




ProLinX[®]
MCM
ProLinX Gateway
Modbus Master/Slave

May 7, 2012

Your Feedback Please

We always want you to feel that you made the right decision to use our products. If you have suggestions, comments, compliments or complaints about our products, documentation, or support, please write or call us.

How to Contact Us

ProSoft Technology

5201 Truxtun Ave., 3rd Floor

Bakersfield, CA 93309

+1 (661) 716-5100

+1 (661) 716-5101 (Fax)

www.prosoft-technology.com

support@prosoft-technology.com

Copyright © 2012 ProSoft Technology, Inc., all rights reserved.

MCM Protocol Manual

May 7 , 2012

ProSoft Technology[®], ProLinx[®], inRAx[®], ProTalk[®], and RadioLinx[®] are Registered Trademarks of ProSoft Technology, Inc. All other brand or product names are or may be trademarks of, and are used to identify products and services of, their respective owners.

ProSoft Technology[®] Product Documentation

In an effort to conserve paper, ProSoft Technology no longer includes printed manuals with our product shipments. User Manuals, Datasheets, Sample Ladder Files, and Configuration Files are provided on the enclosed CD-ROM in Adobe[®] Acrobat Reader file format (.PDFs). These product documentation files may also be freely downloaded from our web site: www.prosoft-technology.com

Important Installation Instructions

Power, Input, and Output (I/O) wiring must be in accordance with Class I, Division 2 wiring methods, Article 501-4 (b) of the National Electrical Code, NFPA 70 for installation in the U.S., or as specified in Section 18-1J2 of the Canadian Electrical Code for installations in Canada, and in accordance with the authority having jurisdiction. The following warnings must be heeded:

WARNING - EXPLOSION HAZARD - SUBSTITUTION OF COMPONENTS MAY IMPAIR SUITABILITY FOR CLASS I, DIV. 2;

WARNING - EXPLOSION HAZARD - WHEN IN HAZARDOUS LOCATIONS, TURN OFF POWER BEFORE REPLACING OR WIRING MODULES

WARNING - EXPLOSION HAZARD - DO NOT DISCONNECT EQUIPMENT UNLESS POWER HAS BEEN SWITCHED OFF OR THE AREA IS KNOWN TO BE NON-HAZARDOUS. THIS DEVICE SHALL BE POWERED BY CLASS 2 OUTPUTS ONLY.

ProLinx® Products Warnings

WARNING – EXPLOSION HAZARD – DO NOT DISCONNECT EQUIPMENT UNLESS POWER HAS BEEN SWITCHED OFF OR THE AREA IS KNOWN TO BE NON-HAZARDOUS.

AVERTISSEMENT – RISQUE D'EXPLOSION – AVANT DE DÉCONNECTER L'EQUIPMENT, COUPER LE COURANT OU S'ASSURER QUE L'EMPLACEMENT EST DÉSIGNÉ NON DANGEREUX.

ProLinx Gateways with Ethernet Ports

Series C ProLinx™ Gateways with Ethernet ports do **NOT** include the HTML Web Server. The HTML Web Server must be ordered as an option. This option requires a factory-installed hardware addition. The HTML Web Server now supports:

- 8 MB file storage for HTML files and associated graphics files (previously limited to 384K)
- 32K maximum HTML page size (previously limited to 16K)

To upgrade a previously purchased Series C model

Contact your ProSoft Technology distributor to order the upgrade and obtain a Returned Merchandise Authorization (RMA) to return the unit to ProSoft Technology.

To order a ProLinx Plus gateway with the -WEB option

Add **-WEB** to the standard ProLinx part number. For example, **5201-MNET-MCM-WEB**.

Markings

Label Markings

<cULus>

E183151

Class I Div 2

Groups A,B,C,D T6

-30 °C ≤ Ta ≤ 60 °C

<Ex>

II 3 G

Ex nA IIC T4

-20 °C ≤ Ta ≤ 50 °C

II – Equipment intended for above ground use (not for use in mines).

3 – Category 3 equipment, investigated for normal operation only.

G – Equipment protected against explosive gasses.

Contents

Your Feedback Please.....	2
How to Contact Us.....	2
ProSoft Technology® Product Documentation.....	2
Important Installation Instructions.....	3
ProLinx® Products Warnings.....	3
ProLinx Gateways with Ethernet Ports.....	3
To upgrade a previously purchased Series C model.....	3
To order a ProLinx Plus gateway with the -WEB option.....	3
Markings.....	4
1 Start Here	9
<hr/>	
1.1 System Requirements.....	10
1.2 Package Contents.....	11
1.3 Mounting the Gateway on a DIN-rail.....	12
1.4 Connecting Power to the Unit.....	13
1.5 Installing ProSoft Configuration Builder Software.....	14
1.6 Using the Online Help.....	15
2 Functional Overview	17
<hr/>	
2.1 Modbus Master/Slave Port(s).....	18
2.2 Serial Port Specifications.....	19
2.3 Modbus Master/Slave Port Specifications.....	20
2.4 Module Internal Database.....	21
2.4.1 Modbus Port Access to Database.....	22
3 MCM Protocol Configuration	23
<hr/>	
3.1 [Modbus Port x].....	25
3.1.1 Enabled.....	25
3.1.2 RS Interface.....	25
3.1.3 Type.....	25
3.1.4 Float Flag.....	25
3.1.5 Float Start.....	26
3.1.6 Float Offset.....	26
3.1.7 Protocol.....	26
3.1.8 Baud Rate.....	27
3.1.9 Parity.....	27
3.1.10 Data Bits.....	27
3.1.11 Stop Bits.....	27
3.1.12 RTS On.....	28
3.1.13 RTS Off.....	28
3.1.14 Minimum Response Delay.....	28
3.1.15 Use CTS Line.....	28
3.1.16 Response Timeout.....	28
3.1.17 Retry Count.....	28
3.1.18 Internal Slave ID.....	29
3.1.19 Bit Input Offset.....	29

3.1.20	Word Input Offset.....	29
3.1.21	Output Offset.....	29
3.1.22	Holding Register Offset.....	30
3.1.23	Use Guard Band Timer.....	31
3.1.24	Guard Band Timeout	32
3.1.25	Minimum Command Delay	32
3.1.26	Error Delay Counter.....	33
3.1.27	Command Control Reg	33
3.2	[Modbus Port X Commands].....	34
3.2.1	Enable.....	35
3.2.2	Internal Address.....	35
3.2.3	Poll Interval	35
3.2.4	Reg Count.....	36
3.2.5	Swap Code	36
3.2.6	Slave Address.....	37
3.2.7	Modbus Function	37
3.2.8	MB Address in Device	38
4	Serial Port Modbus Error and Status Data	39
4.1	Viewing Error and Status Data	40
4.2	Modbus Error and Status Data Area Addresses	41
4.2.1	Modbus Ports: Error and Status	41
4.2.2	Slave Port: General Port STATE and COMM STATE Status.....	42
4.2.3	Master Port: Command List Errors	43
4.2.4	Master Port: Modbus Slave List Status	45
4.3	Error Codes.....	47
4.3.1	Modbus Error Codes.....	47
5	Module Communication Error Codes	48
5.1.1	Command List Error Codes	48
5.1.2	Modbus Configuration Error Word.....	49
6	Communication Port Cables	51
6.1	Port 0, 1, 2, 3: RS-232 - Null Modem (DTE with Hardware Handshaking)	52
6.2	Port 0, 1, 2, 3: RS-232 - Null Modem (DTE without Hardware Handshaking)	53
6.3	Port 0, 1, 2, 3: RS-232 - DTE to DCE Modem Connection	54
6.4	Port 0, 1, 2, 3: RS-422 Interface Connections.....	55
6.5	Port 0, 1, 2, 3: RS-485 Interface Connections.....	56
7	LED Indicators	57
7.1	Base Module LEDs	58
7.2	LEDs for Serial Ports	59
8	Modbus Port Expander (4102/5102-MBS3-MBM)	61
8.1	[Modbus Master Port]	62
8.1.1	Internal Slave ID	62
8.1.2	Protocol.....	62

8.1.3	Baud Rate	63
8.1.4	Parity	63
8.1.5	Data Bits	63
8.1.6	Stop Bits	64
8.1.7	RTS On.....	64
8.1.8	RTS Off.....	64
8.1.9	Minimum Response Delay	64
8.1.10	Use CTS Line	64
8.1.11	Response Timeout	64
8.1.12	Retry Count	65
8.2	[Modbus Slave Port x]	66
8.2.1	Enabled	66
8.2.2	RS Interface	66
8.2.3	Protocol	66
8.2.4	Baud Rate	67
8.2.5	Parity	68
8.2.6	Data Bits	68
8.2.7	Stop Bits	68
8.2.8	RTS On.....	68
8.2.9	RTS Off.....	68
8.2.10	Minimum Response Delay	68
8.2.11	Use CTS Line	69
8.2.12	Use Guard Band Timer	69
8.2.13	Guard Band Timeout	70
8.2.14	Request Timeout	70
8.2.15	Busy Timeout	70
8.2.16	Busy Timeout	71
9	Reference	73
9.1	Modbus Protocol Specification	74
9.1.1	Read Coil Status (Function Code 01)	74
9.1.2	Read Input Status (Function Code 02).....	75
9.1.3	Read Holding Registers (Function Code 03)	76
9.1.4	Read Input Registers (Function Code 04).....	77
9.1.5	Force Single Coil (Function Code 05)	78
9.1.6	Preset Single Register (Function Code 06).....	79
9.1.7	Diagnostics (Function Code 08).....	80
9.1.8	Force Multiple Coils (Function Code 15).....	82
9.1.9	Preset Multiple Registers (Function Code 16)	83
9.1.10	Modbus Exception Responses.....	84
9.2	Moving Data using the Data Map.....	87
10	Support, Service & Warranty	89
	Contacting Technical Support.....	89
10.1	Return Material Authorization (RMA) Policies and Conditions.....	Error! Bookmark not defined.
10.1.1	Returning Any Product	Error! Bookmark not defined.
10.1.2	Returning Units Under Warranty	Error! Bookmark not defined.
10.1.3	Returning Units Out of Warranty	Error! Bookmark not defined.
10.2	LIMITED WARRANTY.....	Error! Bookmark not defined.
10.2.1	What Is Covered By This Warranty	Error! Bookmark not defined.
10.2.2	What Is Not Covered By This Warranty	Error! Bookmark not defined.

10.2.3	Disclaimer Regarding High Risk Activities.....	Error! Bookmark not defined.
10.2.4	Intellectual Property Indemnity	Error! Bookmark not defined.
10.2.5	Disclaimer of all Other Warranties	Error! Bookmark not defined.
10.2.6	Limitation of Remedies **	Error! Bookmark not defined.
10.2.7	Time Limit for Bringing Suit.....	Error! Bookmark not defined.
10.2.8	No Other Warranties.....	Error! Bookmark not defined.
10.2.9	Allocation of Risks	Error! Bookmark not defined.
10.2.10	Controlling Law and Severability	Error! Bookmark not defined.

Index

91

1 Start Here

In This Chapter

❖ System Requirements	10
❖ Package Contents	11
❖ Mounting the Gateway on a DIN-rail	12
❖ Connecting Power to the Unit.....	13
❖ Installing ProSoft Configuration Builder Software.....	14
❖ Using the Online Help	15

1.1 System Requirements

The configuration software for the MCM gateway requires the following minimum hardware and software components:

- Pentium® II 450 MHz minimum. Pentium III 733 MHz (or better) recommended
- 128 Mbytes of RAM minimum, 256 Mbytes of RAM recommended
- 100 Mbytes of free hard disk space (or more based on application requirements)
- 256-color VGA graphics adapter, 800 x 600 minimum resolution (True Color 1024 × 768 recommended)
- CD-ROM drive

Supported operating systems:

- Microsoft Windows 7(32 bit) (64bit not tested)
- Microsoft Windows Vista (not tested)
- Microsoft Windows XP Professional with Service Pack 1 or 2
- Microsoft Windows 2000 Professional with Service Pack 1, 2, or 3 (not tested)
- Microsoft Windows Server 2003 (not tested)

1.2 Package Contents

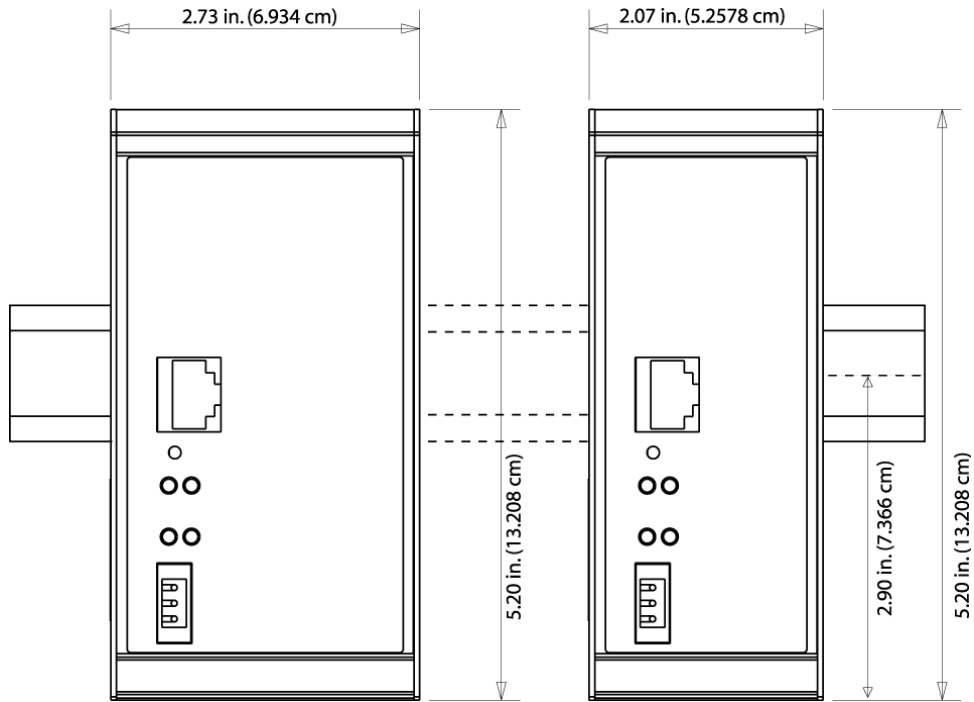
The following components are included with your MCM gateway, and are all required for installation and configuration.

Important: Before beginning the installation, please verify that all of the following items are present.

Qty.	Part Name	Part Number	Part Description
1	MCM gateway	PLX-####	ProLinx communication gateway
1	Cable	Cable #15, RS232 Null Modem	For RS232 Connection from a PC to the CFG Port of the gateway
Varies	Cable	Cable #9, Mini-DIN8 to DB9 Male Adapter	For DB9 Connection to gateway's Port. One DIN to DB-9M cable included per configurable serial port, plus one for gateway configuration
Varies	Adapter	1454-9F	Adapters, DB9 Female to Screw Terminal. For RS422 or RS485 Connections to each serial application port of the gateway
1	ProSoft Solutions CD		Contains sample programs, utilities and documentation for the MCM gateway.

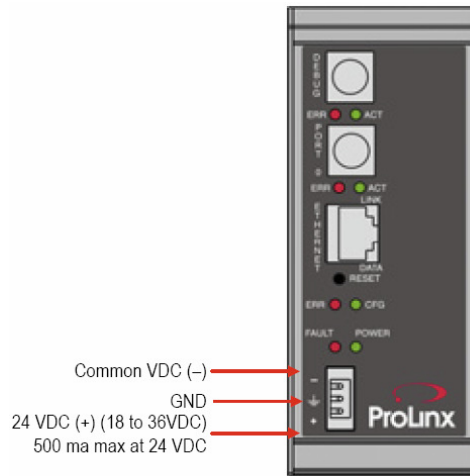
If any of these components are missing, please contact ProSoft Technology Support for replacements.

1.3 Mounting the Gateway on a DIN-rail



ProLinx 5000/6000 Series gateway

1.4 Connecting Power to the Unit



WARNING: Ensure that you do not reverse polarity when applying power to the gateway. This will cause damage to the gateway's power supply.

1.5 Installing ProSoft Configuration Builder Software

You must install the *ProSoft Configuration Builder (PCB)* software to configure the gateway. You can always get the newest version of *ProSoft Configuration Builder* from the ProSoft Technology website.

To install ProSoft Configuration Builder from the ProSoft Technology website

- 1 Open your web browser and navigate to <http://www.prosoft-technology.com/pcb>
- 2 Click the link at the *Current Release Version* section to download the latest version of *ProSoft Configuration Builder*.
- 3 Choose **SAVE** or **SAVE FILE** when prompted.
- 4 Save the file to your *Windows Desktop*, so that you can find it easily when you have finished downloading.
- 5 When the download is complete, locate and open the file, and then follow the instructions on your screen to install the program.

If you do not have access to the Internet, you can install *ProSoft Configuration Builder* from the *ProSoft Solutions Product CD-ROM*, included in the package with your gateway.

To install ProSoft Configuration Builder from the Product CD-ROM

- 1 Insert the *ProSoft Solutions Product CD-ROM* into the CD-ROM drive of your PC. Wait for the startup screen to appear.
- 2 On the startup screen, click **PRODUCT DOCUMENTATION**. This action opens a *Windows Explorer* file tree window.
- 3 Click to open the **UTILITIES** folder. This folder contains all of the applications and files you will need to set up and configure your gateway.
- 4 Double-click the **SETUP CONFIGURATION TOOL** folder, double-click the **PCB_*.EXE** file and follow the instructions on your screen to install the software on your PC. The information represented by the "*" character in the file name is the *PCB* version number and, therefore, subject to change as new versions of *PCB* are released.

Note: Many of the configuration and maintenance procedures use files and other utilities on the CD-ROM. You may wish to copy the files from the *Utilities* folder on the CD-ROM to a convenient location on your hard drive.

1.6 Using the Online Help

Most of the information needed to help you use is provided in a Help System that is always available whenever you are running . The Help System does not require an Internet connection.

To view the help pages, start , open the **HELP** menu, and then choose **CONTENTS**.

2 Functional Overview

In This Chapter

❖ Modbus Master/Slave Port(s)	18
❖ Serial Port Specifications.....	19
❖ Modbus Master/Slave Port Specifications	20
❖ Module Internal Database	21

The Modbus Master/Slave protocol driver may exist in a single port (MCM) or a multiple port (MCM4) implementation. In either case, each port can be independently configured as a Master or Slave port to interface, as required, with other Modbus devices.

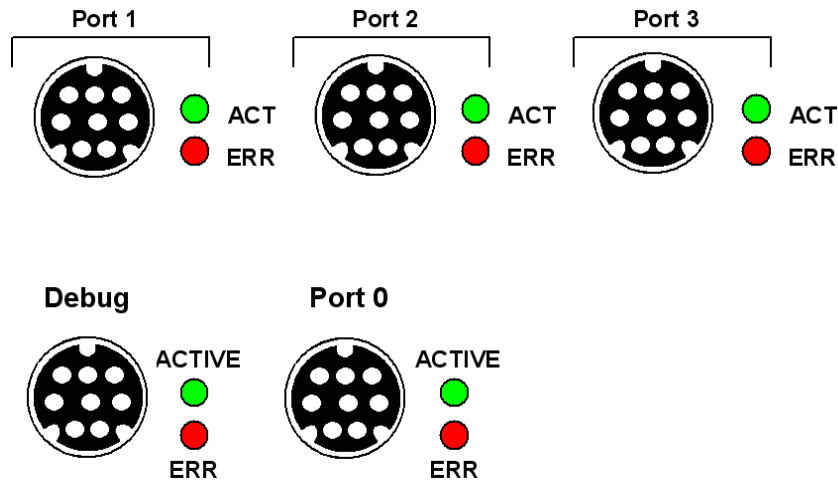
2.1 Modbus Master/Slave Port(s)

The Modbus Master ports can be used to continuously interface with Modbus slave devices over a serial communication interface (RS-232, RS-422 or RS-485). Each Master port may have up to 100 user defined commands which determine the Modbus read/write messages issued to each slave attached to the port.

In addition, the gateway can be configured to poll slave devices that are not responding to commands at a lower priority (slow-poll mode). If a slave device has failed to respond to a command after a user-defined number of retries, the driver will recognize the slave is experiencing a communication failure. In this case, if you have set a non-zero value in the Error Delay Count parameter in the Master port configuration, then, as the driver comes to a command on the Master Command List that should be sent to the failing slave, instead of sending the command, the driver will skip the command and decrement the error delay counter value by one.

When the number of commands have been skipped is equal to the value set in the Error Delay Count parameter, the error delay counter will have been decremented to zero and the driver will send next command for that node in the command list. If the slave responds properly to this command, the driver will resume normal polling of this slave. If the new command fails, the driver will continue to recognize the slave is experiencing communication problems and repeat the process of skipping commands and decrementing the error delay counter before trying to send another command to that slave. This has the effect of putting the slave in a low-priority, slow-poll mode until it begins to communicate normally again.

2.2 Serial Port Specifications



Type	Specifications
Serial Ports	
Serial Port Adapter Cables	One Mini DIN to DB-9M adapter cable included for each configurable serial port
Config Port Connector/ Pinout	DB-9F connector / DTE pinout
Serial Port Isolation	2500V RMS port-to-port isolation per UL 1577. 3000V DC min. port to ground and port to logic power isolation.
Serial Port Protection	RS-485/422 port interface lines TVS diode protected at +/- 27V standoff voltage. RS-232 port interface lines fault protected to +/- 36V power on, +/- 40V power off.

Note: On all ProLinX gateways, data from the application port on the main board, serial Port 0, is not buffered. Packets go directly to and from the serial chipset to the processor. This has the potential to cause the serial communications to become erratic at baud rates above 38,400 baud.

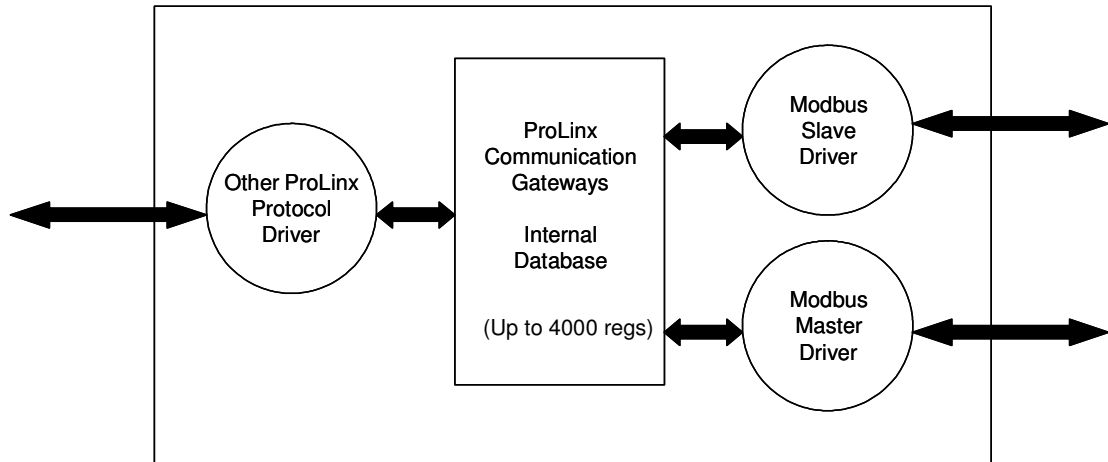
ProLinX gateways with 4 serial ports have a separate serial interface board for serial Ports 1, 2, and 3. These serial ports are buffered and can handle communications up to 115,200 baud.

2.3 Modbus Master/Slave Port Specifications

Type	Specifications
General Parameters	
Internal Database	Up to 4000 registers (words) available.
Communication parameters	Port 0: Baud Rate: 110 to 115K baud possible 110 to 38.4K baud recommended for all applications. Higher baud rates are recommended only for use with very slow Modbus polling rates (e.g. 1 command or less per second) Port 1, 2, 3: Baud Rate: 110 to 115K baud Stop Bits: 1 or 2 Data Size: 5 to 8 bits Parity: None, Even, Odd, Mark, Space RTS Timing delays: 0 to 65535 milliseconds
Modbus Modes	RTU mode (binary) with CRC-16 ASCII mode with LRC error checking
Floating Point Data	Floating point data movement supported, including configurable support for Enron implementation
Modbus Function Codes	1: Read Coil Status 2: Read Input Status 3: Read Holding Registers 4: Read Input Registers 5: Force (Write) Single Coil 6: Preset (Write) Single Register 15: Force(Write) Multiple Coils 16: Force (Write) Multiple Register
Modbus Master	
Command List	Up to 100 command per Master port, each fully configurable for function, slave address, register to/from addressing and word/bit count
Status Data	Error codes available on an individual command basis. In addition, a slave status list is maintained per active Modbus Master port.
Polling of command list	Configurable polling of command list, including continuous and on change of data
Modbus Slave	
Node address	1 to 247 (software selectable)
Status Data	Error codes, counters and port status available per configured slave port starting at memory register 4400.

2.4 Module Internal Database

The internal gateway database is central to the functionality of the gateway. This database is shared between all the ports on the gateway. This permits data from devices on one communication network port to be viewed and controlled by devices on another port or network.



The Master driver uses the database in two ways:

- A read command issued to a slave device by the Master driver will return the slave data into the internal database.
- A write command issued to a slave device by the Master driver uses the data in the internal database as the source data to send to the slave.

The slave driver allows external Master devices to access the database, reading data from the database or writing data to the database. External Modbus Master devices can monitor and control data in this database through any port configured as a slave.

Status and error data from the slave and Master ports generated by the gateway can be mapped into the internal database, where it can be retrieved by either protocol driver.

2.4.1 Modbus Port Access to Database

The Modbus slave support in the gateway permits remote Master applications (that is, HMI software, Quantum processors, and so on) to read from and write to the gateway's database.

When configured as a slave, the internal database of the MCM gateway is used as the source for read requests and the destination for write requests from remote Masters. Access to the database is controlled by the command type received in the incoming message from the remote Master. The following table defines the relationship of the gateway's internal database to the addresses required in the incoming Modbus TCP/IP requests:

Database Address	Modbus Address
0	40001 (five-digit addressing) or 400001 (six-digit addressing)
1000	41001 or 410001
2000	42001 or 420001
3000	43001 or 430001
3999	44000 or 440000

The following virtual addresses are not part of the normal gateway user database and are not valid addresses for standard data. However, these addresses may be used for incoming commands that are requesting 32-bit floating-point data. To use addresses in this upper range requires you to:

- 1 Set the **Float Flag** (page 25) to **YES**
- 2 Set the **Float Start** (page 26) to a database address in the range below
- 3 Set the **Float Offset** (page 26) to a database address in the gateway user memory area shown above.

Remember that, once you do this, all data in addresses above the **Float Offset** address will be treated as 32-bit floating-point data and require two gateway memory registers per Modbus address.

Database Address	Modbus Address
4000	44001 (five-digit addressing) or 440001 (six-digit addressing)
5000	45001 or 450001
6000	46001 or 460001
7000	47001 or 470001
8000	48001 or 480001
9000	49001 or 490001
9999	50000 or 500000

3 MCM Protocol Configuration

In This Chapter

❖ [Modbus Port x]	25
❖ [Modbus Port X Commands]	34

#This is an example of a Master port using Modbus RTU protocol

[Modbus Port 0]

Settings required by both Master and Slave

Enabled : Yes #Port enable flag Y=Yes, N=No
RS Interface : 0 #0=RS-232, 1=RS-485, 2=RS-422
Type : Master #Port type M=Master, S=Slave

Float Flag : No #Use floating data type Y=Yes, N=No
Float Start : 7000 #Register offset in message for floats
Float Offset : 2000 #Internal Address for floats

Protocol : RTU #Modbus Protocol R=RTU, A=ASCII
Baud Rate : 384 #Baud rate for port
Parity : None #N=None, O=Odd, E=Even
Data Bits : 8 #5 to 8 data bits for messages
Stop Bits : 1 #1 or 2 stop bits for messages
RTS On : 0 #Delay after RTS set before message sent (mSec)
RTS Off : 1 #Delay after message before RTS dropped(mSec)
Minimum Response Delay : 2 #Number of mSec to delay before response
Use CTS Line : No #Monitor CTS modem line (Y/N)
Response Timeout : 1000 #mSec for response timeout of command
Retry Count : 2 #retry count for failed requests

Settings required for Slave Driver

Internal Slave ID : 1 #Modbus Node address for this port on network
Bit Input Offset : 0 #Internal DB offset to bit input data
Word Input Offset : 0 #Internal DB offset to word input data
Output Offset : 0 #Internal DB offset to bit output data
Holding Register Offset : 0 #Internal DB offset to holding reg data
Use Guard Band Timer : N #Use packet gap timeout for messages
Guard Band Timeout : 0 #0=Use default for baudrate, or set timeout

Settings required for Master Driver

Minimum Command Delay : 0 #mSec delay between commands
Error Delay Counter : 100 #0-65535 Command cycle count if error
Command Control Reg : -1 #DB start reg for Cmd Control (-1=disable)

[Modbus Port 0 Commands]

#	Internal	Poll	Reg	Swap	Node	Modbus	MB Address
# Enable	Address	Interval	Count	Code	Address	Func	in Device
START							
1	0	0	10	0	1	3	0
1	10	0	10	0	1	4	10
END							

3.1 [Modbus Port x]

The [Modbus Port x] sections of the **CFG** file, where x stands for 0, 1, 2, or 3, set the Modbus Master and slave port communication parameters and define the protocol-specific settings.

3.1.1 *Enabled*

YES or **NO**

This flag specifies if the port on the gateway will be utilized. If the parameter is set to **NO**, the port will not be used. If the parameter is set to **YES**, the port will be used.

3.1.2 *RS Interface*

0 = RS-232

1 = RS-485

2 = RS-422

This parameter is used when Ports 1, 2, and 3 are present to configure the electrical interface for these ports.

RS Interface selections for Port 0 are done with a jumper (electrical shunt) on the back of the gateway.

3.1.3 *Type*

MASTER or **SLAVE**

This parameter defines if the port will emulate a Master or slave device. Choose a value of **MASTER** to make the port function as a Modbus Master. Choose **SLAVE** to make the port function as a Modbus slave.

3.1.4 *Float Flag*

YES or **NO**

This flag specifies how the Slave driver will respond to Function Code 3, 6, and 16 commands (read and write Holding Registers) from a remote Master when it is moving 32-bit floating-point data.

If the remote Master expects to receive or will send one complete 32-bit floating-point value for each count of one (1), then set this parameter to **YES**. When set to **YES**, the Slave driver will return values from two consecutive 16-bit internal memory registers (32 total bits) for each count in the read command, or receive 32-bits per count from the Master for write commands. Example: Count = **10**, Slave driver will send 20 16-bit registers for 10 total 32-bit floating-point values.

If, however, the remote Master sends a count of two (2) for each 32-bit floating-point value it expects to receive or send, or, if you do not plan to use floating-point data in your application, then set this parameter to **NO**, which is the default setting.

You will also need to set the *Float Start* and *Float Offset* parameters to appropriate values whenever the *Float Flag* parameter is set to **YES**.

3.1.5 Float Start

0 TO 65535

Whenever the *Float Flag* parameter is set to **YES**, this parameter determines the lowest Modbus Address, received in commands from a remote Master, to consider as requests to read or write floating-point data. All commands with address values greater than or equal to this value will be considered floating-point data requests. All commands with address values less than this value will be considered normal 16-bit register data requests.

This parameter is used only if the *Float Flag* is set to **YES**. For example, if a value of 7000 is entered, all commands received with addresses of 47001 (or 407001) and above will be considered as requests for floating-point data and 32-bits of data will be returned for each count of one in the command.

You will also need to set the *Float Offset* parameter to an appropriate value whenever the *Float Flag* parameter is set to **YES**.

3.1.6 Float Offset

0 TO 3998

This parameter defines the starting register for floating-point data in the internal gateway database. This parameter is used only if the *Float Flag* is set to **YES**.

For example, if the *Float Offset* value is set to 3000 and the *Float Start* parameter is set to 7000, the data returned as floating-point data for register 47001 (or 407001) will actually come from internal gateway registers 3000 and 3001. If the requested address was 47002 (407002), the data will be returned from internal registers 3002 and 3003. If the requested address was 47101 (407101), the data will be returned from internal registers 3200 and 3201; and so on.

3.1.7 Protocol

RTU

ASCII

This parameter specifies the Modbus protocol version to be used on the port. Valid Protocols are RTU and ASCII.

3.1.8 Baud Rate

This is the baud rate to be used on the port. Enter the baud rate as a value.

Baud Rate	Parameter Value
110	110
150	150
300	300
600	600
1200	12 OR 1200
2400	24 OR 2400
4800	48 OR 4800
9600	96 OR 9600
19,200	19, 192 OR 19200
38,400	38, 384 OR 38400
57,600	57 OR 576
115,200	115 OR 1152

Note: On all ProLinx gateways, data from the application port on the main board, serial Port 0, is not buffered. Packets go directly to and from the serial chipset to the processor. This has the potential to cause the serial communications to become erratic at baud rates above 38,400 baud.

ProLinx gateways with 4 serial ports have a separate serial interface board for serial Ports 1, 2, and 3. These serial ports are buffered, and can handle communications up to 115,200 baud.

3.1.9 Parity

None, Odd, Even

Parity is a simple error checking algorithm used in serial communication. This parameter specifies the type of parity checking to use.

All devices communicating through this port must use the same parity setting.

3.1.10 Data Bits

7 or 8

This parameter sets the number of data bits for each word used by the protocol. All devices communicating through this port must use the same number of data bits.

3.1.11 Stop Bits

1 or 2

Stop bits signal the end of a character in the data stream. For most applications, use one stop bit. For slower devices that require more time to re-synchronize, use two stop bits.

All devices communicating through this port must use the same number of stop bits.

3.1.12 RTS On

0 to **65535** milliseconds

This parameter sets the number of milliseconds to delay after *Ready To Send* (RTS) is asserted before data will be transmitted.

3.1.13 RTS Off

0 to **65535** milliseconds

This parameter sets the number of milliseconds to delay after the last byte of data is sent before the RTS modem signal will be set low.

3.1.14 Minimum Response Delay

0 to **65535**

This parameter is used only when the port is configured as a slave. It sets the number of milliseconds to wait before responding to a command received on the port from a remote Master. This delay is sometimes required to accommodate slower Master devices.

3.1.15 Use CTS Line

YES or **No**

This parameter specifies if the Clear To Send (CTS) modem control line is to be used or not. If the parameter is set to **No**, the CTS line will not be monitored. If the parameter is set to **YES**, the CTS line will be monitored and must be high before the gateway will send data. Normally, this parameter is required when half-duplex modems are used for communication (2-wire). This procedure is commonly referred to as *hardware handshaking*.

3.1.16 Response Timeout

0 to **65535** milliseconds

This parameter sets the command response timeout period in 1 millisecond increments. This is the time that a port configured as a Master will wait for a response from the addressed slave before re-transmitting the command (Retries) or skipping to the next command in the Command List. The value to set depends on the communication network used and the expected response time (plus a little extra) of the slowest device on the network.

3.1.17 Retry Count

0 to **10**

This parameter specifies the number of times a command will be retried if it fails.

3.1.18 Internal Slave ID

0 TO 255

When the port is configured as a slave, this parameter defines the virtual Modbus slave address for the port. Any commands received on the slave port, addressed to the node address entered here, will be processed by the gateway. Each slave device on the network must be assigned a unique address.

3.1.19 Bit Input Offset

0 TO 3999

When the port is configured as a slave, this parameter specifies the internal database address to use as the zero address or starting point for binary Input Status data. Input Status data is read-only data, requested by Modbus Function Code 2 commands (Read Input Status). For example, if this parameter is set to 150 and a Function Code 2 command is received requesting Input Status address 0 (virtual Modbus Address 10001 or 100001), the data returned in the response will be the bit value at register 150, bit 0 in the gateway's database.

3.1.20 Word Input Offset

0 TO 3999

When the port is configured as a slave, this parameter specifies the internal database address to use as the zero address or starting point for Input Register (16-bit integer) data. Input Register data is read-only data, requested by Modbus Function Code 4 commands (Read Input Registers). For example, if this parameter is set to 500 and a Function Code 4 command is received requesting Input Register address 0 (virtual Modbus address 30001 or 300001), the data returned in the response will be the value at register 500 in the gateway's database.

3.1.21 Output Offset

0 TO 3999

When the port is configured as a slave, this parameter specifies the internal database address to use as the zero address or starting point for binary output Coil data. Coil data is read by Modbus Function Code 1 commands (Read Coils) and written by Function Codes 5 (Force Single Coil) or Function Code 15 (Force Multiple Coils). For example, if this parameter is set to 50 and a Function Code 1 command is received requesting Coil address 0 (virtual Modbus Coil address 00001 or 000001), the data returned in the response will be the value at register 50, bit 0 in the gateway's database.

3.1.22 Holding Register Offset

0 to 3999

When the port is configured as a slave, this parameter specifies the internal database address to use as the zero address or starting point for Holding Register (16-bit integer) data. Holding Register data is read by Modbus Function Code 3 commands (Read Holding Registers) and written by Function Code 6 (Preset Single Register) or Function Code 16 (Preset Multiple Registers).

For example, if this parameter is set to 1000 and a Function Code 3 command is received requesting Holding Register address 0 (virtual Modbus address 40001 or 400001), the data returned in the response will be the value at register 1000 in the gateway's database.

3.1.23 Use Guard Band Timer

YES or NO

This parameter specifies if the Guard Band feature (intercharacter delay monitoring and control) is to be used or not. Normally, this feature will not be needed. It is used only when the port is set for Modbus RTU mode as a slave device in a multi-drop, multi-node network (RS-422 or RS-485 wiring). If the remote Master experiences multiple, successive failures when trying to communicate with the gateway on a multi-drop network, setting a small amount of Guard Band Time will usually completely eliminate the failures.

When the port is configured as a slave and when multiple slave nodes exist on the network (multi-drop), without the Guard Band Timer, the gateway might not be able to properly discriminate between a response message returned by another slave and a command from the Master to which it should respond. The *Guard Band Timeout* parameter allows you to set a small delay which can help eliminate communication errors on the Modbus network.

To enable the Guard Band Timeout parameter:

- 1** First of all, enable the the *Use Guard Band Timer* parameter.

Use Guard Band Timer = Y

- 2** Then, set the *Guard Band Timeout*.

Guard Band Timeout = 0

This configuration will set the gateway to use a pre-defined time interval for Modbus message recognition calculated for each baud rate. This should provide optimal performance for most applications.

If you still observe a considerable number of communication errors, try to increase the *Guard Band Timeout* value until the network performance is improved.

3.1.24 Guard Band Timeout

0 TO 65535

A value of 1 to 65535 sets the time in milliseconds. When this period has elapsed and no further data is received, the gateway will handle the last received data sequence as a Modbus message to be processed.

This parameter specifies the *Guard Band Timeout* between packets in multi-drop slave mode. A value of 0 uses the default time, which is roughly equivalent to the time it would take to send 3.5 bytes at the selected baud rate. This is the delay value specified by the Modbus RTU protocol specifications. The following *Guard Band Timeout* values will be automatically used by the gateway when *Use Guard Band* is set to **YES** and the *Guard Band Timeout* is set to the default value of zero (**0**).

Baud	Guard Band Timer (ms)
110	350
150	256
300	128
600	64
1200	32
2400	16
4800	8
9600	4
19200	2
28800	2
38400	2
57600	1
115200	1

3.1.25 Minimum Command Delay

0 TO 65535

This parameter specifies the number of milliseconds to wait between receiving the end of a slave's response to the most recently transmitted command and the issuance of the next command. This parameter can be used to place a delay after each command to avoid sending commands on the network faster than the slaves can be ready to receive them. This parameter does not affect retries of a command, as retries will be issued when a command failure is recognized.

3.1.26 Error Delay Counter

0 to 65535

This parameter specifies the number of poll attempts to be skipped before trying to re-establish communications with a slave that has failed to respond to a command within the time limit set by the *Response Timeout* parameter. After the slave fails to respond, the master will skip sending commands that should have been sent to the slave until the number of skipped commands matches the value entered in this parameter. This creates a sort of *slow poll* mode for slaves that are experiencing communication problems.

3.1.27 Command Control Reg

0 to 3900

This parameter allows you to control the execution of commands in the Command List (page 34) by setting a value of zero (0), one (1), two (2), or three (3) into a 100-register area of the gateway database, beginning at the address entered in this parameter. If a user defines commands in the list with an Enable (page 35) code of zero for all commands, no commands will execute. If the value in the first control register is changed to one (1), command zero will execute continuously. If the value is the control register for a write command is set to two (2), the command will be enabled for conditional writing, which will cause the command to execute whenever the values in the database registers associated with the command change. Use the value of three (3) only for bit-level write commands, FC 5 and 15. If the parameter is set to three(3), the command will execute only if the internal bit data associated with the command changes. It will also clear the bit or bits in the internal database after the write command is built. This Command Control feature can be disabled by setting this parameter to a value of -1 or by omitting the item name from the configuration file, resulting the gateway to using a default value of -1.

3.2 [Modbus Port X Commands]

The [Modbus Port X Commands] (where X can be 0, 1, 2, or 3) sections of the CFG file are used to define a Master serial port *Command List*. This list holds the parameters needed to poll slave devices attached to a Master port. The gateway supports eight types of commands. This permits the gateway to interface with a wide variety of Modbus slave devices.

The *Command List* is formatted differently than the other sections of the configuration file. Commands are present in a block between the labels **START** and **END**. These labels inform the program where the list resides. The module's program will parse all commands after the **START** label until it reaches the **END** label or until the command count entered for the port is reached.

The format of each command in the list is the same. The parameter values entered determine the operation to be performed. The following table lists supported Modbus Functions Codes and outlines their format.

Module Information Data □					□ Device Information Data			
Column #	1	2	3	4	5	6	7	8
Modbus Function Code	Enable Code	Internal Address	Poll Interval Time	Count	Swap Code	Slave Node	Function Code	Device MODBUS Address
FC1	Code	Bit	Seconds	Count	0	Address	1	Bit
FC2	Code	Bit	Seconds	Count	0	Address	2	Bit
FC3	Code	Register	Seconds	Count	Code	Address	3	Register
FC4	Code	Register	Seconds	Count	0	Address	4	Register
FC5	Code	Bit	Seconds	Count	0	Address	5	Bit
FC6	Code	Register	Seconds	Count	Code	Address	6	Register
FC15	Code	Bit	Seconds	Count	0	Address	15	Bit
FC16	Code	Register	Seconds	Count	Code	Address	16	Register

The first part of the record is the Module Information, which relates to the ProLinx module and the second part contains information required to interface to the Modbus slave device. Refer to the Modbus protocol specification and slave device documentation for a full discussion of each function. The **Device Modbus Address** information can be found in the documentation for the slave device.

Command list example:

```
[PORT 0 COMMANDS]
# INTERNAL POLL SWAP SLAVE FUNCTION DEVICE
# ENABLE ADDRESS INTERVAL COUNT CODE NODE CODE ADDRESS
START
 1 0 0 10 0 1 3 0
 1 0 0 10 0 1 16 10
END
```

3.2.1 *Enable*

0, 1, 2 or 3

This field defines whether the command is to be executed and under what conditions.

If the parameter is set to 0, the command is disabled and will not be executed in the normal polling sequence. However, the command can still be activated using Command Control.

Setting the parameter to a value of 1 causes the command to be executed each scan of the *Command List* if the *Poll Interval* is set to zero (0). If the Poll Interval is set to a non-zero value, the command will be executed, when the interval timer for that command expires.

An *Enable* code of 2 is valid only for write commands, FC 5, 6, 15, and 16. If the parameter is set to 2, the command will execute only if the internal data associated with the command changes.

An *Enable* code of 3 is valid only for bit-level write commands, FC 5 and 15. If the parameter is set to 3, the command will execute only if the internal bit data associated with the command changes. It will also clear the bit or bits in the internal database after the write command is built.

Important: Not all revisions of the MCM driver support an *Enable* code of 3. To determine if your gateway supports this *Enable* code, contact ProSoft Technical Support.

3.2.2 *Internal Address*

0 to 3999 (for register-level addressing)

or

0 to 63999 (for bit-level addressing)

This field specifies the database address in the gateway's internal database to use as the destination for data brought in by a read command or as the source for data to be sent out by a write command. The database address is interpreted as a bit address or a 16-bit register (word) address, depending on the Modbus Function Code used in the command.

- For Modbus functions 1, 2, 5, and 15, this parameter is interpreted as a bit-level address.
- For Modbus functions 3, 4, 6, and 16, this parameter is interpreted as a register-level address.

3.2.3 *Poll Interval*

0 to 65535

This parameter specifies the minimum interval between executions of a continuous commands (*Enable* code of 1). The value is in seconds. Therefore, if a value of 10 is entered, the command will execute no more frequently than once every 10 seconds.

3.2.4 *Reg Count*

Regs: **1 to 125**

Coils: **1 to 800**

This parameter specifies the number of 16-bit registers or binary bits to be transferred by the command.

- Functions 5 and 6 ignore this field as they apply only to a single data point.
- For functions 1, 2, and 15, this parameter sets the number of bits (inputs or coils) to be transferred by the command.
- For functions 3, 4, and 16, this parameter sets the number of registers to be transferred by the command.

3.2.5 *Swap Code*

NONE

SWAP WORDS

SWAP WORDS & BYTES

SWAP BYTES

This parameter defines if and how the order of bytes in data received or sent is to be rearranged. This option exists to allow for the fact that different manufacturers store and transmit multi-byte data in different combinations. This parameter is helpful when dealing with floating-point or other multi-byte values, as there is no one standard method of storing these data types. The parameter can be set to rearrange the byte order of data received or sent into an order more useful or convenient for other applications. The following table defines the valid *Swap Code* values and the effect they have on the byte-order of the data.

Swap Code	Description
NONE	No change is made in the byte ordering (1234 = 1234)
SWAP WORDS	The words are swapped (1234=3412)
SWAP WORDS & BYTES	The words are swapped, then the bytes in each word are swapped (1234=4321)
SWAP BYTES	The bytes in each word are swapped (1234=2143)

These swap operations affect 4-byte (or 2-word) groups of data. Therefore, data swapping using these *Swap Codes* should be done only when using an even number of words, such as when 32-bit integer or floating-point data is involved.

3.2.6 Slave Address

0 - Broadcast to all nodes

1 to 255

Use this parameter to specify the slave address of a remote Modbus Serial device through a Modbus Ethernet to Serial converter.

Note: Use the *Node IP Address* parameter to address commands to a remote Modbus TCP/IP device.

Note: Most Modbus devices accept an address in the range of only 1 to 247, so check with the slave device manufacturer to see if a particular slave can use addresses 248 to 255.

If the value is set to zero, the command will be a broadcast message on the network. The Modbus protocol permits broadcast commands for **write** operations. **Do not** use node address 0 for **read** operations.

3.2.7 Modbus Function

1, 2, 3, 4, 5, 6, 15, or 16

This parameter specifies the Modbus Function Code to be executed by the command. These function codes are defined in the Modbus protocol. The following table lists the purpose of each function supported by the module. More information on the protocol is available from www.modbus.org.

Modbus Function Code	Description
1	Read Coil Status
2	Read Input Status
3	Read Holding Registers
4	Read Input Registers
5	Force (Write) Single Coil
6	Preset (Write) Single Register
15	Force Multiple Coils
16	Preset Multiple Registers

3.2.8 MB Address in Device

This parameter specifies the starting Modbus register or bit address in the Slave to be used by the command. Refer to the documentation of each Modbus Slave device for the register and bit address assignments valid for that device.

The Modbus Function Code determines whether the address will be a register-level or bit-level OFFSET address into a given data type range. The offset will be the target data address in the Slave minus the base address for that data type. Base addresses for the different data types are:

- 00001 or 000001 (0x0001) for bit-level Coil data (Function Codes 1, 5, and 15).
- 10001 or 100001 (1x0001) for bit-level Input Status data (Function Code 2)
- 30001 or 300001 (3x0001) for Input Register data (Function Code 4)
- 40001 or 400001 (4x0001) for Holding Register data (Function Codes 3, 6, and 16).

Address calculation examples:

- For bit-level Coil commands (FC 1, 5, or 15) to read or write a Coil 0X address 00001, specify a value of 0 (00001 - 00001 = 0).
- For Coil address 00115, specify 114
(00115 - 00001 = 114)
- For register read or write commands (FC 3, 6, or 16) 4X range, for 40001, specify a value of 0
(40001 - 40001 = 0).
- For 01101, 11101, 31101 or 41101, specify a value of 1100.
(01101 - 00001 = 1100)
(11101 - 10001 = 1100)
(31101 - 30001 = 1100)
(41101 - 40001 = 1100)

Note: If the documentation for a particular Modbus Slave device lists data addresses in hexadecimal (base16) notation, you will need to convert the hexadecimal value to a decimal value to enter in this parameter. In such cases, it is not usually necessary to subtract 1 from the converted decimal number, as this addressing scheme typically uses the exact offset address expressed as a hexadecimal number.

4 Serial Port Modbus Error and Status Data

In This Chapter

- ❖ Viewing Error and Status Data 40
- ❖ Modbus Error and Status Data Area Addresses..... 41
- ❖ Error Codes 47

The most thorough troubleshooting method for debugging the operation of the MCM driver (and the gateway in general) is the powerful Debug port on the gateway. Accessing the Debug capabilities of the gateway is accomplished by connecting a PC to the Debug port and loading a terminal program such as ProSoft Configuration Builder or Windows HyperTerminal.

Note: 4101 series modules that use both serial ports (Debug and Port x) as application ports (for example, 4101-MCM-DFCM) must be reset or power-cycled in order to enable the selected port as a Debug port. To enable the debug port, connect to the port using a null-modem serial cable, and then press and hold the **[A]** key on the keyboard while resetting or power cycling the card. Do not release the key until HyperTerminal shows the debug port menu. Debug mode can be enabled on either port. The upload/download command is only available when connected to the Debug port.

4.1 Viewing Error and Status Data

The following topics list the register addresses that contain error and status data. Use the Database View option from the ProLinx Main Menu to view the contents of each register. The *ProLinx Reference Guide* provides additional information on using this option.

4.2 Modbus Error and Status Data Area Addresses

Modbus error and status data are stored in registers based on port number. Starting register addresses are shown in the following table.

Modbus Port	Starting Address
0	4400
1	4800
2	5200
3	5600

Note: None of the addresses are available in the Modbus address range. In order to view them via a Modbus request, they must be moved into the 0 to 3999 address range using the Data Map section of the configuration file. For additional information on how to move data within the gateway's internal database, see *Moving Data* (page 87).

4.2.1 Modbus Ports: Error and Status

The serial port (Modbus Master/Slave) Error and Status Data areas are discussed in this section.

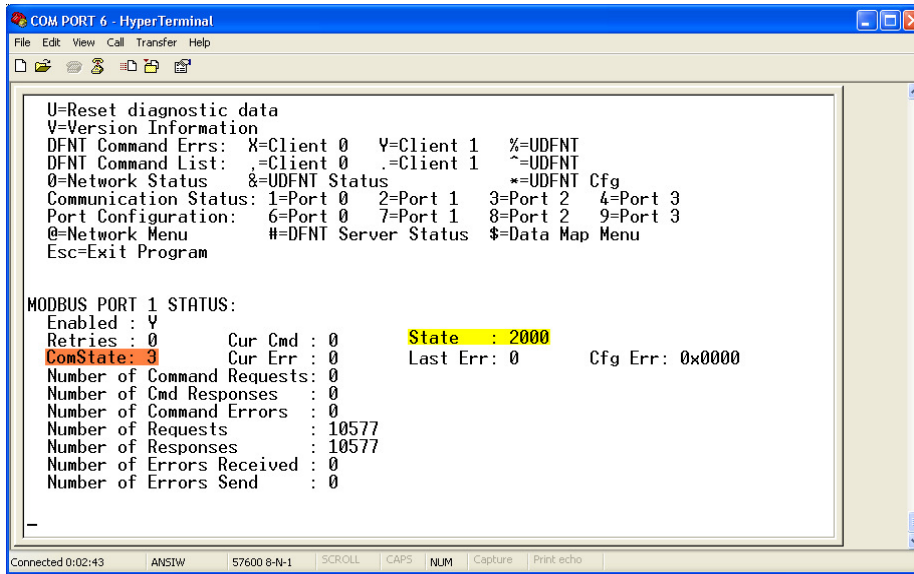
The data area is initialized with zeros whenever the gateway is restarted. This occurs during a cold-start (power-on), reset (reset push-button pressed) or a warm-boot operation (commanded from a debug menu or after downloading a new configuration). The addresses listed are for Port 0 only; but the format is the same for each port. The start address for each port is given in the previous section, *Modbus Error and Status Data Area Addresses* (page 41).

Example Internal Database Address	Offset	Description
4400	0	Number of Command Requests
4401	1	Number of Command Responses
4402	2	Number of Command Errors
4403	3	Number of Requests
4404	4	Number of Responses
4405	5	Number of Errors Sent
4406	6	Number of Errors Received
4407	7	Configuration Error Code
4408	8	Current Error/Index
4409	9	Last Error/Index

Refer to the following *Error Codes* (page 47) section to interpret the status/error codes present in the data area.

4.2.2 Slave Port: General Port STATE and COMM STATE Status

In addition to the port status information stored in the gateway database memory, as already mentioned, there are two additional port status parameters that are not available from the database. The status values, Port STATE (highlighted in yellow) and Port COMM STATE (highlighted in orange), can be found only through the diagnostic menus under "Communication Status" for a port, as shown in the following graphic.



These status values are state registers. They are used as 'scratchpad' areas by the operating firmware to keep track of the current logical state of activities on a slave port. These state registers are constantly changing as the gateway progresses through the various stages needed to process communication on the serial ports. This processing happens faster than can be followed, unless a port error causes the value to remain constant for some noticeable length of time.

The diagnostic screen displays are 'snapshot' type screens; that is, they display the current value at the time they are called but do not continue to automatically update the data displayed. They are not live-data screens. Therefore, to see a change in STATE or COMM STATE, you must repeatedly call the screen by pressing the appropriate menu key to have the screen refreshed. Depending on the refresh timing you may or may not see a change in the displayed values everytime you call for an update.

The STATE register may display any of the following values.

STATE Value	Description
-2	Preparing port, Flushing all buffers, Scan for RTS-Off, or Waiting for port enable signal
-1	Waiting for receipt of data
0	Undefined state
1	Receiving a message from the Master
2	Building a slave response message
2000	Sending slave response to the Master

The COMM STATE register may display any of the following values.

COMM STATE Value	Description
0	Port not sending data (Wait for send)
1	Setting up minimum Response Delay Timer
101	Waiting for Minimum Response Delay to timeout
2	Turning on RTS line and starting RTS-On Delay Timer
3	Waiting for RTS-On Delay timer to timeout
4	Sending data
5	Waiting for all data to be sent and starting RTS-Off Delay Timer
6	Waiting for RTS-Off Delay timeout and then turning OFF RTS line after timeout

4.2.3 Master Port: Command List Errors

The individual command errors for each Master port are returned to the address locations specified in the following table. Each port can have up to 100 commands configured. Each configured command will use one word of these data areas to store a value representing the execution status from the most recent command execution attempt.

Modbus Port	Address Range
0	4410 to 4509
1	4810 to 4909
2	5210 to 5309
3	5610 to 5709

The first word in the defined register location contains the status/error code for the first command in the port's *Command List*. Successive words in the *Command Error List* are associated with corresponding commands in the list. Refer to Error Codes (**page 47**) to interpret the status/error codes present in this data area.

Port 0 Command Error List Layout

The addresses listed are for Port 0 only; but the format is the same for each port. The start address for each port is given in the previous section, Master Port: Command List Errors (page 43).

Internal Database Address (Example)	Offset	Description
4410	0	Command #0 Error Status
4411	1	Command #1 Error Status
4412	2	Command #2 Error Status
4413	3	Command #3 Error Status
4414	4	Command #4 Error Status
...
...
...
4507	97	Command #97 Error Status
4508	98	Command #98 Error Status
4509	99	Command #99 Error Status

Note that the values in the *Command Error List* tables are initialized to zero (0) at power-up, cold boot, and warm boot. If a command executes successfully, the value in the associated register will remain at zero (0), indicating no command error was detected. Any non-zero value in this table indicates the corresponding command experienced an error. The Error Code (page 47) shown will provide valuable troubleshooting information.

The data in this table is dynamic. It is updated each time a command is executed. Therefore, if the command fails once and succeeds on the next attempt, the Error Code from the previously failed attempt will be replaced with zero and lost. Error Codes are not archived in the gateway's database. To see if the port has experienced an error since the most recent restart and what the most recently occurring error was, if any, you can check the Last Error/Index (page 41).

4.2.4 Master Port: Modbus Slave List Status

The slave status list contains the current poll status of each slave device on a Master port. Slaves attached to a Master port can have one of three states.

0	The slave has not defined in the command list for the Master port and will not be polled from the Command List.
1	The slave is configured to be polled by the Master port and the most recent communication attempt was successful.
2	The Master port has failed to communicate with the slave device. Communication with the slave is suspended for a user defined period based on the scanning of the command list.

Slaves are defined to the system when the gateway loads the Master Command List during start-up and initialization. Each slave defined will be set to a state value of 1 in this initial step. If the Master port fails to communicate with a slave device (timeout expired on a command, retries failed), the Master will set the state of the slave to a value of 2 in this status table. This suspends communication with the slave device for a user-specified Error Delay Count (page 33).

When the Master first suspends polling of a particular slave, it creates a Error Delay Counter for this slave address and set the value in that counter equal to the Error Delay Counter parameter in the configuration file. Then, each time a command in the list is scanned that has the address of a suspended slave, the delay counter value for that slave will be decremented. When the value reaches zero, the slave state will be set to 1. This will re-enable polling of the slave.

The first word in the defined register locations contains the status code for slave node address 1. Each successive word in the list is associated with the next node in sequence, up to slave node address 255.

The individual Slave List Status for each Modbus port are returned to the address locations specified in the following table.

Modbus Port	Address Range
0	4510 to 4764
1	4910 to 5164
2	5310 to 5564
3	5710 to 5965=4

Port 0 Slave List Status Layout

The addresses listed are for Port 0 only; but the format is the same for each port. The start address for each port is given in the previous section, Master Port: Modbus Slave List Status. (page 45)

Internal Database Address (Example)	Offset	Description
4510	0	Slave #1 Status
4511	1	Slave #2 Status
4512	2	Slave #3 Status
4513	3	Slave #4 Status
4514	4	Slave #5 Status
.	.	.
.	.	.
.	.	.

Note that the values in the Slave List Status tables are initialized to zero (0) at power-up, cold boot and during warm boot.

4.3 Error Codes

These are error codes that are part of the Modbus protocol or are extended codes unique to this gateway.

4.3.1 Modbus Error Codes

These error codes are generated or returned on both the Master and slave ports. These codes are the standard Modbus errors (page 84).

Code	Description
1	Illegal Function
2	Illegal Data Address
3	Illegal Data Value
4	Failure in Associated Device
5	Acknowledge
6	Busy, Rejected Message

4.4 Module Communication Error Codes

These gateway-specific error codes are also returned from the command polling process and stored in the Command Error List memory area.

Code	Description
-1	CTS modem control line not set before transmit
-2	Timeout while transmitting message
-11	Timeout waiting for response after request
253	Incorrect slave address in response
254	Incorrect function code in response
255	Invalid CRC/LRC value in response

4.4.1 Command List Error Codes

These command-specific error codes are detected during initial command list loading at gateway power-up or reset and are stored in the *Command Error List* memory region.

CODE	Description
-41	Invalid enable code
-42	Internal address > maximum address
-43	Invalid node address (<0 or > 255)
-44	Count parameter set to 0
-45	Invalid function code
-46	All parameters set to 0
-47	All parameters set to -1

4.4.2 Modbus Configuration Error Word

Modbus Configuration Error Word values are stored in protocol-specific registers. The following table lists the register address used for each port that may exist on the gateway.

Modbus Port	Configuration Error Word Register
0	4407
1	4807
2	5207
3	5607

If any of these registers contains a non-zero value, it indicates a problem with the configuration of the associated port. The following table lists the codes, a description of the problem, and which parameters to change to correct the error condition within the configuration file.

Bit	Code	Description
0	0x0001	Invalid Enabled parameter (Yes or No)
1	0x0002	Invalid RS-Interface parameter (0 to 2)
2	0x0004	Invalid Type (Master or Slave)
3	0x0008	Invalid Protocol (RTU or ASCII)
4	0x0010	Invalid Baud Rate
5	0x0020	Invalid Parity (None, Odd, Even)
6	0x0040	Invalid Data Bits (7 or 8 bits)
7	0x0080	Invalid Stop Bits (1 or 2)
8	0x0100	Invalid Use CTS Line (Yes or No)
9	0x0200	Retry Count Invalid (0 to 10)
10	0x0400	Invalid Floating Point Data: Float Flag not Yes or No Float Start less than 0 or Float Offset is Invalid
11	0x0800	Invalid Internal Slave ID (1 to 255) (Slave Only)
12	0x1000	Invalid Entry for Register Offset Data (Slave Only)
13	0x2000	Reserved
14	0x4000	Reserved
15	0x8000	Reserved

5 Communication Port Cables

This section contains information on the cable and pin assignments for the ProLinx gateway's serial ports (RS-232/422/485). The ProLinx gateway will come with one to five serial ports, depending on the configuration purchased. In all cases, the protocol serial ports will have the same pinouts.

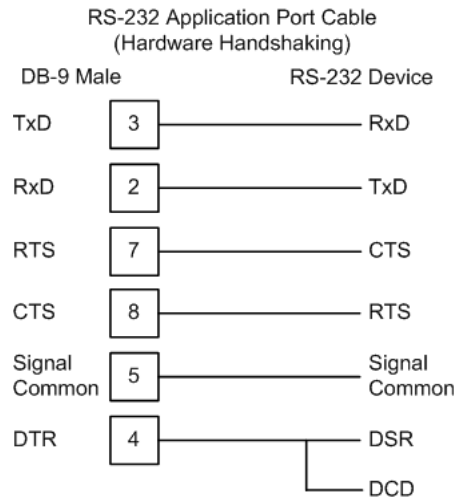
Example: The 5202-MNET-MCM4 gateway contains five serial communication ports; four configurable protocol application ports and one Configuration/ Debug port.

The 5201-MNET-MCM gateway contains two serial communication ports; one configurable protocol application port and one Configuration/Debug port.

Each physical serial port has an eight-pin Mini-DIN jack connector. A six-inch Mini-DIN-8Male to DB-9Male adapter cable is provided for each serial port. The DB-9M provides connections for RS-232, wired as Data Terminal Equipment (DTE), RS-422 and RS-485. The diagrams in the following topics detail the pin assignments for several possible electrical interface connections.

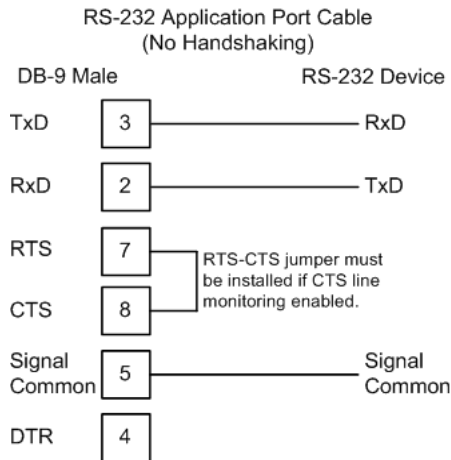
5.1 Port 0, 1, 2, 3: RS-232 - Null Modem (DTE with Hardware Handshaking)

This type of connection is used when the device connected to the gateway requires hardware handshaking (control and monitoring of modem signal lines; Use *CTS* (page 28) parameter set to **YES**).



5.2 Port 0, 1, 2, 3: RS-232 - Null Modem (DTE without Hardware Handshaking)

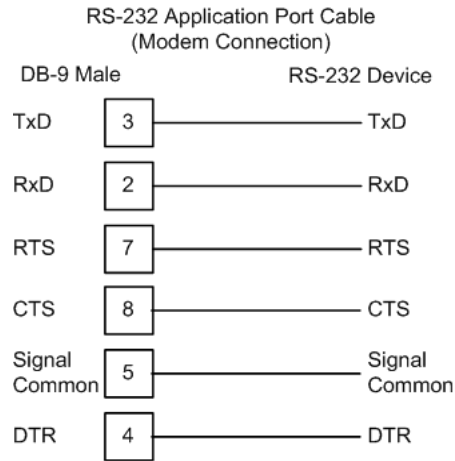
This type of connection can be used to connect the gateway to a computer or field device communication port.



Note: If the port is configured with the *Use CTS* (page 28) set to **YES**, then a jumper is required between the RTS and the CTS line on the gateway connection.

5.3 Port 0, 1, 2, 3: RS-232 - DTE to DCE Modem Connection

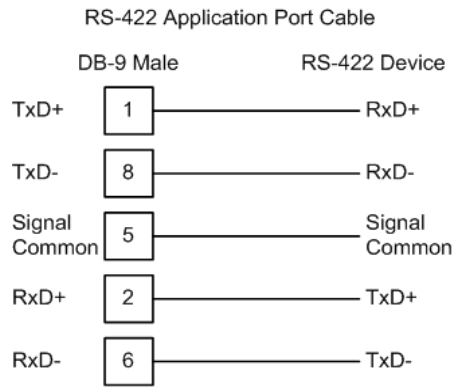
This type of connection is required between the gateway and a modem or other communication device.



The *Use CTS Line* (page 28) parameter for the port configuration should be set to **YES** for most modem applications.

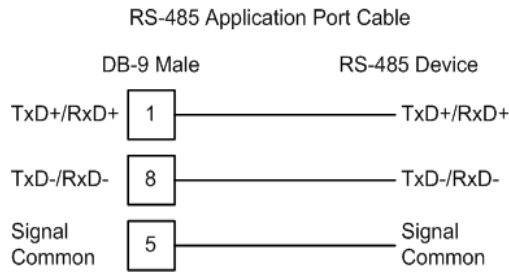
5.4 Port 0, 1, 2, 3: RS-422 Interface Connections

The following illustration applies when the RS-422 interface is selected.



5.5 Port 0, 1, 2, 3: RS-485 Interface Connections

The following illustration applies when the RS-485 interface is selected.



NOTE: This type of connection is commonly called a *RS-485 half-duplex, 2-wire* connection. If you have RS-485 4-wire, full-duplex devices, they can be connected to the gateway's serial ports by wiring together the TxD+ and RxD+ from the two pins of the full-duplex device to Pin 1 on the gateway and wiring together the TxD- and RxD- from the two pins of the full-duplex device to Pin 8 on the gateway. As an alternative, you could try setting the gateway to use the RS-422 interface and connect the full-duplex device according to the RS-422 wiring diagram (page 55). For additional assistance, please contact ProSoft Technical Support.

6 LED Indicators

In This Chapter

- ❖ Base Module LEDs..... 58
- ❖ LEDs for Serial Ports..... 59

LED indicators provide a means of monitoring the operation of the system and individual ports. They are extremely useful for troubleshooting. The gateway provides LEDs to help monitor each port. In addition, system configuration errors, application errors, and fault indications are all indicated by LEDs, providing alerts to possible problems.

6.1 Base Module LEDs

LED	State	Description
Power	Off	Power is not connected to the power terminals or source is insufficient to properly power the gateway (minimum required is 800mA at 24 Vdc)
	Green Solid	Power is connected to the power terminals.
Fault	Off	Normal operation.
	Red Solid	A critical error has occurred. Program executable has failed or has been user-terminated and is no longer running. Press Reset p/b or cycle power to clear error. If not, use the Debug procedures described later in this manual.
Cfg	Off	Normal operation.
	Amber Solid	The unit is in configuration mode. The configuration file is currently being downloaded or, after power-up, is being read, the unit is implementing the configuration values, and initializing the hardware. This will occur during power cycle, or after pressing the reset button. It also occurs after a cold/warm boot command is received.
Err	Off	Normal operation.
	Flashing	An error condition has been detected and is occurring on one of the application ports. Check configuration and troubleshoot for communication errors.
	Solid Red	This error flag is cleared at the start of each command attempt (Master/Client) or on each receipt of data (slave/adaptor/server); so, if this condition exists, it indicates a large number of errors are occurring in the application (due to bad configuration) or on one or more ports (network communication failures).

6.2 LEDs for Serial Ports

ProLinx gateways may have as many as five (5) serial ports. Each of these serial ports has two LEDs indicating status.

LED	Color	Description
Debug - ACT	Off	No activity on the port.
Port 0 - ACT	Green	The port is actively transmitting or receiving data
Port 1 - ACT	Flash	
Port 2 - ACT		
Port 3 - ACT		
Debug - ERR	Off	Normal state. When off and Port Active led is indicating activity, there are no communication errors
Port 0 - ERR		Activity on this LED indicates communication errors are occurring. To determine the exact error, connect the Debug terminal to the Debug port and use the built-in Diagnostic Menus (page 39).
Port 1 - ERR	RED	
Port 2 - ERR	On Solid or	
Port 3 - ERR	Flashing	

7 Modbus Port Expander (4102/5102-MBS3-MBM)

In a typical Modbus serial network implementation, one Modbus Master device would be configured to poll (send commands) to one or more Modbus slave devices. The Modbus protocol specification does not provide a way for multiple Modbus Masters to exist and actively poll slaves on the same physical network. Therefore, more than one Master is usually not able to communicate with one, particular slave.

The ProLinx Modbus Port Expander Communication Gateway is a specialized implementation of the Modbus protocol. It allows multiple Modbus Masters to communicate directly with one Modbus slave, something not normally possible using the Modbus RTU/ASCII serial protocol. It provides this unique capability by multiplexing the inputs received on up to three Modbus slave ports and retransmitting them out one Modbus Master port to one Modbus slave device. When the device responds, the response is routed back to the slave port on which the command was received and returned to the originating Modbus Master.

All internal multiplexing and routing is accomplished automatically by the gateway with no special command configuration required. The only configuration needed is for typical serial port communication, such as baud rate, parity, stop bits, hardware handshaking, and such. The following two sections describe these configuration settings in greater detail.

7.1 [Modbus Master Port]

There is one Master port on the Port Expander gateway. The purpose of this port is to pass all commands received on any of the three slave ports to the single Modbus slave which will be attached to this Master port. There is no way to define a Command List for this Master port. Therefore, it cannot act as a typical Modbus Master and poll one or more slaves on a serial network. It's sole purpose is to pass-through commands received on the gateway's slave ports.

7.1.1 *Internal Slave ID*

0 TO 255

This parameter defines the virtual Modbus slave address for all three slave ports on the Port Expander. It should be set to the same value as the Slave ID of the device attached to the gateway's Master port. Any commands received on the slave ports, addressed to the node address entered here, will be passed by the gateway to the slave attached to the gateway's Master port. Any response received from the slave will be passed back through the gateway's slave port on which it was received.

Note: If the internal slave ID is set to zero, all messages will be passed from the slave port(s) to the master port. This means that the gateway will respond to any command regardless of address. If set to 0 the gateway has to be the only slave attached to any network attached to any of the slave ports on the port expander.

7.1.2 *Protocol*

RTU

ASCII

This parameter specifies the Modbus protocol version to be used on the port. Valid Protocols are RTU and ASCII.

7.1.3 Baud Rate

This is the baud rate to be used on the Master port. It must match the baud rate setting used by the slave that will be connected to this Master port.

To maximize the port multiplexing capability of the gateway, it is recommended, if possible, that you set the baud rate on the Master port and on the attached slave as high as the slave can accommodate, up to 38,400; then set the baud rate for the gateway slave ports and remote Master networks to a lower setting. This will help prevent multiple Masters from overloading the response capability of the single slave and the gateway.

Enter the baud rate as a value, shown in this list.

Baud Rate	Parameter Value
110	110
150	150
300	300
600	600
1200	12 OR 1200
2400	24 OR 2400
4800	48 OR 4800
9600	96 OR 9600
19,200	19, 192 OR 19200
38,400	38, 384 OR 38400
57,600	57 OR 576
115,200	115 OR 1152

Note: On all ProLinx gateways, data from the application port on the main board, serial Port 0, is not buffered. Packets go directly to and from the serial chipset to the processor. This has the potential to cause the serial communications to become erratic at baud rates above 38,400 baud.

ProLinx gateways with 4 serial ports have a separate serial interface board for serial Ports 1, 2, and 3. These serial ports are buffered, and can handle communications up to 115,200 baud.

7.1.4 Parity

None, Odd, Even

Parity is a simple error checking algorithm used in serial communication. This parameter specifies the type of parity checking to use.

All devices communicating through this port must use the same parity setting.

7.1.5 Data Bits

7 or 8

This parameter sets the number of data bits for each word used by the protocol. All devices communicating through this port must use the same number of data bits.

7.1.6 Stop Bits

1 or 2

Stop bits signal the end of a character in the data stream. For most applications, use one stop bit. For slower devices that require more time to re-synchronize, use two stop bits.

All devices communicating through this port must use the same number of stop bits.

7.1.7 RTS On

0 to 65535 milliseconds

This parameter sets the number of milliseconds to delay after *Ready To Send* (RTS) is asserted before data will be transmitted.

7.1.8 RTS Off

0 to 65535 milliseconds

This parameter sets the number of milliseconds to delay after the last byte of data is sent before the RTS modem signal will be set low.

7.1.9 Minimum Response Delay

0 to 65535

This parameter is used only when the port is configured as a slave. It sets the number of milliseconds to wait before responding to a command received on the port from a remote Master. This delay is sometimes required to accommodate slower Master devices.

7.1.10 Use CTS Line

YES or NO

This parameter specifies if the Clear To Send (CTS) modem control line is to be used or not. If the parameter is set to **No**, the CTS line will not be monitored. If the parameter is set to **YES**, the CTS line will be monitored and must be high before the gateway will send data. Normally, this parameter is required when half-duplex modems are used for communication (2-wire). This procedure is commonly referred to as *hardware handshaking*.

7.1.11 Response Timeout

0 to 65535 milliseconds

This parameter sets the command response timeout period in 1 millisecond increments. This is the time that a port configured as a Master will wait for a response from the addressed slave before re-transmitting the command (Retries) or skipping to the next command in the Command List. The value to set depends on the communication network used and the expected response time (plus a little extra) of the slowest device on the network.

7.1.12 Retry Count

0 to 10

This parameter specifies the number of times a command will be retried if it fails.

7.2 [Modbus Slave Port x]

There are three slave ports on the Port Expander gateway. They all share the same Slave ID, which is configured as part of the Master Port settings. These three separate ports allow a single slave device (attached to the Master port) to be connected on three separate physical Modbus serial networks and to be polled by three separate Modbus Master devices.

7.2.1 *Enabled*

YES or NO

This flag specifies if the port on the gateway will be utilized. If the parameter is set to **NO**, the port will not be used. If the parameter is set to **YES**, the port will be used.

7.2.2 *RS Interface*

0 = RS-232

1 = RS-485

2 = RS-422

This parameter is used when Ports 1, 2, and 3 are present to configure the electrical interface for these ports.

RS Interface selections for Port 0 are done with a jumper (electrical shunt) on the back of the gateway.

7.2.3 *Protocol*

RTU

ASCII

This parameter specifies the Modbus protocol version to be used on the port. Valid Protocols are RTU and ASCII.

7.2.4 Baud Rate

This is the baud rate to be used on the slave port. It must match the baud rate setting used by the remote Master serial network that will be connected to this slave port. Each port/network can use different baud rates, if required. It is not necessary to have all the ports on the gateway set to the same baud rate.

To maximize the port multiplexing capability of the gateway, it is recommended, if possible, that you set the baud rate on the Master port and on the attached slave as high as the slave can accommodate, up to 38,400; then set the baud rate for the gateway slave ports and remote Master networks to a lower setting. This will help prevent multiple Masters from overloading the response capability of the single slave and the gateway.

Enter the baud rate as a value, shown in this list.

Baud Rate	Parameter Value
110	110
150	150
300	300
600	600
1200	12 OR 1200
2400	24 OR 2400
4800	48 OR 4800
9600	96 OR 9600
19,200	19, 192 OR 19200
38,400	38, 384 OR 38400
57,600	57 OR 576
115,200	115 OR 1152

Note: On all ProLinx gateways, data from the application port on the main board, serial Port 0, is not buffered. Packets go directly to and from the serial chipset to the processor. This has the potential to cause the serial communications to become erratic at baud rates above 38,400 baud.

ProLinx gateways with 4 serial ports have a separate serial interface board for serial Ports 1, 2, and 3. These serial ports are buffered, and can handle communications up to 115,200 baud.

7.2.5 Parity

None, Odd, Even

Parity is a simple error checking algorithm used in serial communication. This parameter specifies the type of parity checking to use.

All devices communicating through this port must use the same parity setting.

7.2.6 Data Bits

7 or 8

This parameter sets the number of data bits for each word used by the protocol. All devices communicating through this port must use the same number of data bits.

7.2.7 Stop Bits

1 or 2

Stop bits signal the end of a character in the data stream. For most applications, use one stop bit. For slower devices that require more time to re-synchronize, use two stop bits.

All devices communicating through this port must use the same number of stop bits.

7.2.8 RTS On

0 to 65535 milliseconds

This parameter sets the number of milliseconds to delay after *Ready To Send* (RTS) is asserted before data will be transmitted.

7.2.9 RTS Off

0 to 65535 milliseconds

This parameter sets the number of milliseconds to delay after the last byte of data is sent before the RTS modem signal will be set low.

7.2.10 Minimum Response Delay

0 to 65535

This parameter is used only when the port is configured as a slave. It sets the number of milliseconds to wait before responding to a command received on the port from a remote Master. This delay is sometimes required to accommodate slower Master devices.

7.2.11 Use CTS Line

YES or NO

This parameter specifies if the Clear To Send (CTS) modem control line is to be used or not. If the parameter is set to **No**, the CTS line will not be monitored. If the parameter is set to **YES**, the CTS line will be monitored and must be high before the gateway will send data. Normally, this parameter is required when half-duplex modems are used for communication (2-wire). This procedure is commonly referred to as *hardware handshaking*.

7.2.12 Use Guard Band Timer

YES or NO

This parameter specifies if the Guard Band feature (intercharacter delay monitoring and control) is to be used or not. Normally, this feature will not be needed. It is used only when the port is set for Modbus RTU mode as a slave device in a multi-drop, multi-node network (RS-422 or RS-485 wiring). If the remote Master experiences multiple, successive failures when trying to communicate with the gateway on a multi-drop network, setting a small amount of Guard Band Time will usually completely eliminate the failures.

When the port is configured as a slave and when multiple slave nodes exist on the network (multi-drop), without the Guard Band Timer, the gateway might not be able to properly discriminate between a response message returned by another slave and a command from the Master to which it should respond. The *Guard Band Timeout* parameter allows you to set a small delay which can help eliminate communication errors on the Modbus network.

To enable the Guard Band Timeout parameter:

- 1** First of all, enable the the *Use Guard Band Timer* parameter.

Use Guard Band Timer = Y

- 2** Then, set the *Guard Band Timeout*.

Guard Band Timeout = 0

This configuration will set the gateway to use a pre-defined time interval for Modbus message recognition calculated for each baud rate. This should provide optimal performance for most applications.

If you still observe a considerable number of communication errors, try to increase the *Guard Band Timeout* value until the network performance is improved.

7.2.13 Guard Band Timeout

0 to 65535

A value of 1 to 65535 sets the time in milliseconds. When this period has elapsed and no further data is received, the gateway will handle the last received data sequence as a Modbus message to be processed.

This parameter specifies the *Guard Band Timeout* between packets in multi-drop slave mode. A value of 0 uses the default time, which is roughly equivalent to the time it would take to send 3.5 bytes at the selected baud rate. This is the delay value specified by the Modbus RTU protocol specifications. The following *Guard Band Timeout* values will be automatically used by the gateway when *Use Guard Band* is set to **YES** and the *Guard Band Timeout* is set to the default value of zero (**0**).

Baud	Guard Band Timer (ms)
110	350
150	256
300	128
600	64
1200	32
2400	16
4800	8
9600	4
19200	2
28800	2
38400	2
57600	1
115200	1

7.2.14 Request Timeout

Use this default timeout.

7.2.15 Busy Timeout

0 to 65535

This parameter specifies the number of milliseconds to wait for the DF1 Master port to become available. If the DF1 Master port is processing a command list request, the busy flag will be set. The flag will remain busy until the communication transaction is complete. If the port does not become available before the busy timeout expires, the message will be aborted. If the Master port becomes available before this timeout expires, the request will be routed to the Master port.

7.2.16 Busy Timeout

0 to 65535

This parameter specifies the number of milliseconds to wait for the Modbus Master port to become available. If the Modbus Master port is processing a command received on another slave port, the busy flag will be set. The flag will remain set until the communication transaction is complete. If the port does not become available before the busy timeout expires, the message will be aborted and an exception response (busy, try again later) will be returned to the originating Master. If the Master port becomes available before this timeout expires, the request will be routed to the Master port.

8 Reference

In This Chapter

- ❖ Modbus Protocol Specification 74
- ❖ Moving Data using the Data Map 87

8.1 Modbus Protocol Specification

The following pages give additional reference information regarding the Modbus protocol commands supported by the MCM.

8.1.1 Read Coil Status (Function Code 01)

Query

This function allows the user to obtain the ON/OFF status of logic coils used to control discrete outputs from the addressed Slave only. Broadcast mode is not supported with this function code. In addition to the Slave address and function fields, the message requires that the information field contain the initial coil address to be read (Starting Address) and the number of locations that will be interrogated to obtain status data.

The addressing allows up to 2000 coils to be obtained at each request; however, the specific Slave device may have restrictions that lower the maximum quantity. The coils are numbered from zero; (coil number 1 = zero, coil number 2 = one, coil number 3 = two, and so on).

The following table is a sample read output status request to read coils 0020 to 0056 from Slave device number 11.

Adr	Func	Data Start Pt Hi	Data Start Pt Lo	Data # Of Pts Ho	Data # Of Pts Lo	Error Check Field
11	01	00	13	00	25	CRC

Response

An example response to Read Coil Status is as shown in Figure C2. The data is packed one bit for each coil. The response includes the Slave address, function code, quantity of data characters, the data characters, and error checking. Data will be packed with one bit for each coil (1 = ON, 0 = OFF). The low order bit of the first character contains the addressed coil, and the remainder follow. For coil quantities that are not even multiples of eight, the last characters will be filled in with zeros at high order end. The quantity of data characters is always specified as quantity of RTU characters, that is, the number is the same whether RTU or ASCII is used.

Because the Slave interface device is serviced at the end of a controller's scan, data will reflect coil status at the end of the scan. Some Slaves will limit the quantity of coils provided each scan; thus, for large coil quantities, multiple PC transactions must be made using coil status from sequential scans.

Adr	Func	Byte Count	Data Coil Status 20 to 27	Data Coil Status 28 to 35	Data Coil Status 36 to 43	Data Coil Status 44 to 51	Data Coil Status 52 to 56	Error Check Field
11	01	05	CD	6B	B2	OE	1B	CRC

The status of coils 20 to 27 is shown as CD(HEX) = 1100 1101 (Binary). Reading left to right, this shows that coils 27, 26, 23, 22, and 20 are all on. The other coil data bytes are decoded similarly. Due to the quantity of coil statuses requested, the last data field, which is shown 1B (HEX) = 0001 1011 (Binary), contains the status of only 5 coils (52 to 56) instead of 8 coils. The 3 left most bits are provided as zeros to fill the 8-bit format.

8.1.2 Read Input Status (Function Code 02)

Query

This function allows the user to obtain the ON/OFF status of discrete inputs in the addressed Slave PC Broadcast mode is not supported with this function code. In addition to the Slave address and function fields, the message requires that the information field contain the initial input address to be read (Starting Address) and the number of locations that will be interrogated to obtain status data.

The addressing allows up to 2000 inputs to be obtained at each request; however, the specific Slave device may have restrictions that lower the maximum quantity. The inputs are numbered form zero; (input 10001 = zero, input 10002 = one, input 10003 = two, and so on, for a 584).

The following table is a sample read input status request to read inputs 10197 to 10218 from Slave number 11.

Adr	Func	Data Start Pt Hi	Data Start Pt Lo	Data #of Pts Hi	Data #of Pts Lo	Error Check Field
11	02	00	C4	00	16	CRC

Response

An example response to Read Input Status is as shown in Figure C4. The data is packed one bit for each input. The response includes the Slave address, function code, quantity of data characters, the data characters, and error checking. Data will be packed with one bit for each input (1=ON, 0=OFF). The lower order bit of the first character contains the addressed input, and the remainder follow. For input quantities that are not even multiples of eight, the last characters will be filled in with zeros at high order end. The quantity of data characters is always specified as a quantity of RTU characters, that is, the number is the same whether RTU or ASCII is used.

Because the Slave interface device is serviced at the end of a controller's scan, data will reflect input status at the end of the scan. Some Slaves will limit the quantity of inputs provided each scan; thus, for large coil quantities, multiple PC transactions must be made using coil status for sequential scans.

Adr	Func	Byte Count	Data Discrete Input 10197 to 10204	Data Discrete Input 10205 to 10212	Data Discrete Input 10213 to 10218	Error Check Field
11	02	03	AC	DB	35	CRC

The status of inputs 10197 to 10204 is shown as AC (HEX) = 10101 1100 (binary). Reading left to right, this show that inputs 10204, 10202, and 10199 are all on. The other input data bytes are decoded similar.

Due to the quantity of input statuses requested, the last data field which is shown as 35 HEX = 0011 0101 (binary) contains the status of only 6 inputs (10213 to 10218) instead of 8 inputs. The two left-most bits are provided as zeros to fill the 8-bit format.

8.1.3 Read Holding Registers (Function Code 03)

Query

Read Holding Registers (03) allows the user to obtain the binary contents of holding registers 4xxx in the addressed Slave. The registers can store the numerical values of associated timers and counters which can be driven to external devices. The addressing allows up to 125 registers to obtained at each request; however, the specific Slave device may have restriction that lower this maximum quantity. The registers are numbered form zero (40001 = zero, 40002 = one, and so on). The broadcast mode is not allowed.

The example below reads registers 40108 through 40110 from Slave 584 number 11.

Ad r	Fun c	Data Start Reg Hi	Data Start Reg Lo	Data #of Regs Hi	Data #of Regs Lo	Error Check Field
11	03	00	6B	00	03	CRC

Response

The addressed Slave responds with its address and the function code, followed by the information field. The information field contains 1 byte describing the quantity of data bytes to be returned. The contents of the registers requested (DATA) are two bytes each, with the binary content right justified within each pair of characters. The first byte includes the high order bits and the second, the low order bits.

Because the Slave interface device is normally serviced at the end of the controller's scan, the data will reflect the register content at the end of the scan. Some Slaves will limit the quantity of register content provided each scan; thus for large register quantities, multiple transmissions will be made using register content from sequential scans.

In the example below, the registers 40108 to 40110 have the decimal contents 555, 0, and 100 respectively.

Adr	Func	ByteCn t	Hi Data	Lo Data	Hi Data	Lo Data	Hi Data	Lo Data	Error Check Field
11	03	06	02	2B	00	00	00	64	CRC

8.1.4 Read Input Registers (Function Code 04)

Query

Function code 04 obtains the contents of the controller's input registers at addresses 3xxxx. These locations receive their values from devices connected to the I/O structure and can only be referenced, not altered from within the controller. The addressing allows up to 125 registers to be obtained at each request; however, the specific Slave device may have restrictions that lower this maximum quantity. The registers are numbered for zero (30001 = zero, 30002 = one, and so on). Broadcast mode is not allowed.

The example below requests the contents of register 3009 in Slave number 11.

Adr	Func	Data Start Reg Hi	Data Start Reg Lo	Data #of Regs Hi	Data #of Regs Lo	Error Check Field
11	04	00	08	00	01	CRC

Response

The addressed Slave responds with its address and the function code followed by the information field. The information field contains 1 byte describing the quantity of data bytes to be returned. The contents of the registers requested (DATA) are 2 bytes each, with the binary content right justified within each pair of characters. The first byte includes the high order bits and the second, the low order bits.

Because the Slave interface is normally serviced at the end of the controller's scan, the data will reflect the register content at the end of the scan. Each PC will limit the quantity of register contents provided each scan; thus for large register quantities, multiple PC scans will be required, and the data provided will be form sequential scans.

In the example below the register 3009 contains the decimal value 0.

Adr	Func	Byte Count	Data Input Reg Hi	Data Input Reg Lo	Error Check Field
11	04	02	00	00	E9

8.1.5 Force Single Coil (Function Code 05)

Query

This message forces a single coil either ON or OFF. Any coil that exists within the controller can be forced to either state (ON or OFF). However, because the controller is actively scanning, unless the coil is disabled, the controller can also alter the state of the coil. Coils are numbered from zero (coil 0001 = zero, coil 0002 = one, and so on). The data value 65,280 (FF00 HEX) will set the coil ON and the value zero will turn it OFF; all other values are illegal and will not affect that coil.

The use of Slave address 00 (Broadcast Mode) will force all attached Slaves to modify the desired coil.

Note: Functions 5, 6, 15, and 16 are the only messages that will be recognized as valid for broadcast.

The example below is a request to Slave number 11 to turn ON coil 0173.

Adr	Func	Data Coil # Hi	Data Coil # Lo	Data On/off Ind	Data	Error Check Field
11	05	00	AC	FF	00	CRC

Response

The normal response to the Command Request is to re-transmit the message as received after the coil state has been altered.

Adr	Func	Data Coil # Hi	Data Coil # Lo	Data On/ Off	Data	Error Check Field
11	05	00	AC	FF	00	CRC

The forcing of a coil via MODBUS function 5 will be accomplished regardless of whether the addressed coil is disabled or not (*In ProSoft products, the coil is only affected if the necessary ladder logic is implemented*).

Note: The Modbus protocol does not include standard functions for testing or changing the DISABLE state of discrete inputs or outputs. Where applicable, this may be accomplished via device specific Program commands (*In ProSoft products, this is only accomplished through ladder logic programming*).

Coils that are reprogrammed in the controller logic program are not automatically cleared upon power up. Thus, if such a coil is set ON by function Code 5 and (even months later), an output is connected to that coil, the output will be "hot".

8.1.6 Preset Single Register (Function Code 06)

Query

Function (06) allows the user to modify the contents of a holding register. Any holding register that exists within the controller can have its contents changed by this message. However, because the controller is actively scanning, it also can alter the content of any holding register at any time. The values are provided in binary up to the maximum capacity of the controller unused high order bits must be set to zero. When used with Slave address zero (Broadcast mode) all Slave controllers will load the specified register with the contents specified.

Note Functions 5, 6, 15, and 16 are the only messages that will be recognized as valid for broadcast.

Adr	Func	Data Start Reg Hi	Data Start Reg Lo	Data #of Regs Hi	Data #of Regs Lo	Error Check Field
11	06	00	01	00	03	CRC

Response

The response to a preset single register request is to re-transmit the query message after the register has been altered.

Adr	Func	Data Reg Hi	Data Reg Lo	Data Input Reg Hi	Data Input Reg Lo	Error Check Field
11	06	00	01	00	03	CRC

8.1.7 Diagnostics (Function Code 08)

MODBUS function code 08 provides a series of tests for checking the communication system between a Master device and a slave, or for checking various internal error conditions within a slave.

The function uses a two-byte sub-function code field in the query to define the type of test to be performed. The slave echoes both the function code and sub-function code in a normal response. Some of the diagnostics commands cause data to be returned from the remote device in the data field of a normal response.

In general, issuing a diagnostic function to a remote device does not affect the running of the user program in the remote device. Device memory bit and register data addresses are not accessed by the diagnostics. However, certain functions can optionally reset error counters in some remote devices.

A server device can, however, be forced into 'Listen Only Mode' in which it will monitor the messages on the communications system but not respond to them. This can affect the outcome of your application program if it depends upon any further exchange of data with the remote device. Generally, the mode is forced to remove a malfunctioning remote device from the communications system.

Sub-function Codes Supported

Only Sub-function 00 is supported by the MCM gateway.

00 Return Query Data

The data passed in the request data field is to be returned (looped back) in the response. The entire response message should be identical to the request.

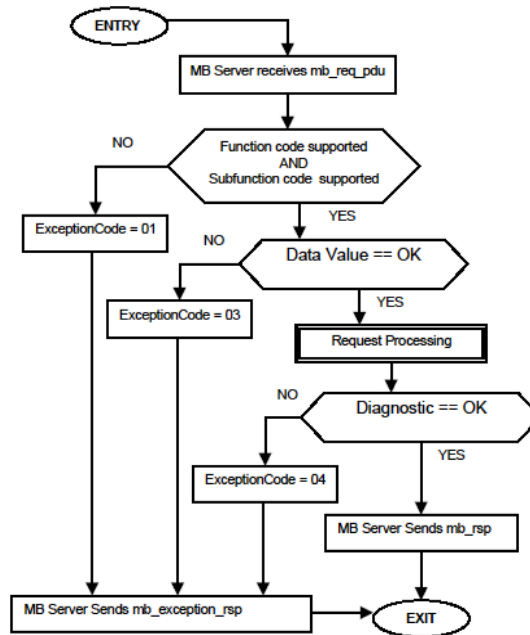
Sub-function	Data Field (Request)	Data Field (Response)
00 00	Any	Echo Request Data

Example and State Diagram

Here is an example of a request to remote device to Return Query Data. This uses a sub-function code of zero (00 00 hex in the two-byte field). The data to be returned is sent in the two-byte data field (A5 37 hex).

Request		Response	
Field Name	(Hex)	Field Name	(Hex)
Function	08	Function	08
Sub-function Hi	00	Sub-function Hi	00
Sub-function Lo	00	Sub-function Lo	00
Data Hi	A5	Data Hi	A5
Data Lo	37	Data Lo	27

The data fields in responses to other kinds of queries could contain error counts or other data requested by the sub-function code.



8.1.8 Force Multiple Coils (Function Code 15)

Query

This message forces each coil in a consecutive block of coils to a desired ON or OFF state. Any coil that exists within the controller can be forced to either state (ON or OFF). However, because the controller is actively scanning, unless the coils are disabled, the controller can also alter the state of the coil. Coils are numbered from zero (coil 00001 = zero, coil 00002 = one, and so on). The desired status of each coil is packed in the data field, one bit for each coil (1= ON, 0= OFF). The use of Slave address 0 (Broadcast Mode) will force all attached Slaves to modify the desired coils.

Note: Functions 5, 6, 15, and 16 are the only messages (other than Loopback Diagnostic Test) that will be recognized as valid for broadcast.

The following example forces 10 coils starting at address 20 (13 HEX). The two data fields, CD =1100 and 00 = 0000 000, indicate that coils 27, 26, 23, 22, and 20 are to be forced on.

Adr	Func	Hi Add	Lo Add	Quantity	Byte Cnt	Data Coil Status 20 to 27	Data Coil Status 28 to 29	Error Check Field
11	0F	00	13	00	0A	02	CD	00 CRC

Response

The normal response will be an echo of the Slave address, function code, starting address, and quantity of coils forced.

Adr	Func	Hi Addr	Lo Addr	Quantity	Error Check Field
11	0F	00	13	00	0A CRC

The writing of coils via Modbus function 15 will be accomplished regardless of whether the addressed coils are disabled or not.

Coils that are unprogrammed in the controller logic program are not automatically cleared upon power up. Thus, if such a coil is set ON by function code 15 and (even months later) an output is connected to that coil, the output will be hot.

8.1.9 Preset Multiple Registers (Function Code 16)

Query

Holding registers existing within the controller can have their contents changed by this message (a maximum of 60 registers). However, because the controller is actively scanning, it also can alter the content of any holding register at any time. The values are provided in binary up to the maximum capacity of the controller (16-bit for the 184/384 and 584); unused high order bits must be set to zero.

Note: Function codes 5, 6, 15, and 16 are the only messages that will be recognized as valid for broadcast.

Adr	Func	Hi Add	Lo Add	Quantity	Byte Cnt	Hi Data	Lo Data	Hi Data	Lo Data	Error Check Field
11	10	00	87	00	02 04	00	0A	01	02	CRC

Response

The normal response to a function 16 query is to echo the address, function code, starting address and number of registers to be loaded.

Adr	Func	Hi Addr	Lo Addr	Quantity	Error Check Field
11	10	00	87	00 02	56

8.1.10 Modbus Exception Responses

When a Modbus Master sends a request to a Slave device, it expects a normal response. One of four possible events can occur from the Master's query:

- If the server device receives the request without a communication error, and can handle the query normally, it returns a normal response.
- If the server does not receive the request due to a communication error, no response is returned. The Master program will eventually process a timeout condition for the request.
- If the server receives the request, but detects a communication error (parity, LRC, CRC, ...), no response is returned. The Master program will eventually process a timeout condition for the request.
- If the server receives the request without a communication error, but cannot handle it (for example, if the request is to read a non-existent output or register), the server will return an exception response informing the Master of the nature of the error.

The exception response message has two fields that differentiate it from a normal response:

Function Code Field: In a normal response, the server echoes the function code of the original request in the function code field of the response. All function codes have a most-significant bit (MSB) of 0 (their values are all below 80 hexadecimal). In an exception response, the server sets the MSB of the function code to 1. This makes the function code value in an exception response exactly 80 hexadecimal higher than the value would be for a normal response.

With the function code's MSB set, the Master's application program can recognize the exception response and can examine the data field for the exception code.

Data Field: In a normal response, the server may return data or statistics in the data field (any information that was requested in the request). In an exception response, the server returns an exception code in the data field. This defines the server condition that caused the exception.

The following table shows an example of a Master request and server exception response.

Request		Response	
Field Name	(Hex)	Field Name	(Hex)
Function	01	Function	81
Starting Address Hi	04	Exception Code	02
Starting Address Lo	A1		
Quantity of Outputs Hi	00		
Quantity of Outputs Lo	01		

In this example, the Master addresses a request to server device. The function code (01) is for a Read Output Status operation. It requests the status of the output at address 1245 (04A1 hex). Note that only that one output is to be read, as specified by the number of outputs field (0001).

If the output address is non-existent in the server device, the server will return the exception response with the exception code shown (02). This specifies an illegal data address for the Slave.

Modbus Exception Codes

Code	Name	Meaning
01	Illegal Function	The function code received in the query is not an allowable action for the Slave. This may be because the function code is only applicable to newer devices, and was not implemented in the unit selected. It could also indicate that the Slave is in the wrong state to process a request of this type, for example because it is unconfigured and is being asked to return register values.
02	Illegal Data Address	The data address received in the query is not an allowable address for the Slave. More specifically, the combination of reference number and transfer length is invalid. For a controller with 100 registers, a request with offset 96 and length 4 would succeed; a request with offset 96 and length 5 will generate exception 02.
03	Illegal Data Value	A value contained in the query data field is not an allowable value for Slave. This indicates a fault in the structure of the remainder of a complex request, such as that the implied length is incorrect. It specifically does not mean that a data item submitted for storage in a register has a value outside the expectation of the application program, because the Modbus protocol is unaware of the significance of any particular value of any particular register.
04	Slave Device Failure	An unrecoverable error occurred while the Slave was attempting to perform the requested action.
05	Acknowledge	Specialized use in conjunction with programming commands. The Slave has accepted the request and is processing it, but a long duration of time will be required to do so. This response is returned to prevent a timeout error from occurring in the Master. The Master can next issue a poll program complete message to determine if processing is completed.
06	Slave Device Busy	Specialized use in conjunction with programming commands. The Slave is engaged in processing a long-duration program command. The Master should retransmit the message later when the Slave is free.
08	Memory Parity Error	Specialized use in conjunction with function codes 20 and 21 and reference type 6, to indicate that the extended file area failed to pass a consistency check. The Slave attempted to read record file, but detected a parity error in the memory. The Master can retry the request, but service may be required on the Slave device.

Code	Name	Meaning
0a	Gateway Path Unavailable	Specialized use in conjunction with gateways, indicates that the gateway was unable to allocate an internal communication path from the input port to the output port for processing the request. Usually means that the gateway is misconfigured or overloaded.
0b	Gateway Target Device Failed To Respond	Specialized use in conjunction with gateways, indicates that no response was obtained from the target device. Usually means that the device is not present on the network.

8.2 Moving Data using the Data Map

The following is an example of the Data Map section of the configuration file.

```
# This section moves data within the database to concentrate information
# for simpler data requests and control. The From Address (source of data) specifies the start
# database location to copy the number of registers set by Register Count to the
# specified To Address (destination of data). When the data is copied, the order of the bytes can be altered
# using the Swap Code field as follows:
#
# SWAP CODE      DEFINITION
0 Bytes left in original order (1234 -> 1234)
1 Words are swapped (1234 -> 3412)
2 Words and bytes are swapped (1234 -> 4321)
3 Bytes in each word are swapped (1234 -> 2143)
[DATA MAP]
# From   To     Register  Swap  Delay
# Address Address  Count    Code  Preset
START
 4000   1000    9        0     1000
 6300   1010   20        0     1002
 4400   1030   20        0     1003
END
```

The *Delay Preset* parameter determines how often the data transfer will happen. The value entered here is in number of internal firmware scan cycles. Since the actual amount of time required for an internal firmware scan varies based on the protocols involved and application-specific configuration variations, there is no direct relationship between scan counts and time interval. Therefore, if you need to move data based on specific timing, you will have to determine this relationship experimentally by trying different *Delay Preset* values and monitoring the time between data moves, until you discover an acceptable setting for your application.

9 Support, Service & Warranty

In This Chapter

- ❖ Contacting Technical Support 89
- ❖ Return Material Authorization (RMA) Policies and Conditions..... 90
- ❖ LIMITED WARRANTY..... **Error! Bookmark not defined.**

Contacting Technical Support

ProSoft Technology, Inc. (ProSoft) is committed to providing the most efficient and effective support possible. Before calling, please gather the following information to assist in expediting this process:

- 1 Product Version Number
- 2 System architecture
- 3 Network details

If the issue is hardware related, we will also need information regarding:

- 1 Module configuration and associated ladder files, if any
- 2 Module operation and any unusual behavior
- 3 Configuration/Debug status information
- 4 LED patterns
- 5 Details about the serial, Ethernet or fieldbus devices interfaced to the module, if any.

Note: *For technical support calls within the United States, an after-hours answering system allows 24-hour/7-days-a-week pager access to one of our qualified Technical and/or Application Support Engineers. Detailed contact information for all our worldwide locations is available on the following page.*

Internet	Web Site: www.prosoft-technology.com/support E-mail address: support@prosoft-technology.com
Asia Pacific (location in Malaysia)	Tel: +603.7724.2080, E-mail: asiapc@prosoft-technology.com Languages spoken include: Chinese, English
Asia Pacific (location in China)	Tel: +86.21.5187.7337 x888, E-mail: asiapc@prosoft-technology.com Languages spoken include: Chinese, English
Europe (location in Toulouse, France)	Tel: +33 (0) 5.34.36.87.20, E-mail: support.EMEA@prosoft-technology.com Languages spoken include: French, English
Europe (location in Dubai, UAE)	Tel: +971-4-214-6911, E-mail: mea@prosoft-technology.com Languages spoken include: English, Hindi
North America (location in California)	Tel: +1.661.716.5100, E-mail: support@prosoft-technology.com Languages spoken include: English, Spanish
Latin America (Oficina Regional)	Tel: +1-281-2989109, E-Mail: latinam@prosoft-technology.com Languages spoken include: Spanish, English
Latin America (location in Puebla, Mexico)	Tel: +52-222-3-99-6565, E-mail: soporte@prosoft-technology.com Languages spoken include: Spanish
Brasil (location in Sao Paulo)	Tel: +55-11-5083-3776, E-mail: brasil@prosoft-technology.com Languages spoken include: Portuguese, English

9.1 Warranty Information

For complete details regarding ProSoft Technology's TERMS & CONDITIONS OF SALE, WARRANTY, SUPPORT, SERVICE AND RETURN MATERIAL AUTHORIZATION INSTRUCTIONS please see the documents on the Product CD/DVD or go to www.prosoft-technology.com/warranty
Documentation is subject to change without notice

Index

I

[Modbus Master Port] • 62
[Modbus Port X Commands] • 32, 33
[Modbus Port x] • 25
[Modbus Slave Port x] • 65

O

00 Return Query Data • 78

A

Allocation of Risks • 92

B

Base Module LEDs • 58
Baud Rate • 27, 63, 66
Bit Input Offset • 29
Busy Timeout • 69, 70

C

Command Control Reg • 32
Command List Error Codes • 49
Communication Port Cables • 51
Connecting Power to the Unit • 13
Contacting Technical Support • 85, 87
Controlling Law and Severability • 92

D

Data Bits • 27, 63, 67
Diagnostics (Function Code 08) • 78
Disclaimer of all Other Warranties • 91
Disclaimer Regarding High Risk Activities • 90

E

Enable • 32, 34
Enabled • 25, 65
Error Codes • 41, 43, 44, 47
Error Delay Counter • 31, 45
Example and State Diagram • 79

F

Float Flag • 22, 25
Float Offset • 22, 26
Float Start • 22, 26
Force Multiple Coils (Function Code 15) • 80
Force Single Coil (Function Code 05) • 76
Functional Overview • 17

G

Guard Band Timeout • 31, 69

H

Holding Register Offset • 29
How to Contact Us • 2

I

Important Installation Instructions • 3
Installing ProSoft Configuration Builder
Software • 14
Intellectual Property Indemnity • 91
Internal Address • 34
Internal Slave ID • 28, 62

L

LED Indicators • 57
LEDs for Serial Ports • 59
Limitation of Remedies ** • 92
LIMITED WARRANTY • 87, 89

M

Markings • 4
Master Port
 Command List Errors • 43, 44
 Modbus Slave List Status • 45, 46
MB Address in Device • 37
MCM Protocol Configuration • 23
Minimum Command Delay • 31
Minimum Response Delay • 28, 64, 67
Modbus Configuration Error Word • 50
Modbus Error and Status Data Area
 Addresses • 41
Modbus Error Codes • 47
Modbus Exception Codes • 83
Modbus Exception Responses • 47, 82
Modbus Function • 36
Modbus Master/Slave Port Specifications • 20
Modbus Master/Slave Port(s) • 18
Modbus Port Access to Database • 22
Modbus Port Expander (4102/5102-MBS3-
 MBM) • 61
Modbus Ports
 Error and Status • 41, 44
Modbus Protocol Specification • 72
Module Communication Error Codes • 49
Module Internal Database • 21
Mounting the Gateway on a DIN-rail • 12
Moving Data using the Data Map • 41, 84

N

No Other Warranties • 92

O

Output Offset • 29

P

Package Contents • 11
Parity • 27, 63, 67

Pinouts • 3, 30
Poll Interval • 34
Port 0 Command Error List Layout • 44
Port 0 Slave List Status Layout • 46
Port 0, 1, 2, 3
 RS-232 - DTE to DCE Modem
 Connection • 54
 RS-232 - Null Modem (DTE with
 Hardware Handshaking) • 52
 RS-232 - Null Modem (DTE without
 Hardware Handshaking) • 53
 RS-422 Interface Connections • 55, 56
 RS-485 Interface Connections • 56
Preset Multiple Registers (Function Code 16)
 • 81
Preset Single Register (Function Code 06) •
 77
ProLinx Gateways with Ethernet Ports • 3
ProLinx® Products Warnings • 3
ProSoft Technology® Product
 Documentation • 2
Protocol • 26, 62, 65

R

Read Coil Status (Function Code 01) • 72
Read Holding Registers (Function Code 03) •
 74
Read Input Registers (Function Code 04) • 75
Read Input Status (Function Code 02) • 73
Reference • 71
Reg Count • 35
Request Timeout • 69
Response Timeout • 28, 64
Retry Count • 28, 64
Return Material Authorization (RMA) Policies
 and Conditions • 87
Returning Any Product • 87
Returning Units Out of Warranty • 88
Returning Units Under Warranty • 87
RS Interface • 25, 65
RTS Off • 28, 64, 67
RTS On • 28, 64, 67

S

Serial Port Modbus Error and Status Data •
 39, 59
Serial Port Specifications • 19
Slave Address • 36
Slave Port
 General Port STATE and COMM STATE
 Status • 42
Start Here • 9
Stop Bits • 27, 64, 67
Sub-function Codes Supported • 78
Support, Service & Warranty • 85
Swap Code • 35
System Requirements • 10

T

Time Limit for Bringing Suit • 92
To order a ProLinx Plus gateway with the -
 WEB option • 3
To upgrade a previously purchased Series C
 model • 3
Type • 25

U

Use CTS Line • 28, 52, 53, 54, 64, 67
Use Guard Band Timer • 30, 68
Using the Online Help • 15

V

Viewing Error and Status Data • 40

W

What Is Covered By This Warranty • 89, 91
What Is Not Covered By This Warranty • 90
Word Input Offset • 29

Y

Your Feedback Please • 2