/Length Header:	N/A	N/A	N/A	N/A
Master Rx Mode Only				
PLC Rx Data Address (Base 1):	N/A	N/A	N/A	N/A
Maximum PLC Update Rate (msec):	N/A	N/A	N/A	N/A
Use Maximum Sized Modbus Messages:	N/A	N/A	N/A	N/A
Master Tx Mode Only				
PLC Tx Data Address (Base 1):	N/A	N/A	N/A	N/A
PLC Tx Poll Rate (msec):	N/A	N/A	N/A	N/A
PLC Tx Poll Message Length (bytes):	N/A	N/A	N/A	N/A
Tx Sequence Number Syncing Enable:	N/A	N/A	N/A	N/A
PLC Tx Consumed Seq Nbr Address (Base 1):	N/A	N/A	N/A	N/A

© Copyright Control Corp.

5.2. Device Interface Configuration Page

Access the *Device Interface Configuration* page by clicking **Ethernet | Device | Socket Connection**. The following table provides information about the options on this page.



Device Interface Configuration Page	
Device Name	A user definable string used to describe the serial interface. Valid characters include a-z, A-Z, 0-9, underscores, spaces and dashes. All other characters are discarded. Up to 80 character ASCII string. The default is blank.
Enable	<p>This setting enables/disables the <i>Device Ethernet Device</i>. Enabling this function allows an Ethernet TCP/IP device to be connected to a PLC and/or application. If both the PLC and application are connected to the device, both can transmit to and receive data from the device socket port. However, the PLC and application cannot communicate directly to each other.</p> <p>The diagram illustrates the following connections:</p> <ul style="list-style-type: none"> Raw/ASCII Ethernet Device ↔ DeviceMaster UP (via Ethernet TCP/IP) DeviceMaster UP ↔ PLC (via Modbus/TCP, Modbus/RTU Serial Master(s), and Modbus/ASCII Serial Master(s)) DeviceMaster UP ↔ Application(s) (via Ethernet TCP/IP)

Device Interface Configuration Page (Continued)	
Listen	<p>Enabling this setting allows the device to connect to the DeviceMaster via an Ethernet TCP/IP socket.</p> <ul style="list-style-type: none"> • Not selected - Disables listening; the DeviceMaster does not accept connection attempts. • Selected - Enables listening; the DeviceMaster accepts connection attempts from the specified Listen Port.
Listen Port	<p>This is the socket port number on the DeviceMaster the application connects to if the Device Listen Enable is selected.</p>
Connect To Mode	<p>This setting specifies if and how the DeviceMaster attempts to connect to the device at the specified Connect IP Address and Connect Port.</p> <ul style="list-style-type: none"> • Never - The DeviceMaster does not attempt to connect to the device. • Connect-Always - The DeviceMaster attempts to connect to the device until a connection is made. • Connect-On-Data - The DeviceMaster does not attempt to connect to the device until there is data to send to the device. Once data is received for the device, the DeviceMaster attempts to connect to the device until a connection is made.
Connect Port	<p>The device socket port number the DeviceMaster connects to if the Device Connect To Mode is set to either Connect-Always or Connect-On-Data.</p>
Connect IP Address	<p>The device IP address the DeviceMaster connects to if the Device Connect To Mode is set to either Connect-Always or Connect-On-Data.</p>
Disconnect Mode	<p>This setting specifies if and how the DeviceMaster disconnects from the device.</p> <ul style="list-style-type: none"> • Never - The DeviceMaster does not disconnect from the device. • Idle - The DeviceMaster disconnects when there has been no transmit or received data between the device and PLC/application for a specified Idle Timer period.
Idle Timer	<p>The idle timeout period in milliseconds that is used if the Device Disconnect Mode is set to Idle.</p>
Rx Timeout Between Packets	<p>Specifies the following information, once the start of a packet is received:</p> <ul style="list-style-type: none"> • How long the DeviceMaster should wait (in milliseconds) before timing-out, if the ETX Rx Detect length is one byte or two bytes and the ETX byte(s) are not received. • The time to wait in milliseconds between Ethernet packets if the ETX Rx Detect length is set to none.

5.3. Device Raw/ASCII Configuration Page

Access the *Device Raw/ASCII Configuration* page by clicking **Ethernet | Device | Raw ASCII Settings**.

The screenshot shows the 'Device 1 Raw/ASCII Configuration' page. It features a navigation bar at the top with 'CONTROL' logo and tabs for Home, Serial, Ethernet, Modbus, Network, Diagnostics, and System. The main content area is divided into four sections:

- Raw/ASCII Data Control:** Includes a dropdown for 'Data-Stream' mode, a text input for 'Cmd/Resp Response Timeout (ms)' set to 200, a text input for 'Cmd/Resp Age Time, Discard Responses After (sec)' set to 10, a text input for 'Cmd/Resp Expected Responses Per Command' set to 1, and a dropdown for 'Cmd/Resp Mode Response To Modbus/TCP Based On' set to 'IP-Address'.
- Device Socket Packet ID Settings:** Includes 'STX (Start of Transmission) Rx Detect' with a dropdown set to 'one byte', 'Byte 1' set to 2, and 'Byte 2' set to 0; and 'ETX (End of Transmission) Rx Detect' with a dropdown set to 'one byte', 'Byte 1' set to 3, and 'Byte 2' set to 0.
- PLC Specific Settings:** Includes 'STX (Start of Transmission) Tx Append' with a dropdown set to 'none', 'Byte 1' set to 0, and 'Byte 2' set to 0; 'ETX (End of Transmission) Tx Append' with a dropdown set to 'none', 'Byte 1' set to 0, and 'Byte 2' set to 0; and a checkbox for 'Strip Rx STX/ETX' which is unchecked.
- Application Specific Settings:** Includes 'STX (Start of Transmission) Tx Append' with a dropdown set to 'none', 'Byte 1' set to 0, and 'Byte 2' set to 0; 'ETX (End of Transmission) Tx Append' with a dropdown set to 'none', 'Byte 1' set to 0, and 'Byte 2' set to 0; and a checkbox for 'Strip Rx STX/ETX' which is unchecked.

At the bottom left, there is a checkbox 'Clone settings to all Ethernet devices' which is unchecked. At the bottom right, there is a 'Save' button. The footer of the page reads '© Copyright Control Corp.'

The following table discusses options on this page.

Device Raw/ASCII Configuration Page	
Raw/ASCII Data Control	
Raw-Data Message Transfer Mode	If you select Data-Stream (default), the serial port operates in Data-Stream mode, see 2.2.4.1. Data-Stream Transfer Mode on Page 19. If you select Command/Response , the serial port operates in Command/Response mode, see 2.2.4.2. Command/Response Mode on Page 20.
Cmd/Resp Response Timeout	The Command/Response mode response timeout setting. The DeviceMaster waits for the response(s) until this time has elapsed before transmitting another message. The default is 200 msec.
Cmd/Resp Age Time, Discard Responses After (seconds)	The Age Time, or elapsed time, when responses destined for Modbus interface(s) are discarded. This typically occurs when a Modbus controller stops communicating and leaves a response in the response queues. The default is 10 seconds.
Cmd/Resp Expected Responses Per Command	The number of responses expected for each transmitted command. The default is 1.
Cmd/Resp Response To Modbus/TCP Based On	Method for returning responses to Modbus/TCP interfaces. <ul style="list-style-type: none"> If you select IP-Address (default), all responses are returned to the IP-Address the original command was received from. This may be the same or different TCP connection. If you select TCP-Connection, all responses are returned to the same TCP connection the original command was received from. <p>Note: <i>TCP-Connection is typically required when multiple controllers are communicating from the same PLC or computer.</i></p>

Device Raw/ASCII Configuration Page (Continued)	
Device Socket Packet ID Settings	
STX (Start of Transmission) Rx Detect	<p>When enabled, the DeviceMaster detects an STX (start of transmission) byte sequence which is configured as one byte or two bytes when it receives an Ethernet packet. The length indicates the number of STX bytes, valid values for length are:</p> <ul style="list-style-type: none"> • none - Disables this function and the DeviceMaster accepts the first byte received after the last ETX byte(s) as the start of the next Ethernet packet. • one byte - Scans Ethernet data for one STX byte and when the DeviceMaster finds an STX byte it collects the data. If the first byte is not the STX byte, it discards the byte. The DeviceMaster continues to discard the bytes until it finds an STX byte. • two bytes - Scans Ethernet data for two STX bytes and when the DeviceMaster finds two STX bytes it collects the data. If the STX bytes cannot be found, it discards the bytes. The DeviceMaster continues to discard the bytes until it finds the two STX bytes. <p>Byte 1 - Specifies the character that represents the first STX byte. The DeviceMaster looks for this character in the first STX byte, if the length is one byte or two bytes. You can specify a value between 0 and 255 in decimal format.</p> <p>Byte 2 - Specifies the character that represents the second STX byte. The DeviceMaster looks for this character in the second STX byte, only if the length is two bytes. You can specify a value between 0 and 255 in decimal format.</p>
ETX (End of Transmission) Rx Detect	<p>When enabled, the DeviceMaster detects an ETX (end of transmission) byte sequence that is configured as one byte or two bytes marking the end of the Ethernet packet. The length indicates the number of ETX bytes, valid values for length are:</p> <ul style="list-style-type: none"> • none - Disables this function and the DeviceMaster uses the Rx Timeout Between Packets to indicate the end of data packet. • one byte - Scans Ethernet data for one ETX byte and when the DeviceMaster finds the ETX byte, it identifies the data as an Ethernet packet. • two bytes - Scans Ethernet data for two ETX bytes and when the DeviceMaster finds the ETX bytes, it identifies the data as an Ethernet packet. <p>Byte 1 - Specifies the character to scan for in the first ETX byte, if the length is one byte or two bytes. You can specify a value between 0 and 255 in decimal format.</p> <p>Byte 2 - Specifies the character to scan for in the second ETX byte, if the length is two bytes. You can specify a value between 0 and 255 in decimal format.</p>
PLC Specific Settings and Application Specific Settings	
STX Tx Append	<p>When enabled, the DeviceMaster appends an STX (start of transmission) byte sequence which is configured as one byte or two bytes to the beginning of the Ethernet packet before it is sent. The length indicates the number of STX bytes, values for length are:</p> <ul style="list-style-type: none"> • none - Disables this function. • one byte - Inserts one STX byte before the data. • two bytes - Inserts two STX bytes before the data. <p>Byte 1 - Specifies the transmit character associated with the first STX byte, if the length is one byte or two bytes. You can specify a value between 0 and 255 in decimal format.</p> <p>Byte 2 - Specifies the transmit character associated with the second STX byte, if the length is two bytes. You can specify a value between 0 and 255 in decimal format.</p>

Device Raw/ASCII Configuration Page (Continued)	
ETX Tx Append	<p>When enabled, the DeviceMaster appends an ETX (end of transmission) byte sequence which is configured as one byte or two bytes to the end of the Ethernet packet before it is sent. The length indicates the number of ETX bytes, valid values for length are:</p> <ul style="list-style-type: none"> • none - Disables this function. • one byte - Inserts one ETX byte at the end of the data. • two bytes - Inserts two ETX bytes at the end of the data. <p>Byte 1 - Specifies the transmit character associated with the first ETX byte, if the length is set to one byte or two bytes. You can specify a value between 0 and 255 in decimal format.</p> <p>Byte 2 - Specifies the transmit character associated with the second ETX byte, if the length is two bytes. You can specify a value between 0 and 255 in decimal format.</p>
Strip Rx STX/ETX	<p>When you select this check box, the DeviceMaster strips STX/ETX characters from received Ethernet packets. Clear the check box when you do not want the DeviceMaster to strip STX/ETX characters from received Ethernet packets.</p> <p>Packets sent from the PLC to the DeviceMaster (over Ethernet), and then sent out the Ethernet port, are not checked for STX/ETX. No STX/ETX character stripping occurs in these Ethernet packets.</p>

5.4. Device Modbus Configuration Page

Access the *Device Modbus Configuration* page by clicking **Ethernet | Device | Modbus Settings**.

Device 1 Modbus Configuration

Serial Modbus Master and Modbus/TCP Interface Settings

Rx (To PLC) Transfer Mode:
 Tx (From PLC) Transfer Mode:
 Maximum Rx Data Packet Size:
 Oversized Rx Packet Handling:
 Rx MS Byte First:
 Tx MS Byte First:
 Disable Non-Filtered To PLC Rx Queue:
 Disable Tx Sequence Number Check:

Modbus/TCP Master Rx/Tx Settings

PLC IP Address:
 PLC Device ID:
 Disable Sequence Number/Length Header:

Master Rx Mode Only

PLC Rx Data Address (Base 1):
 Maximum PLC Update Rate (msec):
 Use Maximum Sized Modbus Messages:

Master Tx Mode Only

PLC Tx Data Address (Base 1):
 PLC Tx Poll Rate (msec):
 PLC Tx Poll Message Length (bytes):
 Tx Sequence Number Syncing Enable:
 PLC Tx Consumed Seq Nbr Address (Base 1):

© Copyright Control Corp.

The following table provides information about the options on this page.

Device Modbus Configuration Page	
Serial Modbus Master and Modbus/TCP Interface Settings	
Rx (To PLC) Transfer Mode	<p>Specifies the Modbus/TCP raw-data receive data transfer mode to the PLC. There are three possible settings.</p> <ul style="list-style-type: none"> • Slave (PLC Polls) – The PLC polls the DeviceMaster for received data by sending read requests on a continual basis. • Master (Write to PLC) – The DeviceMaster writes received data into the specified PLC address using write messages. • Off – Received data is not sent to the PLC. <p>The default is Slave Mode.</p> <p>Note: <i>Slave (PLC Polls) must be selected to interface to serial Modbus masters.</i></p>

Device Modbus Configuration Page(Continued)	
Tx (From PLC) Transfer Mode	<p>Specifies the Modbus/TCP raw-data transmit data transfer mode to the PLC. There are three possible settings.</p> <ul style="list-style-type: none"> • Slave (PLC Writes) – The PLC sends write messages to transmit data. A message received with an incremented sequence number indicates new data to send, unless the Disable Tx Sequence Number Check option is selected. • Master (Poll the PLC) – The DeviceMaster sends read messages to poll the PLC at the specified address, rate, and message length. A message received with an incremented sequence number indicates new data to send, unless the Disable Tx Sequence Number Check option is selected. • Off – Transmit data is not accepted from the PLC. <p>The default is Slave Mode.</p> <p>Note: <i>Slave (PLC Writes) must be selected to interface to serial Modbus masters.</i></p>
Maximum Rx Data Packet Size	<p>Specifies the maximum acceptable size of a received serial or Ethernet packet. Default is 246 bytes. Maximums, <i>Slave Receive mode = 246, Serial receive master mode=1518, Ethernet receive master mode = 2048.</i></p>
Oversize Rx Packet Handling	<p>Specifies how to process oversized received packets.</p> <ul style="list-style-type: none"> • Truncate – truncate the packet to the <i>Maximum Rx Data Packet Size</i>. • Drop – drop the packet. <p>Default = Truncate</p>
Rx MS Byte First	<p>When you select this check box, the DeviceMaster receives the Most Significant (MS) byte of a WORD first. This check box is clear by default. Clear the check box when you need to receive the Least Significant (LS) byte of a WORD first.</p>
Tx MS Byte First	<p>This check box is clear by default. When you select this check box, DeviceMaster transmits the Most Significant (MS) byte of a WORD first. Clear the check box when you need to transmit the Least Significant (LS) byte of a WORD first.</p>
Disable Non-Filtered To PLC Rx Queue	<p>If filtering is disabled, only the last message received is sent to the PLC. This box is clear by default.</p>
Disable Tx Sequence Number Check	<p>Controls the transmit sequence number checking.</p> <ul style="list-style-type: none"> • If selected, the transmit sequence number checking is disabled. All transmit messages are transmitted if the sequence number has been incremented or not. • If not selected, the sequence number is checked and the message only is transmitted if the sequence number has been updated. <p>Default is <i>Not selected</i>.</p>
Modbus/TCP Master Rx/Tx Settings	
PLC IP Address	<p>Specifies the PLC IP Address of the PLC for operating in either <i>Receive</i> or <i>Transmit Master</i> mode.</p>
PLC Device ID	<p>Specifies the PLC Device ID of the PLC for operating in either <i>Receive</i> or <i>Transmit Master</i> mode. The default is 1.</p>
Disable Sequence Number/Length Header	<p>The data is transmitted without the 4 byte header (sequence number and data length).</p>

Device Modbus Configuration Page(Continued)	
Master Tx Mode Only	
PLC Rx Data Address	<p>Specifies the PLC address to write the received data at while operating in the <i>Master Receive Transfer</i> mode. The data area must be comprised of 16 bit words and large enough to contain the largest possible received message plus two 16 bit words for the sequence and length parameters.</p> <p>The address is base 1. Therefore, if your address scheme starts at zero, you need to add one to your address.</p>
Maximum PLC Update Rate	<p>The maximum rate (or minimum time interval) in milliseconds that messages are sent to the PLC in the <i>Master Receive Transfer</i> mode. This setting configures the DeviceMaster to space the messages to the PLC in order to prevent overrunning of data before the PLC can process it.</p>
Use Maximum Sized Modbus Messages	<p>Controls the size of Modbus/TCP messages used to write to the Modbus/TCP slave device.</p> <ul style="list-style-type: none"> • If selected, maximum sized Modbus messages of 242 bytes per message. • If not selected, Modbus messages of no more than 200 bytes are used. <p>The default is <i>Not selected</i>.</p> <p>Note: <i>This option only takes affect when large messages are received. Selecting this option may decrease the number of messages sent to the Modbus/TCP slave, thereby reducing network traffic and latency. However, not all Modbus/TCP slaves support maximum sized Modbus messages. So, this setting must be tested with the Modbus/TCP slave to ensure operability</i></p>
Master Tx Mode Only	
PLC Tx Data Address (Base 1)	<p>Specifies the PLC address to request transmit data messages while operating in the <i>Master Transmit Transfer</i> mode. The data area must be comprised of 16 bit words and contain a sequence number, length, and data to transmit. An updated sequence number indicates new data to transmit. Therefore, the length and data must be written to the transmit data area before updating the sequence number.</p> <p>The address is base 1. Therefore, if your address scheme starts at zero, you need to add one to your address.</p>
PLC Tx Poll Rate (msec)	<p>Specifies the rate, in milliseconds, that the DeviceMaster polls the PLC for transmit data. The default is 100ms</p>
PLC TX Poll Message Length (bytes)	<p>Specifies the length, in bytes, of the transmit message the DeviceMaster requests from the PLC. This must be large enough to contain the largest size data packet plus four bytes for the sequence number and length fields. Any extra bytes received from the PLC are ignored. The default is 250 bytes</p>
Tx Sequence Number Syncing Enable	<p>If the Tx (From PLC) Transfer Mode is operating in <i>Master (Poll the PLC)</i>; this specifies whether or not to enable synchronizing transmit data messages between the PLC and the DeviceMaster. This setting is clear by default.</p>
PLC Tx Consumed Seq Nbr Address (Base 1)	<p>Specifies the PLC memory address at which the DeviceMaster writes the transmit consumed sequence number. This memory address must point to a 16-bit word and, like the other address definitions, is base 1. When the Tx Produced Sequence Number (at the PLC Tx Data Address) and this consumed sequence number are equal, the DeviceMaster has transmitted the last message and is ready for the next transmit message.</p>

5.5. Device Filtering/Data Extraction Configuration Page

Access the *Device Filtering/Data Extraction Configuration* page by clicking **Ethernet | Device | Filtering**.

The options on this page have the same meaning as the *Port Filtering/Data Extraction Configuration* page accessed by clicking **Serial | Port | Filtering**.

The screenshot shows the web interface for the DeviceMaster 4-Port 2E. The top navigation bar includes: CONTROL, Home, Serial, Ethernet, Modbus, Network, Diagnostics, System, DeviceMaster 4-Port 2E, and Logout. Below this is a breadcrumb trail: Overview, Device 1, Device 2, Device 3, Device 4. The main content area is titled "Device 1 Filtering/Data Extraction Configuration" and contains an "Advanced Configuration" section with the following settings:

- To PLC Filter Mode: Off
- To PLC Filter Options (RFID Only):
 - Antenna
 - Filter Value
 - Serial Number
- To PLC Filter Options (RFID/Barcode):
 - Company
 - Product/Location
 - Encoding/Numbering
- To Application Filter Mode: Off
- To Application Filter Options (RFID Only):
 - Antenna
 - Filter Value
 - Serial Number
- To Application Filter Options (RFID/Barcode):
 - Company
 - Product/Location
 - Encoding/Numbering
- RFID Antenna Grouping: None
- RFID Reader Interface Type: Unspecified
- Barcode UPC/EAN Standard 12-14 Digit Format: None
- Barcode UPC/EAN Eight Digit Format: None
- Filter Age Time (Time filtered after last read): 0 (min) 0 (sec) 100 (msec)
- Discard Unrecognized Data (RFID/Barcode): Off

A "Save" button is located at the bottom right of the configuration area. The footer of the page reads "© Copyright Control Corp."

See [4.5. Port Filtering/Data Extraction Configuration \(Raw/ASCII Only\) Page](#) on Page 51 for information about these options.

5.6. Application TCP Configuration

Access the *Device Application TCP Configuration* page by clicking **Ethernet | Device | Application Interface**.

The options on this page have the same meaning as the *Port Application TCP Configuration* page accessed by clicking **Serial | Port | Application Interface**.

The screenshot shows the 'Device 1 Application TCP Configuration' page. The interface includes a top navigation bar with 'CONTROL' logo and tabs for Home, Serial, Ethernet, Modbus, Network, Diagnostics, and System. The current page is 'Application Interface' under the 'Ethernet' tab. Below the navigation, there are tabs for 'Overview', 'Device 1', 'Device 2', 'Device 3', and 'Device 4'. The main content area is divided into two sections: 'Advanced Configuration' and 'Application Rx Packet ID Settings'. The 'Advanced Configuration' section contains the following settings: 'Enable' (checkbox), 'Listen' (checkbox), 'Listen Port' (text input: 8100), 'Connect To Mode' (dropdown: Never), 'Connect Port' (text input: 8110), 'Connect IP Address' (text input: 0.0.0.0), 'Disconnect Mode' (dropdown: Never), and 'Idle Timeout (ms)' (text input: 0). The 'Application Rx Packet ID Settings' section contains: 'Rx Timeout Between Packets (ms)' (text input: 0), 'STX (Start of Transmission) Rx Detect' (dropdown: none, Byte 1: 0, Byte 2: 0 (dec)), and 'ETX (End of Transmission) Rx Detect' (dropdown: none, Byte 1: 0, Byte 2: 0 (dec)). A 'Save' button is located at the bottom right of the configuration area. The footer of the page reads '© Copyright Control Corp.'

See [4.6. Application TCP Configuration \(Raw/ASCII Only\) Page](#) on Page 55 for information about these options.

Chapter 6. Modbus Menus

This chapter discusses the following Modbus web pages:

- [6.1. Known Modbus Slave Devices List Page](#)
- [6.2. Alias Device ID Functionality Overview](#) on Page 72
- [6.3. Modbus Alias Statistics Page](#) on Page 74
- [6.4. Modbus Alias Configuration Page](#) on Page 75

6.1. Known Modbus Slave Devices List Page

You can access the *Known Modbus Slave Device List* page by clicking **Modbus | Modbus Devices**.

Use the *Known Modbus Slave Device List* page to monitor any Modbus slave devices connected to the DeviceMaster. This page is updated every 20 seconds.

Known Modbus Slave Device List

[Reset Statistics](#)

Port 1 Raw Data:
N/A

Port 2 Modbus/ASCII Slave(s):

Device Id	Active?	Tx Requests	Rx Responses	Timeouts	Invalid Responses	Last Rsp Time	Avg Rsp Time	Min Rsp Time	Max Rsp Time
1	no	328	324	4	0	0.08 sec	0.08 sec	0.06 sec	0.09 sec
2	no	328	323	5	0	0.08 sec	0.08 sec	0.06 sec	0.10 sec
3	no	329	324	5	0	0.07 sec	0.08 sec	0.06 sec	0.09 sec
21	yes	9143	9143	0	0	0.06 sec	0.08 sec	0.04 sec	0.11 sec
37	yes	9142	9142	0	0	0.06 sec	0.08 sec	0.05 sec	0.23 sec
43	yes	9142	9142	0	0	0.06 sec	0.08 sec	0.05 sec	0.18 sec

Port 3 Modbus/RTU Slave(s):

Device Id	Active?	Tx Requests	Rx Responses	Timeouts	Invalid Responses	Last Rsp Time	Avg Rsp Time	Min Rsp Time	Max Rsp Time
16	yes	2275	2275	0	0	0.09 sec	0.07 sec	0.04 sec	0.09 sec

Port 4 Modbus/RTU Master:
N/A

© Copyright Control Corp.

Known Modbus Slave Device List Page	
Device Id	Unit identifier associated with this device.
Active?	Status of device. <ul style="list-style-type: none"> • Yes means that the last request did not time out. • No means that the last request timed out.
Tx Requests	Number of Modbus requests transmitted to this device.

Known Modbus Slave Device List Page (Continued)	
Rx Responses	Number of Modbus responses received from this device.
Timeouts	Number of response timeouts associated with this device.
Last Rsp Time	The last response time from Modbus slave device.
Avg Rsp Time	The average response time from Modbus slave device.
Min Rsp Time	The minimum response time from Modbus slave device.
Max Rsp Time	The maximum response time from Modbus slave device.
Tx Broadcasts	Number of broadcast messages transmitted to this device.
Invalid Responses	Number of invalid responses received from this device.

6.2. Alias Device ID Functionality Overview

One of the most common challenges people face when setting up Modbus systems are problems caused by the limited device ID range. The *Alias Device ID* functionality has been developed to help solve those problems.

The Modbus specification has the following limitations:

- Requires all devices attached to the gateway to be addressed by a device ID.
- Allows only 256 device IDs with a range of 0 to 255.
- Not all device IDs can be used for addressing devices.
 - Device ID 0 is reserved for broadcast messages
 - 1-247 are for device addressing
 - 248 to 255 are reserved for such things as gateway functions. The Modbus/TCP firmware reserves device ID 254 for Ethernet TCP/IP raw/ASCII devices and 255 for serial raw/ASCII devices.

The following are common problems that can occur as a result of the device ID limitations:

- A gateway must route Modbus messages based on the device ID. Therefore, it cannot route to multiple Modbus devices with the same device ID.
- It not always possible or practical to change the device ID of serial Modbus slave devices.
- It is not always possible or practical to modify the device IDs on existing Modbus master programs. This is often true when adding a SCADA system to an existing PLC controlled system.
- A Modbus master with one connection, such as serial PLC, requires connectivity to multiple Modbus slave devices with the same device ID and one or more of the slave devices are connected remotely to different gateways.

Note: The [Modbus Router firmware](#) has been designed to provide network-wide Modbus connectivity for serial Modbus masters.

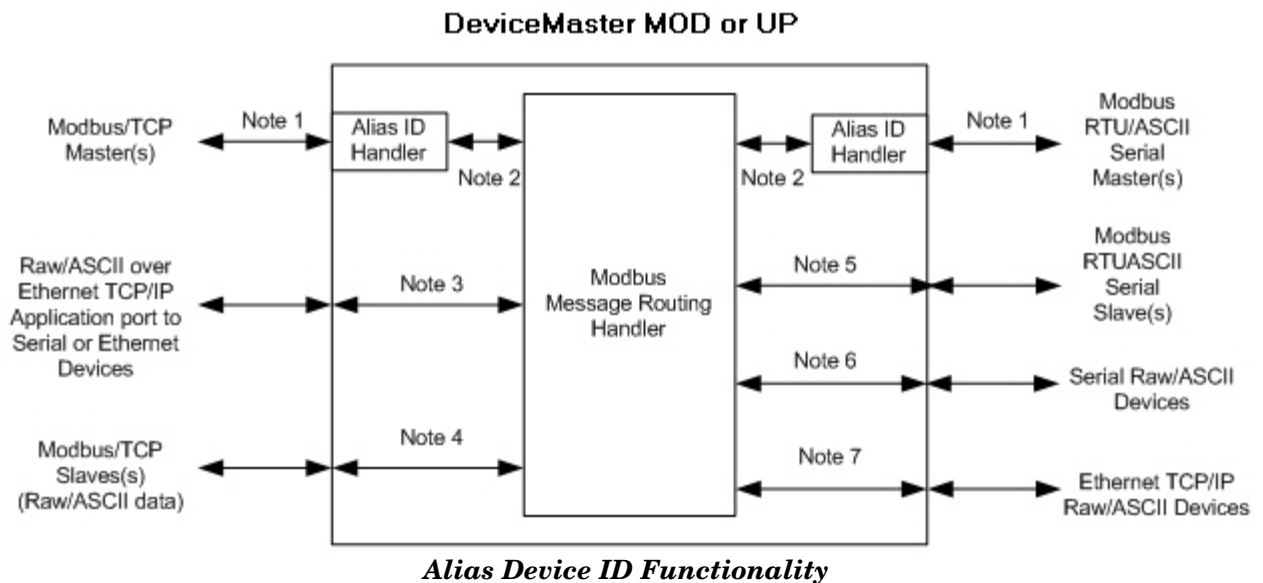
The Alias Modbus Device ID functionality allows modification of device IDs only when messages are received from Modbus masters. When configured, a Modbus message from a master with the specified device ID is converted to the alias device ID, the message is then routed internally using the alias device ID. All responses are returned to the master with the original received message device ID.

The following table demonstrates several device ID aliasing examples:

Received Device ID	Alias Device ID	Routed Message Device ID	Description
1	10	10	Convert messages with received device ID 1 to 10. Route message with device ID 10.
50	5	5	Convert messages with received device ID 50 to 5. Route message with device ID 5.

Received Device ID	Alias Device ID	Routed Message Device ID	Description
100	254	254	Convert messages with received device ID 100 to 254. Route message with device ID 254.
10	10	10	Invalid configuration attempt. No change to device ID is performed.

The functionality is described in the following diagram:



- Note 1:** These are the originally received Modbus messages. All responses are returned with the original device ID.
- Note 2:** Modbus messages sent to and responses received from Modbus Message Routing Handler. Depending on the Alias Device ID configuration, these messages may contain the originally received device ID or the alias device ID.
- Note 3:** The Alias Device ID functionality does not apply to raw/ASCII data received from Ethernet TCP/IP application connections.
- Note 4:** The Alias Device ID functionality does not apply to Modbus/TCP slaves when a raw/ASCII serial or Ethernet TCP/IP device is set to Master Rx and/or Tx mode. This is when the DeviceMaster is writing or reading raw/ASCII data from the Modbus/TCP slave device's memory.
- Note 5:** Modbus messages received from the Modbus Message Routing Handler and sent to Modbus slaves. Depending on the Alias ID configuration, these messages may contain the originally received device ID from the Modbus master or the alias device ID. All responses contain the device ID as received from the Modbus Message Routing Handler.
- Note 6:** Serial raw/ASCII devices must be addressed with a device ID of 255. The device ID of 255 may be either in the original message or derived from the alias device ID configuration.
- Note 7:** Ethernet TCP/IP raw/ASCII devices must be addressed with a device ID of 254. The device ID of 254 may be either in the original message or derived from the alias device ID configuration.

6.3. Modbus Alias Statistics Page

Access the *Modbus Alias Statistics* page by clicking **Modbus | Alias Diagnostics**.

Rx Device ID	Alias Device ID	Modbus/TCP Master	Modbus Serial Master	Modbus/TCP Count	Modbus Serial Count
101	91	yes	no	275	0
102	92	yes	no	276	0
103	96	no	yes	0	81

The following table explains the fields in the *Alias Modbus Device ID Configuration / Status* page.

Modbus Alias Statistics Page	
Rx Device ID	The device ID of the received message from a master.
Alias Device ID	The received device ID is converted to the Alias device ID.
Modbus/TCP Master	Alias applied to Modbus/TCP master messages if set to Yes.
Modbus Serial Master	Alias applied to serial Modbus master messages if set to Yes.
Modbus/TCP Count	The number of Alias conversions performed for this configuration to messages received from Modbus/TCP masters.
Modbus Serial Count	The number of Alias conversions performed for this configuration to messages received from serial Modbus masters.

6.4. Modbus Alias Configuration Page

Access the *Modbus Alias Configuration* page by clicking **Modbus | Alias Configuration**.

The screenshot shows the 'Modbus Alias Configuration' page. At the top, there is a navigation bar with 'CONTROL' logo and menu items: Home, Serial, Ethernet, Modbus (selected), Network, Diagnostics, System, DeviceMaster 1-Port IE, and Logout. Below the navigation bar, there are sub-menu items: Modbus Devices, Alias Diagnostics, and Alias Configuration (selected). The main content area is titled 'Modbus Alias Configuration' and contains an 'Add Alias Configuration' button. Below this is a table with the following data:

Rx Device ID	Alias Device ID	Modbus/TCP Master	Modbus Serial Master	Delete
1	50	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Below the table, there is a 'Delete All' button with a checkbox and a 'Save' button. The footer of the page reads '© Copyright Control Corp.'

The following configuration options apply:

Modbus Alias Configuration Page	
Rx Device ID	The device ID (also often called the unit ID) of the received message from a master. The range is from 1 to 247.
Alias Device ID	The alias device ID to convert the received device ID to. These values are permitted: 1-247,254,255.
Modbus/TCP Master	If selected, applies the alias device ID configuration to messages received from Modbus/TCP masters.
Modbus RTU/ASCII Serial Master	If selected, applies the alias device ID configuration to messages received from serial Modbus masters.

Section 7. Network Menus

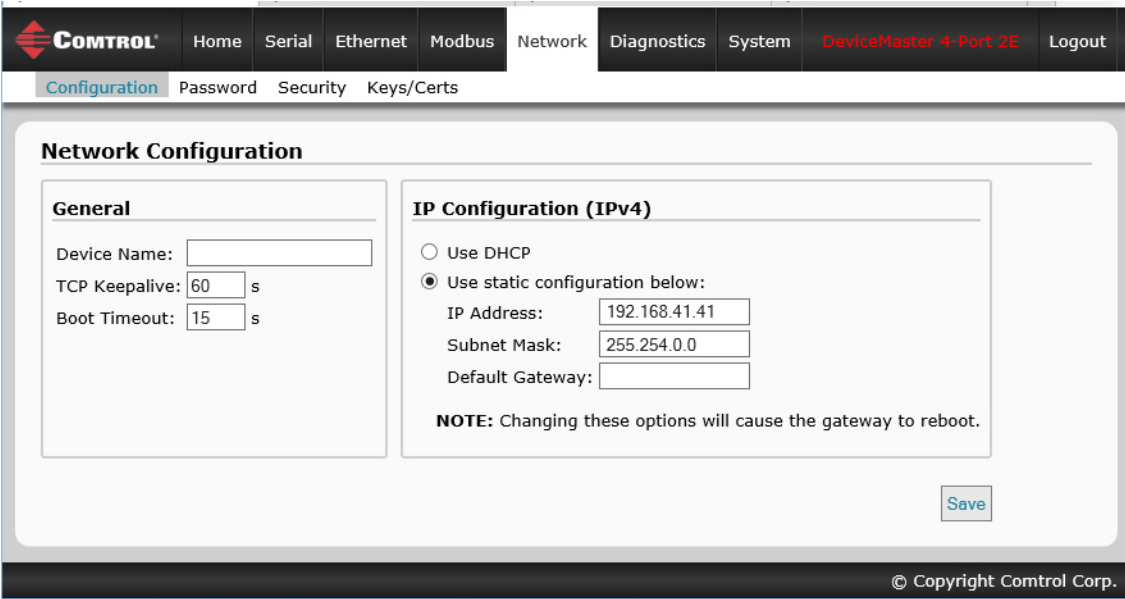
This section discusses the pages under the *Network* menu, which includes:

- [7.1. Network Configuration Page](#) on Page 77
- [7.2. Password Page](#) on Page 78
- [7.3. Security Page](#) on Page 80
- [7.4. Key and Certificate Management Page](#) on Page 81

7.1. Network Configuration Page

You can use the *Network Configuration* page to change the DeviceMaster network configuration after using PortVision DX for initial network configuration.

Click the **Network** tab to access this page.



The screenshot shows the 'Network Configuration' page in the Control interface. The top navigation bar includes 'CONTROL', 'Home', 'Serial', 'Ethernet', 'Modbus', 'Network', 'Diagnostics', 'System', 'DeviceMaster 4-Port 2E', and 'Logout'. Below this, a sub-menu shows 'Configuration', 'Password', 'Security', and 'Keys/Certs'. The main content area is titled 'Network Configuration' and is divided into two sections: 'General' and 'IP Configuration (IPv4)'. The 'General' section has fields for 'Device Name', 'TCP Keepalive: 60 s', and 'Boot Timeout: 15 s'. The 'IP Configuration (IPv4)' section has radio buttons for 'Use DHCP' and 'Use static configuration below:'. The 'Use static configuration below:' option is selected, and it includes fields for 'IP Address: 192.168.41.41', 'Subnet Mask: 255.254.0.0', and 'Default Gateway:'. A 'NOTE' states: 'Changing these options will cause the gateway to reboot.' A 'Save' button is located at the bottom right of the configuration area. The footer of the page reads '© Copyright Control Corp.'

The following table provides information about the **Network Configuration** fields.

Network Configuration Page	
General	
Device Name	You can enter a 16-character Device Name to identify this DeviceMaster on the <i>Home</i> page.
TCP Keepalive Default = 60	<p>The TCP protocol has a keepalive feature where the two network stacks periodically ping each other to make sure the connection is still up.</p> <p>Upon loss of a TCP/IP connection, the network stack starts a timer. If the TCP/IP connection is still lost after the number of seconds set by the TCP Keepalive value, then the DeviceMaster closes the connection and frees all the ports associated with the connection.</p> <p>If the DeviceMaster was the originator of the first connection, it will then try to re-connect the TCP/IP connection. This allows the DeviceMaster to be connected and ready to send/receive data even after a network disturbance.</p> <p>For most networks the default value should not need to be changed.</p>
Boot Timeout Default = 15	<p>Allows you to change the bootloader time-out value before the default application, Modbus Router loads.</p> <p>You may need to increase this time-out value to 45 for compatibility with spanning tree devices (normally switches). If you change the time-out value to 0, this prevents Modbus Router from loading.</p>
IP Configuration (IPv4)	
Use DHCP	<p>Configures the DeviceMaster to use DHCPv4 mode.</p> <p>If you select Use DHCP, the IPv4 Address field below is disabled and set to 0.0.0.0.</p> <p>See your System Administrator to acquire a unique reserved IPv4 address if you are using DHCP. They will need the MAC address of the unit to provide you with an IPv4 address.</p>
Use static configuration below	<p>Configures the DeviceMaster with the static IPv4 address information that you provide in the IPv4 Address, IPv4 Netmask, and IPv4 Gateway fields below.</p> <p>The DeviceMaster is shipped from the factory with the following default IPv4 network settings:</p> <ul style="list-style-type: none"> • IPv4 address = 192.168.250.250 • IPv4 Netmask = 255.255.0.0 • IPv4 Gateway address = 192.168.250.1

7.2. Password Page

You can easily set up a password to secure the DeviceMaster.

There is no password set from the factory.

The screenshot shows the Control web interface. At the top, there is a navigation bar with the Control logo and menu items: Home, Serial, Ethernet, Modbus, Network, Diagnostics, System, DeviceMaster 4-Port 2E, and Logout. Below this is a sub-menu with Configuration, Password (highlighted), Security, and Keys/Certs. The main content area is titled "Password Configuration" and contains three input fields: "Old Password:", "New Password:", and "Confirm New Password:". A "Save" button is located below the input fields. The footer of the page reads "© Copyright Control Corp."

Use the following information to configure a password for this DeviceMaster.

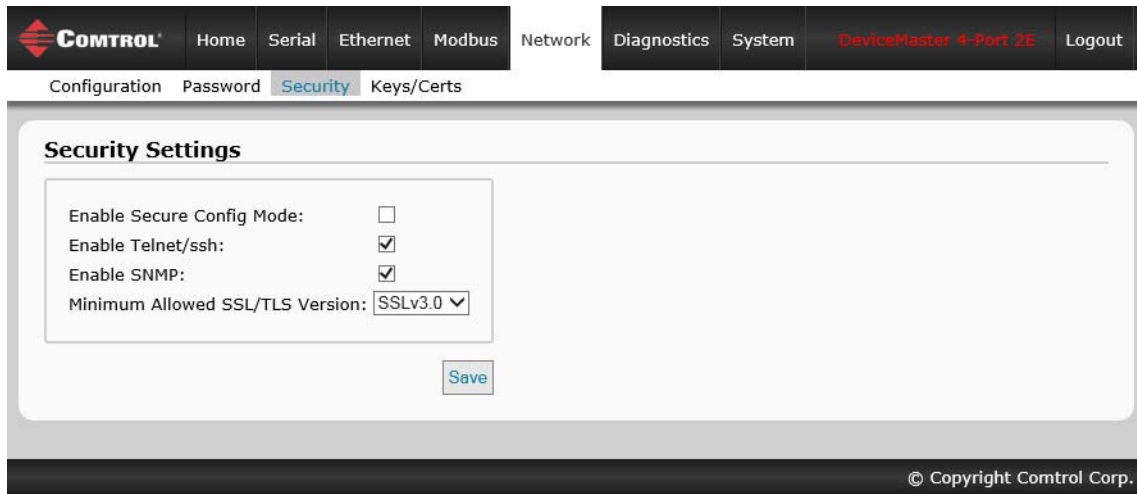
1. If necessary, click **Network | Password**.
2. If changing an existing password, enter that password in the **Old Password** field.
3. Enter a new password.
4. Enter the password in the **Confirm New Password** field.
5. Click the **Save** button.

To log into the DeviceMaster, you must enter the following:

- **admin** for the username
- The configured password for the password

7.3. Security Page

This table discusses Security Settings options.



Security Settings Page	
Enable Secure Config Mode	<p>If Secure Config mode is enabled, unencrypted access to administrative and diagnostic functions is disabled. Secure Config mode changes DeviceMaster behavior as follows:</p> <ul style="list-style-type: none"> • Telnet access to administrative and diagnostic functions is disabled. SSH access is still allowed. • Unencrypted access to the web server via port 80 (http:// URLs) is disabled. Encrypted access to the web server via port 443 (https:// URLs) is still allowed. • Administrative commands that change configuration or operating state which are received using the Control proprietary TCP driver protocol on TCP port 4606 are ignored. • Administrative commands that change configuration or operating state that are received using the Control MAC mode proprietary Ethernet protocol number 0x11FE are ignored.
Enable Telnet/ssh	This option enables or disables the telnet security feature after you click Save and the DeviceMaster has been rebooted. This option is enabled by default.
Enable SNMP	This option enables or disables the SNMP security feature after you click Save and the DeviceMaster has been rebooted. This option is enabled by default.
Minimum Allowed SSL/TLS Version	<p>You can select the appropriate version for your environment.</p> <ul style="list-style-type: none"> • SSLv3.0 • TLSv1.0 (default) • TLSv1.1 • TLSv1.2

Use the following steps to change security settings in the DeviceMaster.

1. Click the **Network | Security**.
2. Click the appropriate check boxes in the *Security Settings* page to enable or disable security accordingly.
3. After making changes, you must click **Save**.
4. You may need to configure security keys or certificates depending on your choices using the next subsection.

7.4. Key and Certificate Management Page

For secure operation, the DeviceMaster uses a set of three keys and certificates. These keys and certificates are user configurable.

Note: All DeviceMaster units are shipped from the factory with identical configurations. They all have the identical self-signed, Control Server RSA Certificates, Server RSA Keys, and Server DH Keys.

For maximum data and access security, you should configure all DeviceMaster units with custom certificates and keys.

The screenshot shows the 'Key and Certificate Management' page in the DeviceMaster web interface. The page has a dark navigation bar at the top with the 'CONTROL' logo and menu items: Home, Serial, Ethernet, Modbus, Network, Diagnostics, System, DeviceMaster 4-Port 2E, and Logout. Below the navigation bar is a secondary menu with 'Configuration', 'Password', 'Security', and 'Keys/Certs' (which is highlighted). The main content area is titled 'Key and Certificate Management' and contains three rows of settings, each with a 'Browse' and 'Delete' button:

RSA Key pair used by SSL and SSH servers:	Factory	Browse	Delete
RSA Server Certificate used by SSL servers:	Factory	Browse	Delete
DH Key pair used by SSL servers:	Factory	Browse	Delete

Below these settings is a 'Save' button. A 'Note' section follows, stating: 'Key and certificate changes will take effect after a reboot. Files must be in DER format. The RSA key and RSA certificate are used together by clients to authenticate the identity of the server. If you update one without updating the other, clients will be unable to authenticate the server and you will receive warnings from the web browser and other SSL clients.'

© Copyright Control Corp.

Key and Certificate Management Page	
<p>RSA Key pair used by SSL and SSH servers</p>	<p>This is a private/public key pair that is used for two purposes:</p> <ul style="list-style-type: none"> • It is used by some cipher suites to encrypt the SSL/TLS handshaking messages. Possession of the private portion of this key pair allows an eavesdropper to both decrypt traffic on SSL/TLS connections that use RSA encryption during handshaking. • It is used to sign the Server RSA Certificate in order to verify that the DeviceMaster MOD or DeviceMaster UP is authorized to use the server RSA identity certificate. <p>Note: <i>Possession of the private portion of this key pair allows somebody to pose as the DeviceMaster.</i></p> <p>If the Server RSA Key is to be replaced, a corresponding RSA identity certificate must also be generated and uploaded or clients are not able to verify the identity certificate.</p>
<p>RSA Server Certificate used by SSL servers</p>	<p>This is the RSA identity certificate that the DeviceMaster uses during SSL/TLS handshaking to identify itself. It is used most frequently by SSL server code in the DeviceMaster when clients open connections to the DeviceMaster's secure web server or other secure TCP ports.</p> <p>If a DeviceMaster serial port configuration is set up to open (as a client) a TCP connection to another server device, the DeviceMaster also uses this certificate to identify itself as an SSL client if requested by the server.</p> <p>In order to function properly, this certificate must be signed using the Server RSA Key. This means that the server RSA certificate and server RSA key must be replaced as a pair.</p>
<p>DH Key pair used by SSL servers</p>	<p>This is a private/public key pair that is used by some cipher suites to encrypt the SSL/TLS handshaking messages.</p> <p>Note: <i>Possession of the private portion of the key pair allows an eavesdropper to decrypt traffic on SSL/TLS connections that use DH encryption during handshaking.</i></p>

Use the following steps to update security keys and certificates in the DeviceMaster.

1. Click the **Network | Keys/Cert.**
2. Click **Browse** to locate the key or certificate file, highlight the file, and click **Open**.
3. Click **Upload** when you return to the Key and Certificate Management page.

The key or certificate notation changes from **factory** or **none** to **User** when the DeviceMaster is secure.

Note: *You do not need to click **Save**, but changes will not take effect until the DeviceMaster is rebooted.*

Chapter 8. Diagnostic Menus

You can access the following diagnostic and statistics pages from the *Diagnostics* menu:

This section discusses the following pages:

- [8.1. Serial Communication Statistics Page](#) on Page 83
- [8.2. Ethernet Device Statistics Page](#) on Page 86
- [8.3. Modbus/TCP and Serial Modbus Master Statistics](#) on Page 88
- [8.4. Serial Interface Logs](#) on Page 91
- [8.5. Ethernet Device Logs](#) on Page 92
- [8.6. System Log](#) on Page 93

8.1. Serial Communication Statistics Page

The default *Diagnostics* menu page is the *Serial Communication Statistics* page.

Serial Communication Statistics

[Reset Statistics](#)

Serial Interface	Port 1	Port 2	Port 3	Port 4
Tx Byte Count:	8508	487560	18944	14513
Tx Pkt Count:	254	28680	2368	613
Rx Byte Count:	78254	1534120	56875	4904
Rx Pkt Count:	1944	28576	2275	613
Parity Error Count:	0	0	0	0
Framing Error Count:	0	0	0	0
Overrun error count:	0	0	0	0
To PLC Dropped Packet Count:	685	N/A	N/A	N/A
To PLC Truncated Packet Count:	0	N/A	N/A	N/A
Tx Unexpected Seq Errors:	1	N/A	N/A	N/A
Invalid Modbus Message/Response Count:	N/A	0	0	0
Device Timeouts:	N/A	0	0	N/A
Cmd/Resp Mode Response Discard:	N/A	N/A	N/A	N/A
Filtering Statistics				
Valid Data Items Sent to PLC Interface:	0	N/A	N/A	N/A
Valid Data Items Filtered From PLC:	0	N/A	N/A	N/A
Invalid Data Items Discarded From PLC:	0	N/A	N/A	N/A
Valid Data Items Sent to App Interface:	0	N/A	N/A	N/A
Valid Data Items Filtered From App:	0	N/A	N/A	N/A
Invalid Data Items Discarded From App:	0	N/A	N/A	N/A
RFID Tags With Unknown Formats:	0	N/A	N/A	N/A

© Copyright Comtrol Corp.

Note: This screen shot is not displaying Application Connection Statistics, which only display if it is configured.

Serial Communications Statistics Page																																				
Reset Statistics	This button clears the serial port statistics, which sets the value to 0 for all ports.																																			
Serial Interface																																				
Tx Byte Count	Displays the number of bytes sent out of the serial port.																																			
Tx Pkt Count	Displays the number of serial packets sent out of the serial port.																																			
Rx Byte Count	Displays the number of bytes received over the serial port.																																			
Rx Pkt Count	Displays the number of packets received over the serial port.																																			
Parity Error Count	Displays the number of received serial packets dropped due to parity errors.																																			
Framing Error Count	Displays the number of received serial packets dropped due to framing errors.																																			
Overrun Error Count	Displays the number of received serial packets dropped due to overrun error incidents.																																			
To PLC Dropped Packet Count	Displays the number of received serial packets intended for the PLC dropped: <ul style="list-style-type: none"> • No STX byte(s) found • No ETX byte(s) found • Time-outs • Packet too large • Receive buffer queue overflows 																																			
To PLC Truncated Packet Count	Displays the number of received packets that were truncated before being sent to the PLC.																																			
Tx Unexpected Seq Errors	Displays the number of <i>Unexpected Transmit Sequence Number</i> errors. The DeviceMaster increments this number when the DeviceMaster receives a transmit message with a sequence number that is not equal to either the previous transmit sequence number or the previous transmit sequence number plus one. (If the Disable Tx Sequence Number Check option is not selected, the DeviceMaster expects this sequence number to be incremented by one with each new transmit message.)																																			
Invalid Modbus Message/Response Count	Displays the number of invalid Modbus To-Master messages or Modbus To-Slaves responses that were received on this port.																																			
Device Timeouts	The number of Command/Response or Modbus To-Slaves messages that timed out waiting for a response.																																			
Cmd/Resp Mode Response Discards	Displays the number of raw-data Command/Response mode responses that were discarded as a result of either: <ul style="list-style-type: none"> • The connection to the controller was closed. • The response timed out after the Age Time had been reached. 																																			
Filtering Statistics	<p>Note: <i>Filtering Statistics are only displayed if filtering is enabled on or more serial ports.</i></p> <p style="text-align: center;">Filtering Statistics</p> <table style="margin-left: auto; margin-right: auto;"> <tbody> <tr> <td>Valid Data Items Sent to PLC Interface:</td> <td style="text-align: right;">0</td> <td style="text-align: right;">0</td> <td style="text-align: right;">0</td> <td style="text-align: right;">0</td> </tr> <tr> <td>Valid Data Items Filtered From PLC:</td> <td style="text-align: right;">0</td> <td style="text-align: right;">0</td> <td style="text-align: right;">0</td> <td style="text-align: right;">0</td> </tr> <tr> <td>Invalid Data Items Discarded From PLC:</td> <td style="text-align: right;">0</td> <td style="text-align: right;">0</td> <td style="text-align: right;">0</td> <td style="text-align: right;">0</td> </tr> <tr> <td>Valid Data Items Sent to App Interface:</td> <td style="text-align: right;">0</td> <td style="text-align: right;">0</td> <td style="text-align: right;">0</td> <td style="text-align: right;">0</td> </tr> <tr> <td>Valid Data Items Filtered From App:</td> <td style="text-align: right;">0</td> <td style="text-align: right;">0</td> <td style="text-align: right;">0</td> <td style="text-align: right;">0</td> </tr> <tr> <td>Invalid Data Items Discarded From App:</td> <td style="text-align: right;">0</td> <td style="text-align: right;">0</td> <td style="text-align: right;">0</td> <td style="text-align: right;">0</td> </tr> <tr> <td>RFID Tags With Unknown Formats:</td> <td style="text-align: right;">0</td> <td style="text-align: right;">0</td> <td style="text-align: right;">0</td> <td style="text-align: right;">0</td> </tr> </tbody> </table>	Valid Data Items Sent to PLC Interface:	0	0	0	0	Valid Data Items Filtered From PLC:	0	0	0	0	Invalid Data Items Discarded From PLC:	0	0	0	0	Valid Data Items Sent to App Interface:	0	0	0	0	Valid Data Items Filtered From App:	0	0	0	0	Invalid Data Items Discarded From App:	0	0	0	0	RFID Tags With Unknown Formats:	0	0	0	0
Valid Data Items Sent to PLC Interface:	0	0	0	0																																
Valid Data Items Filtered From PLC:	0	0	0	0																																
Invalid Data Items Discarded From PLC:	0	0	0	0																																
Valid Data Items Sent to App Interface:	0	0	0	0																																
Valid Data Items Filtered From App:	0	0	0	0																																
Invalid Data Items Discarded From App:	0	0	0	0																																
RFID Tags With Unknown Formats:	0	0	0	0																																

Serial Communications Statistics Page (Continued)																																				
Valid Data Items Sent To PLC Interface	Displays the number of valid string, RFID, or barcode data sent to the PLC. Applies when filtering is enabled.																																			
Valid Data Items Filtered From PLC	Displays the number of valid string, RFID, or barcode data filtered from (not sent) to the PLC. Applies when filtering is enabled.																																			
Invalid Data Items Discarded From PLC	Displays the number of invalid RFID or barcode data not sent to the PLC. Applies when RFID or barcode filtering is enabled.																																			
Valid Data Items Sent To App Interface	Displays the number of valid string, RFID, or barcode data sent to the application. Applies when filtering is enabled.																																			
Valid Data Items Filtered From App	Displays the number of valid string, RFID, or barcode data filtered from (not sent) to the application. Applies when filtering is enabled.																																			
Invalid Data Items Discarded From App	Displays the number of invalid RFID or barcode data not sent to the PLC. Applies when RFID or barcode filtering is enabled.																																			
RFID Tags With Unknown Formats	Data received that was in the general form of 64 or 96 bit RFID tags, but was not in any of the EPCglobal formats. Applies only when RFID filtering is enabled.																																			
Application Connection Statistics	<p><i>Note: Application Connection Statistics are only displayed if an application connection is enable on one or more serial ports.</i></p> <table style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th colspan="5" style="text-align: center;">Application Connection Statistics</th> </tr> </thead> <tbody> <tr> <td>TX Byte Count:</td> <td style="text-align: right;">0</td> <td style="text-align: right;">0</td> <td style="text-align: right;">0</td> <td style="text-align: right;">0</td> </tr> <tr> <td>TX Pkt Count:</td> <td style="text-align: right;">0</td> <td style="text-align: right;">0</td> <td style="text-align: right;">0</td> <td style="text-align: right;">0</td> </tr> <tr> <td>To Application Dropped Packet Count:</td> <td style="text-align: right;">0</td> <td style="text-align: right;">0</td> <td style="text-align: right;">0</td> <td style="text-align: right;">0</td> </tr> <tr> <td>RX Byte Count:</td> <td style="text-align: right;">0</td> <td style="text-align: right;">0</td> <td style="text-align: right;">0</td> <td style="text-align: right;">0</td> </tr> <tr> <td>RX Pkt Count:</td> <td style="text-align: right;">0</td> <td style="text-align: right;">0</td> <td style="text-align: right;">0</td> <td style="text-align: right;">0</td> </tr> <tr> <td>To Device Dropped Packet Count:</td> <td style="text-align: right;">0</td> <td style="text-align: right;">0</td> <td style="text-align: right;">0</td> <td style="text-align: right;">0</td> </tr> </tbody> </table>	Application Connection Statistics					TX Byte Count:	0	0	0	0	TX Pkt Count:	0	0	0	0	To Application Dropped Packet Count:	0	0	0	0	RX Byte Count:	0	0	0	0	RX Pkt Count:	0	0	0	0	To Device Dropped Packet Count:	0	0	0	0
Application Connection Statistics																																				
TX Byte Count:	0	0	0	0																																
TX Pkt Count:	0	0	0	0																																
To Application Dropped Packet Count:	0	0	0	0																																
RX Byte Count:	0	0	0	0																																
RX Pkt Count:	0	0	0	0																																
To Device Dropped Packet Count:	0	0	0	0																																
TX Byte Count	Displays the number of bytes sent out the application socket port.																																			
TX Pkt Count	Displays the number of packets sent out the application socket port.																																			
To Application Dropped Packet Count	Displays the number of received serial or Ethernet device packets intended for the application dropped: <ul style="list-style-type: none"> • No STX byte(s) found • No ETX byte(s) found • Time-outs • Packet to large • Receive buffer queue overflows • Application connection is off line 																																			
To PLC Truncated Packet Count	Displays the number of received packets that were truncated before being sent to the PLC.																																			
RX Byte Count	Displays the number of bytes received over the application socket port.																																			
RX Pkt Count	Displays the number of packets received over the application socket port.																																			
To Device Dropped Packet Count	Displays the number of dropped packets that were intended for the device.																																			

8.2. Ethernet Device Statistics Page

Access the *Ethernet Device Statistics* page by clicking **Diagnostics | Ethernet Statistics**.

Ethernet Device Statistics

[Reset Statistics](#)

Ethernet Interface	Device 1	Device 2	Device 3	Device 4
Remote Connection Status:	(no connection)	(no connection)	(no connection)	(no connection)
Tx Byte Count:	0	0	0	0
Tx Pkt Count:	0	0	0	0
Rx Byte Count:	0	0	0	0
Rx Pkt Count:	0	0	0	0
To PLC Dropped Packet Count:	0	0	0	0
To PLC Truncated Packet Count:	0	0	0	0
Tx Unexpected Seq Errors:	0	0	0	0
Cmd/Resp Mode Device Timeouts:	0	0	0	0
Cmd/Resp Mode Response Discards:	0	0	0	0

© Copyright Control Corp.

Note: This screen shot does not display *Filtering or Application Interface Statistics* because those features were not configured.

Ethernet Device Statistics Page	
Reset Statistics	This button clears the socket port statistics, which sets the value to 0 for all ports.
Ethernet Interface	
Remote Connection Status	Displays the connected Ethernet device IP address and port.
Tx Byte Count	Displays the number of bytes sent out the device socket port.
Tx Pkt Count	Displays the number of packets sent out the device socket port.
Rx Byte Count	Displays the number of bytes received over the device socket port.
Rx Pkt Count	Displays the number of packets received over the device socket port.
To PLC Dropped Packet Count	Displays the number of dropped packets that were intended for the PLC.
To PLC Truncated Packet Count	Displays the number of received packets that were truncated before being sent to the PLC.
Tx Unexpected Sequence Errors	Same as the serial port statistics (Page 84).
Cmd/Resp Mode Device Timeouts	The number of Command/Response messages that timed out waiting for a response.
Cmd/Resp Mode Response Discards	Displays the number of Command/Response mode responses that were discarded as a result of either: The connection to the controller was closed. The response timed out after the Age Time had been reached.

Ethernet Device Statistics Page (Continued)	
Filtering Statistics	<i>Note: Filtering Statistics are only displayed if filtering is enabled on one or more Ethernet device interface.</i>
Valid Data Items Sent To PLC Interface	Displays the number of valid string, RFID, or barcode data sent to the PLC. Applies when filtering is enabled.
Valid Data Items Filtered From PLC	Displays the number of valid string, RFID, or barcode data filtered from (not sent) to the PLC. Applies when filtering is enabled.
Invalid Data Items Discarded From PLC	Displays the number of invalid RFID or barcode data not sent to the PLC. Applies when RFID or barcode filtering is enabled.
Valid Data Items Sent To App Interface	Displays the number of valid string, RFID, or barcode data sent to the application. Applies when filtering is enabled.
Valid Data Items Filtered From App	Displays the number of valid string, RFID, or barcode data filtered from (not sent) to the application. Applies when filtering is enabled.
Invalid Data Items Discarded From Application	Displays the number of invalid RFID or barcode data not sent to the PLC. Applies when RFID or barcode filtering is enabled.
RFID Tags With Unknown Formats	Data received that was in the general form of 64 or 96 bit RFID tags, but was not in any of the EPCglobal formats. Applies only when RFID filtering is enabled.
Application Connection Statistics	<i>Note: Application Connection Statistics are only displayed if an Application connection is enabled on or more Ethernet device interfaces.</i>
TX Byte Count	Displays the number of bytes sent out the application socket port.
TX Pkt Count	Displays the number of packets sent out the application socket port.
To Application Dropped Packet Count	Displays the number of received serial or Ethernet device packets intended for the application dropped: <ul style="list-style-type: none"> • No STX byte(s) found • No ETX byte(s) found • Time-outs • Packet too large • Receive buffer queue overflows • Application connection is offline
To PLC Truncated Packet Count	Displays the number of received packets that were truncated before being sent to the PLC.
RX Byte Count	Displays the number of bytes received over the application socket port.
RX Pkt Count	Displays the number of packets received over the application socket port.
To Device Dropped Packet Count	Displays the number of dropped packets that were intended for the device.

8.3. Modbus/TCP and Serial Modbus Master Statistics

Access the *Modbus/TCP and Serial Modbus Master Statistics* page by clicking **Diagnostics | PLC Diagnostics**. It is intended to help with debugging PLC programs, monitoring the PLC interface, and solving configuration problems.

Modbus/TCP and Serial Modbus Master Statistics

[Reset Statistics](#)

Modbus/TCP Slave Mode Specific Statistics	
Messages Received From PLC:	2060079
Responses Sent To PLC:	2060079
Raw Serial Port Data Messages Received From PLC:	2029187
Raw Socket Port Data Messages Received From PLC:	0
Modbus RTU/ASCII Messages Received From PLC:	30892
Modbus RTU/ASCII Broadcasts Received From PLC:	0
Invalid Command Lengths:	0
Invalid Message Data Errors:	2
Unknown Request Destination IDs:	0
Invalid Request Protocol Types:	0
Unsupported Modbus Function Codes:	0
Modbus/TCP Master Mode Specific Statistics	
Messages Sent To PLC:	0
Responses Received From PLC:	0
Invalid Response Data Errors:	0
Error Responses:	0
Unexpected Response Function Codes:	0
Unknown Response Destination IDs:	0
Invalid Response Protocol Types:	0
Failed Modbus/TCP Connection Attempts:	0
Modbus/TCP Connection Problems:	0
No Available Modbus/TCP Connection Errors:	0
Non-Mode Specific Statistics/Diagnostics	
Oversized Received Data Packet Errors:	0
Improper Configuration Errors:	0
System Resource Errors:	0
Writes To Offline Ethernet Device On Socket 1	0
Writes To Offline Ethernet Device On Socket 2	0
Writes To Offline Ethernet Device On Socket 3	0
Writes To Offline Ethernet Device On Socket 4	0
First Error Description:	Modbus Read Request, Invalid address of 4096
Last Error Description:	Modbus Write Request - Unexpected transmit write sequence number, serial port 1, expected 1, rcvd 1691

© Copyright Control Corp.

Modbus/TCP and Serial Modbus Master Statistics Page

Reset Statistics	Selecting this button clears the values displayed on this page.
Modbus/TCP Slave Mode Specific Statistics	
Messages Received From PLC	Displays the total number of raw data and Modbus RTU/ASCII messages received from the PLC.
Responses Sent to PLC	Displays the total number of raw data and Modbus RTU/ASCII responses sent to the PLC.
Raw Serial Port Data Messages Received From PLC	Displays the number of serial port raw data messages received.
Raw Socket Port Data Messages Received From PLC	Displays the number of socket port raw data messages received.

Modbus/TCP and Serial Modbus Master Statistics Page (Continued)	
Modbus RTU/ASCII Messages Received From PLC	Displays the number of Modbus RTU/ASCII device specific messages received.
Modbus RTU/ASCII Broadcasts Received From PLC	Displays the number of Modbus RTU/ASCII broadcast messages received.
Invalid Command Lengths	Displays the number of messages received with invalid command lengths.
Invalid Message Data Errors	Displays the number of messages received with invalid message data errors. These errors occur when the DeviceMaster receives a message that cannot be processed due to improper message data.
Unknown Request Destination IDs	Displays the number of messages received with unknown request destination IDs.
Invalid Request Protocol Types	Displays the number of messages received with invalid protocol errors. This occurs when a message is received with a protocol other than the Modbus protocol value of zero.
Unsupported Modbus Function Codes	Displays the number of messages received with unsupported function codes.
Modbus/TCP Master Mode Specific Statistics	
Messages Sent To PLC	Displays the total number of raw data messages sent to the PLC.
Responses Received From PLC	Displays the total number of raw data responses received from the PLC.
Invalid Response Data Errors	Displays the number of response data errors to polling requests returned from the PLC. Possible causes include: <ul style="list-style-type: none"> • Incorrectly formed transmit data memory format. (Possibly missing the sequence number and/or length fields.) • More data to transmit indicated via the length field than was returned in the message. • Transmit sequence number error. Sequence number increased by more than one. This could indicate an unsent transmit message. • Insufficient polling length configuration. (DeviceMaster polling length is too small.) • Attempting to transmit too large of message for a single Modbus/TCP message. (More than 246 bytes of data.)
Error responses	Displays the number of responses received from the PLC with errors indicated. This occurs when the PLC returns a response with an error indication. This may be caused by such things as: <ul style="list-style-type: none"> • Invalid PLC address configuration • Improper PLC configuration
Unexpected Response Function Codes	Displays the number of unexpected response function codes from either a <i>Master Receive</i> or <i>Master Transmit</i> mode message. This occurs when a response was received without an expected function code.
Unknown Response Destination Ids	Displays the number of responses with unknown destination IDs. This occurs when the PLC returns a response with an unknown destination ID.
Invalid Response Protocol Types	Displays the number of responses with invalid protocol errors. This occurs when a response is returned with a protocol other than the Modbus protocol value of zero.
Failed Modbus/TCP Connection Attempts	Displays the number of failed Modbus/TCP connection attempts to the specified PLC IP address.

Modbus/TCP and Serial Modbus Master Statistics Page (Continued)	
Modbus/TCP Connection Problems	<p>Displays the number of Modbus/TCP connection attempt problems. This occurs when the device responds and the connection is made, but there are problems setting up the connection options.</p> <p>Possible problems include:</p> <ul style="list-style-type: none"> • Setting the TCP connection to <i>TCP_NODELAY</i>. • Setting the socket connection to <i>SO_OOBINLINE</i>. • Setting the socket connection to <i>SO_KEEPALIVE</i>
No Available Modbus/TCP Connection Errors	<p>Displays the number of connections aborted when there are no available Modbus/TCP connections. This error occurs when the maximum number of 32 Modbus/TCP connections has been reached and the DeviceMaster is attempting to form another Modbus/TCP connection.</p>
Non-Mode Specific Statistics/Diagnostics	
Oversized Received Data Packet Errors	<p>Displays the number of received serial or Ethernet data packets that were larger than the configured maximum receive data packet.</p>
Improper Configuration Errors	<p>Displays the number of improper configuration errors. These errors occur when the DeviceMaster receives a message that cannot be performed due to an invalid configuration.</p>
System Resource Errors	<p>Displays the number of system resource errors. These errors indicate a system error on the DeviceMaster such as an inoperable serial port or a full transmit queue. These errors typically occur when the PLC(s) are sending data to the DeviceMaster faster than the DeviceMaster can process it.</p>
Writes to Offline Ethernet Device on Socket N	<p>Displays the number of write attempts by a PLC to the Ethernet device when the device was offline.</p>
First Error Description	<p>Text description of the first error that occurred.</p>
Last Error Description	<p>Text description of the last or most recent error that occurred.</p>

8.4. Serial Interface Logs

Access the *Serial Interface Logs* page by clicking **Diagnostics | Serial Logs**.

The *Serial Interface Logs* page provides a log of received and transmitted serial port messages. Up to 128 bytes per message and up to 128 messages are logged. It is intended to help with debugging serial connectivity problems, determining the proper start and end of transmission bytes, and diagnosing device problems.

The format is as follows:

Pkt(n) : ddd:hh:mm:ss.ms Tx/Rx:<Data>

Where:

ddd – days since last system restart

hh – hours since last system restart

ms – minutes since last system restart

ss – seconds since last system restart

mm – milliseconds since last system restart

<Data> - Data packet received.

- For Modbus slave data, all data bytes shown in hex (xxh) format.
- For Raw/ASCII and Modbus/ASCII data
- ASCII characters displayed as characters
- Non-ASCII displayed in hex (xxh) format

Serial Interface Logs

[Reset Log](#)

Port 1 Rx/Tx Packets (first 128 packets, max of 128 bytes):

Pkt	ddd hh:mm:ss.ms	Tx/Rx	Data
1	001 00:06:36.709	Rx	!"\$%&'()*+,-./0123456789;<=>?@ABCDEFGHIJKLMNQRSTUvwxyz[\]^_`abcdefghijklmnopqrstuvwxyz!"\$%&'()*+,-./0123456789

Port 2 Rx/Tx Packets (first 128 packets, max of 128 bytes):

Pkt	ddd hh:mm:ss.ms	Tx/Rx	Data
1	001 00:06:28.158	Tx	(3Ah)(30h)(37h)(30h)(33h)(30h)(30h)(31h)(42h)(30h)(30h)(30h)(41h)(44h)(31h)(0Dh)(0Ah)
2	001 00:06:28.229	Rx	(3Ah)(30h)(37h)(30h)(33h)(31h)(34h)(30h)(37h)(31h)(34h)(30h)(37h)(31h)(35h)(30h)(37h)(31h)(36h)(30h)(37h)(31h)(37h)(30h)(37

Port 3 Rx/Tx Packets (first 128 packets, max of 128 bytes):

Pkt	ddd hh:mm:ss.ms	Tx/Rx	Data
1	001 00:06:32.159	Tx	(46h)(03h)(00h)(05h)(00h)(0Ah)(DAh)(BBh)
2	001 00:06:32.229	Rx	(46h)(03h)(14h)(46h)(10h)(46h)(11h)(46h)(12h)(46h)(13h)(46h)(14h)(46h)(15h)(46h)(16h)(46h)(17h)(46h)(18h)(46h)(19h)(F1h)(AD

Port 4 Rx/Tx Packets (first 128 packets, max of 128 bytes):

Pkt	ddd hh:mm:ss.ms	Tx/Rx	Data
1	001 00:06:32.159	Rx	(46h)(03h)(00h)(05h)(00h)(0Ah)(DAh)(BBh)
2	001 00:06:32.230	Tx	(46h)(03h)(14h)(46h)(10h)(46h)(11h)(46h)(12h)(46h)(13h)(46h)(14h)(46h)(15h)(46h)(16h)(46h)(17h)(46h)(18h)(46h)(19h)(F1h)(AD

© Copyright Control Corp.

8.5. Ethernet Device Logs

The *Ethernet Device Interface Logs* page is accessed using the Display Ethernet Device Logs option, which provides a log of received and transmitted Ethernet device messages. Up to 128 bytes per message and up to 128 messages are logged. It is intended to help with debugging Ethernet connectivity problems, determining the proper start and end of transmission bytes, and diagnosing device problems.

The format is as follows: **Pkt(n): ddd:hh:mm:ss.ms Tx/Rx:<Data>**

Where:

ddd – days since last system restart

hh – hours since last system restart

mm – minutes since last system restart

ss – seconds since last system restart

ms – milliseconds since last system restart

<Data> - Data packet received.

- ASCII characters displayed as characters
- Non-ASCII displayed in hex (xxh) format

The screenshot shows the 'Ethernet Device Interface Logs' page within the Control software. The interface includes a top navigation bar with 'CONTROL' logo and menu items: Home, Serial, Ethernet, Modbus, Network, Diagnostics, System, DeviceMaster 4-Port 2E, and Logout. Below this is a secondary navigation bar with 'Serial Statistics', 'Ethernet Statistics', 'PLC Diagnostics', 'Serial Logs', 'Ethernet Logs' (highlighted), and 'System Log'. The main content area is titled 'Ethernet Device Interface Logs' and contains a 'Reset Log' button. Below the button, there are four entries for Device 1, Device 2, Device 3, and Device 4, each showing 'Rx/Tx Packets (first 128 packets, max of 128 bytes):' followed by 'No data'. The footer of the page reads '© Copyright Control Corp.'

8.6. System Log

The *System Log* page provides system level information, which is updated every 10 seconds.

- Click the **Refresh** button to view the latest system log information.
- Click the **Clear** button to restart a new system log file.
- Click the **Save Logfile** button to save the system log file, if requested by Technical Support.

The screenshot shows the web interface for the DeviceMaster 4-Port 2E. The top navigation bar includes the CONTROL logo and menu items: Home, Serial, Ethernet, Modbus, Network, Diagnostics, System, DeviceMaster 4-Port 2E, and Logout. Below this, a secondary navigation bar highlights the System Log page, with other options like Serial Statistics, Ethernet Statistics, PLC Diagnostics, Serial Logs, and Ethernet Logs. The main content area is titled "System Log" and contains a scrollable text box with the following log entries:

```
sent - IP: 192.168.41.41 Net Mask: 255.254.0.0 Gateway: 0.0.0.0
MAC Admin: Current IP info request received, response sent - IP: 192.168.41.41 Net Mask: 255.254.0.
MAC Admin: Current IP info request received, response sent - IP: 192.168.41.41 Net Mask: 255.254.0.
MAC Admin: Current IP info request received, response sent - IP: 192.168.41.41 Net Mask: 255.254.0.
MAC Admin: Current IP info request received, response sent - IP: 192.168.41.41 Net Mask: 255.254.0.
MAC Admin: Current IP info request received, response sent - IP: 192.168.41.41 Net Mask: 255.254.0.
MAC Admin: Current IP info request received, response sent - IP: 192.168.41.41 Net Mask: 255.254.0.
MAC Admin: Current IP info request received, response sent - IP: 192.168.41.41 Net Mask: 255.254.0.
MAC Admin: Current IP info request received, response sent - IP: 192.168.41.41 Net Mask: 255.254.0.
MAC Admin: Current IP info request received, response sent - IP: 192.168.41.41 Net Mask: 255.254.0.
MAC Admin: Current IP info request received, response sent - IP: 192.168.41.41 Net Mask: 255.254.0.
MAC Admin: Current IP info request received, response sent - IP: 192.168.41.41 Net Mask: 255.254.0.
MAC Admin: Current IP info request received, response sent - IP: 192.168.41.41 Net Mask: 255.254.0.
MAC Admin: Current IP info request received, response sent - IP: 192.168.41.41 Net Mask: 255.254.0.
```

At the top right of the log area are three buttons: Refresh, Clear, and Save Logfile. The bottom of the interface shows the copyright notice: © Copyright Control Corp.

Section 9. System Menus

This section discusses the web pages under the **System** menu:

- [9.1. Update Firmware](#) on Page 95
- [9.2. Configuration File Page](#) on Page 98
- [9.3. System Snapshot Page](#) on Page 100
- [9.4. Restore Defaults Page](#) on Page 100
- [9.5. Reboot](#) on Page 101

9.1. Update Firmware

You can upload firmware (Modbus/TCP or Bootloader) using the **System | Update Firmware** page. You must first unpackage the firmware from the **.msi** file.

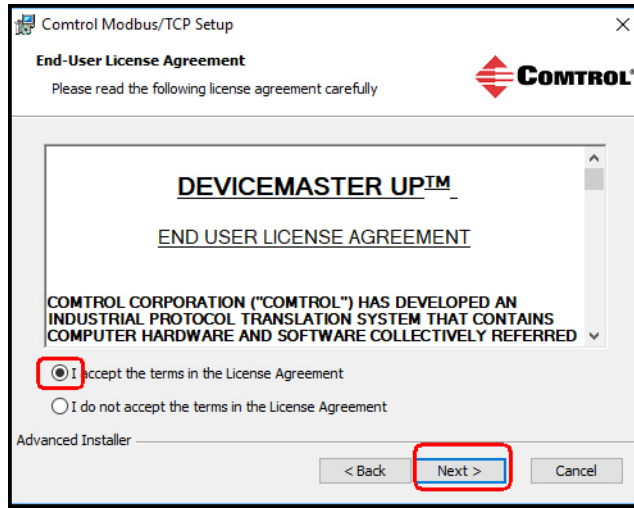
Note: *Optionally, you can use PortVision DX to upload firmware after unpackaging the .msi file.*

Use the following procedure to upload the latest firmware onto the DeviceMaster.

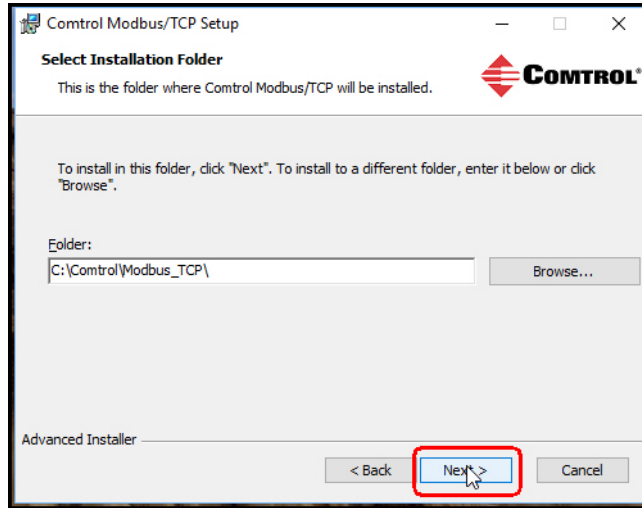
1. If necessary, download the firmware from the [Control download site](#).
2. Execute the **Modbus_Router_x.x.msi** file.
3. Click the **Next** button.



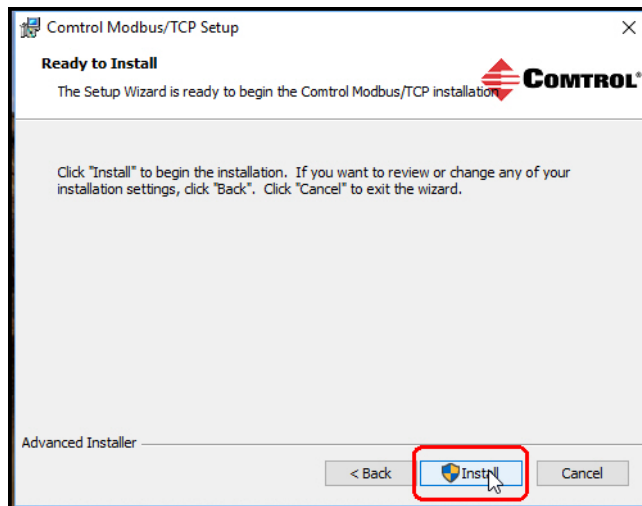
4. After reviewing the license, click **I accept the terms in the License Agreement** and the **Next** button.



5. Click the **Next** button or browse to the location you want the files stored.



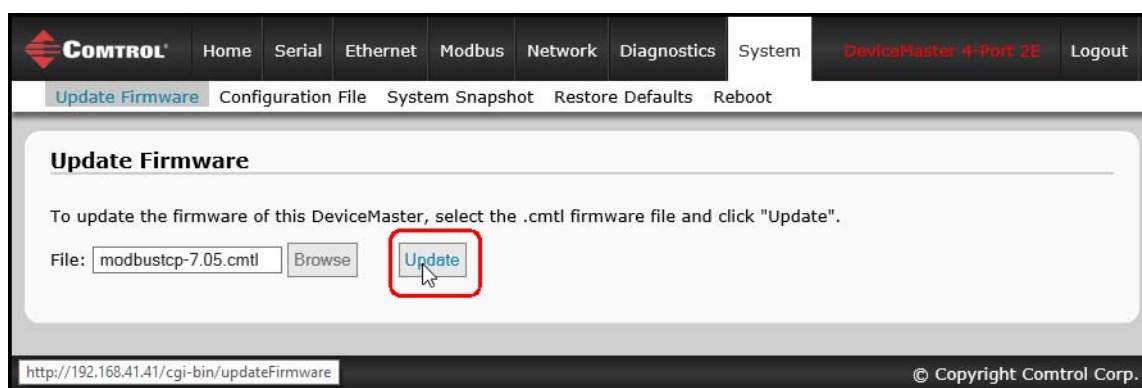
6. Click the **Install** button.



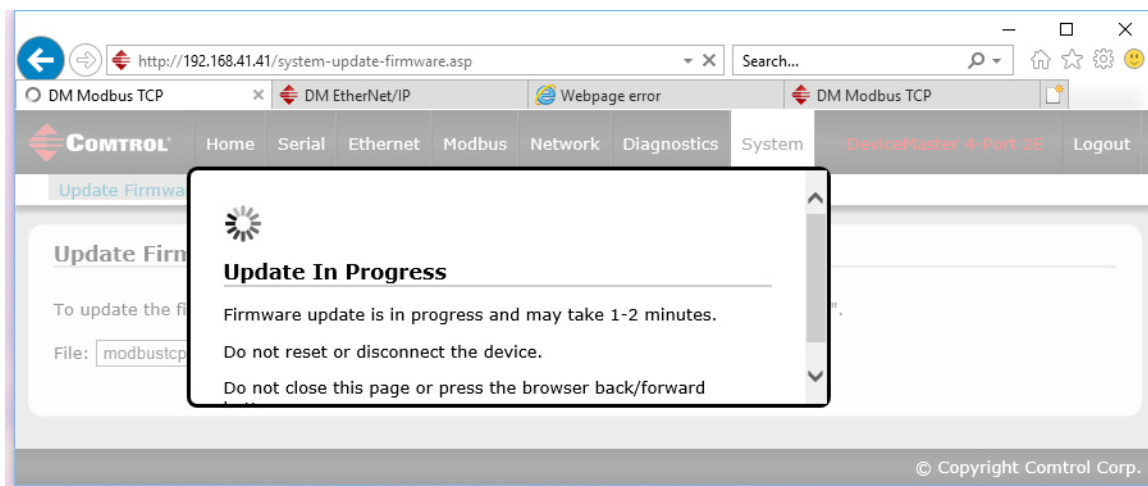
7. Click **Yes** to the **Do you want to allow this app to make changes to your device** pop up message.
8. Click the **Finish** button.
9. Open your web browser and enter the IP address of the DeviceMaster.
10. Click the **System** menu, which opens the *Update Firmware* page.
11. Click the **Browse** button, navigate to the file, select it and click the **Open** button.



12. Click the **Update** button.



An *Update In Progress* pop up notifies you with the upload duration, not to reset or disconnect the device or to close the page.



9.2. Configuration File Page

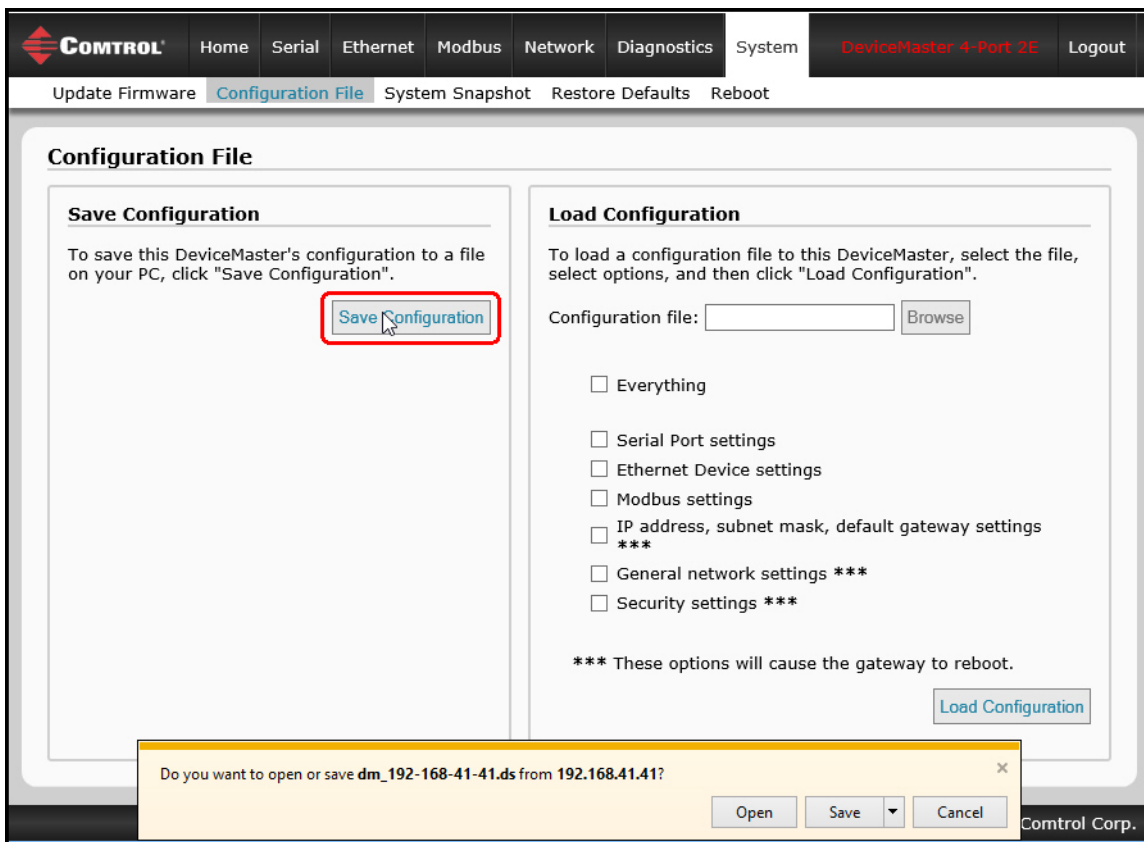
You can use the **Save Configuration** option to save a DeviceMaster configuration file for recovery purposes or to quickly configure other DeviceMasters that require the same configuration using the **Load Configuration** option.

Note: Optionally, you can use PortVision DX to save and load configuration files.

9.2.1. Saving a Configuration File

You can use this procedure to save a DeviceMaster configuration file.

1. Click **System | Configuration File**.
2. Click the **Save Configuration** button.
3. Save the configuration file following your browser prompts.



9.2.2. Loading a Configuration File

You can use this procedure to load a previously saved DeviceMaster configuration file.

1. Click **System | Configuration File**.
2. Click the **Browse** button, highlight the configuration file you want to load, and click the **Open** button.
3. Click the **Load Configuration** button.

The screenshot shows the 'Configuration File' page in the DeviceMaster web interface. The page is divided into two main sections: 'Save Configuration' and 'Load Configuration'. The 'Load Configuration' section is active and shows a file named 'dm_192-168-41-41.ds' selected. Below the file name are several checkboxes for configuration options: 'Everything' (unchecked), 'Serial Port settings' (checked), 'Ethernet Device settings' (checked), 'Modbus settings' (checked), 'IP address, subnet mask, default gateway settings ***' (unchecked), 'General network settings ***' (checked), and 'Security settings ***' (unchecked). A note at the bottom of the 'Load Configuration' section states: '*** These options will cause the gateway to reboot.' The 'Load Configuration' button is highlighted with a mouse cursor.

COMTROL Home Serial Ethernet Modbus Network Diagnostics System DeviceMaster 4-Port 2E Logout

Update Firmware Configuration File System Snapshot Restore Defaults Reboot

Configuration File

Save Configuration

To save this DeviceMaster's configuration to a file on your PC, click "Save Configuration".

[Save Configuration](#)

Load Configuration

To load a configuration file to this DeviceMaster, select the file, select options, and then click "Load Configuration".

Configuration file: [Browse](#)

Everything

Serial Port settings

Ethernet Device settings

Modbus settings

IP address, subnet mask, default gateway settings ***

General network settings ***

Security settings ***

*** These options will cause the gateway to reboot.

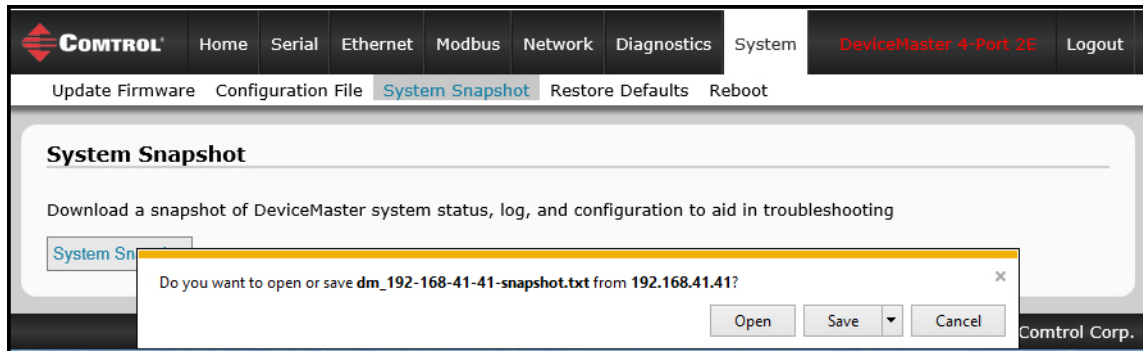
[Load Configuration](#)

© Copyright Control Corp.

9.3. System Snapshot Page

You can use the *System Snapshot* page to download a snapshot of the device status, log, and configuration. You may find the information can help you diagnose a problem with the DeviceMaster. In addition, this information may be requested by technical support if you have called for assistance.

1. Open your browser and enter the IP address of the DeviceMaster.
2. Click **System | System Snapshot**.
3. Click the **Device Snapshot** button.

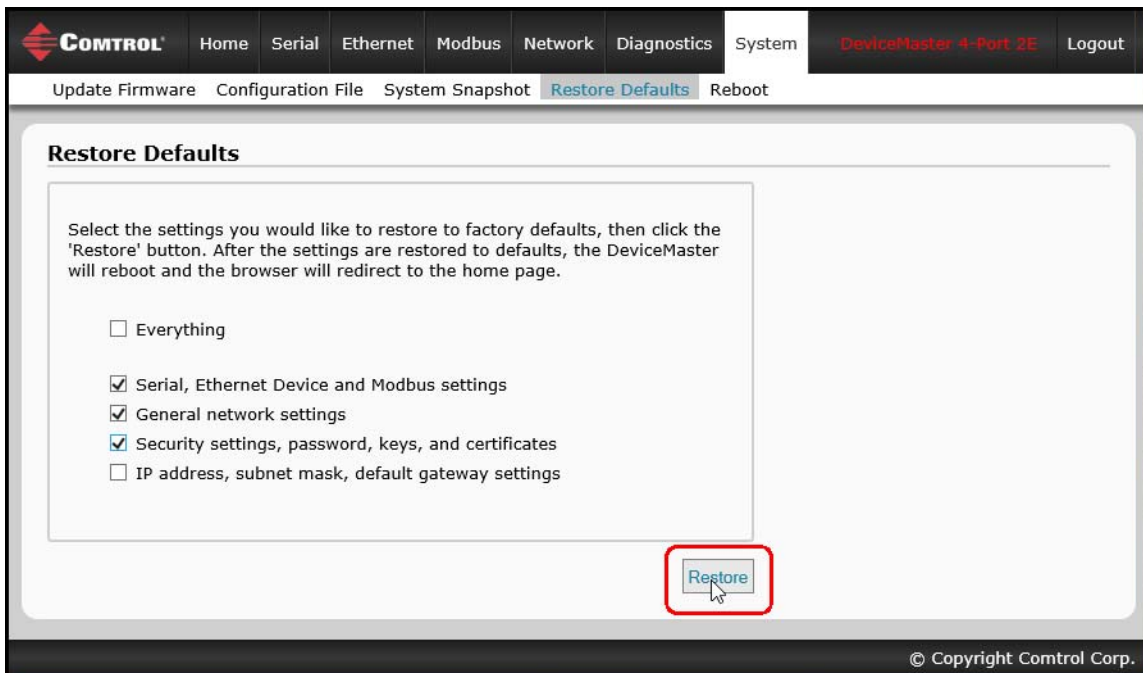


4. Save the file using the method for your browser.

9.4. Restore Defaults Page

You can easily some or all of your settings to factory defaults by using the procedure below.

1. Open your browser and enter the IP address of the DeviceMaster.
2. Click **System | Restore Defaults**.



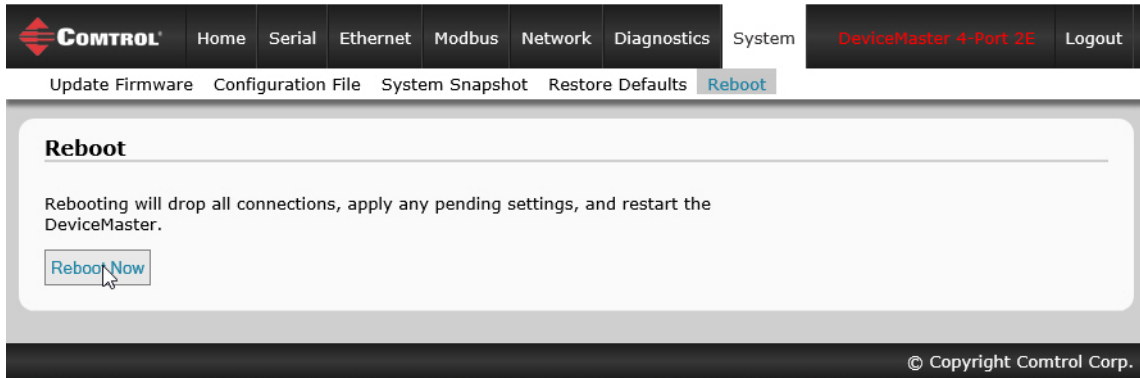
3. Select **Everything** or the specific setting or settings that you want to restore.
4. Click the **Restore** button.

5. The DeviceMaster reboots and re-opens the web interface.

9.5. Reboot

You can reboot the DeviceMaster remotely using the *Reboot* web page.

1. Click **System | Reboot**.
2. Click the **Reboot Now** button or wait the 10 seconds for it automatically reboot.



Chapter 10. Troubleshooting and Technical Support

You should review the *Troubleshooting* chapter in the [Modbus Hardware Installation and Configuration Guide](#) before calling Technical Support because they will request that you perform many of the procedures or verifications before they can help you diagnose a problem.

- [10.1. Troubleshooting Checklist](#) on Page 103
- [10.2. General Troubleshooting](#) on Page 104

If you cannot diagnose the problem, you can contact [10.3. Technical Support](#) on Page 104.

10.1. Troubleshooting Checklist

The following checklist may help you diagnose your problem:

- Verify that you are using the correct types of cables on the correct connectors and that all cables are connected securely.

Note: *Most customer problems reported to Control Technical Support are eventually traced to cabling or network problems.*

- Isolate the DeviceMaster from the network by connecting the device directly to a NIC in a host system.
- Verify that the Ethernet hub and any other network devices between the system and the DeviceMaster are powered up and operating.
- Reset the power on the DeviceMaster and watch the **PWR** or **Status** light activity.

PWR or Status LED	Description
5 sec. off, 3 flashes, 5 sec. off, 3 flashes...	RedBoot™ checksum failure.
5 sec. off, 4 flashes, 5 sec. off, 4 flashes...	SREC load failure.

- If the device has a power switch, turn the device's power switch off and on, while watching the LED diagnostics.
- If the DeviceMaster does not have a power switch, disconnect and reconnect the power cord.
- Verify that the network IP address, subnet mask, and gateway is correct and appropriate for the network. If IP addressing is being used, the system should be able to ping the DeviceMaster.
- Verify that the IP address programmed into the DeviceMaster matches the unique reserved IP configured address assigned by the system administrator.
- If using DHCP, the host system needs to provide the subnet mask and gateway.
- Reboot the system and the DeviceMaster.
- If you have a spare DeviceMaster, try replacing the device.

10.2. General Troubleshooting

This table illustrates some general troubleshooting tips.

Note: Make sure that you have reviewed the [10.1. Troubleshooting Checklist](#) on Page 103.

General Condition	Explanation/Action
PWR or Status LED flashing	Indicates that boot program has not downloaded to the unit. 1. Reboot the system. 2. Make sure that you have downloaded the most current firmware for your protocol: http://www.comtrol.com/support/download.asp . Note: If the PWR or Status LED is still flashing, contact Technical Support.
PWR or Status LED not lit	Indicates that power has not been applied or there is a hardware failure. Contact Technical Support.
Cannot ping the device through Ethernet hub	Isolate the DeviceMaster from the network. Connect the device directly to the NIC in the host system (see Page 103).
Cannot ping or connect to the DeviceMaster	The default IP address is often not accessible due to the subnet masking from another network unless 192.168 is used in the network. In most cases, it is necessary to program in an address that conforms to your network.
DeviceMaster continuously reboots when connected to some Ethernet switches or routers	Invalid IP information may also cause the switch or router to check for a gateway address. Lack of a gateway address is a common cause.

10.3. Technical Support

It contains troubleshooting procedures that you should perform before contacting Technical Support since they will request that you perform, some or all of the procedures before they will be able to help you diagnose your problem. If you need technical support, use one of the following methods.

Control Contact Information	
Downloads	ftp://ftp.comtrol.com/html/up_modbus_tcp_main.htm
Web site	http://www.comtrol.com
Phone	763.957.6000

Appendix A. Programming the PLC via Concept

A.1. Overview

After reviewing [Chapter 2. Programming Interface](#) on Page 15, you can use information in the [A.2. Concept Program Screens](#) subsection on Page 106 to help set up your PLC and program the various messages.

A.1.1. What is Concept?

Concept is the PLC programming software package designed to support the Schneider Electric Momentum, Quantum, and Compact PLCs. It does not support any other Schneider Electric PLC or any other manufacturers PLC. However, the example PLC program and installation process may be helpful when working with other PLCs

A.1.2. Requirements

The following requirements must be met to run the example programs.

- The Modbus/TCP firmware must be installed on the DeviceMaster and configured as described in the [DeviceMaster MOD or UP Hardware Installation and Configuration Guide](#).
- The DeviceMaster must be installed on the same Ethernet network segment as the PLC.
- Concept must be installed on your computer.
- The instructions in this guide require that you have some familiarity with this programming application.
- A loopback plug is required for the first port on the DeviceMaster when running an example PLC program. See the [DeviceMaster MOD or UP Hardware Installation and Configuration Guide](#), for more information if you need to build loopback plugs.

The PLC program examples ([A.2.3.1. LPBKCNCP](#) on Page 114 and [A.2.3.2. SCANCNCP](#) on Page 114) are optional. You can copy the PLC program examples from the CD or download the latest program examples from the Internet.

A.1.3. Example Program Considerations (Raw Data)

The example programs are for raw data only.

- While the receive and transmit sequence numbers are cleared on the DeviceMaster at the start of the program, the only requirement is that the sequence numbers be in sync between the PLC and DeviceMaster.
- The DeviceMaster should be reset before starting SCANCNCP example program due to PLC program execution scheduling. If the DeviceMaster is not reset, the sequence numbers may be out of sync. This may result in receiving outdated serial data as well as an unexpected transmission of serial data. A *Transmit Unexpected Sequence Number* error may also occur.

Statistics retrieval is not included in the example programs, but can be easily added by inserting a request statistics message.

A.2. Concept Program Screens

The following screens are intended to aid the PLC programmer in setting up their PLC and programming the various messages.

A.2.1. Processor and Ethernet Setup

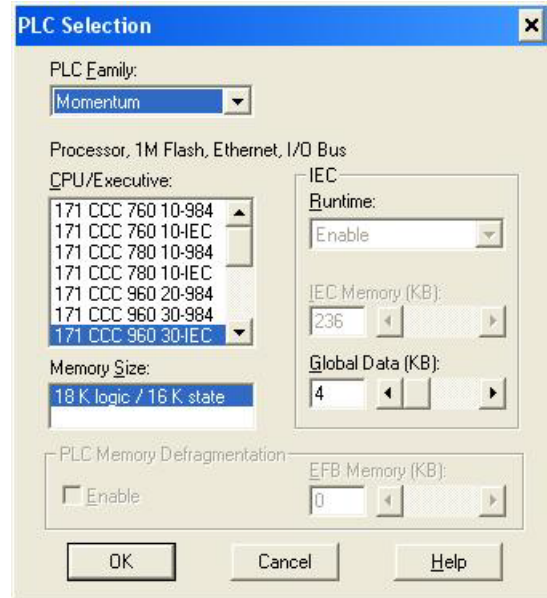
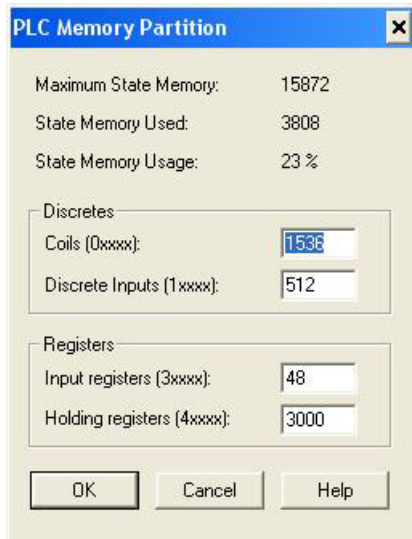
The Processor and Ethernet communications port needs to be set up properly in order for Modbus/TCP to function. It is highly recommended to read and follow your PLC manufacturer’s documentation.

The following documents are recommended for the Concept programming software.

- *Concept User Manual – 840 USE 503 00*
- *Concept IEC Block Library Part: Comm – 840 USE 504 00*

In addition to that information, it is recommended that the following settings be made to allow Modbus/TCP to function properly on a Schneider Electric Momentum, Quantum, or Compact PLC.

1. Verify that the correct processor type has been selected.
2. Verify the proper memory is configured to interface to the DeviceMaster. At least 256 registers must be available.



3. Select the proper extension for Modbus/TCP Ethernet. This is generally 1 for Momentum.



4. Set the Network Configuration. The following is recommended for this screen:
 - Select the **Specify IP Address** option.
 - Set the **Internet Address**.
 - Set the **Subnet Mask**.
 - Set the **Gateway**.
 - Optionally set a diagnostic block.
- In this screen, the I/O Scanner can be configured to directly access the serial port communications on the DeviceMaster. For more information, see [2.3. I/O Scanner \(Raw Data\)](#) on Page 25.

	Slave IP Address	Unit ID	Health Timeout (ms)	Rep Rate (ms)	Link Type	Read Ref Master	Read Ref Slave	Read Length	Last Value (Input)	Write Ref Master	Write Ref Slave	Write Length	Description
1													
2													
3													
4													
5													
6													
7													
8													
9													
10													
11													

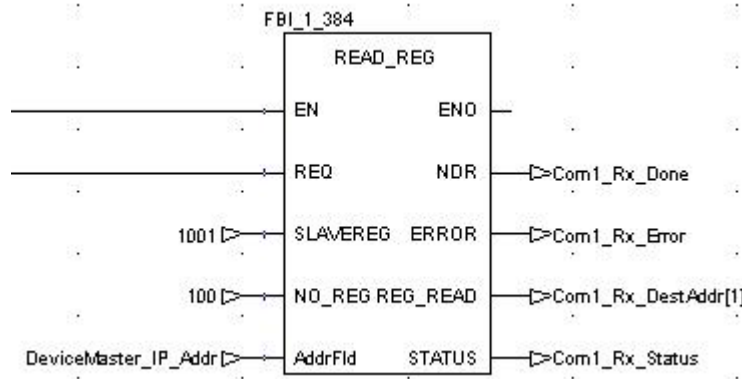
A.2.2. Message Screens

The following message screens are discussed in the upcoming subsections.

- [Read Serial Data via Read Holding Registers Message](#) on Page 108
- [Transmit Serial Data via Write Multiple Registers Message](#) on Page 109
- [Set Receive Sequence Number via Write Multiple Registers Message](#) on Page 110
- [Set Transmit Sequence Number via Write Multiple Registers Message](#) on Page 111
- [Read Serial Port Statistics via Read Holding Registers Message](#) on Page 112
- [Modbus/TCP Slot/Index and DeviceMaster IP Address Definition](#) on Page 113

A.2.2.1. Read Serial Data via Read Holding Registers Message

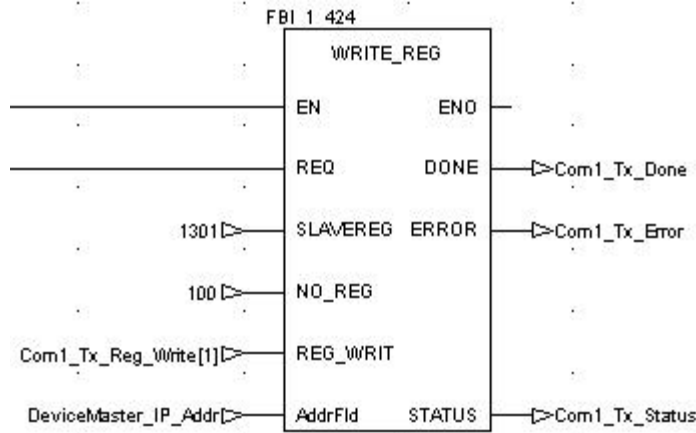
The following screen depicts a Read Holding Registers message used to receive raw serial data in ladder logic.



<i>Where:</i>	
SLAVEREG	Refers to the port receive data address (+1 for use with Concept).
NO_REG	The maximum receive message size in 16 bit words. The maximum size of 100 includes two words for the sequence number and length and then up to 196 bytes of serial data.
NDR	The done flag.
ERROR	The error flag.
REG_READ	The destination address to place the received data.
STATUS	The message status word.
AddrFld	This contains the PLC Modbus/TCP slot/index and IP address of the DeviceMaster.

A.2.2.2. Transmit Serial Data via Write Multiple Registers Message

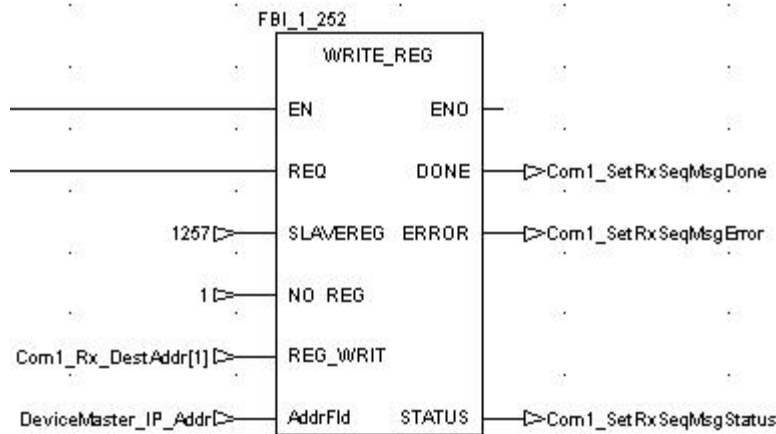
The following screen depicts a *Write Multiple Registers* message used to transmit raw serial data in ladder logic.



Where:	
SLAVEREG	Refers to the port transmit data address (+1 for use with Concept).
NO_REG	The maximum receive message size in 16 bit words. The maximum size of 100 includes two words for the sequence number and length and then up to 196 bytes of serial data.
REG_WRIT	The memory location where the data message to transmit resides on the PLC. (This includes the sequence number, length in bytes, and serial data to transmit.)
DONE	The done flag.
ERROR	The error flag.
STATUS	The message status word.
AddrFld	This contains the PLC Modbus/TCP slot/index and IP address of the DeviceMaster.

A.2.2.3. Set Receive Sequence Number via Write Multiple Registers Message

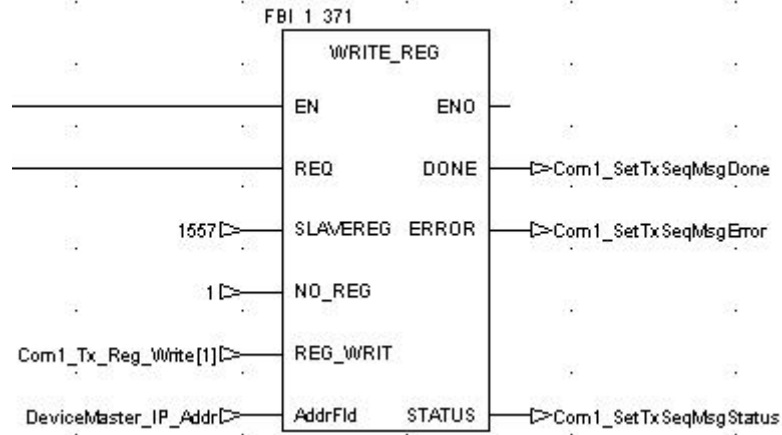
The following screen depicts a *Write Multiple Registers* message used to initialize the receive data sequence number in ladder logic.



Where:	
SLAVEREG	Refers to the port receive sequence number address (+1 for use with Concept).
NO_REG	Set to 1.
REG_WRIT	The memory location where the receive sequence number resides on the PLC.
DONE	The done flag.
ERROR	The error flag.
STATUS	The message status word.
AddrFld	This contains the PLC Modbus/TCP slot/index and IP address of the DeviceMaster.

A.2.2.4. Set Transmit Sequence Number via Write Multiple Registers Message

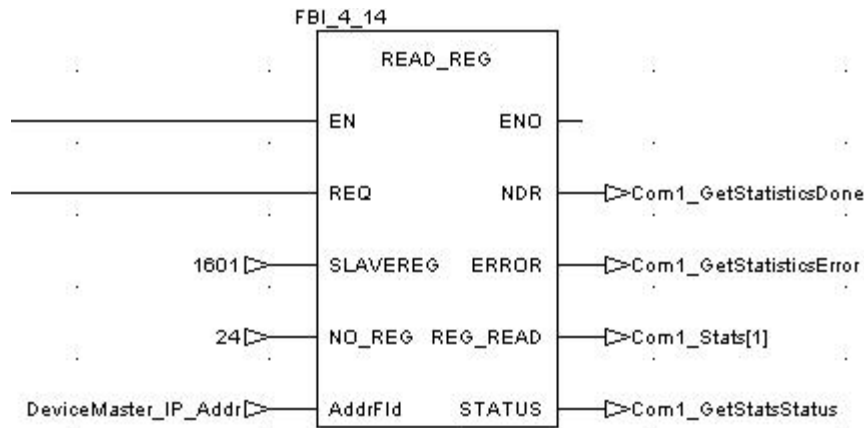
The following screen depicts a *Write Multiple Registers* message used to initialize the transmit data sequence number in ladder logic.



Where:	
SLAVEREG	Refers to the port transmit sequence number address (+1 for use with Concept).
NO_REG	Set to 1.
REG_WRIT	The memory location where the transmit sequence number resides on the PLC.
DONE	The done flag.
ERROR	The error flag.
STATUS	The message status word.
AddrFld	This contains the PLC Modbus/TCP slot/index and IP address of the DeviceMaster.

A.2.2.5. Read Serial Port Statistics via Read Holding Registers Message

The following screen depicts a *Read Holding Registers* message used to retrieve the serial port statistics in ladder logic.

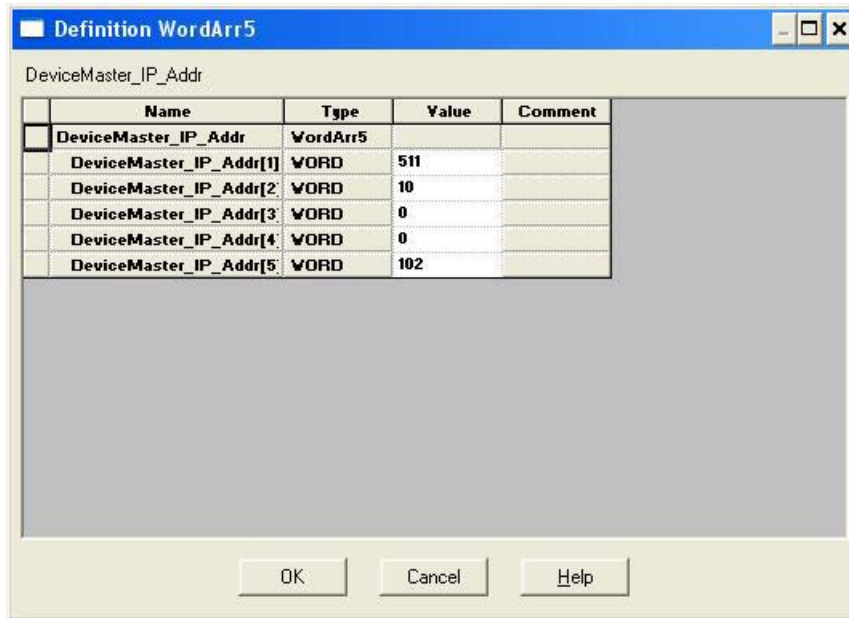


Where:

SLAVEREG	Refers to the serial port statistics address on the DeviceMaster (+1 for use with Concept).
NO_REG	Set to 24 (size of statistics data in words).
NDR	The done flag.
ERROR	The error flag.
REG_READ	The destination address on the PLC to place the statistics data. (Must be at least 24 words in length.)
STATUS	The message status word.
AddrFld	This contains the PLC Modbus/TCP slot/index and IP address of the DeviceMaster.

A.2.2.6. Modbus/TCP Slot/Index and DeviceMaster IP Address Definition

The following screen displays the AddrFld used in all Concept Modbus/TCP messages.



Where:	The first entry (511 or 1FF hex) denotes the following:
Byte 1	<ul style="list-style-type: none"> • LS byte must be 255 (FF hex) to indicate to the DeviceMaster that this is raw/ASCII serial data. • MS byte: Momentum PLC = 1 • Quantum PLC = Slot number of Ethernet card • Compact PLC = Slot number of Ethernet card
Byte 2	MS byte of IP Address.
Byte 3	Second byte of IP Address.
Byte 4	Third byte of IP Address.
Byte 5	LS byte of IP Address.

A.2.3. Concept Example Programs



Disclaimer: Control supplies example PLC programs for demonstration purposes only. They are intended for the sole purpose of an example loop-back demonstration in a controlled lab environment. They are not intended for use in a production environment and may not function correctly on all PLCs. Control does not warrant these example programs or any part thereof. The user assumes all liability for any modification to and use of a modified example program.

The following PLC programs have been included with the released binary. They are designed to interface to a DeviceMaster 1-port or port one of a DeviceMaster 2- or 4-port. Additional programming is required to interface to additional ports on a DeviceMaster 2- or 4-port.

Note: *The following example programs were developed with version 2.6 of Concept and a Schneider Electric Momentum PLC.*

A.2.3.1. LPBKCNCNP

This example program demonstrates a loop-back PLC program using *Read Holding Registers* and *Write Multiple Registers* messages in a standard *polling* type receive method. This program initializes receive and transmit data sequence numbers at startup and then loops raw data via a loop-back plug on the serial port. The data is transmitted and received and the sequence numbers are incremented.

The following files apply:

- LPBKCNCNP.SEC – Ladder logic and variable definitions in Concept programming format.
- LPBKCNCNP.CCF – Configuration file.
- LPBKEXPL.RDF – Reference data template file.

See [Appendix B. LPBKCNCNP Example Program](#) on Page 123 for more information.

A.2.3.2. SCANCNCNP

This example program demonstrates a loop-back PLC program using the I/O Scanner utility on the Concept PLC programming software. This program uses the I/O Scanner to send and receive serial data at a predefined rate. The data is transmitted and received and the sequence numbers are incremented.

The following files apply:

- SCANCNCNP.SEC – Ladder logic and variable definitions in Concept programming format
- SCANCNCNP.CCF – Configuration file
- LPBKEXPL.RDF – Reference data template file.

Note: *The DeviceMaster should be reset before starting a PLC program using the I/O Scanner due to PLC program execution scheduling. If the DeviceMaster is not reset, the sequence numbers may be out of sync. This may result in receiving outdated raw serial data as well as an unexpected transmission of serial data. A Transmit Unexpected Sequence Number error may also occur.*

See [Appendix C. SCANCNCNP Example Program](#) on Page 129 for more information.

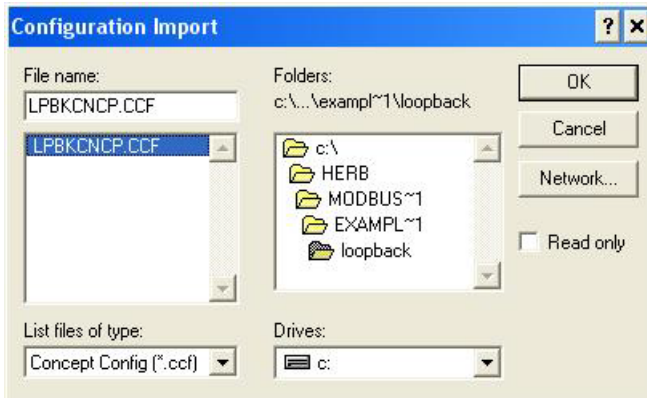
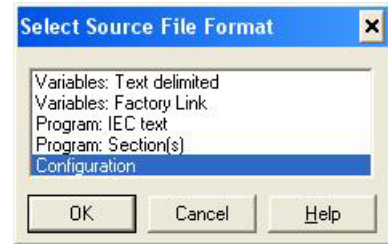
A.2.3.3. Setting up and Running the Concept Example Programs

The following steps are required to set up and run the Concept example programs.

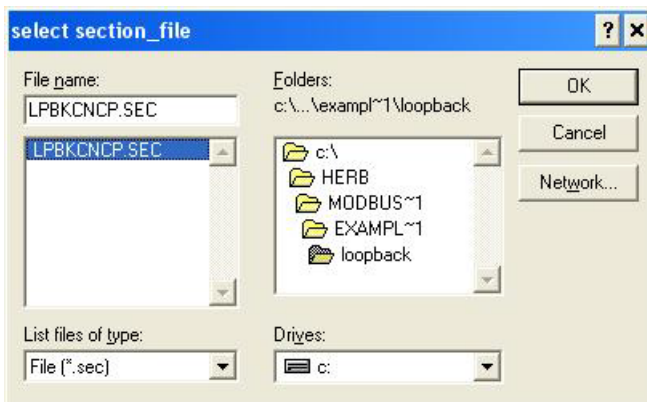
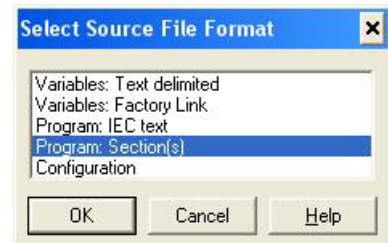
If you have not done so, configure the DeviceMaster by setting the IP address, mask, and gateway for your network and load the Modbus/TCP binary file.

1. Attach the loop-back plug on the serial port.
 - a. Attach the PLC and DeviceMaster to the same Ethernet subnet.
2. Open the *Serial Configuration* web page by entering the DeviceMaster IP address in your web browser to configure the DeviceMaster serial port.
3. Set the following *Serial Port Settings*:
 - **Mode:** RS-232
 - **Baud:** 57600
 - **Parity:** none
 - **Data Bits:** 8
 - **Stop Bits:** 1
 - **Flow Control:** none
 - **DTR:** off
 - **Rx Timeout Between Packets:** 200
4. Set the following *General Protocol Settings*:
 - **Serial Port Protocol:** Raw-Data
 - **Discard Rx Packets With Errors:** Enable
5. Set the following *Raw-Data Settings*:
 - **STX Rx Detect:** one byte, Byte 1 = 2
 - **ETX Rx Detect:** one byte, Byte 1 = 3
 - **STX Tx Append:** one byte, Byte 1 = 2
 - **ETX Tx Append:** one byte, Byte 1 = 3
 - **Strip Rx Stx/ETX:** Enable
 - **Rx MS Byte First:** *Optional*
 - **Tx MS Byte First:** *Optional*
6. Select **Enable** for the **Reset Port** option.
7. Select the **Save** option.
8. Choose either the loop-back (**LPBKCNCP**) or I/O Scanner (**SCANCNCP**) example programs.
9. Load the **.SEC**, **.CCF**, and **.RDF** files into the desired directory.
10. Open Concept.
11. Open a new project by selecting **File->New Project**.
 - a. In the *Create a new project database* pane, enter a file name under *File name*.
 - b. Navigate to the directory where you wish to place the new project.
 - c. Select **OK**.

12. Import the configuration by selecting **File->Import...**
 - a. In the *Select Source File Format* pane, select **Configuration**.
 - b. Select **OK**.
 - c. In the *Configuration Import* pane, select the **.CCF** file.
 - d. Select **OK**.
 - e. Select **OK** in the *Configuration Import completed* pane.

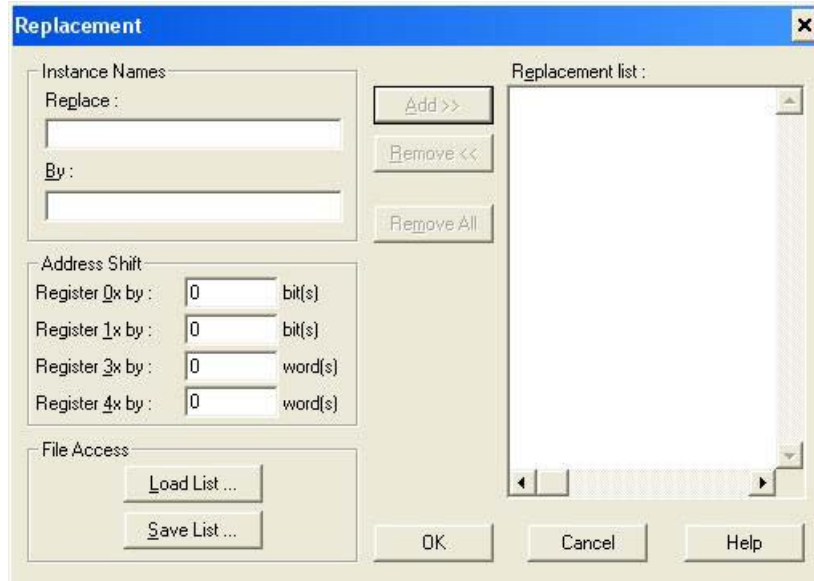


13. Import the program sections by selecting **File->Import...**
 - a. In the *Select Source File Format* pane, select **Program Section(s)**.
 - b. Select **OK**.
 - c. In the *select section_file* pane, select the **.SEC** file.
 - d. Select **OK**.



- e. Select **YES** when asked whether to *save project file first*.

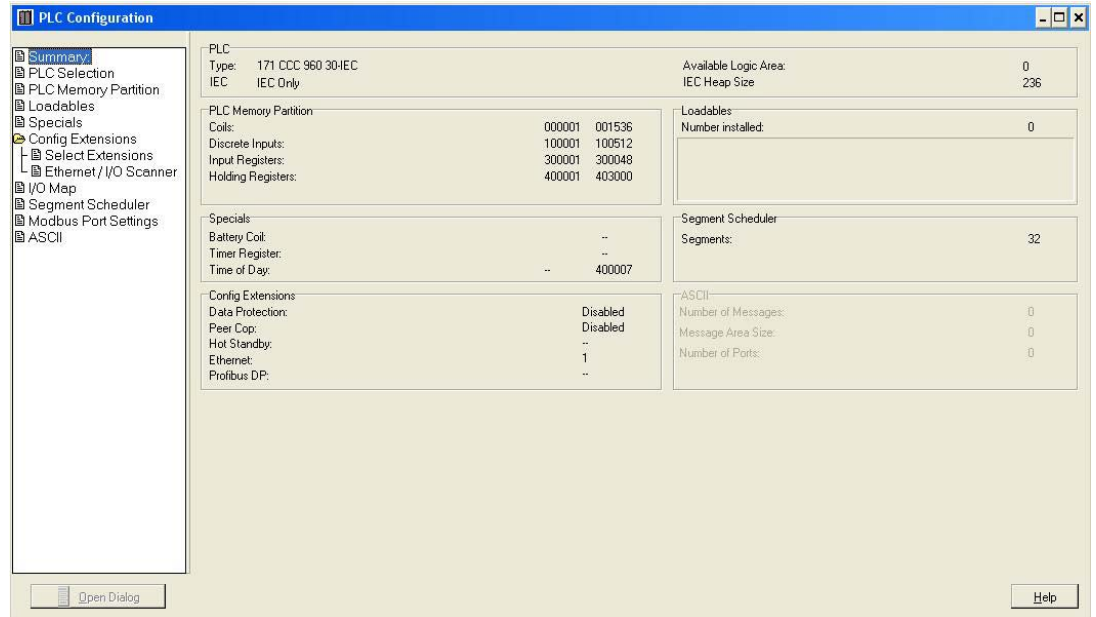
- f. In the *Replacement* pane, select **OK**.



- g. Select **OK** in the status window.

14. Modify the configuration for your PLC.

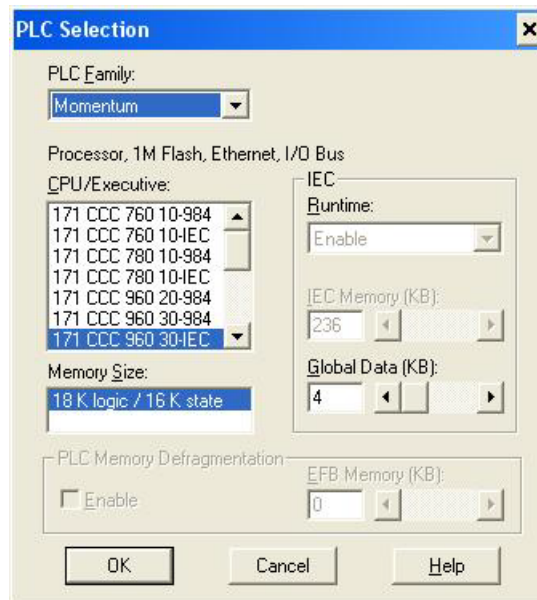
- a. Select **Project->Configurator**.



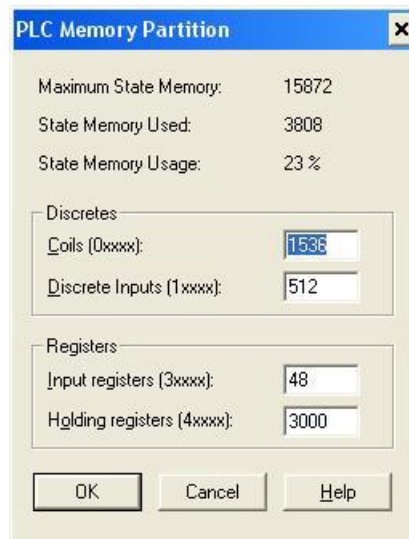
In the *PLC Selection* pane:

- b. Select your **PLC Family**.
 c. Select your **CPU/Executive**.
 d. Select your **Memory Size**.

- e. Select **OK**.

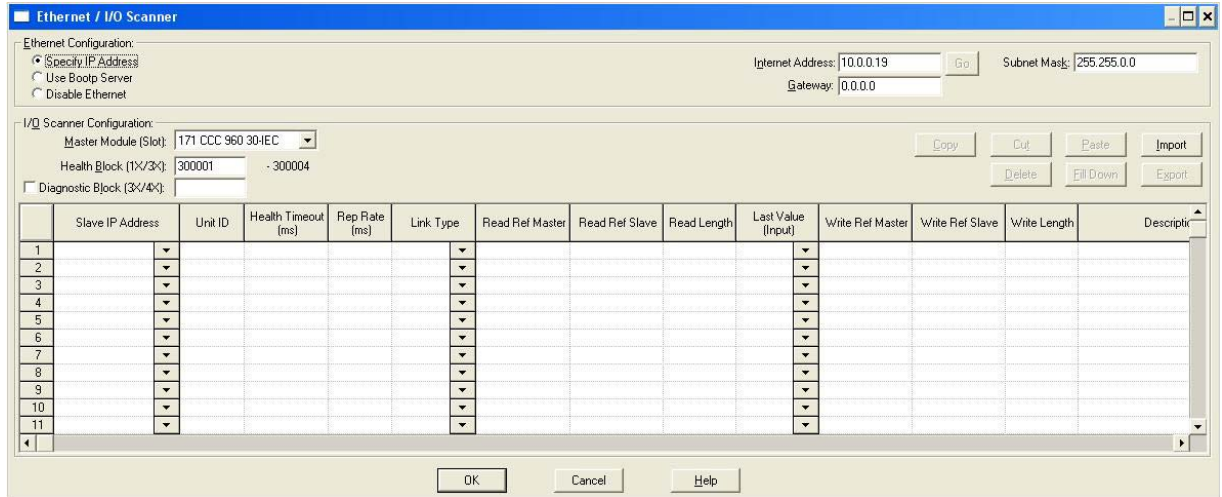


15. In the *PLC Memory Partition* pane:



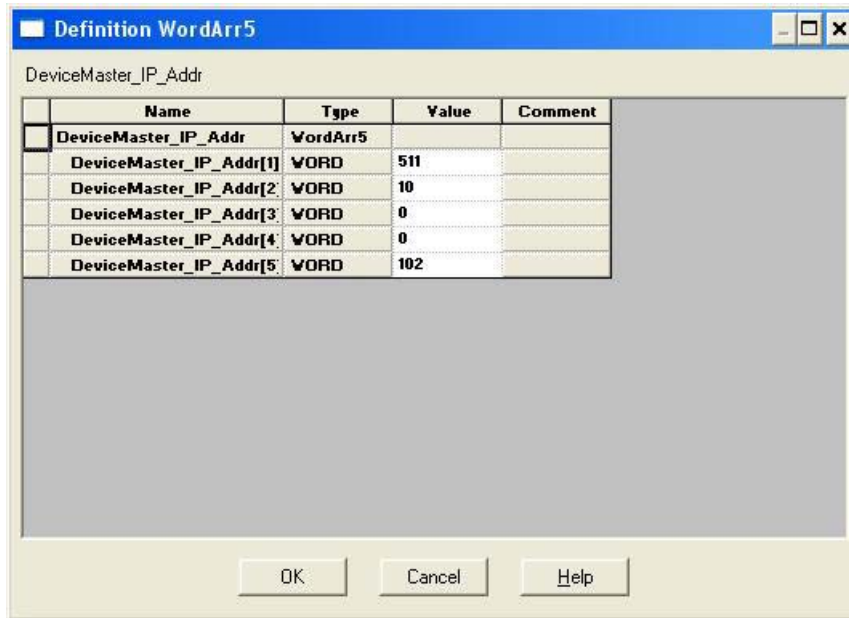
- a. Verify that your PLC has sufficient memory. The example program requires 3000 holding registers.
 - b. Select **OK**.
16. In the *Config Extensions->Ethernet / IO Scanner* pane:
 - a. Select your method of specifying an IP address.
If you are specifying an IP Address:
 - Enter the PLC IP Address under **Internet Address**.
 - Enter the Gateway address.
 - Enter the Subnet Mask.
 - b. Verify the **Master Module** is configured properly.

- c. Select OK.

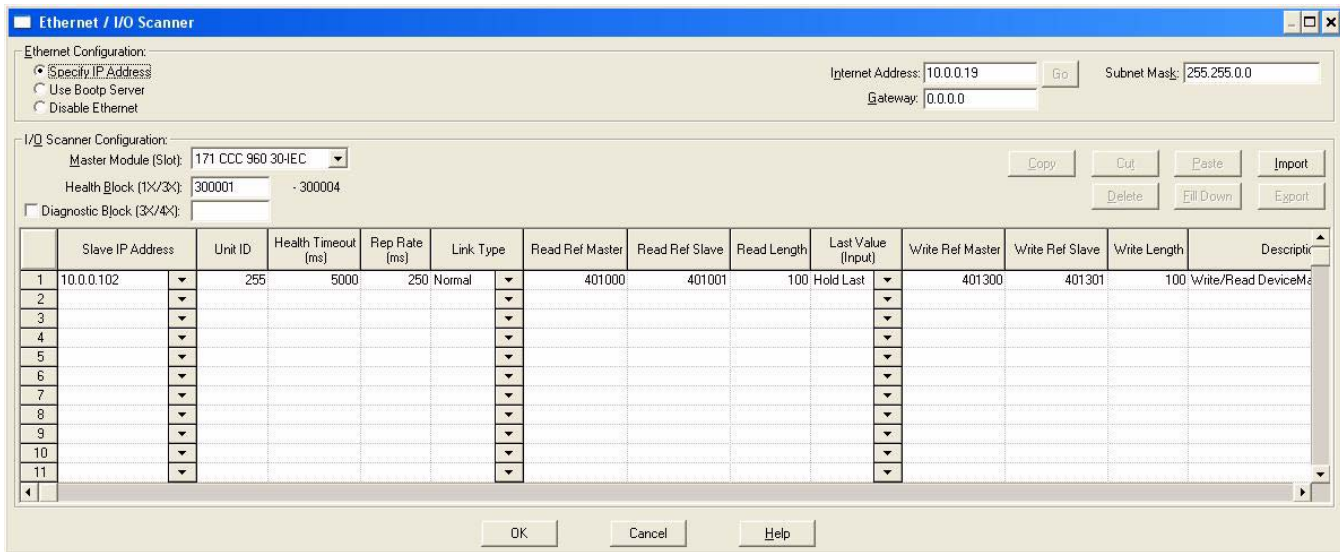


17. Modify the DeviceMaster IP Address in the *DeviceMaster_IP_Addr* array:

- a. Select **Project->Variable Declarations...**
- b. Click the **DeviceMaster_IP_Addr->Set...** variable, which is the slot/device index number.
 For the Schneider Electric Momentum PLC, the first word does not change.
 For the Schneider Electric Quantum or Compact PLC, set the upper eight bits to the slot number of the Ethernet module.
- c. WORDS two through five contain the IP address in the standard 255.255.255.255 mask format.



18. For the SCANCNCP example program only, modify the DeviceMaster IP address in the I/O scanner window. Change the default 10.0.0.102 IP address to that of your DeviceMaster.



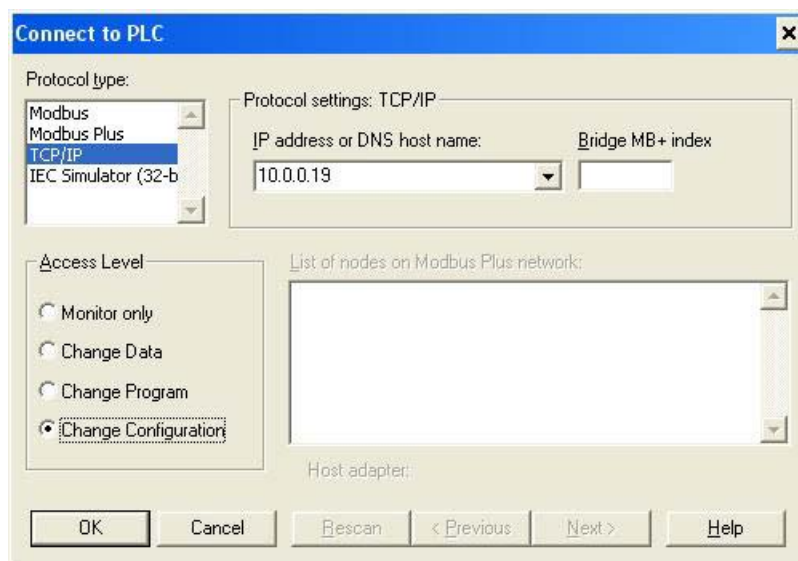
19. Analyze the program by selecting **Project->Analyze Program**.

- a. There should be no errors.
- b. There may be a few *Multi-assignment* warnings, but these can be ignored.

20. Connect to the PLC by selecting **Online->Connect...**

Configure the *Connect to PLC* pane.

- a. Under *Protocol Type*, select **TCP/IP**.
- b. Under *IP Address or DNS host name*, enter the PLC IP address.
- c. Select **Change Configuration**.
- d. Select **OK**.



21. Download the PLC program by selecting **Online->Download...**

In the *Download Controller* pane:

- a. Select **ALL**.
- b. Select **Download...**

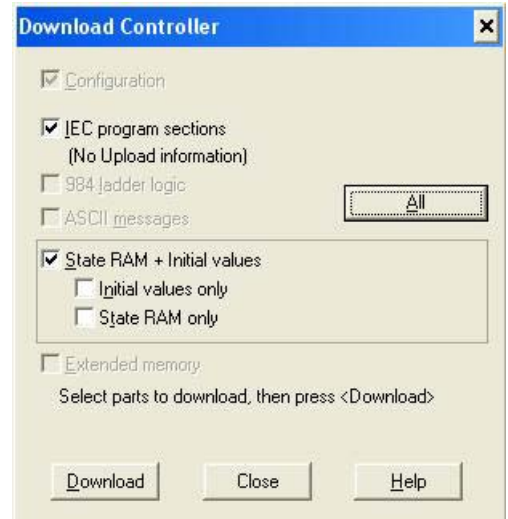
22. Set up the program monitoring:

- a. Select the **Open Reference Data Template** button on the menu.

In the *Open Reference Data Template* pane:

- Locate and select the **LPBKEXPL.RDF** template file supplied with the example files.
- Select **OK**.

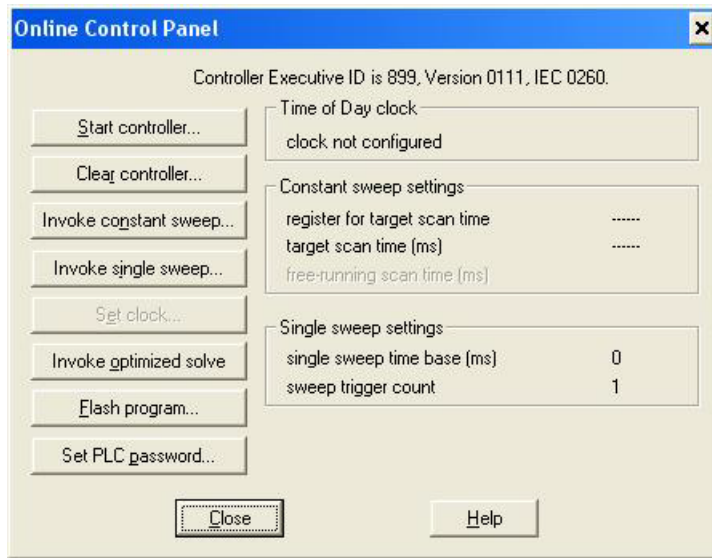
Note: *There may be warnings when loading the reference template file and running the SCANCNCP example program. These warnings can be ignored.*



	Variable Name	Data Type	Address	Value	Set Value	Format	Disable	Cyclic Set	Animation Status
1	Com1_ReadData	BOOL		Off		Bool	<input type="checkbox"/>	<input type="checkbox"/>	
2	Com1_TransmitData	BOOL		Off		Bool	<input type="checkbox"/>	<input type="checkbox"/>	
3	Com1_RxProdSeqNum	UINT		0		Uns Dec	<input type="checkbox"/>	<input type="checkbox"/>	
4	Com1_RxConSeqNum	UINT		0		Uns Dec	<input type="checkbox"/>	<input type="checkbox"/>	
5	Com1_TxProdSeqNum	UDINT		0		Uns Dec	<input type="checkbox"/>	<input type="checkbox"/>	
6	Com1_RxTotalMsgs	UDINT		0		Uns Dec	<input type="checkbox"/>	<input type="checkbox"/>	
7	Com1_Rx_Error_Cnt	DINT		0		Dec [32	<input type="checkbox"/>	<input type="checkbox"/>	
8	Com1_Tx_Error_Cnt	DINT		0		Dec [32	<input type="checkbox"/>	<input type="checkbox"/>	
9	SystemTimerIn	TIME		1s		Time	<input type="checkbox"/>	<input type="checkbox"/>	
10	SystemTimerOut	TIME		0s		Time	<input type="checkbox"/>	<input type="checkbox"/>	
11							<input type="checkbox"/>	<input type="checkbox"/>	
12	Com1_Rx_DestAddr[1]	WORD	401000	0		Hex	<input type="checkbox"/>	<input type="checkbox"/>	
13			401001	0		Dec	<input type="checkbox"/>	<input type="checkbox"/>	
14			401002	0		Dec	<input type="checkbox"/>	<input type="checkbox"/>	
15			401003	0		Dec	<input type="checkbox"/>	<input type="checkbox"/>	
16			401004	0		Dec	<input type="checkbox"/>	<input type="checkbox"/>	
17			401005	0		Dec	<input type="checkbox"/>	<input type="checkbox"/>	
18			401006	0		Dec	<input type="checkbox"/>	<input type="checkbox"/>	
19			401007	0		Dec	<input type="checkbox"/>	<input type="checkbox"/>	
20			401008	0		Dec	<input type="checkbox"/>	<input type="checkbox"/>	
21			401009	0		Dec	<input type="checkbox"/>	<input type="checkbox"/>	
22			401010	0		Dec	<input type="checkbox"/>	<input type="checkbox"/>	
23			401011	0		Dec	<input type="checkbox"/>	<input type="checkbox"/>	
24			401012	0		Dec	<input type="checkbox"/>	<input type="checkbox"/>	
25			401013	0		Dec	<input type="checkbox"/>	<input type="checkbox"/>	
26			401014	0		Dec	<input type="checkbox"/>	<input type="checkbox"/>	
27			401015	0		Dec	<input type="checkbox"/>	<input type="checkbox"/>	

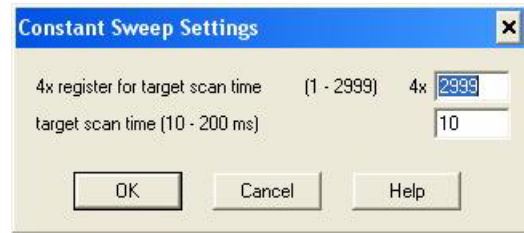
- b. Animate the reference template by clicking once on the template and then selecting **Online->Animate**.
- c. Click once on the **RxTxDataSection** section. Select **Online->Animate Booleans**.

23. Start the processor by selecting **Online->Online Control Panel...**



In the *Online Control Panel*, select **Invoke Constant Sweep...**

- a. Verify the 4x register is 2999. (This is an unused register.)
- b. Verify the sweep rate is 10 msec.
- c. Select **OK**.
- d. Select **Start Controller**.
- e. Select **Close**.



24. Observe the data being transmitted and received.

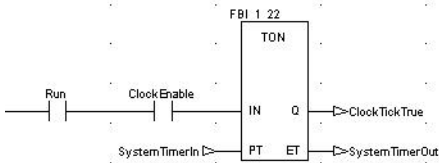
Appendix B. LPBKCNCP Example Program

The following is the ladder logic for the LPBKCNCP example program:

DISCLAIMER

Control supplies example PLC programs for demonstration purposes only. This PLC program is intended for the sole purpose of an example loop-back demonstration in a controlled lab environment. This example PLC program is not intended to be used in a production environment and may not function correctly on all PLCs. Control does not warrant this example program or any part thereof. The user assumes all liability for any modification to and use of a modified example program.

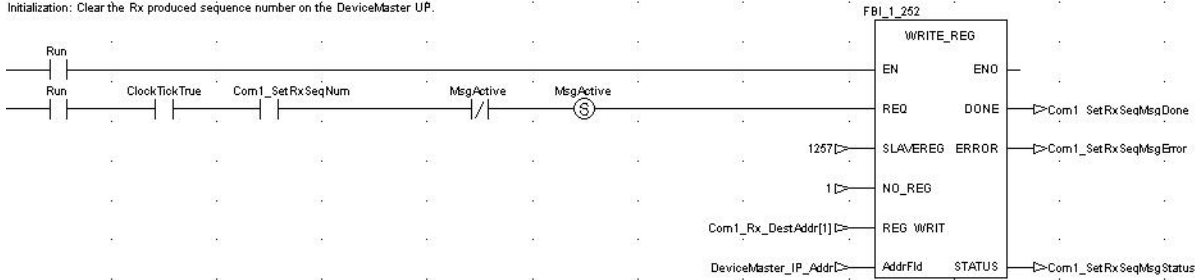
This is the loop-back timer. It controls how fast data is transmitted and received.



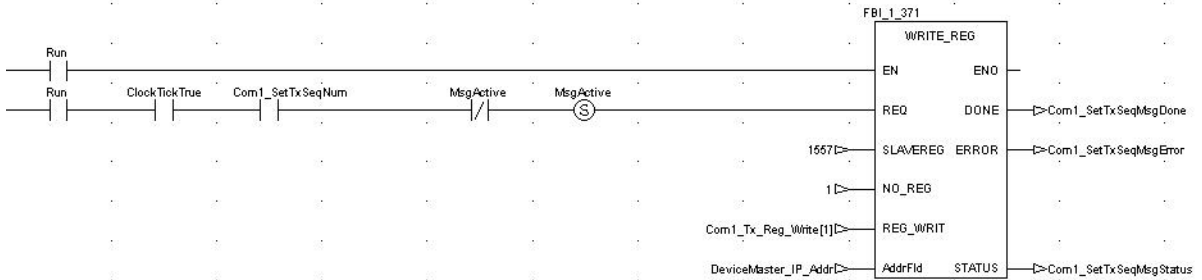
Timer handling: Re-enable clock after resetting to restart.



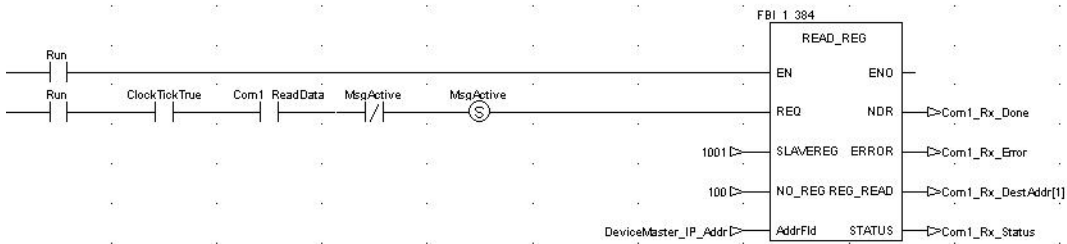
Initialization: Clear the Rx produced sequence number on the DeviceMaster UP.



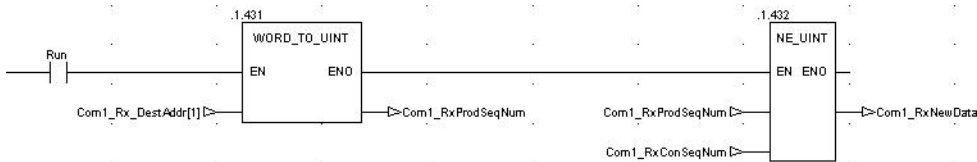
Initialization: Clear the Tx produced sequence number on the DeviceMaster UP.



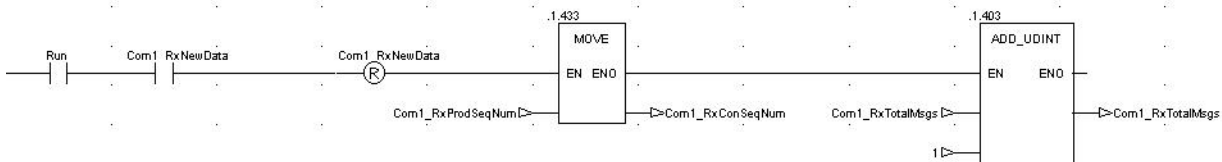
Operational loop: Read the latest data from port 1 on the DeviceMaster UP.



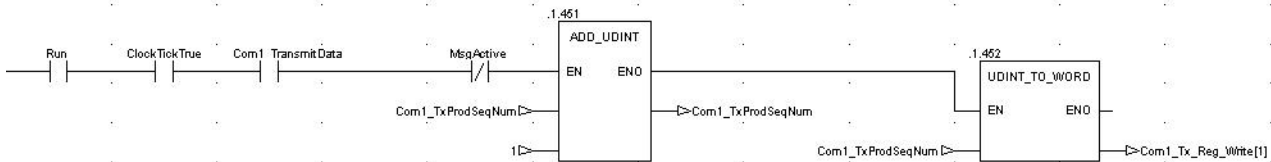
Operational loop: Check for new data received here. If new data is received, the sequence number (first 16 bit word) will have been incremented.

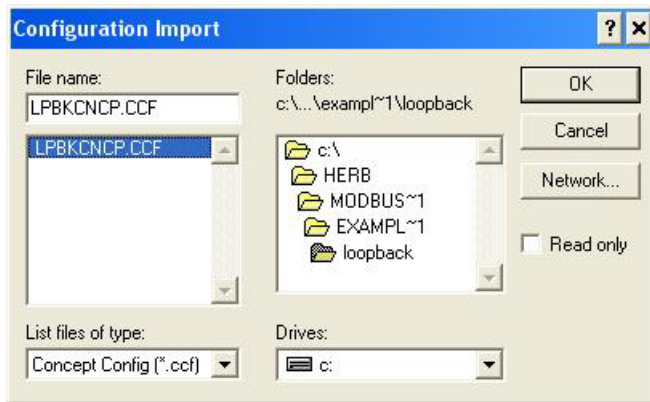


Operational loop: If the new data flag is set, process new data here (Just bumping a total message counter as an example)

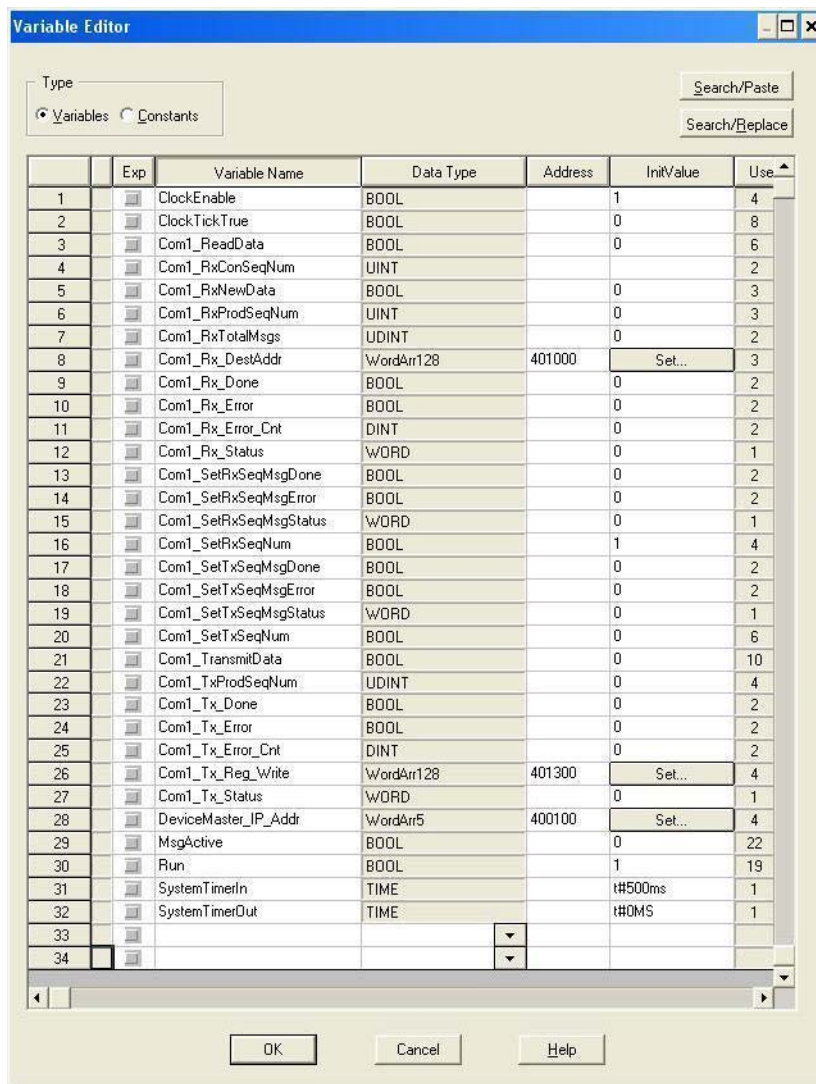


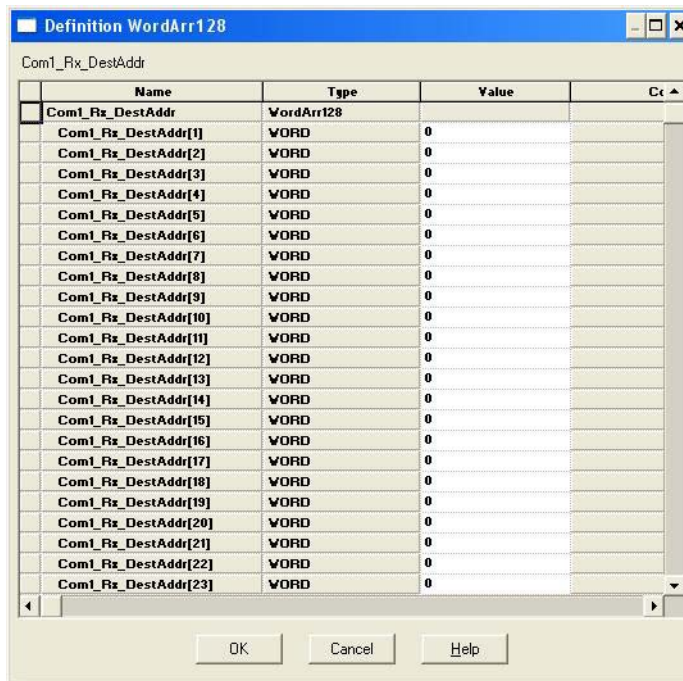
Operational loop: If there is new data to transmit, load the data into the message (starting at the third word), and then increment the Transmit Sequence number.



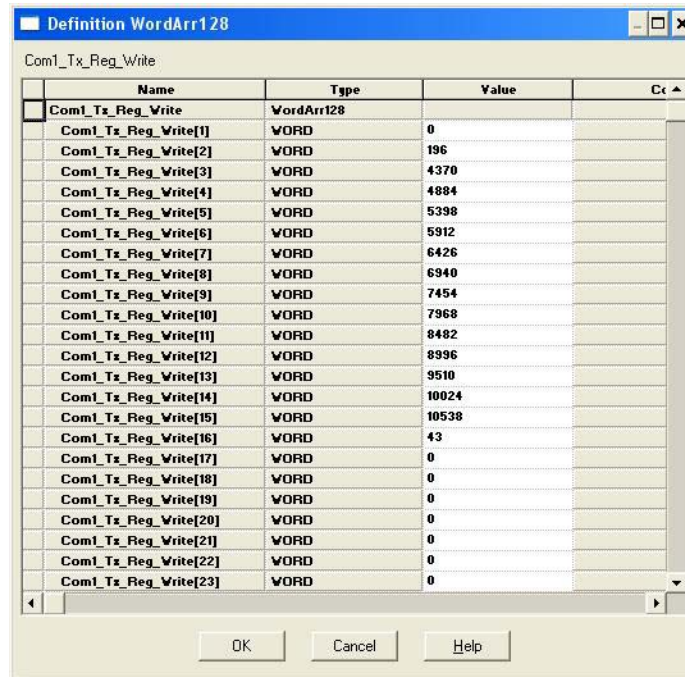


The following screens depict the variables for the LPBKCNCP example program.



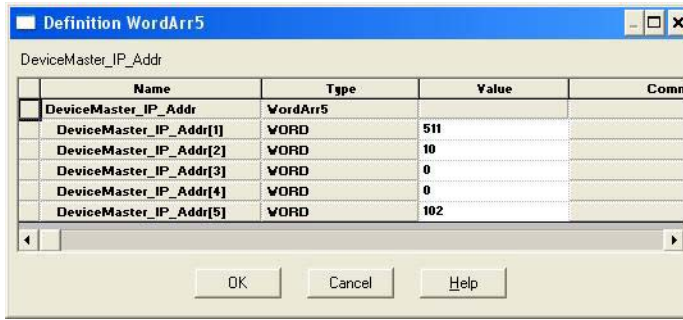


Note: There is no valid data in this variable array until received data is requested from the DeviceMaster.



Where:

- The first word contains the sequence number starting at zero.
- The second word contains the length that is set to the maximum number of bytes transmitted by the Concept software package. (100 words = 200 bytes. 196 bytes maximum transmit data size.)
- Words three to 100 contain transmit data.



Appendix C. SCANCNCP Example Program

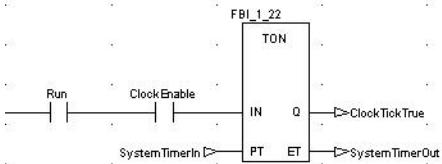
The following is the ladder logic for the SCANCNCP example program.

DISCLAIMER

Control supplies example PLC programs for demonstration purposes only. This PLC program is intended for the sole purpose of an example loop-back demonstration in a controlled lab environment. This example PLC program is not intended to be used in a production environment and may not function correctly on all PLCs. Control does not warrant this example program or any part thereof. The user assumes all liability for any modification to and use of a modified example program.

NOTE: The DeviceMaster UP must be reset before starting this example program or any other PLC program that utilizes the PLC I/O scanner utility to communicate to the DeviceMaster UP.

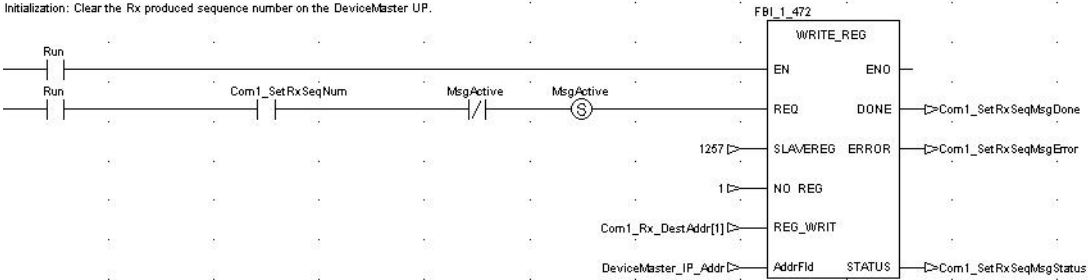
This is the loop-back timer. It controls how fast the transmit data message will be updated so that new serial data is transmitted and received.



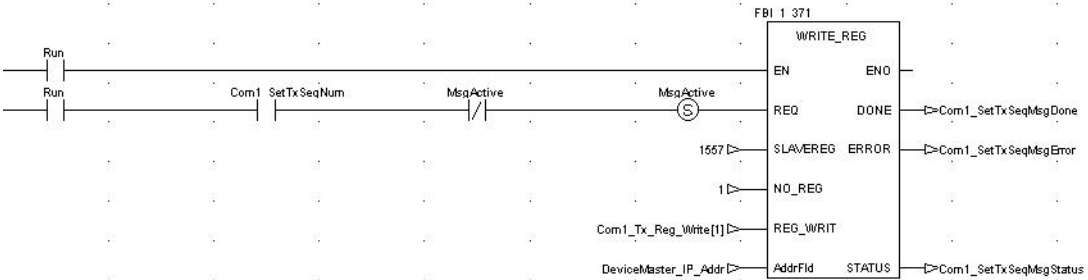
Re-enable clock after resetting to restart.



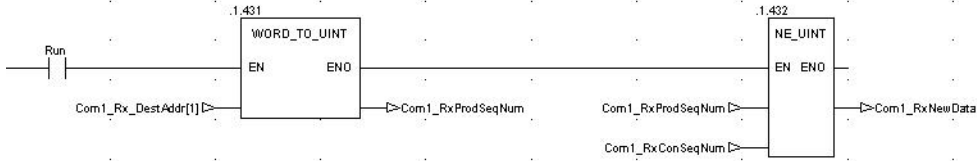
Initialization: Clear the Rx produced sequence number on the DeviceMaster UP.



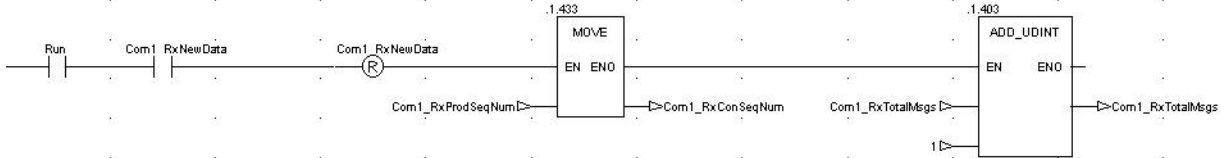
Initialization: Clear the Tx produced sequence number on the DeviceMaster UP.



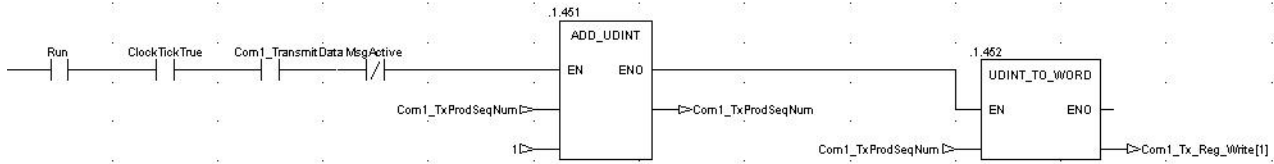
Operational loop: Check for new data received here. If new data is received, the sequence number (first 16 bit word) will have been incremented.



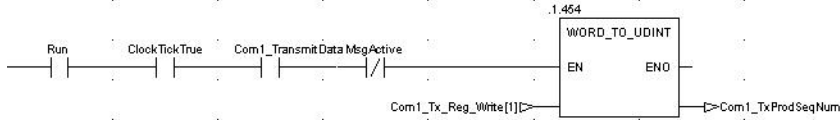
Operational loop: If the new received data flag is set, process new received data here (Just bumping a total message counter as example)



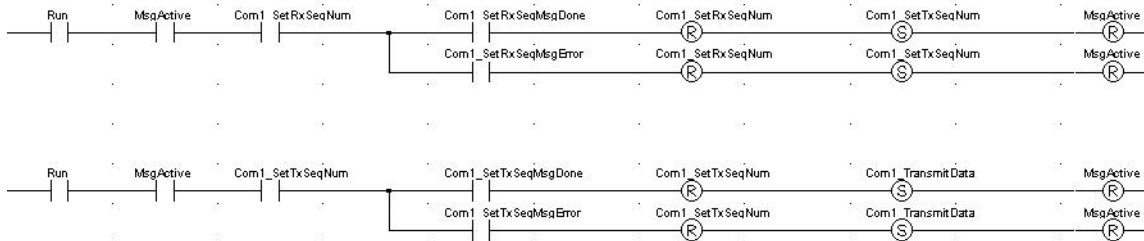
Operational loop: If there is new data to transmit, load the data into the message (starting at the third word), and then increment the Transmit Sequence number.



Operational loop: Handle rollover of 16 bit integer sequence number.



Operational loop: Wait for active messages to complete here. (Initialization messages only for this example program.)



Reset timer.



The I/O Scanner screen displays.

Ethernet Configuration:

Specify IP Address
 Use Bootp Server
 Disable Ethernet

Internet Address: 10.0.0.19 Go Subnet Mask: 255.255.0.0
 Gateway: 0.0.0.0

I/O Scanner Configuration:

Master Module (Slot): 171 CCC 960 304EC Copy Cut Paste Import
 Health Block (1X/3X): 300001 - 300004 Delete Fill Down Export
 Diagnostic Block (3X/4X)

	Slave IP Address	Unit ID	Health Timeout (ms)	Rep Rate (ms)	Link Type	Read Ref Master	Read Ref Slave	Read Length	Last Value (Input)	Write Ref Master	Write Ref Slave	Write Length	Description
1	10.0.0.102	255	5000	250	Normal	401000	401001	100	Hold Last	401300	401301	100	Write/Read DeviceM...
2													
3													
4													
5													
6													
7													
8													
9													
10													
11													

OK Cancel Help

The variable definitions are the same as for the LPBKSCAN program in [Appendix B. LPBKCNCPC Example Program](#) on Page 123.

