



User Manual for
HE800ETN200 and HE800ETN300

SmartStack™

Ethernet Modules

PREFACE

This manual explains how to use SmartStack Ethernet Modules (ETN200 and ETN300).

Copyright (C) 2005 Horner APG, LLC, 640 North Sherman Drive Indianapolis, Indiana 46201. All rights reserved. No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language or computer language, in any form by any means, electronic, mechanical, magnetic, optical, chemical, manual or otherwise, without the prior agreement and written permission of Horner APG, LLC.

All software described in this document or media is also copyrighted material subject to the terms and conditions of the Horner Software License Agreement.

Information in this document is subject to change without notice and does not represent a commitment on the part of Horner APG, LLC.

Cscape, SmartStack, and CsCAN are trademarks of Horner APG, LLC.

Ethernet/IP is a trademark of the Open DeviceNet Vendor Association (ODVA), Inc.

Ethernet Global Data, EGD, Service Request Transfer Protocol, SRTP, CIMPLICITY and Series 90 are trademarks of GE Fanuc Automation North America, Inc.

Internet Explorer is a trademark of Microsoft Corporation.

KEPware is a trademark of KEPware, Inc.

Firefox is a trademark of the Mozilla Foundation.

FileX, NetX and ThreadX are trademarks of Express Logic, Inc.

For User Manual updates, contact Horner APG Technical Support Division, at (317) 916-4274 or visit our website at www.heapg.com.

LIMITED WARRANTY AND LIMITATION OF LIABILITY

Horner APG, LLC ("HE-APG") warrants to the original purchaser that the SmartStack Ethernet Module manufactured by HE-APG is free from defects in material and workmanship under normal use and service. The obligation of HE-APG under this warranty shall be limited to the repair or exchange of any part or parts which may prove defective under normal use and service within two (2) years from the date of manufacture or eighteen (18) months from the date of installation by the original purchaser whichever occurs first, such defect to be disclosed to the satisfaction of HE-APG after examination by HE-APG of the allegedly defective part or parts. THIS WARRANTY IS EXPRESSLY IN LIEU OF ALL OTHER WARRANTIES EXPRESSED OR IMPLIED INCLUDING THE WARRANTIES OF MERCHANTABILITY AND FITNESS FOR USE AND OF ALL OTHER OBLIGATIONS OR LIABILITIES AND HE-APG NEITHER ASSUMES, NOR AUTHORIZES ANY OTHER PERSON TO ASSUME FOR HE-APG, ANY OTHER LIABILITY IN CONNECTION WITH THE SALE OF THIS SmartStack Ethernet Module. THIS WARRANTY SHALL NOT APPLY TO THIS SmartStack Ethernet Module OR ANY PART THEREOF THAT HAS BEEN SUBJECT TO ACCIDENT, NEGLIGENCE, ALTERATION, ABUSE, OR MISUSE. HE-APG MAKES NO WARRANTY WHATSOEVER IN RESPECT TO ACCESSORIES OR PARTS NOT SUPPLIED BY HE-APG. THE TERM "ORIGINAL PURCHASER", AS USED IN THIS WARRANTY, SHALL BE DEEMED TO MEAN THAT PERSON FOR WHOM THE SmartStack Ethernet Module IS ORIGINALLY INSTALLED. THIS WARRANTY SHALL APPLY ONLY WITHIN THE BOUNDARIES OF THE CONTINENTAL UNITED STATES.

In no event, whether as a result of breach of contract, warranty, tort (including negligence) or otherwise, shall HE-APG or its suppliers be liable of any special, consequential, incidental or penal damages including, but not limited to, loss of profit or revenues, loss of use of the products or any associated equipment, damage to associated equipment, cost of capital, cost of substitute products, facilities, services or replacement power, down time costs, or claims of original purchaser's customers for such damages.

To obtain warranty service, return the product to your distributor with a description of the problem, proof of purchase, post paid, insured and in a suitable package.

ABOUT PROGRAMMING EXAMPLES

Any example programs and program segments in this manual or provided on accompanying media are included solely for illustrative purposes. Due to the many variables and requirements associated with any particular installation, Horner APG cannot assume responsibility or liability for actual use based on the examples and diagrams. It is the sole responsibility of the system designer utilizing the SmartStack Ethernet Module to appropriately design the end system, to appropriately integrate the SmartStack Ethernet Module and to make safety provisions for the end equipment as is usual and customary in industrial applications as defined in any codes or standards which apply.

Note: The programming examples shown in this manual are for illustrative purposes only. Proper machine operation is the sole responsibility of the system integrator.

REVISIONS TO THIS MANUAL

Section 1.1:	Introduced ETN300 and inserted Table 1.1 and Table 1.2.
Section 1.2:	Inserted Ethernet Network Example section with revised Figure 1.1.
Section 1.3:	Revised Table 1.3 to add interoperability information for Ethernet/IP, FTP and HTTP Protocols.
Section 1.4:	Revised Table 1.4 to add Ethernet Cable Type Recommendation as well as Ethernet/IP, FTP and HTTP Protocol Specifications.
Section 2.3.1:	Added note regarding source port usage and revised Table 2.1 to document FTP and HTTP Protocol destination port usage.
Section 3.1:	Replaced Figure 3.5 with new Ethernet Module Configuration dialog image.
Section 6.8.1:	Replaced Figure 6.7 with new Ethernet Module Configuration dialog image.
Section 6.8.2:	Replaced Figure 6.14 with new Ethernet Module Configuration dialog image.
Section 8.1:	Revised 4 th paragraph and Table 8.1 to describe Expanded Modbus References, which can be used by a Modbus TCP Master to access extended registers that are supported by the new ETN300 Ethernet Module.
Chapters 1-9:	Changed most "ETN200" references to "Ethernet Module" since this User Manual now documents both ETN200 and ETN300 Ethernet Modules.
Chapter 10:	Added FTP Protocol chapter.
Chapter 11:	Added HTTP Protocol chapter.
Index:	Revised for completeness and uniformity.

Table of Contents

REVISIONS TO THIS MANUAL	5
CHAPTER 1: INTRODUCTION	9
1.1 Ethernet Module Overview	9
1.2 Ethernet Network Example.....	10
1.3 Ethernet System Requirements and Interoperability.....	10
1.4 Ethernet Module Specifications	11
1.5 Additional Technical Resources	11
CHAPTER 2: INSTALLATION	13
2.1 Built-In Ethernet Module	13
2.2 User-Installed Ethernet Module	13
2.3 Network Administrator Installation Notes.....	14
2.3.1 UDP and TCP Ports	14
2.3.2 EGD Unicast and Multicast IP Addressing	14
2.3.3 Internet Connectivity.....	15
2.3.4 IMPORTANT: Wireless Network Considerations	15
CHAPTER 3: GENERAL CONFIGURATION	17
3.1 Ethernet Module Configuration.....	17
3.2 Ethernet Module IP Address.....	21
3.2.1 Static IP Address.....	21
3.2.2 Static IP Address with CAN ID	21
3.2.3 IP Address from OCS Register	21
CHAPTER 4: CSCAN OVER ETHERNET PROTOCOL	23
4.1 CsCAN over Ethernet Overview	23
4.2 CsCAN over Ethernet Configuration.....	23
4.3 CsCAN over Ethernet Operation	24
4.4 CsCAN over Ethernet Downloading Precautions	24
4.4.1 How to Prevent Losing Communication	25
4.4.2 How to Recover from Lost Communication.....	25
4.5 CsCAN over Ethernet Security	26
CHAPTER 5: INTERNET CONTROL MESSAGE PROTOCOL (ICMP)	29
5.1 ICMP Overview.....	29
5.2 ICMP Configuration	29
5.3 ICMP Operation	30
CHAPTER 6: ETHERNET GLOBAL DATA PROTOCOL (EGD)	31
6.1 EGD Overview	31
6.2 EGD Terminology	31
6.3 EGD Configuration.....	32
6.4 EGD Produced Exchange Configuration	34
6.4.1 Creating EGD Produced Exchanges.....	35
6.4.2 Defining EGD Produced Exchange I/O Blocks	36
6.5 EGD Consumed Exchange Configuration.....	37
6.5.1 Creating EGD Consumed Exchanges.....	38
6.5.2 Defining EGD Consumed Exchange I/O Blocks	39
6.6 EGD Operation	40
6.7 EGD Status Words	40
6.7.1 EGD Produced Exchange Status Words	40
6.7.2 EGD Consumed Exchange Status Words	41
6.8 EGD Example 1	42
6.8.1 EGD Example 1 – Configuring Node 1	42
6.8.2 EGD Example 1 – Configuring Node 2	46
6.8.3 EGD Example 1 – Starting EGD Communication between Node 1 and Node 2	50
6.9 EGD Example 2	51

6.9.1	EGD Example 2 – Adding a Status Block	51
6.9.2	EGD Example 2 – Adding an OCS Timestamp Block.....	52
6.9.3	EGD Example 2 – Adding a Filler Block.....	53
CHAPTER 7: SERVICE REQUEST TRANSFER PROTOCOL (SRTP).....		55
7.1	SRTP Overview	55
7.2	SRTP Configuration.....	56
7.3	SRTP Operation	56
CHAPTER 8: MODBUS TCP SLAVE PROTOCOL.....		57
8.1	Modbus Overview	57
8.2	Modbus Configuration.....	58
8.3	Modbus Operation	58
CHAPTER 9: ETHERNET/IP PROTOCOL		59
9.1	Ethernet/IP Overview.....	59
9.2	Ethernet/IP Configuration	60
9.3	Ethernet/IP Operation	61
CHAPTER 10: FTP SERVER PROTOCOL		63
10.1	FTP Overview	63
10.2	FTP Configuration.....	63
10.3	FTP Operation	64
10.4	FTP File Accessing.....	64
CHAPTER 11: HTTP SERVER PROTOCOL.....		65
11.1	HTTP Overview	65
11.2	HTTP Configuration	65
11.3	HTTP Operation.....	66
11.4	Web Content - Storing	66
11.5	Web Content - Dynamic Data.....	67
11.5.1	Static Html Files.....	67
11.5.2	Dynamic Html Files – Reading OCS Register Data (readreg.htm)	67
11.5.3	Dynamic Html Files – Writing OCS Register Data (writereg.htm)	69
11.6	Web Content - Authoring	71
11.6.1	Web Server Example Applications	71
11.6.2	Web Authoring Reference Material	71
11.6.2	Web Authoring Tools.....	71
INDEX		73

CHAPTER 1: INTRODUCTION

1.1 Ethernet Module Overview

Ethernet Modules are built-in or user-installed into Horner OCS controllers to provide advanced Ethernet communication capabilities. Although this manual specifically addresses ETN200 and ETN300 Ethernet Modules, it is useful to describe them in the context of legacy Ethernet Modules (ETN100 and ETN116).

In this regard, Table 1.1 illustrates which OCS Models are compatible with which Ethernet Module types and Table 1.2 summarizes the protocols and features supported by each Ethernet Module type.

Table 1.1 – OCS Model / Ethernet Module Compatibility						
OCS Model	Ethernet Module Types					
	ETN100	ETN116	ETN200		ETN300	
	User-Installed	Built-In	User-Installed	Built-In	User-Installed	Built-In
OCS100	√		√			
OCS200	√		√			
RCS210	√		√			
OCS110	√		√		√	
OCS210	√		√		√	
OCS250	√		√		√	
RCS116	√		√		√	
RCS250	√		√		√	
OCS301		√				
OCS351		√				
OCS451				√		
OCS551				√		
OCS651				√		
NX221						√
NX251						√
❖ QX451						√
❖ QX551						√
❖ QX651						√

❖ Future Products

Table 1.2 – Ethernet Module Protocols and Features					
Protocol / Feature	Protocol / Feature Description	Ethernet Module Types			
		ETN100	ETN116	ETN200	ETN300
ICMP Ping	Internet Control Message Protocol	√	√	√	√
EGD (Peer)	GE Fanuc Ethernet Global Data	√	√	√	√
SRTP Server	GE Fanuc Service Request Transfer Protocol	√	√	√	√
Modbus TCP Slave	Modbus over Ethernet	√	√	√	√
CsCAN TCP Server	Horner APG CsCAN over Ethernet			√	√
Ethernet / IP Server	ODVA CIP over Ethernet			√	√
➤ FTP Server	File Transfer Protocol			√	√
➤ HTTP Server	HyperText Transfer Protocol (Web Server)			√	√
Half / Full Duplex	Auto-Negotiated Ethernet Mode			√	√
10 / 100 MHz	Auto-Negotiated Ethernet Speed			√	√
Extended Registers	Access to %R2049-%R9999				√

➤ FTP & HTTP protocols are supported only by OCS Models with built-in Ethernet and Compact Flash, such as OCS451, OCS551, OCS651, NX221, NX251, QX451, QX551 and QX651.

1.2 Ethernet Network Example

Ethernet Module configuration is done through Cscape Programming Software. Figure 1.1 illustrates an example of a network containing ETN200 and ETN300 Ethernet Modules.

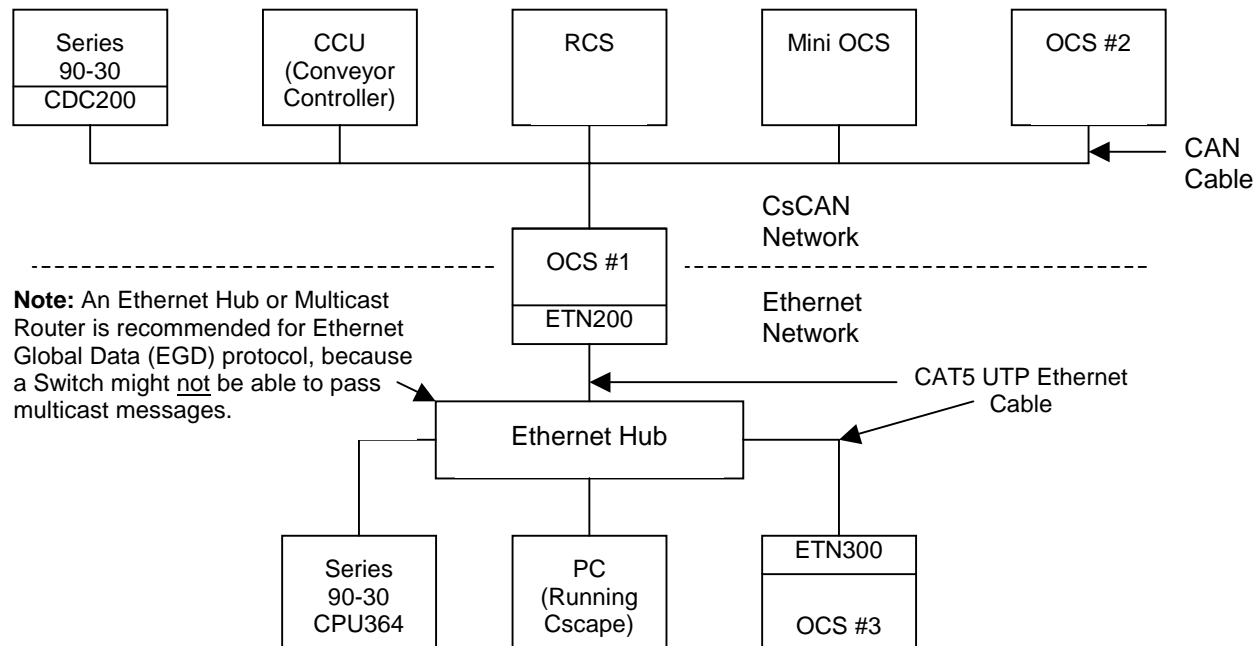


Figure 1.1 – Example of an Ethernet Network

1.3 Ethernet System Requirements and Interoperability

Full Ethernet functionality, as described in this User Manual, requires:

- PC running Cscape Programming Software Version 7.0c or later (for configuration)
- OCS controller with built-in or user-installed ETN200 or ETN300 Ethernet Module
- FTP & HTTP protocols supported only by OCS Models with built-in Ethernet and Compact Flash

As of the writing of this User Manual, the *server* protocols, supported by the ETN200 and ETN300 SmartStack Ethernet Modules, have been tested for interoperability with Cscape and several 3rd party *client* software packages, as shown in Table 1.3.

Table 1.3 – Ethernet Module Server Protocol Interoperability							
Client Software Package		Ethernet Module Server Protocol					
Name	Version	CsCAN	SRTP	Modbus	EIP	FTP	HTTP
Cscape Programming Software	7.0c	√					
GE Fanuc Cimplicity HMI	3.2		√				
GE Fanuc Cimplicity PE	5.0		√				
	6.1		√	√			
KEPware OPC / DDE Server	4.84.227		√	√			
Pyramid EIP Scanner Demo	3.51				√		
Microsoft Internet Explorer	5.50.4522					√	√
	6.0.2900					√	√
SmartFTP	1.5.990					√	
Mozilla Firefox	1.0.7						√

1.4 Ethernet Module Specifications

Table 1.4 – Specifications	
General	
Ethernet Speeds	10 BaseT Ethernet (10 MHz) 100 BaseTx Fast Ethernet (100 MHz)
Ethernet Modes	Half or Full Duplex
Ethernet Auto-Negotiation	Both 10 / 100 MHz and Half / Full Duplex
Ethernet Connector Type	Shielded RJ-45
Ethernet Cable Type Recommendation	CAT5 (or better) UTP
Application Protocols	
ICMP	Ping Only
CsCAN TCP Server	Maximum Connections = 8
EGD (Ethernet Global Data)	Maximum Exchanges = 127 Maximum Data Bytes per Exchange = 1,400
SRTP Server	Maximum Connections = 16
Modbus TCP Slave	Maximum Connections = 16
Ethernet / IP Server	Maximum Connections = 2
FTP Server	Maximum Connections = 4
HTTP Server	Maximum Connections = 1

1.5 Additional Technical Resources

It is assumed that the user has a working knowledge of Ethernet networks and Horner OCS / RCS controllers. The following references are available to assist the user in these areas.

For a technical summary of Ethernet and other information, refer to:

www.techfest.com/networking/lan/ethernet.htm

For Horner OCS / RCS controller technical support, refer to:

www.heapg.com/Support/index.htm

For additional assistance, contact Technical Support at the following locations.
Please visit our website for manual updates.

North America:
(317) 916-4274
www.heapg.com

Europe:
(+) 353-21-4321-266
www.horner-apg.com

NOTES

CHAPTER 2: INSTALLATION

2.1 Built-In Ethernet Module

As shown in Table 1.1 (page 9), for some OCS Models, the SmartStack Ethernet Module is built-in to the OCS unit, requiring no user-installation. For example, Figure 2.1 shows the location of the RJ-45 Ethernet connector on OCS451, OCS551 and OCS651 units.

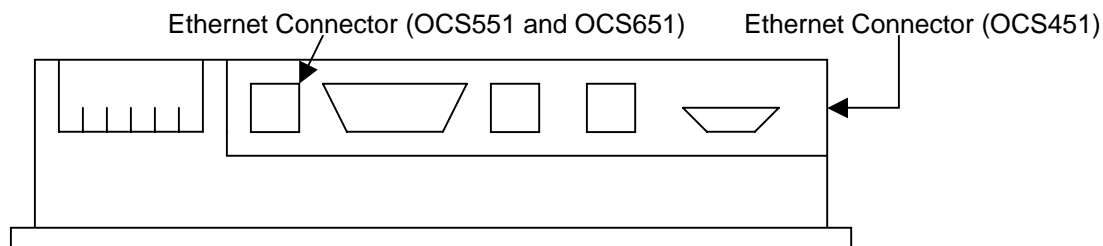


Figure 2.1 – SVGA Color Graphics OCS Bottom View

2.2 User-Installed Ethernet Module

For other OCS or RCS Models, the SmartStack Ethernet Module must be user-installed on the back of the OCS / RCS unit. Figure 2.2 shows the locations of the Ethernet RJ-45 connector and LED indicators on the user-installed SmartStack Ethernet Module.

Note: Normally, a user-installed Ethernet Module occupies the first I/O slot on the back of the OCS or RCS unit. The only exception to this rule occurs if a FOX100 Module is also installed. In this case, the FOX100 Module occupies the first I/O slot, and the Ethernet Module is installed into the second I/O slot.

Caution: For proper functioning and to avoid possible damage, do not install more than four SmartStack Modules on the back of an OCS or RCS controller.

For more details on installing SmartStack I/O modules, refer to Chapter 2 of the Control Station Hardware Manual (MAN0227), which also provides a handy checklist concerning panel box layout and minimum clearance requirements.

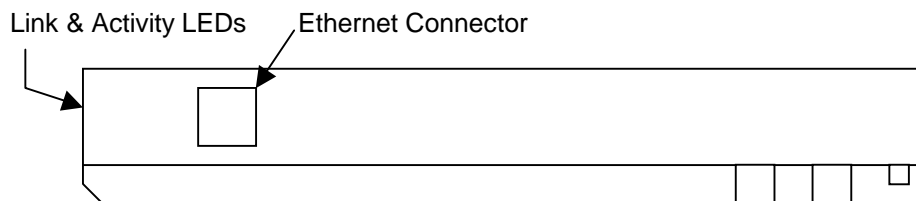


Figure 2.2 – User-Installed Ethernet Module Side View

2.3 Network Administrator Installation Notes

When connecting an Ethernet Module to a local network, the following information is provided to the Network Administrator, as an aid in configuring Ethernet Hubs, Routers, Switches, Gateways and Servers.

2.3.1 UDP and TCP Ports

Each protocol supported by the Ethernet Module (except ICMP) uses one or more UDP and/or TCP Ports as the destination port for all messaging, as shown in Table 2.1. (**Note:** Any port can be used as the source port.) Required port usage for the supported protocols should be taken into account when configuring Ethernet Routers and Gateways.

Table 2.1 – Ethernet Protocol UDP and TCP Port Usage				
Ethernet Protocol	UDP Port		TCP Port	
	Hexadecimal	Decimal	Hexadecimal	Decimal
EGD	4746	18246		
SRTP			4745	18245
Modbus			01F6	502
CsCAN			4845	18501
Ethernet / IP	08AE	2222	AF12	44818
FTP			0014 and 0015	20 and 21
HTTP			0050	80

2.3.2 EGD Unicast and Multicast IP Addressing

When using Ethernet Global Data (EGD) protocol for peer-to-peer communication, there are two methods for sending data: (1) send to a single device or (2) send to a group of devices.

When sending to a single device (method 1), EGD protocol uses **Unicast IP Addressing**. This means that the IP header's 32-bit Destination IP Address will contain the intended recipient's unique IP Address.

When sending to a group of devices (method 2), EGD protocol uses **Multicast IP Addressing**. This means that the IP header's 32-bit Destination IP Address will contain one of the 32 Multicast IP Addresses shown in Table 2.2.

Table 2.2 – EGD Multicast IP Addressing	
Group ID	Multicast IP Address
1	224.0.7.1
2	224.0.7.2
:	:
:	:
32	224.0.7.32

Ethernet Switches normally don't support Multicast IP Addressing, while Ethernet Hubs do support Multicast IP Addressing. Some Ethernet Routers, known as Multicast Routers, do support Multicast IP Addressing, by using Internet Group Management Protocol (IGMP).

Note: For those customers wanting to use Multicast Routers to connect EGD devices, the Ethernet Module automatically handles IGMP communication with Multicast Routers.

For more information regarding EGD protocol, refer to Chapter 6 of this manual.

2.3.3 *Internet Connectivity*

Since the Ethernet Module uses a standard TCP/IP protocol stack (powered by NetX and ThreadX), it can communicate beyond the local network, and on the Internet, for all protocols except EGD. To do so, the Ethernet Module must be configured with the IP Address of a Network Gateway server, which allows communication outside the local network. See **Default Gateway** configuration under step 5 of Section 3.1 (page 19) in this manual for details.

Note: As network complexity increases, due to Ethernet Hubs, Routers, Switches, Gateways, and the Internet, the worst-case network delay increases. In many cases, the client software must be configured to account for this time lag. For example, Cscape's **Timeout** can be adjusted as shown under step 2 of Section 4.2 (page 23) in this manual.

2.3.4 *IMPORTANT: Wireless Network Considerations*

When using wireless equipment with industrial networks, make sure the system is designed and installed by personnel that have been trained to use wireless networks in industrial environments. Site surveys, selection of equipment and installation can be critical in network performance.

In general the 802.11**b**- based equipment is not a good choice in industrial environments. The frequencies and modulation techniques used in the "b" standard are very susceptible to multi-path interference in industrial environments (large metal objects, dense walls and floors, etc.) The 802.11**a** and 802.11**g** are less susceptible to this interference.

Using UDP based protocols such as Ethernet Global Data (EGD) must be carefully considered when using wireless networks. Wireless networks are more likely to lose or damage communication packets. Many UDP based protocols, including EGD, do not detect and retransmit lost or damaged packets and depend on periodic data transmissions to compensate for this lost data. If your system requires a UDP protocol with a wireless network, make sure it is designed such that random periods without refreshed data do not adversely affect the operation of your system.

Depending on the architecture of the wireless network and the protocols used, wireless networks often produce collisions and extra data packets that are not experienced when using a traditional wire and switch-based network. The extra collisions and traffic coupled with the typically lower bandwidth and higher latency of wireless networks can cause degradation in performance.

NOTES

CHAPTER 3: GENERAL CONFIGURATION

Note: The following configuration is required for all applications regardless of the protocols used. Additional configuration procedures must be performed for each protocol used as described in the configuration sections of the next several chapters.

3.1 Ethernet Module Configuration

To configure the Ethernet Module, use Cscape Programming Software to perform the following six steps:

1. On the main Cscape screen, select the **Controller** menu and its **I/O Configure** sub-menu to open the I/O Configuration dialog (Figure 3.1 or Figure 3.2).
2. If configuring a different OCS Model than the one shown in the I/O Configuration dialog, click on the topmost **Config** button, select the desired OCS Model, and then click **OK**.

Note: Figure 3.1 shows a typical I/O Configuration dialog for OCS Models, which have standard FOX100 and Ethernet Modules built-in. For these models, step 3 should be skipped.

Figure 3.2 shows a typical I/O Configuration dialog for other OCS Models, in which the Ethernet Module is a user-installable option. For these models, step 3 must be performed, to configure the Ethernet Module into a SmartStack I/O slot.

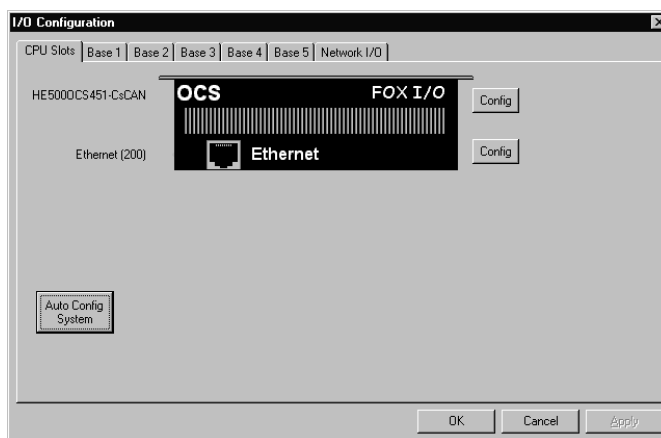


Figure 3.1 – I/O Configuration Dialog – OCS Models With Built-In Ethernet

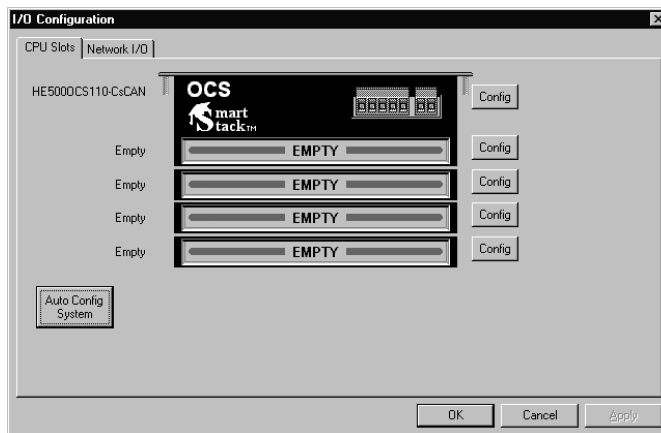


Figure 3.2 – I/O Configuration Dialog – OCS Models With User-Installable Ethernet

3. To configure a user-installable Ethernet Module into a SmartStack I/O slot, do the following:
 - a. Referring to Figure 3.2 above, click on the **Config** button to the right of the desired I/O slot, which will open the Add I/O Module dialog, as shown in Figure 3.3.
 - b. Click the **Comm** tab and select the desired **HE800ETNxxx** item from the list.
 - c. Click **OK**. The Ethernet Module is now configured into an I/O slot, as shown in Figure 3.4.

Note: Normally, a user-installed Ethernet Module occupies the first I/O slot on the back of the OCS or RCS unit. The only exception to this rule occurs if a FOX100 Module is also installed. In this case, the FOX100 Module occupies the first I/O slot, and the Ethernet Module is installed into the second I/O slot.

Caution: For proper functioning and to avoid possible damage, do not install more than four SmartStack Modules on the back of an OCS or RCS controller.

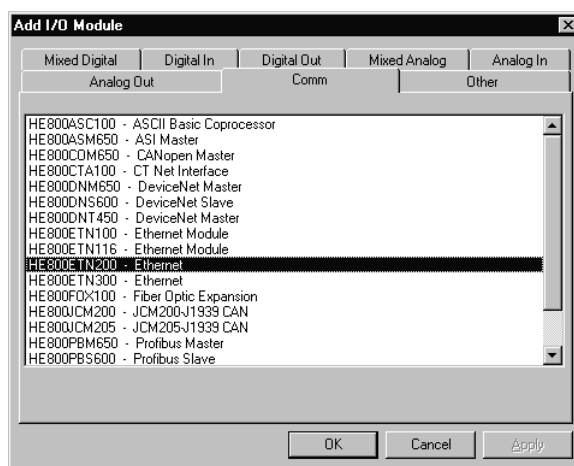


Figure 3.3 – Add I/O Module Dialog – Comm Tab and Ethernet Module Selected

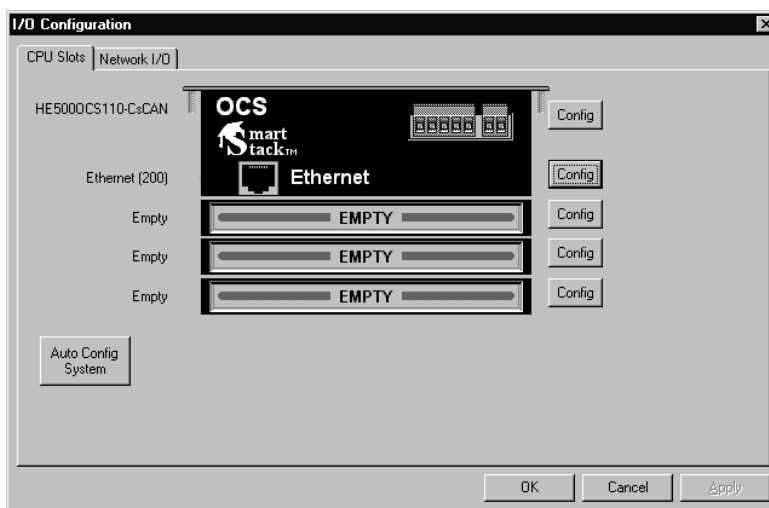


Figure 3.4 – I/O Configuration Dialog – User-Installed Ethernet Module

- Now it is time to configure the Ethernet Module as appropriate for the application. Referring to Figure 3.1 or 3.4 above, click the **Config** button to the right of the Ethernet Module, and then select the Module Setup tab, revealing the Ethernet Module Configuration dialog (Figure 3.5).

Figure 3.5 – Ethernet Module Configuration Dialog - Module Setup Tab Selected

- Configure the Ethernet Module parameters as follows:

IP Address: Enter the static IP Address for the Ethernet Module being configured.

Note: IP Addresses are entered as four numbers, each ranging from 0 to 255. These four numbers are called octets and they are always separated by decimal points.

Use CAN ID for last Octet: Check this box to replace the low octet (rightmost number) of the static IP Address with the OCS unit's CAN Network ID.

Get IP from IP Addr Register: Check this box if the Ethernet Module's IP Address will be *read from* the 32-bit register indicated by the IP Addr Register edit box. Otherwise, the static IP Address will be used and will be *written to* the register indicated by the IP Addr Register edit box.

Net Mask: Enter the Net Mask (sometimes called Subnet Mask) being used by all nodes on the local network. Typical local networks use Class C IP Addresses, in which case the low octet (rightmost number) is used to uniquely identify each node on the local network. In this case, the default Net Mask value of 255.255.255.0 should be used.

Default Gateway: Enter the IP Address of a Gateway Server on the local network that allows for communication outside of the local network. To prevent the Ethernet Module from communicating outside the local network, set the Default Gateway IP Address to 0.0.0.0 (the default setting).

Status Register: Enter an OCS Register reference (such as %R100) to indicate which 16-bit OCS register will have the Ethernet Status word written to it. Table 3.1 shows how this register value is formatted and explains the meaning of each bit in the Status Word.

Table 3.1 - Ethernet Status Word Register Format															
High Byte								Low Byte							
Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
0	0	Dup	Spd	0	Rx	Tx	Link	TCP Connections							
Status Bit(s)			Status Indication					Status Values							
								Minimum		Maximum					
0			Reserved					Always 0							
Dup			Link Duplex (Auto-Negotiated)					0 = Half Duplex		1 = Full Duplex					
Spd			Link Speed (Auto-Negotiated)					0 = 10 MHz		1 = 100 MHz					
Rx			Receive State					0 = Inactive		1 = Active					
Tx			Transmit State					0 = Inactive		1 = Active					
Link			Link State					0 = Down		1 = Up					
TCP Connections			Total Number of Active TCP Connections (CsCAN, SRTP, Modbus, EIP, FTP, HTTP)					0		40					

Version Register: Enter an OCS Register reference (such as %R101) to indicate which 16-bit OCS register will have the Ethernet Firmware Version written to it. The value stored in the Version Register is: (Ethernet Firmware Version * 100). For example, for Ethernet Firmware Version 4.30, the Version register will contain 430.

Note: The Ethernet Module writes to the Status and Version Registers even when the OCS is not in RUN mode. Therefore, when the OCS is not in RUN mode, User Screens and Cscape Data Watch can still be used to monitor Ethernet Status and Firmware Version.

IP Addr Register: Enter an OCS Register reference (such as %R102) to indicate which 32-bit OCS register will be read or written to obtain or indicate the current IP Address of the Ethernet Module being configured. For more details, see the above discussion on the **Get IP from IP Addr Register** check box and see Section 3.2 below.

Protocol Support: Configure each protocol that will be used in the application, as described in the configuration sections of the following chapters.

- Click **OK** to accept the new Ethernet Module configuration.

3.2 Ethernet Module IP Address

The Ethernet Module obtains its IP Address in one of three different ways, depending on how the **Use CAN ID for last Octet** and the **Get IP from IP Addr Register** checkboxes (Figure 3.5) are configured, as described in the following three sections.

3.2.1 Static IP Address

- ☐ **Use CAN ID for last Octet**
- ☐ **Get IP from IP Addr Register**

In this mode, the Ethernet Module's IP Address comes from the **IP Address** parameter only, and does not use the CAN Network ID for the low octet and does not use **IP Addr Register** to obtain an IP Address from an OCS register. In this case the Ethernet Module writes the static IP Address to the 32-bit OCS register indicated by the **IP Addr Register** parameter.

3.2.2 Static IP Address with CAN ID

- ☒ **Use CAN ID for last Octet**
- ☐ **Get IP from IP Addr Register**

In this mode, the Ethernet Module's IP Address comes from a combination of the **IP Address** parameter and the OCS/RCS CAN Network ID. The most significant (leftmost) three octets of the IP Address come from the **IP Address** parameter. The least significant (rightmost) octet of the IP Address is taken from the OCS (or RCS) CAN Network ID. In this case the Ethernet Module writes the *adjusted* IP Address to the 32-bit OCS register indicated by the **IP Addr Register** parameter.

3.2.3 IP Address from OCS Register

- ☐ **Use CAN ID for last Octet**
- ☒ **Get IP from IP Addr Register**

In this mode, the Ethernet Module's IP Address comes from an OCS register. The **IP Addr Register** parameter indicates which 32-bit OCS register to read the IP Address from. The static **IP Address** parameter is not used in this situation, except to set the **Default IP Address** in non-volatile memory.

Note: Every time an I/O configuration is successfully downloaded to an OCS with an Ethernet Module, the static **IP Address**, **Net Mask** and **Default Gateway** parameters are stored in non-volatile memory. In the event of a future unsuccessful I/O configuration download, the Ethernet Module will communicate using these 3 stored parameters. This is done in an effort to minimize potential loss of communication, which would require direct on-site intervention to correct.

NOTES

CHAPTER 4: CSCAN OVER ETHERNET PROTOCOL

4.1 CsCAN over Ethernet Overview

This chapter describes CsCAN TCP Server Communication protocol, also known as CsCAN over Ethernet protocol.

CsCAN over Ethernet protocol allows a CsCAN Host Programming Tool, such as Cscape, to access an OCS unit, as though it were connected directly to the OCS programming serial port.

4.2 CsCAN over Ethernet Configuration

The Ethernet Module requires no protocol-specific configuration for CsCAN over Ethernet protocol. Only the general Ethernet Module Configuration previously described in Chapter 3 is required. As long as the Ethernet Module has been assigned an IP Address and Net Mask, it will automatically respond to CsCAN over Ethernet messages sent to it by a PC running Cscape.

To connect Cscape to an OCS unit's Ethernet Module, using CsCAN over Ethernet protocol, perform the following two steps:

1. On the main Cscape screen, select the **Tools** menu and its **Options** sub-menu to open the Program Options dialog (Figure 4.1).

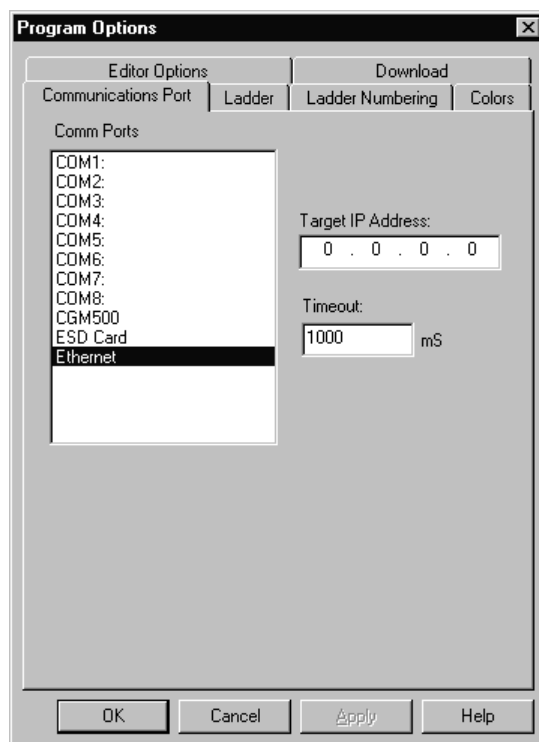


Figure 4.1 – Program Options Dialog – Communications Port Tab and Ethernet Selected

2. Select the Communication Ports tab, select **Ethernet** from the **Comm Ports** list, and then set the **Target IP Address** and **Timeout** parameters as follows:

Target IP Address: Enter the IP Address previously assigned to the target Ethernet Module. Please refer to Chapter 3 regarding how to assign an IP Address to an Ethernet Module.

Timeout: Enter a number between 1000 and 65,000 (in milliseconds) for the maximum expected network round-trip communication time. This value determines how long Cscape will wait for a response after it sends a CsCAN over Ethernet protocol command to the Ethernet Module.

Note: For most local network applications, the default **Timeout** value of 1000 is sufficient.

However, there are some network considerations, which may require the **Timeout** value to be increased to facilitate reliable communication. This includes heavily loaded networks, complex networks with multiple levels of routers and switches, and Internet communication.

Also, because CsCAN communication is affected by ladder code scan rate, the **Timeout** value may have to be further increased to compensate for applications with very slow scan rates.

4.3 CsCAN over Ethernet Operation

As stated previously, the Ethernet Module allows a CsCAN Host Programming Tool, such as Cscape, to use CsCAN over Ethernet to perform all standard supervisory control, monitoring and programming functions with the OCS, as though it were connected directly to the OCS programming serial port.

These standard supervisory functions include the Ethernet Module's ability to handle pass-through communication with the CsCAN Nodes attached to the OCS CAN port. This feature is known as CsCAN single-point programming. For example, referring back to Figure 1.1 (page 10) the PC (running Cscape) can use OCS #1's ETN200 Module as a gateway, to easily access all of the CsCAN Nodes on OCS #1's CsCAN network.

4.4 CsCAN over Ethernet Downloading Precautions

When downloading a new Ethernet configuration to a target OCS, using CsCAN over Ethernet protocol, extra care should be taken.

In particular, when downloading **I/O and Network Configuration**, the Ethernet Module configuration is also downloaded, which has the potential to change the target device's IP Address, or it could even remove the Ethernet Module configuration entirely, resulting in loss of communication and a failed download.

For this reason, when the Cscape user modifies the I/O Configuration, and then starts to download it using CsCAN over Ethernet, Cscape issues a warning message, as shown in Figure 4.2.

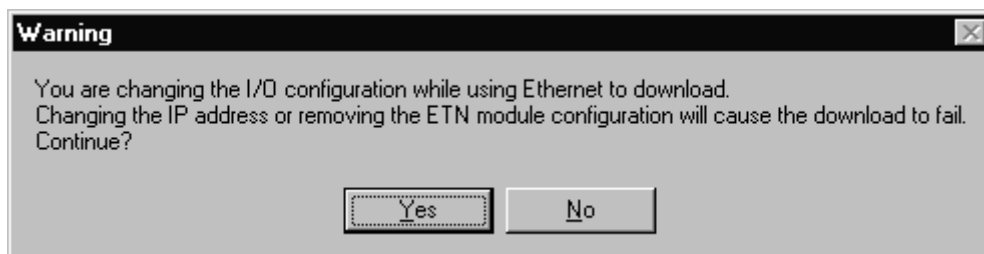


Figure 4.2 – I/O Configuration Download Warning

4.4.1 How to Prevent Losing Communication

Before using CsCAN over Ethernet to download a new I/O Configuration, to an OCS with an Ethernet Module installed, the application programmer should:

1. Make sure the new I/O Configuration contains an Ethernet Module configuration.
2. Make sure the new Ethernet configuration will not change the IP Address.

Refer to Chapter 3 of this User Manual on how to check and/or correct Ethernet Module configuration and pay particular attention to step 5 of the Ethernet Module configuration process in Section 3.1 (page 19), and to Section 3.2 (page 21), which explains in detail how the Ethernet Module obtains an IP Address.

If the new Ethernet configuration specifies *Static IP Address* (Section 3.2.1 [page 21]), make sure the **IP Address** parameter (Figure 3.5 [page 19]) matches Cscape's **Target IP Address** (Figure 4.1).

If the new Ethernet configuration specifies *Static IP Address with CAN ID* (Section 3.2.2 [page 21]), make sure the IP Address, which will be built from the combination of the **IP Address** parameter (Figure 3.5 [page 19]) and the OCS/RCS CAN Network ID, matches Cscape's **Target IP Address** (Figure 4.1).

If the new Ethernet configuration specifies *IP Address from OCS Register* (Section 3.2.3 [page 21]), make sure the OCS register indicated by the **IP Addr Register** parameter (Figure 3.5 [page 19]) contains an IP Address, and that it matches Cscape's **Target IP Address** (Figure 4.1). If necessary, use Cscape's Data Watch facility to set the OCS register to the correct IP Address before downloading.

4.4.2 How to Recover from Lost Communication

If the Ethernet Module's IP Address changes as a result of using CsCAN over Ethernet to download a new user program, activity will halt at the end of the I/O Configuration download, and communication between Cscape and the target OCS will be lost. Then, after several seconds, Cscape will display a Communication Timeout error.

If this happens, it is possible to recover Cscape communication with the target OCS, as follows:

1. Referring to Chapter 3 of this User Manual (especially Section 3.2 [page 21]) and to Section 4.4.1 above, determine the Ethernet Module's new IP Address, by using Cscape to re-examine the I/O Configuration just downloaded.
2. Referring to Figure 4.1, change Cscape's **Target IP Address** to match the Ethernet Module's new IP Address.
3. Try performing the download again.
4. If this does not succeed, try changing Cscape's **Target IP Address** (Figure 4.1) to match the static **IP Address** parameter (even if it is grayed out).
5. If all else fails, connect a PC running Cscape directly to the target OCS unit's programming serial port, repeat the download, and then use Data Watch to examine the register indicated by the **IP Addr Register** parameter (Figure 3.5 [page 19]), to discover the Ethernet Module's new IP Address.

Note: If the application programmer wants to use CsCAN over Ethernet to *intentionally* change an Ethernet Module's IP Address, do this by (1) starting the download, (2) waiting for Cscape to timeout and (3) performing steps 1, 2 and 3 of the recovery process, as described above.

4.5 CsCAN over Ethernet Security

To prevent the use of CsCAN over Ethernet protocol to gain unauthorized access to an OCS or RCS, Cscape Programming Software and OCS Ethernet Firmware have the ability to password-protect CsCAN over Ethernet communication. To implement CsCAN over Ethernet Security, use Cscape Programming Software, to perform the following nine steps:

1. Open the user program previously created for the target OCS or RCS controller.
2. If the user program has already been set up with security passwords, first log-in as the Administrator. To do this, select the **Tools** menu and its **Security** sub-menu, click **Log-in**, enter the Administrator password, and then click **OK**. Cscape will then acknowledge the Administrator log-in; click **OK** again.
3. On the main Cscape screen, select the **Tools** menu and its **Security** sub-menu, and then click **Change Passwords**, to open the **Security Passwords** dialog, as shown in Figure 4.3.

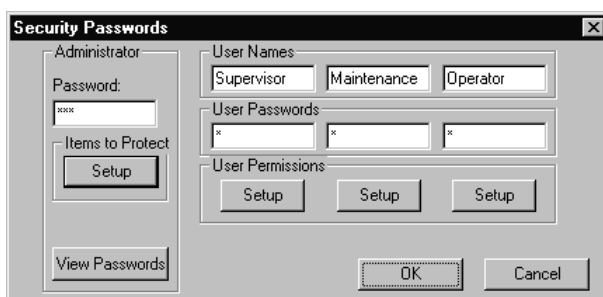


Figure 4.3 – Security Passwords Dialog

4. Click on the **View Passwords** button to view the Administrator and User passwords.
5. Change passwords and user names as desired for the application. Note that passwords are numeric values between 1 and 999,999, while the user names can be any text from 1 to 15 characters long.
6. Click on the **Items to Protect Setup** button to open the Administrator's **Security Settings** dialog, as shown in Figure 4.4.



Figure 4.4 – Administrator's Security Settings Dialog

7. To password-protect CsCAN over Ethernet protocol, make sure the **CsCAN TCP (Ethernet)** checkbox is checked in the Administrator's **Security Settings** dialog, as shown in Figure 4.4. Then click **OK**.
8. The Administrator password is always authorized for all protected features. If desired, one or more of the user passwords can also be authorized for CsCAN over Ethernet communication. To do this, refer to Figure 4.3 and click one of the **User Permissions Setup** buttons to open that user's **Security Settings** dialog, which will be similar to Figure 4.4. Then check the **CsCAN TCP (Ethernet)** checkbox and click **OK**.
9. After downloading the user program to the target OCS or RCS, both Cscape and the OCS (or RCS) controller will enforce the new password security for CsCAN over Ethernet protocol.

NOTES

CHAPTER 5: INTERNET CONTROL MESSAGE PROTOCOL (ICMP)

5.1 ICMP Overview

ICMP is used for diagnostic purposes only, to determine if another device exists on the Ethernet network.

Using ICMP, the Ethernet Module sends Ping Echo Requests to another device, and expects the other device to answer with Ping Echo Responses. The Ethernet Module measures the round-trip time of each Ping Echo Request / Response exchange and puts the result (in milliseconds) into an OCS register.

In addition, when the Ethernet Module receives a Ping Echo Request from another device, it answers with a Ping Echo Response.

Note: Although ICMP supports other network diagnostics, the Ethernet Module only supports *Ping*.

5.2 ICMP Configuration

If ICMP protocol will be used in the application, ICMP Configuration must be performed, in addition to the general Ethernet Module Configuration previously described in Chapter 3. To configure ICMP protocol, use Cscape Programming Software to perform the following five steps:

1. Open the Ethernet Module Configuration dialog (Figure 3.5 [page 19]), as described in Chapter 3.
2. Enable ICMP by checking the **ICMP (Ping)** checkbox in the Module Configuration dialog (Figure 3.5).
3. Click on the **Config** button next to the **ICMP (Ping)** checkbox to open the ICMP Configuration dialog (Figure 5.1).

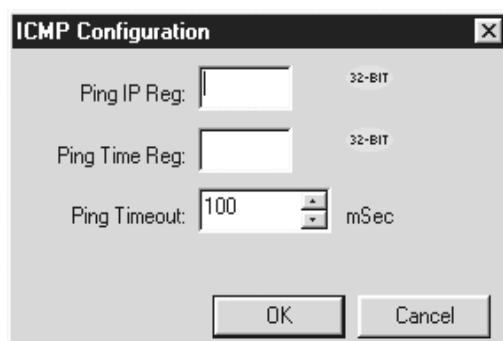


Figure 5.1 – ICMP Configuration Dialog

4. Set up the ICMP Configuration parameters as follows:

Ping IP Reg: Enter an OCS Register reference (such as %R200) to indicate which 32-bit OCS register will be read to obtain the IP Address of the network device to send Ping Echo Requests to.

Ping Time Reg: Enter an OCS Register reference (such as %R202) to indicate which 32-bit OCS register will be written with the Ping Echo Request / Response round-trip time (in milliseconds).

Ping Timeout: Enter a number between 100 and 100,000 for how often (in milliseconds) the Ethernet Module should send Ping Echo Requests.

5. Click **OK** to accept the new ICMP Configuration.

5.3 ICMP Operation

To start Ping Echo Requests, use Ladder Code, User Screens or Cscape Data Watch to write an IP Address to the OCS register indicated by **Ping IP Reg.** To check the resulting Ping Echo Response times, read the OCS register indicated by **Ping Time Reg.** If the round-trip time exceeds the configured **Ping Timeout** (or if there is no response at all), the reported response time will be -1.

To stop Ping Echo Requests, write 0.0.0.0 to the OCS register indicated by **Ping IP Reg.**

Note that ICMP protocol is active even when the OCS is not in RUN mode. Therefore, when the OCS is not in RUN mode, User Screens and Cscape Data Watch can still be used to control and monitor ICMP Pinging.

CHAPTER 6: ETHERNET GLOBAL DATA PROTOCOL (EGD)

6.1 EGD Overview

Ethernet Global Data (EGD) protocol is a GE Fanuc Automation protocol, which is designed for simple, efficient data exchanges between peer devices on a network.

EGD protocol communicates using the UDP transport layer. Although this method of data transfer is very efficient, it has no specific way to detect and recover lost data packets. However, since all EGD data transfers are periodic, lost data packets will be repeated when their user-configured time periods expire.

Caution: EGD protocol is not intended for *one-shot event notification* or for *applications with critical data*, which can't withstand being delayed as described above.

Each device on an EGD network can be configured as a Producer, as a Consumer, or both. A Producer is a device that transmits Exchanges (blocks of data) to one or more Consumers. A Consumer is a device that receives Exchanges from one or more Producers.

A Producer can transmit Exchanges directly to a specific Consumer, by sending them to the Consumer's IP Address (using Unicast IP Addressing). A Producer can also transmit Exchanges to a Group of Consumers, by sending them to a Group ID (using Multicast IP Addressing). See Section 2.3.2 (page 14) for more details regarding Unicast and Multicast IP Addressing.

An Ethernet Module supports up to 127 concurrent Exchanges, each of which can be either a Producer or a Consumer of data.

6.2 EGD Terminology

Before configuring an Ethernet Module for EGD protocol, it is essential that the application programmer understand the key EGD terms, which are shown in Table 6.1.

Table 6.1 – EGD Terminology	
Term	Definition
Exchange	A block of data sent by a Producer and received by one or more Consumers
Exchange Number	A number (1 to 16,383), which along with the IP Address of the Producer, is used to uniquely identify an Exchange on an EGD network
Producer	An EGD network device configured to transmit one or more Exchanges
Consumer	An EGD network device configured to receive one or more Exchanges
Produced Exchange	A block of data that a Producer sends to a Consumer or to a Group of Consumers
Consumed Exchange	A block of data that a Consumer or Group of Consumers receives from a Producer
Group	One or more Consumers that are configured to receive Exchanges, which have been sent by a Producer to a specific Group ID
Group ID	A number (1 to 32), which is used to identify a Group of Consumers
Production Period	A value (in milliseconds) that specifies how often a Produced Exchange is transmitted to the network
Update Timeout	A value (in milliseconds) that specifies how long a Consumer will wait to receive an Exchange, before considering it late. (Note: As rule of thumb, a Consumed Exchange's Update Timeout is normally set to at least twice the corresponding Produced Exchange's Production Period, plus 10 milliseconds.)

6.3 EGD Configuration

If EGD protocol will be used in the application, EGD Configuration must be performed, in addition to the general Ethernet Module Configuration previously described in Chapter 3. To configure EGD protocol, use Cscape Programming Software to perform the following six steps:

1. Open the Ethernet Module Configuration dialog (Figure 3.5 [page 19]) as described in Chapter 3.
2. Enable EGD by checking the **EGD (Ethernet Global Data)** checkbox in the Module Configuration dialog (Figure 3.5).
3. Click on the **Config** button next to the **EGD (Ethernet Global Data)** checkbox to open the Ethernet Global Data Configuration dialog (Figure 6.1 or Figure 6.2).

Note: The Ethernet Global Data Configuration dialog allows EGD Produced and Consumed Exchanges to be configured. To configure Produced Exchanges, select the Produced Exchanges tab (Figure 6.1); to configure Consumed Exchanges, select the Consumed Exchanges tab (Figure 6.2)

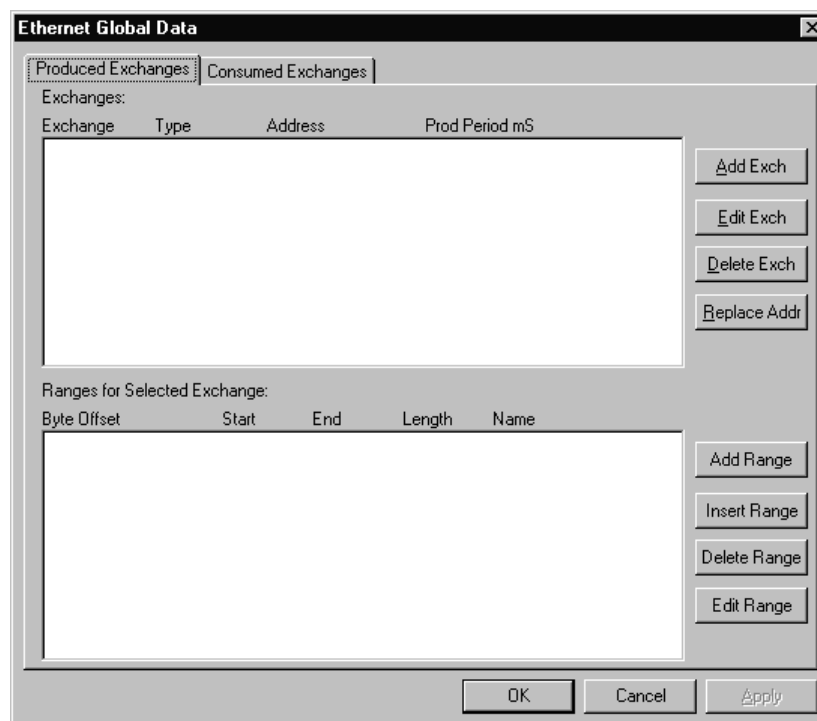


Figure 6.1 – Ethernet Global Data (EGD) Configuration Dialog – Produced Exchange Tab Selected

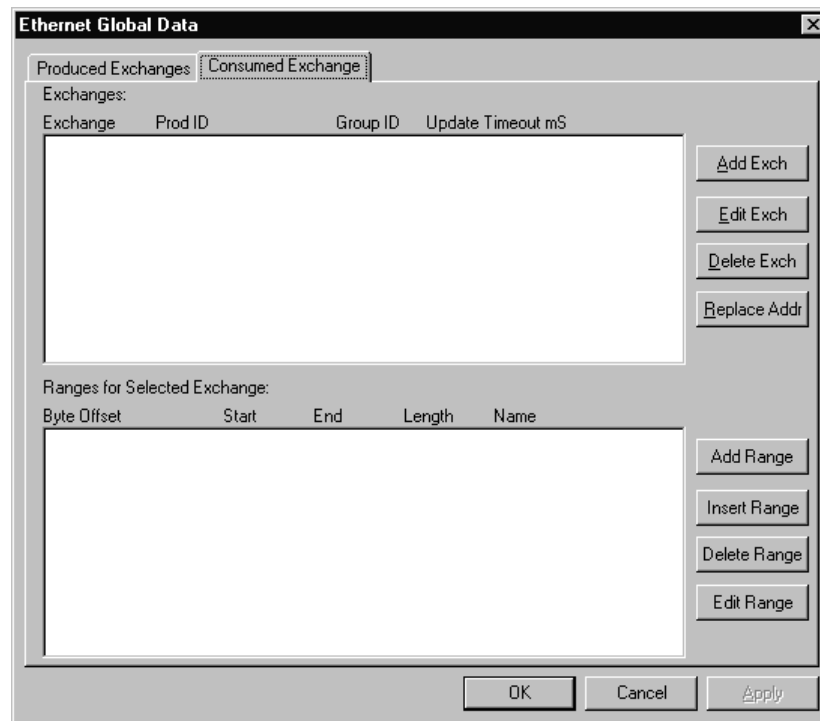


Figure 6.2 – Ethernet Global Data (EGD) Configuration Dialog – Consumed Exchange Tab Selected

4. Follow the steps in Section 6.4 to configure Produced Exchanges, as necessary for the application.
5. Follow the steps in Section 6.5 to configure Consumed Exchanges, as necessary for the application.
6. Click **OK** to accept the new EGD Configuration.

6.4 EGD Produced Exchange Configuration

To configure EGD Produced Exchanges, open the Ethernet Global Data Configuration dialog (Figure 6.1) as described in Section 6.3, and select the **Produced Exchanges** tab, where:

1. In the upper window, one or more Produced Exchanges can be created (Section 6.4.1).
2. In the lower window, I/O Blocks can be defined for each Produced Exchange (Section 6.4.2).

When creating a Produced Exchange, the application programmer selects an Exchange Number for it, determines whether the Exchange will be sent to a specific Consumer or to a Group of Consumers, chooses which Consumer or Group of Consumers will receive the Exchange, and sets how often the Exchange will be sent to the EGD network.

When defining I/O Blocks for a Produced Exchange, the application programmer selects what type and how much *information* will be associated with the Exchange. For Produced Exchanges, there are two types of I/O Blocks to choose from: Data and Status. Table 6.2 shows these I/O Block Types along with their definitions:

Table 6.2 – Produced Exchange I/O Block Type Definitions	
Type	Definition
Data Block	Block of consecutive OCS registers to be sent periodically to the EGD network
Status Block	16-bit OCS register to be written with the Produced Exchange's Status Word

When defining **Data Blocks** for a Produced Exchange, the maximum total OCS register data the Exchange can send to the EGD network is 1400 bytes. This means that up to a total of 700 16-bit registers (%R, %AI, AQ, etc.), or 11,200 1-bit registers (%M, %T, %I, %Q, etc.), or a combination thereof can be defined for a Produced Exchange.

Note: The Ethernet Module allows a total of up to 256 Data Blocks to be defined for all Produced Exchanges combined. This means that if 127 Produced Exchanges are configured, each can have an average of about 2 Data Blocks defined.

When a **Status Block** is defined for a Produced Exchange, exactly 2 bytes of register data are written with the Produced Exchange's Status Word. See Section 6.7 for general information regarding EGD Status Words, and Section 6.7.1 for specific information regarding EGD Produced Exchange Status Words.

Note: The Ethernet Module maintains just one 16-bit status word for each Produced Exchange. For this reason, there is never any need to define more than one Status Block for a given Produced Exchange.

6.4.1 Creating EGD Produced Exchanges

To create EGD Produced Exchanges, perform the following six steps:

1. In the Ethernet Global Data Configuration dialog (Figure 6.1), click on the **Add Exch** button to open the Add / Edit Produced Exchange dialog (Figure 6.3).

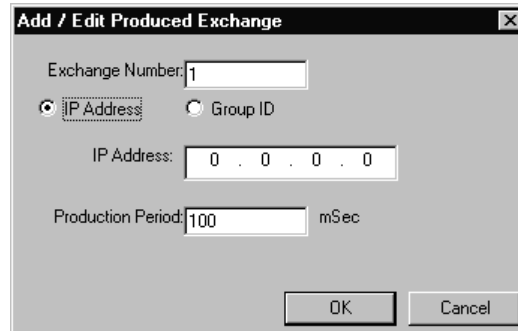
The image shows a dialog box titled "Add / Edit Produced Exchange". It contains several input fields and radio buttons. At the top, "Exchange Number:" is followed by a text box containing the value "1". Below this, there are two radio buttons: "IP Address" (which is selected) and "Group ID". Under the "IP Address" radio button, there is an "IP Address:" label followed by a text box containing "0 . 0 . 0 . 0". Under the "Group ID" radio button, there is no visible text box. At the bottom, there is a "Production Period:" label followed by a text box containing "100" and the unit "mSec". At the very bottom of the dialog are "OK" and "Cancel" buttons.

Figure 6.3 – Add / Edit Produced Exchange Dialog – IP Address Selected

2. Configure the Produced Exchange parameters as follows:

Exchange Number: Enter a number between 1 and 16,383, which will be used to identify the Exchange to be sent.

IP Address Radio Button: Select this option if the Exchange will be sent to a specific Consumer. (This will cause the next edit box to be for entering IP Address, instead of Group ID.)

Group ID Radio Button: Select this option if the Exchange will be sent to a Group of Consumers. (This will cause the next edit box to be for entering Group ID, instead of IP Address.)

IP Address Edit Box: If the IP Address radio button was selected, enter the IP Address of the specific Consumer that will receive the Produced Exchange.

Group ID Edit Box: If the Group ID radio button was selected, enter the Group ID number (1 to 32) of the Group of Consumers that will receive the Produced Exchange.

Production Period: Enter a number (in milliseconds) for how often the Produced Exchange will be sent to the EGD network.

3. Click **OK** to accept the new Produced Exchange configuration, which will now be displayed in the upper window of the Ethernet Global Data Configuration dialog (Figure 6.1).
4. To add another Produced Exchange to the list, click on the **Add Exch** button again.
5. To edit or delete a Produced Exchange, highlight it in the upper window of the Ethernet Global Data Configuration dialog (Figure 6.1), and then click on the **Edit Exch** button or the **Delete Exch** button.
6. To quickly change the **IP Address** in multiple Produced Exchanges, click on the **Replace Addr** button. This will open a dialog to allow the user to search and replace the **IP Address** parameter, in all Produced Exchanges simultaneously.

6.4.2 Defining EGD Produced Exchange I/O Blocks

After creating a Produced Exchange (Section 6.4.1), one or more I/O Blocks should be defined for it. An I/O Block specifies what type and how much information will be associated with the Produced Exchange.

To define I/O Blocks for a Produced Exchange, perform the following six steps:

1. In the upper window of the Ethernet Global Data Configuration dialog (Figure 6.1), highlight one of the Produced Exchanges, and then click the **Add Range** button to open the Add I/O Range to Exchange dialog (Figure 6.4).

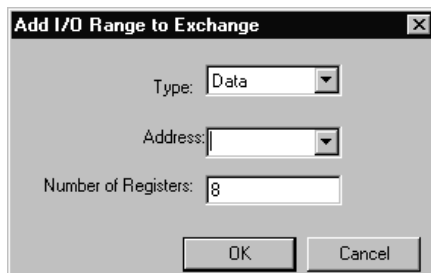


Figure 6.4 – Add I/O Range to Exchange Dialog

2. Configure the I/O Block parameters as follows:

Type: Select Data Type to define a block of OCS registers, which the Produced Exchange will periodically read and send to the EGD network. Select Status Type to define a 16-bit OCS register, which will be written with the Produced Exchange's Status Word.

Address: Enter an OCS Register reference (such as %R1000), for the first OCS register, in a block of OCS registers, that will be periodically read and sent (Data Type), or for a 16-bit OCS register that will be written with status information (Status Type).

Number of Registers: If Data Type was selected, enter the number of registers to be periodically read and sent. If Status Type was selected, this edit box changes to **Number of Bytes**, and always has a fixed value of 2.

3. Click **OK** to accept the Produced Exchange's new I/O Block, which will be now be displayed in the lower window of the Ethernet Global Data Configuration dialog (Figure 6.1).

Note: The OCS registers, specified in Data Block definitions, are sent to the EGD network in top-to-bottom order, as they appear in the lower window of the Ethernet Global Data Configuration dialog (Figure 6.1).

4. To add another I/O Block, click the **Add Range** button again and repeat steps 2 and 3. The new I/O Block will appear at the end of the list.
5. To insert an I/O Block into the middle of the list, highlight one of the I/O Block items in the list and then click on the **Insert Range** button. In this case, the new I/O Block will be inserted just before the highlighted I/O Block.
6. To edit or delete an I/O Block in the list, highlight it and then click on the **Edit Range** or **Delete Range** button.

6.5 EGD Consumed Exchange Configuration

To configure EGD Consumed Exchanges, open the Ethernet Global Data Configuration dialog (Figure 6.2) as described in Section 6.3, and select the **Consumed Exchanges** tab, where:

1. In the upper window, one or more Consumed Exchanges can be created (Section 6.5.1).
2. In the lower window, I/O Blocks can be defined for each Consumed Exchange (Section 6.5.2).

When creating a Consumed Exchange, the application programmer selects an Exchange Number for it, determines whether to receive the Exchange as a single Consumer or as a member of a Group of Consumers, chooses which Producer to receive the Exchange from, and sets how often to expect the Exchange to be received.

When defining I/O Blocks for a Consumed Exchange, the application programmer selects what type and how much *information* will be associated with the Exchange. For Consumed Exchanges, there are five types of I/O Blocks to choose from: Data, Status, Timestamp, OCS Timestamp, and Filler. Table 6.3 shows these I/O Block Types along with their definitions:

Table 6.3 – Consumed Exchange I/O Block Type Definitions	
Type	Definition
Data Block	Block of consecutive OCS registers to be written with received data
Status Block	16-bit OCS register to be written with the Consumed Exchange's status word
Timestamp Block	8-byte binary timestamp indicating when the Producer sampled the data
OCS Timestamp Block	14-byte OCS-format timestamp indicating when the Producer sampled the data
Filler Block	Specifies a block of received data to ignore (skips unwanted data)

When defining **Data Blocks** for a Consumed Exchange, the maximum total OCS register data the Exchange can receive from the EGD network is 1400 bytes. This means that up to a total of 700 16-bit registers (%R, %AI, AQ, etc.), or 11,200 1-bit registers (%M, %T, %I, %Q, etc.), or a combination thereof can be defined for a Consumed Exchange.

Note: The Ethernet Module allows a total of up to 256 Data Blocks to be defined for all Consumed Exchanges combined. This means that if 127 Consumed Exchanges are configured, each can have an average of about 2 Data Blocks defined.

When a **Status Block** is defined for a Consumed Exchange, exactly 2 bytes of register data are written with the Consumed Exchange's Status Word. See Section 6.7 for general information regarding EGD Status Words, and Section 6.7.2 for specific information regarding EGD Consumed Exchange Status Words.

For a **Timestamp Block**, exactly 8 bytes of register data will be written with two 32-bit binary timestamp values, containing the number of seconds and nanoseconds since January 1, 1970.

For an **OCS Timestamp Block**, exactly 14 bytes of register data will be written with a 7-word OCS-format timestamp, consisting of second, minute, hour, day, month, year and millisecond words. This OCS Timestamp can be displayed on the OCS screen, can be loaded into the OCS time-of-day clock, and can be more easily processed by ladder logic.

Note: The binary Timestamp Block is rarely used, and there is never any need to define more than one Status Block or OCS Timestamp Block for a given Consumed Exchange.

A **Filler Block** is used to skip unwanted data sent by the Producer. For example, if the Producer sends 40 data bytes in an Exchange, and the Consumer only needs the first and last 10 bytes, a Filler Block would be defined, in the appropriate slot in the list, to skip the middle 20 bytes of data.

6.5.1 Creating EGD Consumed Exchanges

To create EGD Consumed Exchanges, perform the following six steps:

1. In the Ethernet Global Data Configuration dialog (Figure 6.2), click on the **Add Exch** button to open the Add / Edit Consumed Exchange dialog (Figure 6.5).

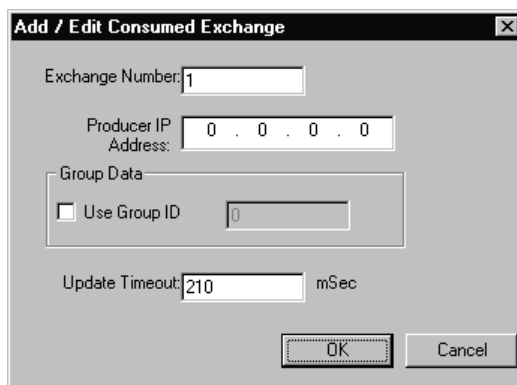
The image shows a dialog box titled "Add / Edit Consumed Exchange". It contains several input fields: "Exchange Number" with the value "1", "Producer IP Address" with the value "0 . 0 . 0 . 0", a "Group Data" section with a checkbox "Use Group ID" and an adjacent input field with the value "0", and "Update Timeout" with the value "210" and the unit "mSec". At the bottom right are "OK" and "Cancel" buttons.

Figure 6.5 – Add / Edit Consumed Exchange Dialog

2. Configure the Consumed Exchange parameters as follows:

Exchange Number: Enter a number between 1 and 16,383, which will be used to identify the Exchange to be received.

Producer IP Address: Enter the IP Address of the Producer to receive the Exchange from.

Use Group ID: Check this box to receive an Exchange sent by a Producer to a Group ID.

Group ID: If the Use Group ID box is checked, enter a number from 1 to 32 in the edit box, to identify which of the 32 possible Groups of Consumers to become a member of.

Update Timeout: Enter a number (in milliseconds) for how long to wait to receive the Exchange, before considering it to be late. Typically, this should be set to twice the corresponding Production Timeout, plus 10 milliseconds.

3. Click **OK** to accept the new Consumed Exchange configuration, which will be now be displayed in the upper window of the Ethernet Global Data Configuration dialog (Figure 6.2).
4. To add another Consumed Exchange to the list, click on the **Add Exch** button again.
5. To edit or delete a Consumed Exchange, highlight it in the upper window of the Ethernet Global Data Configuration dialog (Figure 6.2), and then click on the **Edit Exch** or **Delete Exch** button.
6. To quickly change the **Producer IP Address** in multiple Consumed Exchanges, click on the **Replace Addr** button. This will open a dialog to allow the user to search and replace the **Producer IP Address** parameter, in all Consumed Exchanges simultaneously.

6.5.2 Defining EGD Consumed Exchange I/O Blocks

After creating a Consumed Exchange (Section 6.5.1), one or more I/O Blocks should be defined for it. An I/O Block specifies what type and how much information will be associated with the Consumed Exchange.

To define I/O Blocks for a Consumed Exchange, perform the following six steps:

1. In the upper window of the Ethernet Global Data Configuration dialog (Figure 6.2), highlight one of the Consumed Exchanges, and then click the **Add Range** button to open the Add I/O Range to Exchange dialog (Figure 6.4).

2. Configure the I/O Block parameters as follows:

Type: Select Data Type to define a block of OCS registers, which the Consumed Exchange will write with received data. Select Status Type to define a 16-bit OCS register, which will be written with the Consumed Exchange's Status Word. Select OCS Timestamp Type to define a 14-byte OCS register block to write with the received data's OCS-format timestamp. Select Filler Type, to skip unwanted received data.

Address: Enter an OCS Register reference (such as %R2000), for the first OCS register, in a block of OCS registers, that will be written with received data (Data Type), or for a 16-bit OCS register that will be written with status information (Status Type), or for the first OCS register in a block of OCS registers that will be written with the 14-byte timestamp (OCS Timestamp Type). For Filler Type, the Address edit box is not used.

Number of Registers: If Data Type was selected, enter the number of registers to be written with received data. Otherwise, this edit box changes to **Number of Bytes**, and has a fixed value of 2 for Status Type or 14 for OCS Timestamp Type. For Filler Type, enter the number of unwanted received data bytes to skip.

3. Click **OK** to accept the Consumed Exchange's new I/O Block, which will be now be displayed in the lower window of the Ethernet Global Data Configuration dialog (Figure 6.2).

Note: The OCS registers, specified in Data Block definitions, are filled with received data in top-to-bottom order, as they appear in the lower window of the Ethernet Global Data Configuration dialog (Figure 6.2). In this respect, Filler Blocks are placeholders for Data Blocks and can be thought of as Data Blocks whose received data bytes are discarded.

4. To add another I/O Block, click on the **Add Range** button again and repeat steps 2 and 3. The new I/O Block will appear at the end of the list.
5. To insert an I/O Block into the middle of the list, highlight one of the I/O Block items in the list and then click on the **Insert Range** button. In this case, the new I/O Block will be inserted just before the highlighted I/O Block.
6. To edit or delete an I/O Block in the list, highlight it and then click on the **Edit Range** or **Delete Range** button.

6.6 EGD Operation

Unlike other protocols, EGD protocol stops completely when the OCS (or RCS) is not in RUN mode. In this case, EGD Produced Exchanges are not transmitted, and all received Consumed Exchanges are ignored.

As soon as an OCS (or RCS), containing a configured Ethernet Module, is placed into RUN mode, it will start exchanging EGD messages with other EGD devices on the network, as follows:

1. The Ethernet Module will transmit each configured Produced Exchange (see Section 6.4) periodically, according to its Production Period (in milliseconds).
2. The Ethernet Module will expect to receive each configured Consumed Exchange (see Section 6.5) at least as often as its Update Timeout (in milliseconds).
3. The Ethernet Module will maintain a Status Word for each Produced and Consumed Exchange, as described in Section 6.7, and will report these Status Words in OCS (or RCS) registers, if configured to do so (see Sections 6.4 and 6.5).

6.7 EGD Status Words

EGD Status Words allow an Ethernet Global Data user to obtain the operating status of each EGD Exchange. The set of EGD Status Word values, implemented in the Ethernet Module, is a subset of the ones used in GE Fanuc EGD devices. This is due to the fact that the Ethernet Module does not support dynamically defined (at run time) EGD Exchanges.

Please note that both Produced and Consumed Exchange Status Words are written to local OCS registers. This is the only case where a Produced Exchange can be configured to write to a local register.

The Status Word for a Produced Exchange is updated each time the Exchange's Production Period expires. The Status Word for a Consumed Exchange is updated when new data arrives for consumption or when the Exchange's Update Timeout expires.

In normal operation, each EGD Exchange's Status Word will always be 1 (OK), implying that new data was successfully Produced or Consumed. If the application needs to be notified when a data transfer has occurred on a given Exchange, the ladder program should clear the Exchange's Status Word register to 0 (IDLE) each time a non-zero event is detected.

6.7.1 EGD Produced Exchange Status Words

The Status Word for an EGD Produced Exchange can take on the following values:

- 0 IDLE - No new status event has occurred. The Ethernet Module initializes all Status Words to 0, only at power-up and each time the OCS enters RUN mode. Subsequently, the OCS application ladder program can write the value 0 to the Status Word, as an aid in knowing when Ethernet Global Data production occurs (see Status Word 1).
- 1 OK - Data Produced. New EGD network data has been transmitted. A transition to 1 indicates production of data occurred since the last OCS ladder scan.
- 2 NO LINK - The Ethernet link was down when EGD data production was attempted. This will occur, for example, if the Ethernet cable is unplugged from the OCS.

6.7.2 EGD Consumed Exchange Status Words

The Status Word for an EGD Consumed Exchange can take on the following values:

- 0 IDLE - No new status event has occurred. The Ethernet Module initializes all Status Words to 0, only at power-up and each time the OCS enters RUN mode. Subsequently, the OCS application ladder program can write the value 0 to the Status Word, as an aid in knowing when Ethernet Global Data consumption occurs (see Status Words 1 and 7).
- 1 OK - Data Consumed. New EGD network data has been received as expected (before the configured Update Timeout expired). A transition to 1 (or to 7) indicates consumption of data occurred since the last OCS ladder scan.
- 2 NO LINK - The Ethernet link was down when EGD data consumption was expected. This will occur, for example, if the Ethernet cable is unplugged from the OCS.
- 4 NO SYNC - SNTP Error. The Ethernet Interface in the device producing the exchange is configured for network time synchronization (Network Time Sync parameter is set to SNTP), but is not synchronized to an SNTP server. Therefore, the timestamp associated with this data is not synchronized to the network. Note: This error condition is considered the least important of all the error codes in this section. If another error condition exists, its status code will appear in the Status Word.
- 6 OVERDUE - Data Refresh Error. The Update Timeout has expired without receiving the expected data. Some possible causes for this error are: (1) the Producer has stopped producing the data, (2) the Consumed Exchange's configured Producer IP Address and/or Group ID do not exactly match those of the corresponding Produced Exchange, (3) or the Consumed Exchange's configured Update Timeout is too short (it should normally be twice the producer's Production Period, plus 10 milliseconds).
- 7 TARDY - Data Consumed Late. New EGD network data has been received, but it is later than expected (after the configured Update Timeout expired). A transition to 7 (or to 1) indicates consumption of data occurred since the last OCS ladder scan. The most probable cause for this error is that the Consumed Exchange's configured Update Timeout is too short (it should normally be twice the producer's Production Period, plus 10 milliseconds).

6.8 EGD Example 1

Figure 6.6 shows a simple example, in which Node 1 uses Exchange 1 to transmit 10 words of data to Node 2 every 100 mS, and Node 2 uses Exchange 2 to transmit 5 words of data to Node 1 every 200 mS.

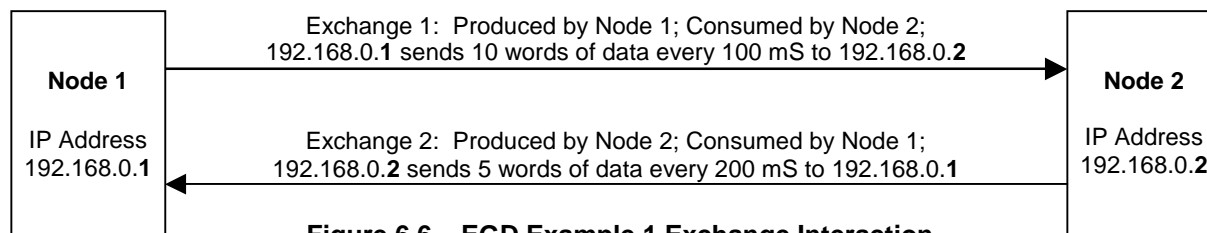


Figure 6.6 – EGD Example 1 Exchange Interaction

6.8.1 EGD Example 1 – Configuring Node 1

To configure Node 1 for EGD Example 1, as shown in Figure 6.6, perform the following six steps:

1. On the main Cscope screen, select **New** on the **File** menu to start a new user program. Then open the Ethernet Module Configuration dialog (Chapter 3, Figure 3.5 [page 19]), and fill in the parameters for Node 1, as shown in Figure 6.7 below.

In this example, Node 1 will have a Static IP Address (Section 3.2.1 [page 21]), and the %R1, %R2 and %R3 registers will be used to report the Ethernet Module's status, firmware version, and IP Address (step 5 in Section 3.1 [page 19]). Note that since an IP Address is 32-bits long, Node 1's IP Address (192.168.0.1) will actually be written into %R3 and %R4.

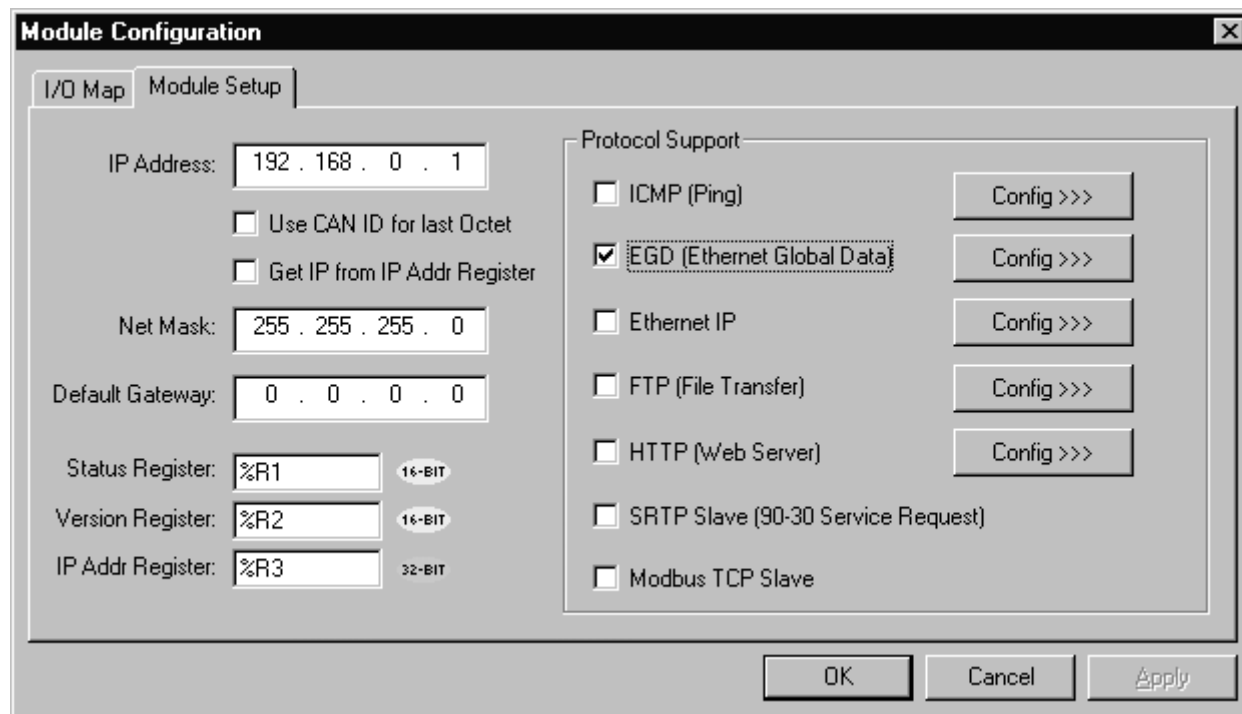


Figure 6.7 - Ethernet Module Configuration - Node 1

2. Click on the **Config** button next to the **EGD (Ethernet Global Data)** checkbox to open the Ethernet Global Data Configuration dialog (Figure 6.1 [page 32]) and click on the **Add Exch** button to create a Produced Exchange. Then, fill in the parameters for Node 1, as shown in Figure 6.8 below, and click **OK**.

In this example, Node 1 will transmit Exchange 1 to a specific Consumer (Node 2), instead of to a Group of Consumers. Also, the Production Period is set to 100, which will cause Node 1 to transmit Exchange 1 every 100 mS.

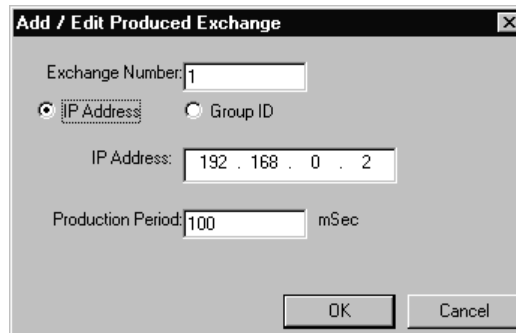


Figure 6.8 – Creating Produced Exchange 1 - Node 1

3. Click on the **Add Range** button, in the EGD Configuration dialog, to define a Data Block for Produced Exchange 1. Then fill in the parameters, as shown in Figure 6.9 below, and click **OK**.

In this example, Node 1 will use Exchange 1, to transmit 10 words of data taken from %R100 through %R109.

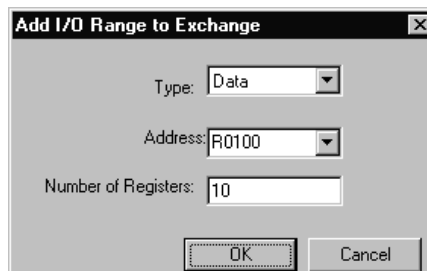


Figure 6.9 – Produced Exchange 1 Data Block Definition – Node 1

At this point, the Produced Exchanges tab in the EGD Configuration dialog should be as shown in Figure 6.10.

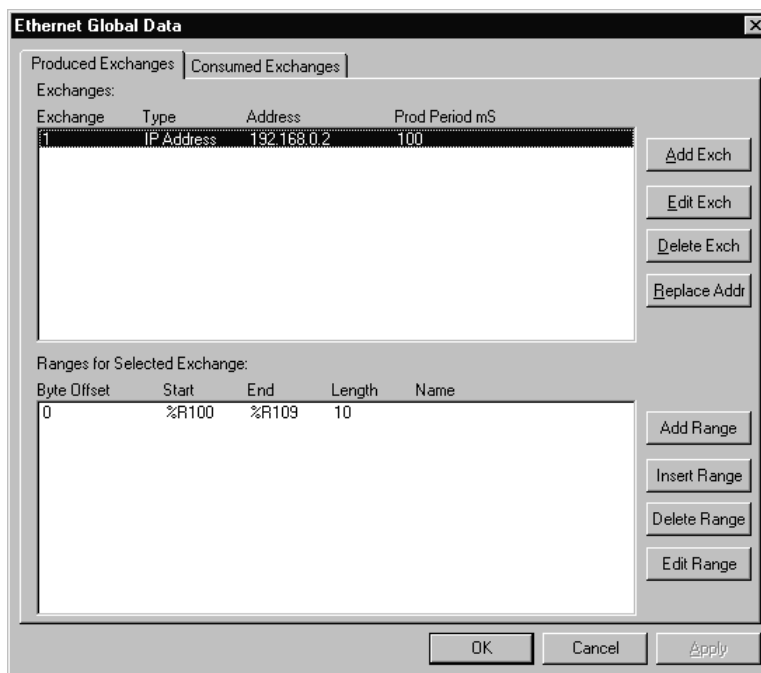


Figure 6.10 – Configured Produced Exchange 1 - Node 1

4. Now that Exchange 1 has been configured as a Produced Exchange for Node 1, it is time to configure Exchange 2 as a Consumed Exchange for Node 1. To do this, first select the **Consumed Exchanges** tab (Figure 6.2 [page 33]) and click on the **Add Exch** button to create a Consumed Exchange. Then fill in the parameters, as shown in Figure 6.11 below, and click **OK**.

In this example, Node 1 will be the only Consumer to receive Exchange 2 from Node 2 and will not be a member of a Group of Consumers. Also, the Update Timeout is set to 410, which means Node 1 will expect to receive Exchange 2 from Node 2 *at least* every 410 mS. Note this is twice the time, plus 10 mS, that Node 2 will be configured to transmit Exchange 2 as recommended in Table 6.1.

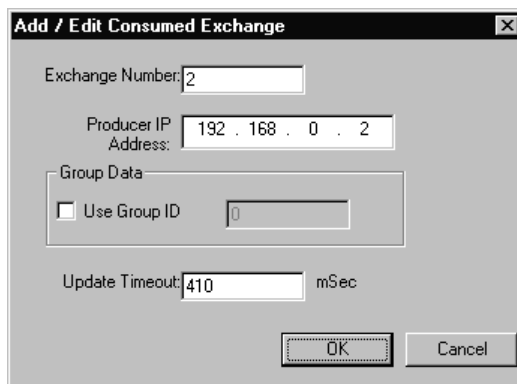


Figure 6.11 - Creating Consumed Exchange 2 - Node 1

5. Click on the **Add Range** button, in the EGD Configuration dialog, to define a Data Block for Consumed Exchange 2. Then fill in the parameters, as shown in Figure 6.12 below, and click **OK**.

In this example, Node 1 will use Exchange 2, to receive 5 data words into %R200 through %R204.

Add I/O Range to Exchange

Type: Data

Address: R0200

Number of Registers: 5

OK Cancel

Figure 6.12 – Consumed Exchange 2 Data Block Definition – Node 1

At this point, the Consumed Exchanges tab in the EGD Configuration dialog should be as shown in Figure 6.13.

Ethernet Global Data

Produced Exchanges Consumed Exchanges

Exchanges:

Exchange	Prod ID	Group ID	Update Timeout mS
2	192.168.0.2	None	410

Buttons: Add Exch, Edit Exch, Delete Exch, Replace Addr

Ranges for Selected Exchange:

Byte Offset	Start	End	Length	Name
0	%R200	%R204	5	

Buttons: Add Range, Insert Range, Delete Range, Edit Range

Buttons: OK, Cancel, Apply

Figure 6.13 – Configured Consumed Exchange 2 - Node 1

6. Node 1 configuration is now complete. Click **OK**, save the user program using an appropriate filename, such as **EGD Node 1.csp**, and then continue with Section 6.8.2 to perform Node 2 configuration.

6.8.2 EGD Example 1 – Configuring Node 2

To configure Node 2 for EGD Example 1, as shown in Figure 6.6, perform the following six steps:

1. On the main Cscape screen, select **New** on the **File** menu to start a new user program. Then open the Ethernet Module Configuration dialog (Chapter 3, Figure 3.5 [page 19]), and fill in the parameters for Node 2, as shown in Figure 6.14 below.

In this example, Node 2 will have a Static IP Address (Section 3.2.1 [page 21]), and the %R1, %R2 and %R3 registers will be used to report the Ethernet Module's status, firmware version, and IP Address (step 5 in Section 3.1 [page 19]). Note that since an IP Address is 32-bits long, Node 2's IP Address (192.168.0.2) will actually be written into %R3 and %R4.

Module Configuration

I/O Map | **Module Setup**

IP Address: 192 . 168 . 0 . 2

☐ Use CAN ID for last Octet

☐ Get IP from IP Addr Register

Net Mask: 255 . 255 . 255 . 0

Default Gateway: 0 . 0 . 0 . 0

Status Register: %R1 16-BIT

Version Register: %R2 16-BIT

IP Addr Register: %R3 32-BIT

Protocol Support

☐ ICMP (Ping) Config >>>

☒ EGD (Ethernet Global Data) Config >>>

☐ Ethernet IP Config >>>

☐ FTP (File Transfer) Config >>>

☐ HTTP (Web Server) Config >>>

☐ SRTP Slave (90-30 Service Request)

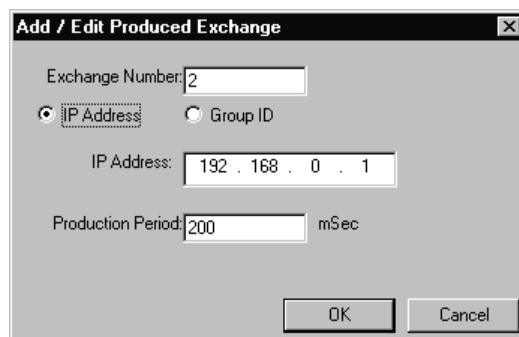
☐ Modbus TCP Slave

OK Cancel Apply

Figure 6.14 - Ethernet Module Configuration - Node 2

2. Click on the **Config** button next to the **EGD (Ethernet Global Data)** checkbox to open the Ethernet Global Data Configuration dialog (Figure 6.1 [page 32]) and click on the **Add Exch** button to create a Produced Exchange. Then, fill in the parameters for Node 2, as shown in Figure 6.15 below, and click **OK**.

In this example, Node 2 will transmit Exchange 2 to a specific Consumer (Node 1), instead of to a Group of Consumers. Also, the Production Period is set to 200, which will cause Node 2 to transmit Exchange 2 every 200 mS.



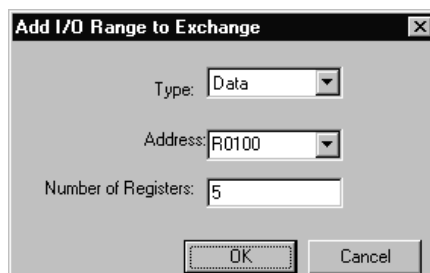
The screenshot shows a dialog box titled "Add / Edit Produced Exchange". It contains the following fields and controls:

- Exchange Number:** A text box containing the value "2".
- Radio Buttons:** Two radio buttons are present: "IP Address" (which is selected) and "Group ID".
- IP Address:** A text box containing the value "192 . 168 . 0 . 1".
- Production Period:** A text box containing the value "200", followed by a label "mSec".
- Buttons:** "OK" and "Cancel" buttons are located at the bottom right of the dialog.

Figure 6.15 – Creating Produced Exchange 2 - Node 2

3. Click on the **Add Range** button, in the EGD Configuration dialog, to define a Data Block for Produced Exchange 2. Then fill in the parameters, as shown in Figure 6.16 below, and click **OK**.

In this example, Node 2 will use Exchange 2, to transmit 5 words of data taken from %R100 through %R104.



The screenshot shows a dialog box titled "Add I/O Range to Exchange". It contains the following fields and controls:

- Type:** A dropdown menu with "Data" selected.
- Address:** A dropdown menu with "R0100" selected.
- Number of Registers:** A text box containing the value "5".
- Buttons:** "OK" and "Cancel" buttons are located at the bottom right of the dialog.

Figure 6.16 – Produced Exchange 2 Data Block Definition – Node 2

At this point, the Produced Exchanges tab in the EGD Configuration dialog should be as shown in Figure 6.17.

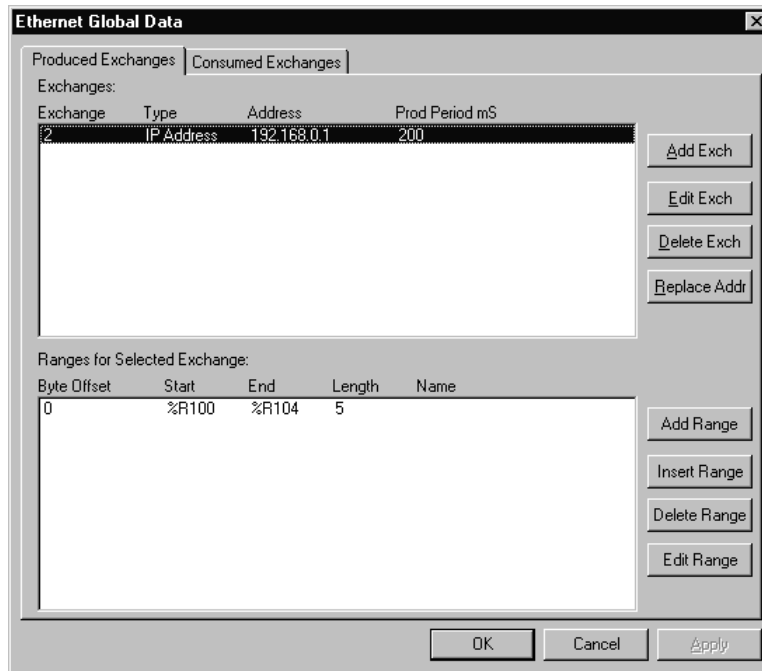


Figure 6.17 – Configured Produced Exchange 2 - Node 2

- Now that Exchange 2 has been configured as a Produced Exchange for Node 2, it is time to configure Exchange 1 as a Consumed Exchange for Node 2. To do this, first select the **Consumed Exchanges** tab (Figure 6.2 [page 33]) and click on the **Add Exch** button to create a Consumed Exchange. Then fill in the parameters, as shown in Figure 6.18 below, and click **OK**.

In this example, Node 2 will be the only Consumer to receive Exchange 1 from Node 1, and will not be a member of a Group of Consumers. Also, the Update Timeout is set to 210, which means Node 2 will expect to receive Exchange 1 from Node 1 *at least* every 210 mS. Note this is twice the time, plus 10 mS, that Node 1 was configured to transmit Exchange 1, as recommended in Table 6.1.

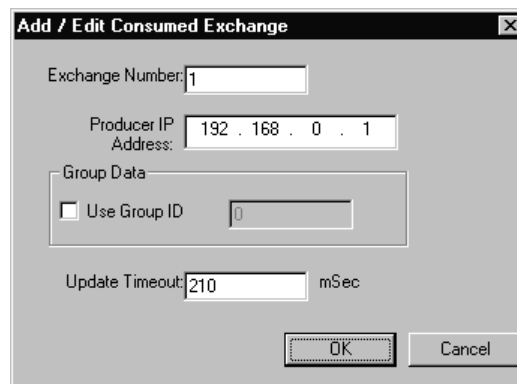


Figure 6.18 - Creating Consumed Exchange 1 - Node 2

5. Click on the **Add Range** button, in the EGD Configuration dialog, to define a Data Block for Consumed Exchange 1. Then fill in the parameters, as shown in Figure 6.19 below, and click **OK**.

In this example, Node 2 will use Exchange 1, to receive 10 data words into %R200 through %R209.

Figure 6.19 – Consumed Exchange 1 Data Block Definition – Node 2

At this point, the Consumed Exchanges tab in the EGD Configuration dialog should be as shown in Figure 6.20.

Exchange	Prod ID	Group ID	Update Timeout mS
1	192.168.0.1	None	210

Byte Offset	Start	End	Length	Name
0	%R200	%R209	10	

Figure 6.20 – Configured Consumed Exchange 1 - Node 2

6. Node 2 configuration is now complete. Click **OK**, save the user program using an appropriate filename, such as **EGD Node 2.csp**, and then continue with Section 6.8.3, to start EGD communication between Node 1 and Node 2.

6.8.3 EGD Example 1 – Starting EGD Communication between Node 1 and Node 2

Now that both Node 1 and Node 2 have been configured for EGD Example 1 (Figure 6.6), start them communicating as follows:

1. Open the saved Node 1 user program (**EGD Node 1.csp**), and download it to Node 1.
2. Open the saved Node 2 user program (**EGD Node 2.csp**), and download it to Node 2.
3. Connect both Node 1 and Node 2 to an Ethernet network and put them both in RUN mode.

At this point, Node 1 and Node 2 should be exchanging EGD data as follows:

1. Every 100 mS, Node 1 will read its %R100 through %R109 registers and send them via Exchange 1 to be received by Node 2 into Node 2's %R200 through %R209 registers.
2. Every 200 mS, Node 2 will read its %R100 through %R104 registers and send them via Exchange 2 to be received by Node 1 into Node 1's %R200 through %R204 registers.

6.9 EGD Example 2

Building upon EGD Example 1, EGD Example 2 will demonstrate how to add Status, Timestamp, and Filler Blocks to Exchanges in Node 1.

6.9.1 EGD Example 2 – Adding a Status Block

To get a Produced or Consumed Exchange's Status Word (Section 6.7 [page 40]) into an OCS register, define a Status Block for the Exchange. To do this for Node 1's Produced Exchange 1, for example, perform the following two steps:

1. On the main Cscape screen, select **Open** from the **File** menu and open the saved Node 1 user program (**EGD Node 1.csp**). Then open the Ethernet Module Configuration dialog (Figure 6.7 [page 40]) and click on the **Config** button next to the **EGD (Ethernet Global Data)** checkbox to open the EGD Configuration dialog (Figure 6.10 [page 44]).
2. Make sure Exchange 1 is highlighted in the upper window (it should be the only Produced Exchange in this example) and click on the **Add Range** button, to define a Status Block for Produced Exchange 1. Then fill in the parameters, as shown in Figure 6.21 below, and click **OK**.

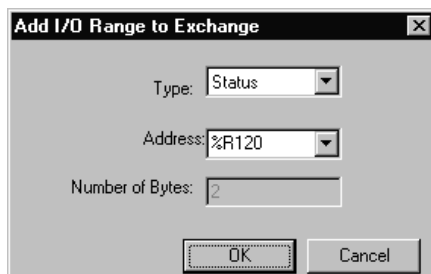


Figure 6.21 – Produced Exchange 1 Status Block Definition – Node 1

In this example, when the status of Node 1's Produced Exchange changes, its new Status Word (Section 6.7 [page 40]), will be written into the Node 1 register %R120. Notice that the **Number of Bytes** parameter is not configurable and will always be 2 for a Status Block.

At this point, the Produced Exchanges tab in the EGD Configuration dialog should be as shown in Figure 6.22.

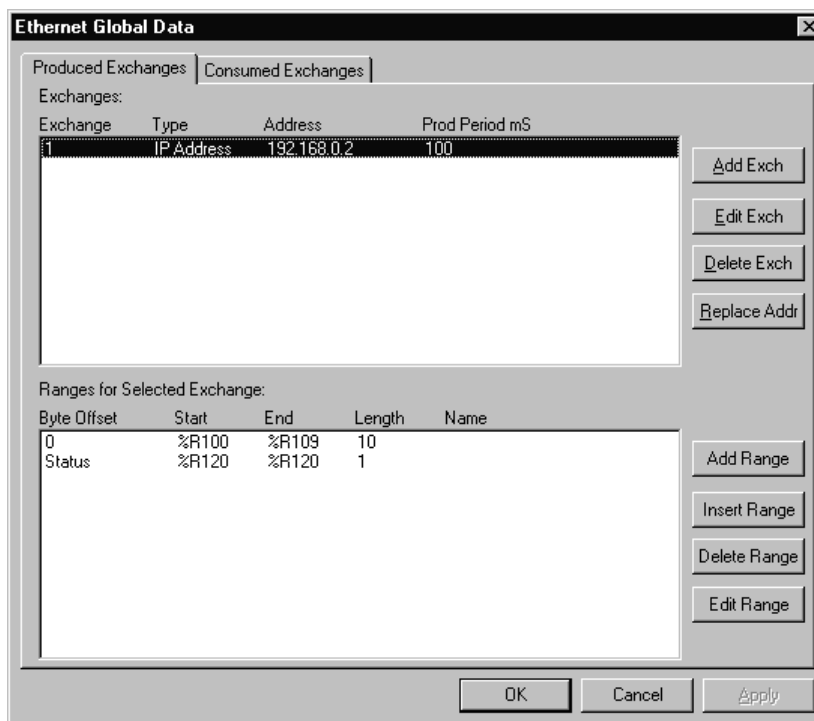


Figure 6.22 – Produced Exchange 1 with Status Block – Node 1

6.9.2 EGD Example 2 – Adding an OCS Timestamp Block

In EGD protocol, a timestamp is sent with every Produced Exchange, indicating when the Producer sampled the data being sent. For a Consumer of the Exchange to get this information into an OCS register, define an OCS Timestamp Block for the Consumed Exchange. To do this for Node 1's Consumed Exchange 2, for example, perform the following 2 steps:

1. In the EGD Configuration dialog (Figure 6.22), select the **Consumed Exchanges** tab.
2. Make sure Exchange 2 is highlighted in the upper window (it should be the only Consumed Exchange in this example) and click on the **Add Range** button, to define an OCS Timestamp Block for Consumed Exchange 2. Then fill in the parameters, as shown in Figure 6.23 below, and click **OK**.

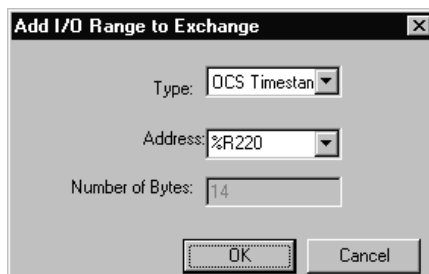


Figure 6.23 – Consumed Exchange 2 OCS Timestamp Block Definition – Node 1

In this example, when Node 1 receives Consumed Exchange 2 from Node 2, its timestamp will be converted into a 14-byte OCS-format timestamp, and will be written into the Node 1 registers %R220 through %R226. Notice that the **Number of Bytes** parameter is not configurable and will always be 14 for an OCS Timestamp Block.

At this point, the Consumed Exchanges tab in the EGD Configuration dialog should be as shown in Figure 6.24.

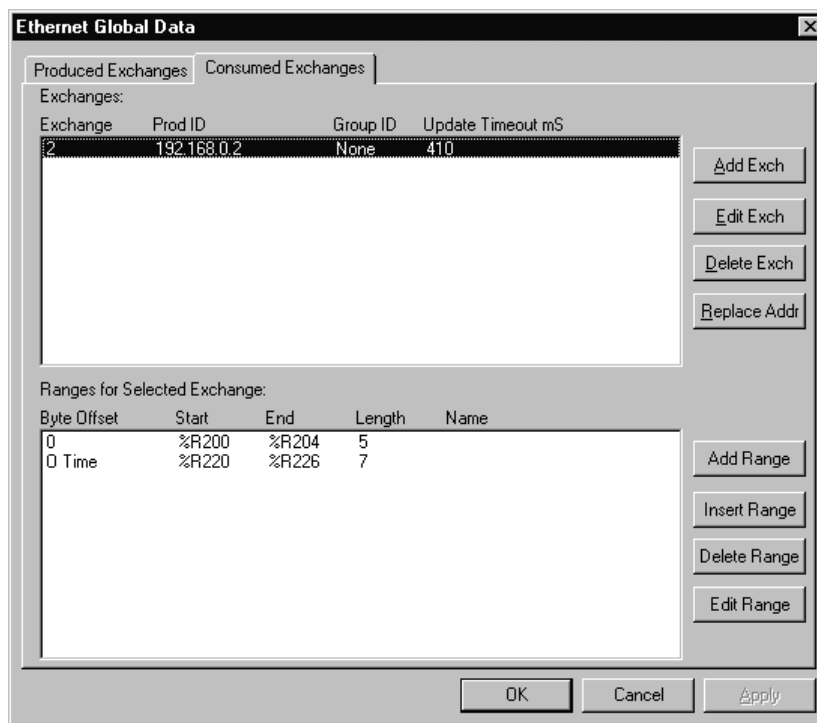


Figure 6.24 – Consumed Exchange 2 with Data and OCS Timestamp Blocks – Node 1

6.9.3 EGD Example 2 – Adding a Filler Block

Sometimes a Consumer does not need all of the data sent by a Producer in an Exchange. In this case, the Consumed Exchange's Data Block should be defined to receive fewer registers than will be sent by the Producer of the Exchange.

However, if the partial data needed by the Consumer is not at the *beginning* of the data received in an Exchange, there must be some way to skip (ignore) the extra data. For this reason, the Ethernet Module supports Filler Blocks for Consumed Exchanges.

For example, if Node 1 only needs the 5th word of the 5 data words sent by Node 2 in Exchange 2, skip the 1st four data words by defining a Filler Block for Node 1's Consumed Exchange 2. To do this, perform the following 2 steps:

1. Referring the Figure 6.24, highlight Exchange 2 in the upper window, highlight the first entry in the lower window, and click on the **Edit Range** button. Change the Data Block into a Filler Block, by editing the parameters as shown in Figure 6.25a below, and then click **OK**.

2. With Exchange 2 still highlighted in the upper window, click on the **Add Range** button, to define a new Data Block for Consumed Exchange 2. Then fill in the parameters, as shown in Figure 6.25b below, and click **OK**.

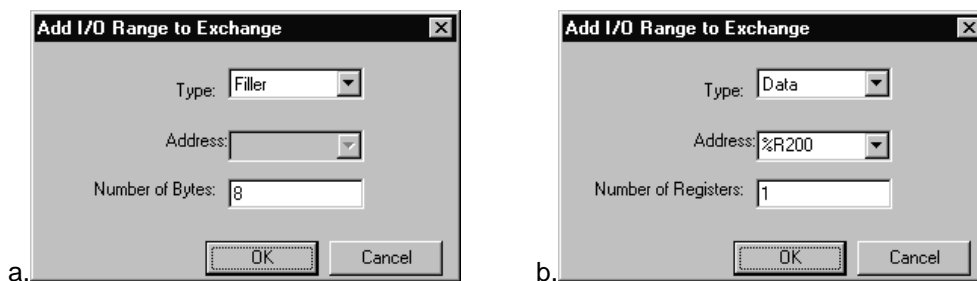


Figure 6.25 – Consumed Exchange 2 Filler and Data Block Definitions – Node 1

In this example, the first 4 data words (8 bytes) received by Node 1 in Exchange 2 will be skipped, and the 5th data word will be written to Node 1 register %R200.

At this point, the Consumed Exchanges tab in the EGD Configuration dialog should be as shown in Figure 6.26.

Note: OCS Timestamp and Status Blocks contain overhead information, and are not considered to be part of an Exchange's data. Therefore the position of the OCS Timestamp Block in the lower window of Figure 6.26, does not affect the sequence of the data received in the Exchange. Only Data Blocks and Filler Blocks are position-sensitive.

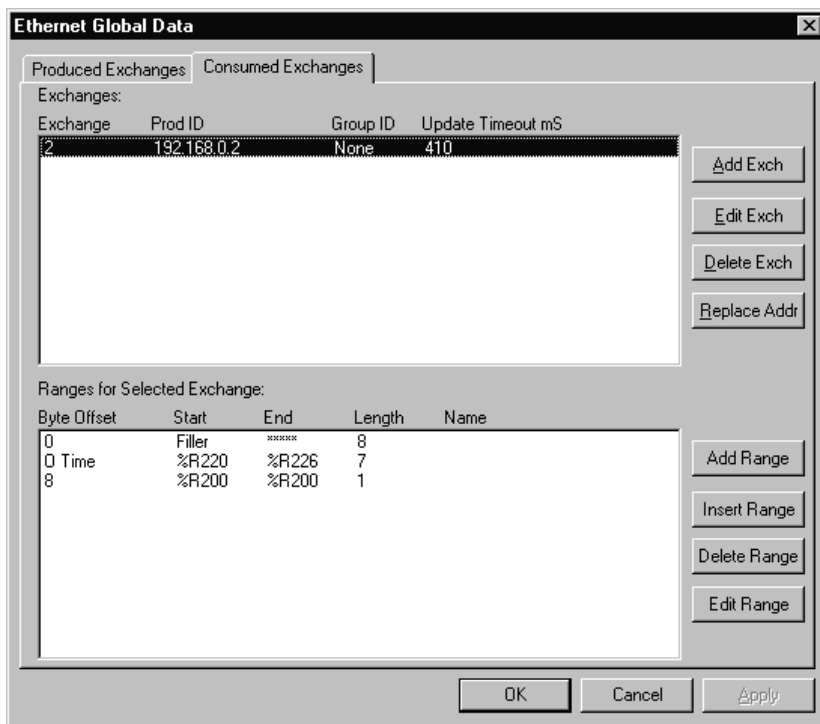


Figure 6.26 - Consumed Exchange 2 with Filler, OCS Timestamp and Data Blocks - Node 1

CHAPTER 7: SERVICE REQUEST TRANSFER PROTOCOL (SRTP)

7.1 SRTP Overview

Service Request Transfer Protocol (SRTP) is a GE Fanuc Automation protocol, which allows a remote SRTP Client to request services from an SRTP Server. In this context, the Ethernet Module acts as an SRTP Server, which responds to requests from one or more SRTP Clients.

Since SRTP was originally designed to support services provided by GE Fanuc Series 90 PLC products, the Ethernet Module's SRTP protocol, for OCS and RCS products, does not support all possible SRTP services. The Ethernet Module's implementation of SRTP is mainly limited to those services required for the exchange of PLC register data.

The following SRTP service requests are supported by the Ethernet Module:

Request Code	Service Name
0	PLC_SSTAT
1	PLC_LSTAT
4	READ_SMEM
7	WRITE_SMEM
33	CHG_PRIV_LEVEL
67	RET_CONFIG_INFO
79	SESSION_CONTROL
97	PLC_FEATURES_SUPP

The READ_SMEM and WRITE_SMEM requests are supported for the following register types:

Selector	Register	Type
8	%R	16 bit
10	%AI	16 bit
12	%AQ	16 bit
16	%I	8 bit
18	%Q	8 bit
20	%T	8 bit
22	%M	8 bit
30	%S	8 bit
70	%I	1 bit
72	%Q	1 bit
74	%T	1 bit
76	%M	1 bit
84	%S	1 bit

7.2 SRTP Configuration

If SRTP protocol will be used in the application, the general Ethernet Module Configuration, previously described in Chapter 3, must be performed, and SRTP must be enabled. To enable SRTP protocol, use Cscape Programming Software, to perform the following two steps:

1. Open the Ethernet Module Configuration dialog (Figure 3.5 [page 19]) as described in Chapter 3.
2. Enable SRTP by checking the **SRTP Slave (90-30 Service Request)** checkbox in the Module Configuration dialog (Figure 3.5).

No additional configuration is required for SRTP protocol. As long as SRTP is enabled and the Ethernet Module has been assigned an IP Address and Net Mask, it will respond to service requests from an SRTP Client, such as a PC running GE Fanuc CIMPLICITY Plant Edition software.

7.3 SRTP Operation

With respect to SRTP protocol, the Ethernet Module's behavior depends on whether the OCS is in RUN mode or not.

When the OCS is in RUN mode, the Ethernet Module responds normally to SRTP Client requests.

When the OCS is not in RUN mode, the Ethernet Module still responds to SRTP Client requests, with the following differences:

1. When a WRITE_SMEM request is received from an SRTP Client, to write a block of OCS registers, a reply is sent to the client, but the data is not written to OCS registers.
2. When a READ_SMEM request is received from an SRTP Client, to read a block of OCS registers, a reply is sent to the client, but the data returned to the client is the last known values of the requested OCS registers, and is not actually read from the OCS registers.
3. For all requests received from an SRTP Client, the reply will indicate to the client that the OCS is not in RUN mode, by returning the RUN DISABLED code in the Piggyback Status word.

CHAPTER 8: MODBUS TCP SLAVE PROTOCOL

8.1 Modbus Overview

Modbus TCP is a Master / Slave protocol, which allows a remote Modbus TCP Master (client) to request services from a Modbus TCP Slave (server). In this context, the Ethernet Module acts as a Modbus TCP Slave, which responds to requests from one or more Modbus TCP Masters.

All Modbus requests that contain the Ethernet Module's IP Address are serviced. The master needs to be configured with the Ethernet Module's IP Address, and most masters also require the slave unit number. Since each Ethernet Module must have its own unique IP Address, the slave unit number is not relevant and is discarded by the Ethernet Module.

To access OCS registers, a Modbus TCP Master must be configured with the appropriate register type and offset. This is usually accomplished with one of two methods:

The first method uses either **Traditional Modbus References** or **Expanded Modbus References**, in which the high digit represents the register type and the lower digits represent the register offset (starting with register 1 for each type). Since only four Modbus register types (0, 1, 3 and 4) can be represented in this manner, the Ethernet Module's Modbus implementation packs several OCS register types into each Modbus register type. Starting addresses of each OCS register type are shown in the **Traditional Modbus Reference** and **Expanded Modbus Reference** columns of Table 8.1.

The second method requires the Modbus TCP Master to be configured with a specific **Modbus Command and Offset**. The supported Modbus commands and the associated offsets are also illustrated in Table 8.1.

Table 8.1 - Modbus Master Mapping					
OCS Reference	Maximum Range	Traditional Modbus Reference	Expanded Modbus Reference	Modbus Command(s)	Modbus Offset
%Q1	2048	00001	000001	Read Coil Status (1) Force Single Coil (5) Force Multiple Coils (15)	00000
%M1	2048	03001	003001		03000
%T1	2048	06001	006001		06000
%QG1	256	09001	009001		09000
%I1	2048	10001	100001	Read Input Status (2)	00000
%IG1	256	13001	103001		03000
%S1	256	14001	104001		04000
%K1	256	15001	105001		05000
%AI1	512	30001	300001	Read Input Register (4)	00000
%AIG1	32	33001	303001		03000
%SR1	32	34001	304001		04000
%AQ1	512	40001	400001	Read Holding Registers (3) Preset Single Register (6) Preset Multiple Registers (16)	00000
➤ %R1	2048	43001	403001		03000
%AQG1	32	46001	406001		06000
❖ %R1	9999	-	410001		10000

- All Ethernet modules support legacy register accessing (%R1-2048).
- ❖ Only ETN300 modules support extended register accessing (%R1-9999).

8.2 Modbus Configuration

If Modbus TCP Slave protocol will be used in the application, the general Ethernet Module Configuration, previously described in Chapter 3, must be performed, and Modbus must be enabled. To enable Modbus protocol, use Cscape Programming Software, to perform the following two steps:

1. Open the Ethernet Module Configuration dialog (Figure 3.5 [page 19]), as described in Chapter 3.
2. Enable Modbus by checking the **Modbus TCP Slave** checkbox in the Module Configuration dialog (Figure 3.5).

No additional configuration is required for Modbus protocol. As long as Modbus is enabled and the Ethernet Module has been assigned an IP Address and Net Mask, it will respond to service requests from a Modbus TCP Master (client), such as a PC running GE Fanuc CIMPLICITY Plant Edition software.

8.3 Modbus Operation

With respect to Modbus TCP Slave protocol, the Ethernet Module's behavior depends on whether the OCS is in RUN mode or not.

When the OCS is in RUN mode, the Ethernet Module responds normally to Modbus TCP Master requests.

When the OCS is not in RUN mode, the Ethernet Module still responds to Modbus TCP Master requests, with the following differences:

1. When a request is received from a Modbus TCP Master to write a block of OCS registers, a reply is sent to the Master, but the data is not written to OCS registers.
2. When a request is received from a Modbus TCP Master, to read a block of OCS registers, a reply is sent to the Master, but the data returned to the Master is all zeroes.
3. A Modbus TCP Master can determine if the OCS is RUN mode in either of the following two ways:
 - a. By reading the OCS Always On register (%S7), which is **1** if the OCS is in RUN mode and **0** if the OCS is not in RUN mode (because of item 2 above).
 - b. By issuing a Read Exception Status request, which returns an Exception Status of **0** if the OCS is in RUN Mode or **1** if the OCS is not in RUN mode.

CHAPTER 9: ETHERNET/IP PROTOCOL

9.1 Ethernet/IP Overview

Ethernet/IP protocol is ODVA's Common Industrial Protocol (CIP) over Ethernet. Figure 9.1 illustrates the protocol layers. Ethernet/IP starts at the Ethernet Physical Layer, and moves up through the IP, TCP/UDP, and Encapsulation Layers. It is beyond the scope of this User Manual to discuss the specifics of the Ethernet/IP protocol. See the Ethernet/IP specification at <http://www.odva.org> for more details of the Ethernet/IP protocol.

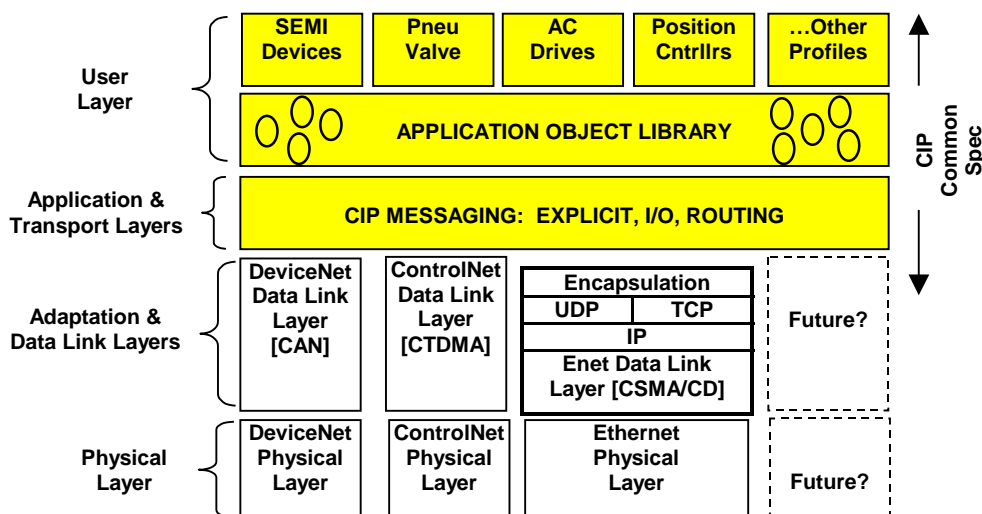


Figure 9.1 - Ethernet/IP Protocol Layers

The Ethernet Module provides Ethernet/IP server capability, and implements the following CIP objects:

- Identity → Instance #1
- Connection Manager → Instance #1
- Message Router → Instance #1
- Assembly → Instance #100 (0x64) for Consumed data
- Assembly → Instance #101 (0x65) for Produced data
- TCP/IP → Instance #1
- Ethernet Link → Instance #1

The Ethernet Module has an Unconnected Message Manager (UCMM), which forwards the following services on to the addressed object:

- Forward Open (Connection Manager object service)
- Forward Close (Connection Manager object service)
- Get Attribute Single
- Set Attribute Single

It will forward the above services on to the addressed object, however the addressed object may or may not support the service.

The Ethernet Module supports Class 3 connections for explicit messaging, and Class 1 connections for I/O (Implicit) messaging.

The Ethernet Module can be configured to produce and consume 0 bytes of data up to 256 bytes of data.

The TCP connections used for Ethernet/IP have an inactivity timeout of 60 seconds. In the event of an inactivity timeout, the TCP connection will automatically close.

9.2 Ethernet/IP Configuration

If Ethernet/IP protocol will be used in the application, Ethernet/IP configuration must be performed, in addition to the general Ethernet Module Configuration previously described in Chapter 3. To configure Ethernet/IP protocol, use Cscape Programming Software to perform the following five steps:

1. Open the Ethernet Module Configuration dialog (Figure 3.5 [page 19]) as described in Chapter 3.
2. Enable Ethernet/IP by checking the **Ethernet IP** checkbox in the Module Configuration dialog (Figure 3.5).
3. Click on the **Config** button next to the **Ethernet IP** checkbox to open the Ethernet/IP Configuration dialog (Figure 9.2).
4. Set the location and size of Produced and Consumed data.
5. Also, set the location of the 16-bit Ethernet/IP Status word.

The screenshot shows the 'Ethernet IP Configuration' dialog box. It is divided into three main sections. The first section, 'Produced (Sent from Controller to Network)', contains two input fields: 'Starting Register' and 'Number of Words' (with a value of 0). The second section, 'Consumed (Sent from Network to Controller)', also contains two input fields: 'Starting Register' and 'Number of Words' (with a value of 0). The third section, 'Status', contains a 'Register' input field and a '16-BIT' label. At the bottom of the dialog are 'OK' and 'Cancel' buttons.

Figure 9.2 - Ethernet IP Configuration Dialog

9.3 Ethernet/IP Operation

After configuring the Ethernet Module as described in section 9.2, it is ready to respond to Ethernet/IP requests. The Ethernet Module handles unconnected requests anytime.

In order to exchange I/O data with the Ethernet Module, a class 1 connection pair must be established. The connection pair consists of a class 1 originator-to-target connection and a class 1 target-to-originator connection, both of which should be set for point-to-point Transport Type. The data sizes must match the Ethernet Module's configured Consumed and Produced data sizes, respectively. The application path should be set as follows:

- The originator-to-target connection point should be set to 100 (0x64), which stands for the assembly class (4), instance 100 (0x64).
- The target-to-originator connection point should be set to 101 (0x65), which stands for the assembly class (4), instance 101 (0x65).

There is NO configuration assembly instance. All of the class 1 connection pair setup data is transmitted through a forward open service to the connection manager.

Once the class 1 connection pair is established, I/O data is exchanged to/from the Ethernet Module through these connections. The Consumed data (data coming from the scanner) and Produced data (data going to the scanner) are available at the locations within the OCS register map as configured.

The TCP connections used for all encapsulated messages and explicit messages have an inactivity timeout of 60 seconds. If the TCP connections are required to stay open, a NOP encapsulated command can be used at a periodic rate to keep the TCP connection open.

When the OCS is not in RUN mode, Produced data is all zeros, and Consumed data is not written to the configured registers within the OCS. When the OCS is in RUN mode, the configured Produced OCS registers are read and sent as Produced data, while Consumed data is received and written to the configured Consumed OCS registers.

The Status word provides Ethernet/IP connection status. The upper byte of the word contains the Class 3 (Explicit) connection count and the lower byte contains the Class 1 (IO) connection count.

Note: When the Status word indicates no connections, the Consumed OCS registers contain old data.

NOTES

CHAPTER 10: FTP SERVER PROTOCOL

10.1 FTP Overview

File Transfer Protocol (FTP) is a standard Client/Server Internet protocol, based on RFC959, which supports efficient and reliable file transfers over a TCP/IP network. In this context, the Ethernet Module acts as an FTP Server, which responds to file transfer requests from one or more FTP Clients.

Note: As indicated in Table 1.2 (page 9), FTP protocol is supported only by OCS Models, which have built-in Ethernet *and* an OCS file system with removable media (such as Compact Flash). For OCS Models that do not have an OCS file system, the **FTP (File Transfer)** checkbox will be grayed out in Cscape's Ethernet Module Configuration dialog.

The Ethernet Module's FTP Server supports both anonymous and authenticated file transfers between an FTP Client and the OCS file system. Authenticated file transfers require that the FTP Client provide one of two possible User Name and Password pairs, before the OCS file system can be accessed.

Properly authenticated, an FTP Client can access OCS file system functions, which include file read, file create, file delete, file rename, file write, directory read, directory create and directory delete.

10.2 FTP Configuration

If FTP File Transfer will be used in the application, FTP Configuration must be performed, in addition to the general Ethernet Module Configuration previously described in Chapter 3. To configure FTP protocol, use Cscape Programming Software to perform the following five steps:

1. Open the Ethernet Module Configuration dialog (Figure 3.5 [page 19]), as described in Chapter 3.
2. Enable FTP by checking the **FTP (File Transfer)** checkbox in the Ethernet Module Configuration dialog (Figure 3.5). Until this is done, the OCS will not respond to *any* FTP Client requests.
3. Click on the **Config** button next to the **FTP (File Transfer)** checkbox to open the FTP Configuration dialog (Figure 10.1).

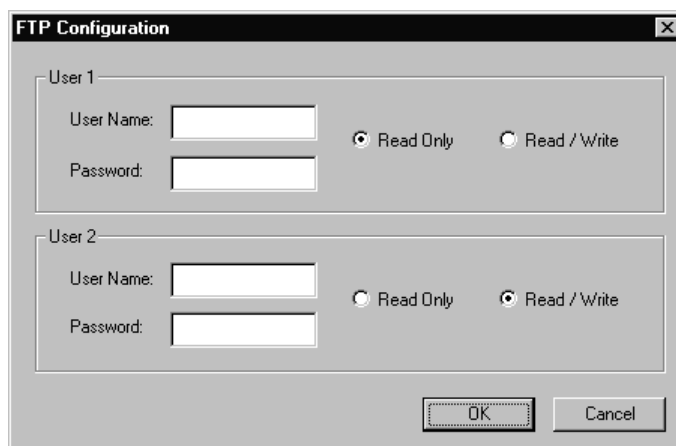


Figure 10.1 – FTP Configuration Dialog

4. Optionally, set up the FTP Configuration parameters for **User 1** and/or **User 2** as follows:

User Name: Enter an alphanumeric string (up to 40 characters) for the FTP Client's User Name.
Password: Enter an alphanumeric string (up to 40 characters) for the FTP Client's Password.
Read Only: Select this option to give the FTP Client read-only access to the OCS file system.
Read / Write: Select this option to give the FTP Client read-write access to the OCS file system.

Note: User Names and Passwords are case-sensitive and, by default, User Name and Passwords for both User 1 and User 2 are empty. In this case, *anonymous* FTP Clients can access OCS files, but file access will be read-only. In a typical application, both User 1 and User 2 should be configured - one with read-only access and the other with read-write access.

5. Click **OK** to accept the new FTP protocol configuration.

10.3 FTP Operation

After performing Ethernet Module FTP Configuration, as described in section 10.2, a 3rd party FTP Client (such as a PC running SmartFTP) can be used to access the OCS file system.

The target Ethernet Module's configured IP Address, FTP User Name (if any) and FTP Password (if any), must be entered into the FTP Client, to establish an FTP connection before any file accesses can be initiated. Please refer to documentation provided with the 3rd party FTP Client, regarding how to install, configure and operate the FTP Client.

When configuring an FTP Client, the user should be aware that the FTP protocol standard defines many optional features. Similar to many Unix implementations, the Ethernet Module FTP Server imposes constraints on some of these options, as shown in Table 10.1 below.

Table 10.1 – Ethernet Module FTP Server Option Constraints		
FTP Option	Supported	Not Supported
File Transfer Type:	Binary Type	ASCII Type & EBCDIC Type
File Format:	Non-Print Format	Telnet Format & Carriage Control Format
File Structure:	File Structure	Record Structure & Page Structure
Transmission Mode:	Stream Mode	Block Mode & Compressed Mode
Data Connection Mode:	Active (PORT) Mode	Passive (PASV) Mode

In addition, since the Ethernet Module FTP Server will automatically disconnect an FTP Client after about 3 minutes of inactivity, the user might need to configure the FTP Client to periodically send NOOP (No Operation) commands to the FTP Server, to keep the connection alive.

10.4 FTP File Accessing

When using FTP to exchange files with an OCS, the user should know that the OCS file system implements an "8.3" filename format, which means all file and directory names must consist of up to 8 characters, followed by an optional dot, and an optional extension with up to 3 characters.

Also, the OCS file system allows multiple concurrent file accessing. For example, an FTP Client can read a file at the same time the OCS ladder program is logging data to another file. It is also possible for both the FTP Client and OCS ladder functions to read the same file at the same time.

If there is a file access conflict, such as an FTP Client attempting to delete a file that is currently being read or written by ladder, the file delete request is denied and an error response is sent to the FTP Client.

CHAPTER 11: HTTP SERVER PROTOCOL

11.1 HTTP Overview

HyperText Transfer Protocol (HTTP) is a standard Client/Server Internet protocol, based on RFC1945, which transfers web content over a TCP/IP network. In this context, the Ethernet Module acts as an HTTP Server (web content provider), which responds to requests from HTTP Clients (web browsers).

Note: As indicated in Table 1.2 (page 9), HTTP protocol is supported only by OCS Models, which have built-in Ethernet *and* an OCS file system with removable media (such as Compact Flash). For OCS Models that do not have an OCS file system, the **HTTP (Web Server)** checkbox will be grayed out in Cscope's Ethernet Module Configuration dialog.

The Ethernet Module's HTTP Server supports optional HTTP Client authentication, requiring the web browser operator to enter a User Name and Password, before web content can be accessed.

Properly authenticated, an HTTP Client can access web content previously stored in the OCS file system. This web content can include standard static HTML files and other web resource files. In addition, the HTTP Server allows the HTTP Client to display and edit OCS register data, by embedding specially named dynamic HTML files (**readreg.htm** and **writereg.htm**) into a static HTML file.

11.2 HTTP Configuration

If the HTTP Web Server will be used in the application, HTTP Configuration must be performed, in addition to the general Ethernet Module Configuration previously described in Chapter 3. To configure HTTP protocol, use Cscope Programming Software to perform the following five steps:

1. Open the Ethernet Module Configuration dialog (Figure 3.5 [page 19]), as described in Chapter 3.
2. Enable HTTP by checking the **HTTP (Web Server)** checkbox in the Ethernet Module Configuration dialog (Figure 3.5). Until this is done, the OCS will not respond to *any* HTTP Client requests.
3. Click on the **Config** button next to the **HTTP (Web Server)** checkbox to open the HTTP Configuration dialog (Figure 11.1).

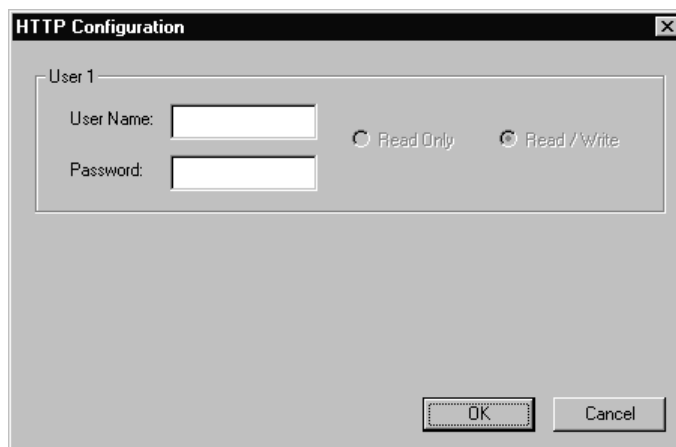


Figure 11.1 – HTTP Configuration Dialog

4. Optionally, set up the HTTP Configuration parameters for **User 1** as follows:

User Name: Enter an alphanumeric string (up to 40 characters) for the HTTP Client's User Name.
Password: Enter an alphanumeric string (up to 40 characters) for the HTTP Client's Password.
Read Only: This option is always deselected and grayed out.
Read / Write: This option is always selected and grayed out.

Note: User Names and Passwords are case-sensitive and, by default, the User Name and Password are empty. In this case, *all* HTTP Clients can access the OCS web content. In a typical application, the User Name and Password should be configured for security purposes.

5. Click **OK** to accept the new HTTP protocol configuration.

11.3 HTTP Operation

After performing Ethernet Module HTTP Configuration, as described in section 11.2, a 3rd party HTTP Client (such as a PC running Internet Explorer or Firefox) can be used to browse OCS web content.

To start browsing web resources stored in the OCS file system, enter the appropriate URL (Uniform Resource Locator) into the HTTP Client (web browser). As a minimum, the URL should include the target Ethernet Module's IP Address and a filename. For example, if the OCS Ethernet Module's IP address is 192.168.254.128, the following URL will access the **index.htm** file in the OCS file system root directory:

`http://192.168.254.128/index.htm`

Note: The HTTP Server does not automatically default to **index.htm**, if no file is specified in the URL.

If the Ethernet Module was configured with a User Name and Password, the browser will prompt for the User Name and Password to be entered, before opening the web page. Please refer to documentation provided with the 3rd party HTTP Client, regarding how to install, configure and operate the HTTP Client.

The Ethernet Module HTTP Server implements HTTP/1.0 and does not support persistent connections. Therefore, each time a web page (or other resource) is transferred, the following sequence takes place:

1. The HTTP Client (web browser) opens a connection with the HTTP Server and sends a request.
2. The HTTP Server sends a response to the HTTP Client and then closes the connection.

Since the connection only stays open long enough to complete one request, multiple HTTP Clients can concurrently access OCS web content, even though they are actually *sharing* a single connection.

11.4 Web Content - Storing

When storing web content files in the OCS file system, the user should know that the OCS file system implements an "8.3" filename format, which means all file and directory names must consist of up to 8 characters, followed by an optional dot, and an optional extension with up to 3 characters.

Web content files can be stored into the OCS file system media (such as Compact Flash), by temporarily plugging the media into a PC with an installed card reader, or via Ethernet using FTP (File Transfer Protocol). See Chapter 10 for information on using the Ethernet Module FTP Server.

To keep the web content files organized, multiple files, directories and subdirectories can be created on the OCS file system media as desired. Normally, it is a good practice to store an **index.htm** file in the root directory as a "home page", but this is not required.

11.5 Web Content - Dynamic Data

An OCS HTML file is either *static* or *dynamic*. The application engineer creates a static HTML file and stores it on the OCS file system media. As a result of special markups in the static HTML file, the HTTP Server creates dynamic HTML files to access OCS Registers. This technique allows a static HTML file to read and write OCS registers dynamically.

11.5.1 Static Html Files

Custom HTML files are created to suit the application. These files can be created with a simple text editor or by using a 3rd party HTML Editor.

Within the static HTML file, <iframe> and <form> tags respectively read and write OCS Register data, by referencing two specially named dynamic HTML files (**readreg.htm** and **writereg.htm**).

11.5.2 Dynamic Html Files – Reading OCS Register Data (readreg.htm)

OCS Register data is read by placing an <iframe> tag, that references **readreg.htm**, within a static HTML file. The following example <iframe> tag reads and displays OCS Register %R11 as an Unsigned Integer every two seconds:

```
<iframe src="readreg.htm;reg-R11;fmt-UINT;rfs-2"></iframe>
```

The required **src** attribute is a quoted string containing the **readreg.htm** reference and two or three parameters, all separated by semicolons. The **reg**, **fmt** and **rfs** parameters specify the OCS Register to read, the display format to use and optionally how often to refresh the data. These parameters are described in more detail in the following 3 sections:

11.5.2.1 **reg** Parameter – OCS Register Read Reference

Examples: reg-R11
 reg-R11.3

The required **reg** parameter is followed by a dash and an OCS Register reference. Register reference numbers start at 1 and the maximum depends on the register type and Ethernet Module type, as show in the following table:

Table 11.1 – OCS Register Read References		
OCS Register Type	Maximum Reference (ETN200)	Maximum Reference (ETN300)
➤ %R	R2048	R9999
➤ %AI, %AQ	AI512, AQ512	AI512, AQ512
%I, %Q, %M, %T	I2048, Q2048, M2048, T2048	I2048, Q2048, M2048, T2048
%S	S16	S16

- For BOOL formatting (see next section), a bit number can be appended to 16-bit register references (%R, %AI and %AQ). For example, **reg-R11.3** references bit 3 of register %R11. Valid bit numbers are 1 through 16.

11.5.2.2 **fmt** Parameter – OCS Register Read Format

Examples: **fmt**-BOOL
 fmt-BOOL-ON-OFF
 fmt-BOOL-YES-NO
 fmt-BIN
 fmt-HEX
 fmt-INT
 fmt-UINT
 fmt-DINT
 fmt-UDINT
 fmt-REAL
 fmt-REAL-E
 fmt-ASCII-10
 fmt-IPADR

The required **fmt** parameter is followed by a dash and a display format type. Valid format types are **BOOL**, **BIN**, **HEX**, **INT**, **UINT**, **DINT**, **UDINT**, **REAL**, **ASCII** and **IPADR**. Three of these format types (**BOOL**, **REAL** and **ASCII**) have optional or required modifiers as follows:

The **BOOL** format reads and displays a single bit, and two optional modifiers indicate the text to display for the bit's logical On and Off states. The text "TRUE" and "FALSE" will be displayed if the modifiers are omitted. In other words, **fmt-BOOL** is equivalent to **fmt-BOOL-TRUE-FALSE**.

The **REAL** format reads a 32-bit floating-point value, and the optional **-E** modifier can be used to display data in exponential (scientific) notation. The **fmt-REAL** format displays a variable number of digits to the left of the decimal point and 6 digits to the right of the decimal point (e.g. -12345.000000). On the other hand, the **fmt-REAL-E** format only shows 1 digit to the left of the decimal point and shows 6 digits to the right of the decimal point followed by a 4-character power of 10 (e.g. -1.234500E+04).

The **ASCII** format accesses a string of ASCII characters, and a string size modifier from **-1** to **-31** is required, indicating how many ASCII characters to read and display. For example, the **fmt-ASCII-10** format reads and displays a string of 10 ASCII characters.

11.5.2.3 **rfs** Parameter – OCS Register Read Auto-Refresh

Example: **rfs**-2

The optional **rfs** parameter specifies how often the <iframe> should automatically refresh the data on the browser screen. For example, **rfs-2** will cause the data to be refreshed every 2 seconds.

The number of seconds specified can range from 0 to the limits of the browser. A refresh time of 0 will refresh the data as often as possible, but this is not recommended because of the volume of Ethernet traffic it can create. If the optional **rfs** parameter is omitted, the data will only be refreshed when the user manually refreshes their web browser screen.

The **rfs** parameter uses a technique called "client-pull" to tell the web browser to do the periodic refresh. It should be noted that not all web browsers support "client pull" and those that do support it can usually be configured to disable it for security purposes.

Note: When using an **Internet Explorer** browser to view OCS web pages that employ the **rfs** parameter, the browser plays the currently configured **Start Navigation** sound (a short "tick" by default) for each refresh. To disable the "ticking", (1) open the **Windows Control Panel**, (2) click the **Sounds** or **Sound and Audio Devices** icon or link and (3) turn off the **Start Navigation** sound by selecting **(None)** for it in the drop-down list.

11.5.3 Dynamic Html Files – Writing OCS Register Data (writereg.htm)

OCS Register data is written by placing a <form> tag, that references **writereg.htm**, within a static HTML file. The following example <form> tag sets up an Unsigned Integer write to OCS Register %R11:

```
<form action="writereg.htm;reg-R11;fmt-UINT" method="post">...</form>
```

Between the <form> and </form> tags, there should be one or more <input> tags that enable a browser operator to enter or select data and write it to the register.

For example, in the following complete <form> markup, the first <input> tag creates a text input box, containing a default value of 0, which allows the user to enter a new value for %R11. The second <input> tag creates a **Send** button, which can be clicked to submit the new value (write it to the register):

```
<form action="writereg.htm;reg-R11;fmt-UINT" method="post">  
  <input type="text" name="reg" value="0">  
  <input type="submit" value="Send">  
</form>
```

In the following example, the first <input> tag creates an **ON** button that can be clicked to turn %T22 On, while the second <input> tag creates an **OFF** button that can be clicked to turn %T22 Off.

```
<form action="writereg.htm;reg-T22;fmt-BOOL-ON-OFF" method="post">  
  <input type="submit" name="reg" value="ON">  
  <input type="submit" name="reg" value="OFF">  
</form>
```

Important: In the above two examples, the OCS HTTP Server expects the **name="reg"** attributes to appear exactly as shown.

The <form> tag's required **action** attribute is a quoted string containing the **writereg.htm** reference and two parameters, all separated by semicolons. The **reg** and **fmt** parameters specify the OCS Register to write and the data entry/select format to use. These parameters are described in more detail in the following 2 sections:

11.5.3.1 **reg** Parameter – OCS Register Write Reference

Examples: reg-R11
 reg-R11.3

The required **reg** parameter is followed by a dash and an OCS Register reference. Register references start at 1 and the maximum reference depends on the register type and Ethernet Module type, as show in the following table:

Table 11.2 – OCS Register Write References		
OCS Register Type	Maximum Reference (ETN200)	Maximum Reference (ETN300)
➤ %R	R2048	R9999
➤ %AI, %AQ	AI512, AQ512	AI512, AQ512
%I, %Q, %M, %T	I2048, Q2048, M2048, T2048	I2048, Q2048, M2048, T2048

- For BOOL formatting (see next section), a bit number can be appended to 16-bit register references (%R, %AI and %AQ). For example, **reg-R11.3** references bit 3 of register %R11. Valid bit numbers are 1 through 16.

11.5.3.2 **fmt** Parameter – OCS Register Write Format

Examples: **fmt-BOOL**
 fmt-BOOL-ON-OFF
 fmt-BOOL-YES-NO
 fmt-BIN
 fmt-HEX
 fmt-INT
 fmt-UINT
 fmt-DINT
 fmt-UDINT
 fmt-REAL
 fmt-ASCII-10

The required **fmt** parameter is followed by a dash and a data entry format type. Valid format types are **BOOL**, **BIN**, **HEX**, **INT**, **UINT**, **DINT**, **UDINT**, **REAL** and **ASCII**. Three of these format types (**BOOL**, **REAL** and **ASCII**) have optional or required modifiers as follows:

The **BOOL** format writes a single bit, and two optional modifiers indicate the text that must be submitted to set the bit to its logical On and Off states. If these modifiers are omitted, they default to “TRUE” and “FALSE”, so that **fmt-BOOL** is equivalent to **fmt-BOOL-TRUE-FALSE**.

The **REAL** format writes a 32-bit floating-point value, and data can be entered in either standard decimal notation or in scientific notation as desired, with or without the **-E** modifier. The **-E** modifier is not necessary and is ignored if present.

The **ASCII** format accesses a string of ASCII characters, and a string size modifier from **-1** to **-31** is required, indicating how many ASCII characters will be written when the form is submitted. For example, the **fmt-ASCII-10** format writes a string of 10 ASCII characters. If the user enters an ASCII string that is smaller than the specified string size, the submitted string will be padded with spaces.

11.6 Web Content - Authoring

Web content can range from very simple to extremely complex depending on application needs. As such, it is well beyond the scope of this User Manual to attempt to provide a tutorial on the subject.

11.6.1 Web Server Example Applications

One of the best ways to learn how to use the HTTP Web Server is to study working examples. OCS HTTP Web Server Examples are available for download on the Horner APG, LLC web site at:

<http://www.heapg.com/Support/downloads.htm>

11.6.2 Web Authoring Reference Material

Those who are relatively new to web content development would be well advised to obtain and study one or more published books on the subject. Here are two good places to start:

Crowder, David A. and Bailey, Andrew.
Creating Web Sites Bible, Second Edition.
Indianapolis, IN: Wiley Publishing, Inc., 2004.
ISBN: 0-7645-7498-1

Powel, Thomas A.
HTML & XHTML: The Complete Reference, Fourth Edition.
Emeryville, CA: McGraw-Hill/Osborne, 2003.
ISBN: 0-07-222942-X

11.6.2 Web Authoring Tools

There are several 3rd party tools available to assist in the development of web content. These tools come in four basic types: Text Editors, HTML Editors, WYSIWYG Programs and Word Processors.

Text Editors require intimate knowledge of HTML (HyperText Markup Language) and are therefore not for the faint of heart. However, many experienced web developers prefer to work using a plain text editor.

HTML Editors are specially enhanced text editors designed specifically to make web page creation easier, but they still require a great deal of HTML knowledge.

WYSIWYG (What You See Is What You Get) Programs allow HTML code creation using drag and drop techniques, and thus promise to minimize the author's need to understand HTML. However, when problems arise, there is no substitute for getting into the generated HTML code to see what is going on.

Word Processors should be avoided for web content design. They are not designed for this purpose and are notorious for creating bad HTML code.

Table 11.3 contains suggestions for web authoring tools:

Table 11.3 – Web Content Authoring Tools			
Tool Name	Tool Type	Platform(s)	Web Site
Homesite	HTML Editor	Windows	www.macromedia.com/software/homesite
Dreamweaver	WYSIWYG Program	Windows MAC	www.macromedia.com/software/dreamweaver

NOTES

INDEX

B

Built-In Ethernet Module 13

C

Configuration

 CsCAN Protocol 23
 EGD Protocol 32
 Ethernet Module 17
 Ethernet/IP Protocol 60
 FTP Protocol 63
 HTTP Protocol 65
 ICMP Protocol 29
 Modbus Protocol 58
 SRTP Protocol 56

CsCAN Protocol

 Configuration 23
 Downloading Precautions 24
 Operation 24
 Overview 23
 Preventing Loss of Communication 25
 Recovering from Lost Communication 25
 Security 26

D

Default Gateway 20

E

EGD Protocol

 Configuration 32
 Configuring Consumed Exchanges 37
 Configuring Produced Exchanges 34
 Creating Consumed Exchanges 38
 Creating Produced Exchanges 35
 Defining Consumed Exchange I/O Blocks 39
 Defining Produced Exchange I/O Blocks 36
 Example 1 42
 Example 2 51
 Operation 40
 Overview 31
 Status Words 40
 Terminology 31
 Unicast and Multicast IP Addressing 14

Ethernet Module

 Built-In 13
 Configuration 17
 Default Gateway 20
 Interoperability 10
 IP Address 19, 21

 IP Address Register 20
 Net Mask 19
 Network Example 10
 Overview 9
 Specifications 11
 Status Register 20
 System Requirements 10
 User-Installed 13
 Version Register 20
 Ethernet Technical Resources 11
 Ethernet/IP Protocol
 Configuration 60
 Operation 61
 Overview 59

F

FTP Protocol

 Configuration 63
 File Accessing 64
 Operation 64
 Overview 63

H

HTTP Protocol

 Configuration 65
 Operation 66
 Overview 65
 Web Content - Authoring 71
 Web Content - Dynamic Data 67
 Web Content - Storing 66

I

ICMP Protocol

 Configuration 29
 Operation 30
 Overview 29

Internet Connectivity 15

IP Address 19, 21

M

Modbus Protocol

 Configuration 58
 Master Mapping 57
 Operation 58
 Overview 57

N

Net Mask 19

O

Operation	
CsCAN Protocol	24
EGD Protocol	40
Ethernet/IP Protocol	61
FTP Protocol.....	64
HTTP Protocol.....	66
ICMP Protocol.....	30
Modbus Protocol.....	58
SRTP Protocol	56
Overview	
CsCAN Protocol	23
EGD Protocol	31
Ethernet Module	9
Ethernet/IP Protocol	59
FTP Protocol.....	63
HTTP Protocol.....	65
ICMP Protocol.....	29
Modbus Protocol.....	57
SRTP Protocol	55

P

Port Usage	
UDP and TCP	14

S

Specifications	11
SRTP Protocol	
Configuration.....	56
Operation	56
Overview	55

T

TCP Port Usage	14
Technical Resources	
Ethernet.....	11
Web Content Authoring.....	71

U

UDP Port Usage	14
User-Installed Ethernet Module.....	13