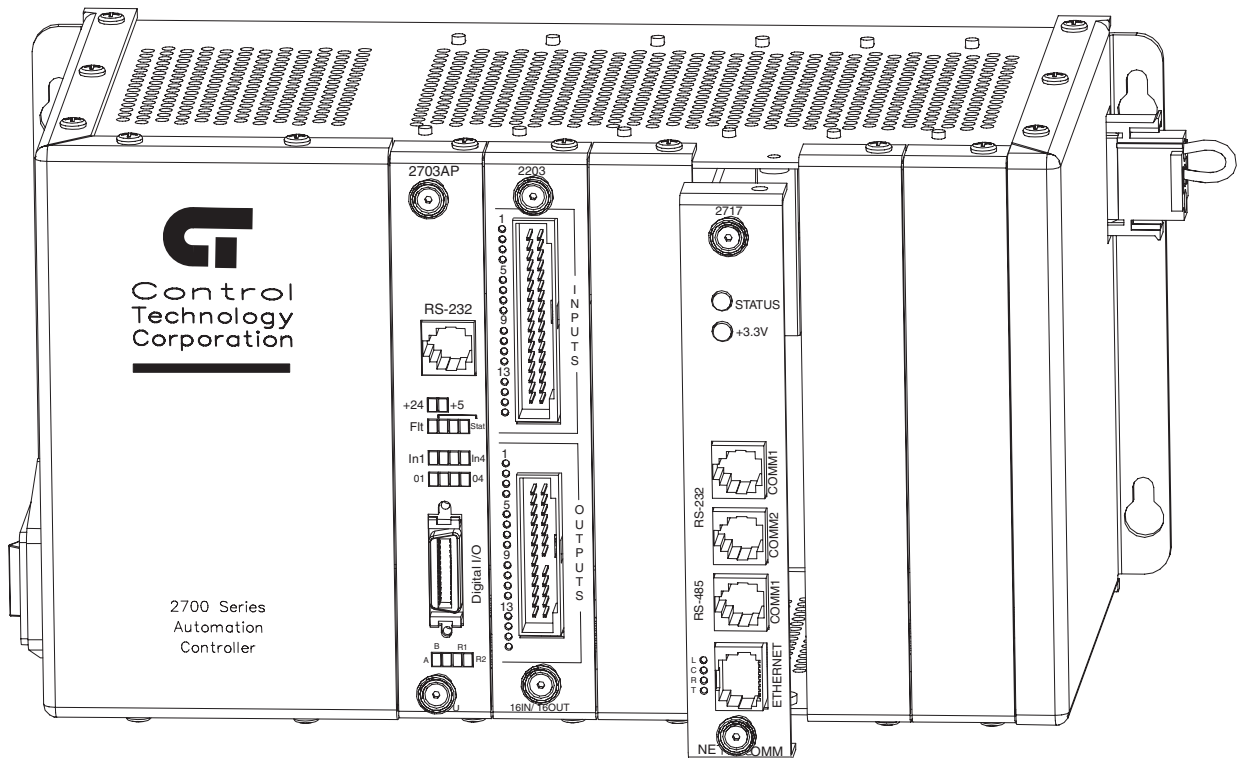




Model 2717 Ethernet Communications Module Installation Guide



Doc. No. 2717IG
Revision F
March 2007

The information in this document is subject to change without notice. The software described in this document is provided under license agreement and may be used or copied only in accordance with the terms of the license agreement.

The information, drawings, and illustrations contained herein are the property of Control Technology Corporation. No part of this manual may be reproduced or distributed by any means, electronic or mechanical, for any purpose other than the purchaser's personal use, without the express written consent of Control Technology Corporation.

The following are trademarks of Control Technology Corporation:

- Quickstep
- CTC Monitor
- CTC Utilities

Windows is a trademark of Microsoft Corporation.

Contents

Notes to Readers	1
Related Documents	2
Formatting Conventions	2
Contacting Control Technology Corporation	3
Errata Sheets	3
Your Comments	3
 1 Getting Started	
System Overview	7
2717 Description	8
Connectors and Pinout Diagrams	9
Specifications	10
Board Handling Precautions	12
Installing the 2717 Module	13
Port Addressing	14
Host Operation with the RS-232 and RS-485 Ports	14
Computer - Controller Communications	15
RS-232 Communications	15
RS-232 Connections	15
Connecting to a D Connector	16
RS-485 Connections	17
Ethernet Connections	17
 2 Configuring the 2717 Module	
File Transfer Protocol (FTP)	21
Supported FTP Commands	21
Special Reserved Commands	22
Caveats	22
Updating Flash Memory with FTP	23
2717.INI Initialization File	24
[NETWORK]	24
[SECURITY]	25
[PEER_INITIALIZATION]	26

Sample 2717.ini File	27
Modifying IP information	30
Using CTC Monitor to set an IP address	30
Dip Switch Settings	31

3 Network Protocols

Supported Protocols	35
CTNet Binary Protocol (Server)	35
Peer-to-Peer Client/Server Protocol	36
Ethernet Protocol	36
10Base-T	36
100Base-T (Fast Ethernet)	36
Network Specifications	37
File Transfer Protocol (FTP)	38
Anonymous FTP	38
FTP Connections	38
ModBus Protocol	39
ModBus TCP/IP	39
ModBus Client Protocol	40
ModBus Server Protocol	40
Transmission Control Protocol/Internet Protocol (TCP/IP)	41
Packet Transmission and Routable Protocols	41
TCP/IP Services	41
Client/Server Computing and TCP/IP	42
TCP/IP Layers	42
TCP/IP Protocol Suite	43
TCP/IP Addressing	44
Network Classes	45
Constructing Addresses in Binary Notation	46
Subnets	46
Subnet Masks	46
Applying Subnet Masks To A Network	47
Default Masks	48
Custom Masks - Classic Rules	48
Custom Masks - CIDR Rules	49
User Datagram Protocol (UDP)	50
Datagrams	50
Headers	50
Services	51
Client/Server Computing	52
What is a Client?	52
What is a Server?	52
Client/Server Model Defined	52
Internet Applications	52
Pros and Cons of Client/Server Computing	53
Benefits	53
Drawbacks	54

Setting up an Intranet with CTC Controllers	55
Virtual Private Networks	58
Remote Access Service (RAS)	60
Point-to-Point Tunneling Protocol (PPTP)	60
Layer Two Tunneling Protocol (L2TP)	61

4 Special Registers

Special Purpose Registers	65
Network and Communications Registers	65
20000 - CTNet/Ethernet Device Node Number Register - R/W	65
20005 - MAC Address Register - Upper Four Bytes - R/W	65
20006 - MAC Address Register - Lower Four Bytes - R/W	65
20010, 20014 - Serial Port Baud Rate - R/W	65
20011, 20015 - Serial Port Data Length - R/W	65
20012, 20016 - Serial Port Parity - R/W	66
20048-20051 - IP Address Registers - R/W	66
20064-20067 - Subnet Mask Registers - R/W	66
20080-20083 - Gateway Address Registers - R/W	66
20096 - Serial E2PROM Update Register - R/W	66
20102 - On-board Millisecond Timer - Read-only	66
Serial Port Redirection TCP Protocol (Server) - Register 20120 - R/W	67
Peer-to-Peer Protocol Registers	67
21XX0 - First Octet IP Address Register (Most Significant) - R/W ...	67
21XX1 - Second Octet IP Address Register - R/W	67
21XX2 - Third Octet IP Address Register - R/W	67
21XX3 - Fourth Octet IP Address Register (Least Significant) - R/W	67
21XX4 - Start Register - R/W	68
21XX5 - Sequential Number Register - R/W	68
21XX6 - Poll Timer Register - R/W	68
21XX7 - Status Flag Register - Read-Only	68
21XX8 - Index Offset Register - R/W	69
21XX9 - Data Registers/Peer Request Time-Out Register - R/W	70

Glossary	71
-----------------------	-----------

Bibliography	77
---------------------------	-----------

This page is intentionally left blank.

Notes to Readers

The *Model 2717 Installation Guide* provides the following information:

- System Overview -- describes the 2717's basic features.
- Description and Connections -- an overview of the 2717's basic functions; pinout diagrams for all connectors.
- Specifications -- general and performance specifications.
- Hardware/Firmware Revision Levels -- lists the hardware and firmware revisions for the 2717 module and several CTC controllers.
- Board Handling Precautions-- contains general guidelines on handling printed circuit boards with ESD devices.
- Installation -- describes how to install the 2717 module in a CTC controller.
- Communications -- describes RS-232 / RS-485 port addressing and how the 2717's ports function.
- FTP and Flash Memory - lists the FTP commands supported by the 2717; describes how to format and program flash memory.
- .INI File Structure - describes the 2717's initialization file and its structure.
- Network Protocols - CTNet, UDP, Modbus, TCP/IP, and Ethernet; what they are, how they function, and how they are used by the 2717 module.
- Application Notes -- contains technical information on the 2717's features; also contains sample Quickstep programs that illustrate how the 2717 functions.
- Special Purpose Registers -- how to use the 2717's on-board registers in special applications.

Related Documents

The following documents contain additional information:

- For information on Quickstep, refer to the *Quickstep™ Language and Programming Guide* or the *Quickstep™ User Guide*.
- For information on the registers in your controller, refer to the *Register Reference Guide* (available at www.ctc-control.com).
- For information on Microsoft Windows or your PC, refer to the manuals provided by the vendor.

Formatting Conventions

The following conventions are used in this book:

ALL CAPS BOLDFACE	Identifies DOS, Windows, and installation program names.
Boldface	Indicates information you must enter, an action you must perform, or a selection you can make on a dialog box or menu.
<i>Italics</i>	Indicates a word requiring an appropriate substitution. For example, replace <i>filename</i> with an actual file name.
Text_Connected_With_Underlines	Indicates symbolic names used in Quickstep programs. Step Names are ALL_CAPITALS. Other symbolic names can be Initial_Capitals or lower_case.
SMALL CAPS	Identifies the name of Quickstep instructions in text.
Courier font	Identifies step names, comments, output changes, and Quickstep instructions appearing in the Quickstep editor.
Art Code 2217F1	Identifies the file name of a particular graphic image.

Contacting Control Technology Corporation

Control Technology Corporation is located in Massachusetts. Our business hours are 8:30 AM to 5:00 PM. EST (Eastern Standard Time).

Contact Method	Address or Number
E-Mail:	
Technical Support:	help@ctc-control.com
Technical Publications:	techpubs@ctc-control.com
Website:	www.ctc-control.com
Telephone:	508.435.9595 and 800.282.5008
FAX:	508.435.2373
Mail:	Control Technology Corporation 25 South Street Hopkinton, MA 01748

Errata Sheets

Refer to the Support area of Control Technology's web site (www.ctc-control.com) for any errata information on this product.

Your Comments

Suggestions and comments about this or any other Control Tech document can be e-mailed to the Technical Publications Group at techpubs@ctc-control.com.

This page is intentionally left blank.

Getting Started

System Overview	7
2717 Description	8
Connectors and Pinout Diagrams	9
Specifications	10
Board Handling Precautions	12
Installing the 2717 Module	13
Port Addressing	14
Computer - Controller Communications	15

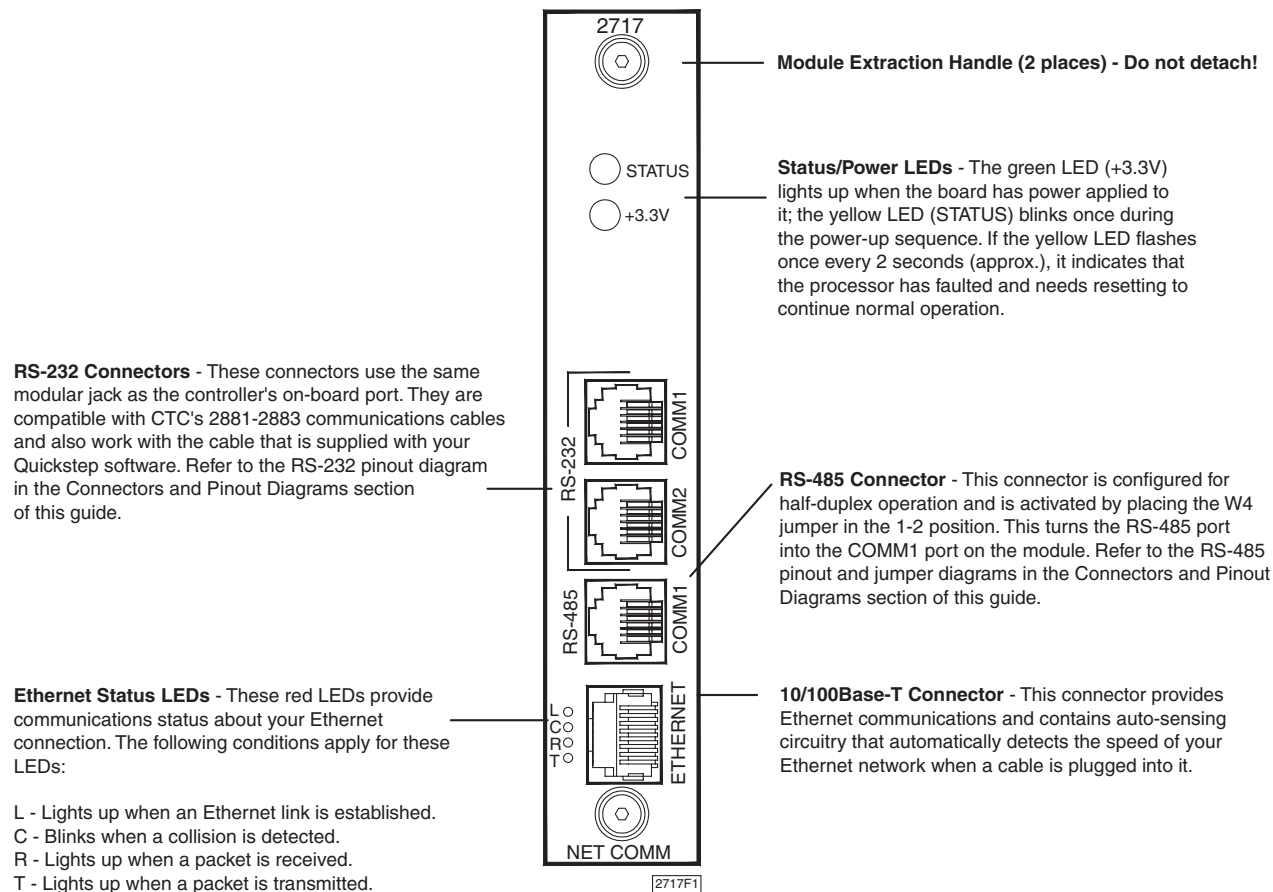
This page is intentionally left blank.

System Overview

The Model 2717 Ethernet Communications Module is packed with useful features that are essential in any industrial control application. The 2717 supports many popular networking protocols and is accessible from local intranets or from remote locations. The module is easy to configure and program and can act as a client or server on your network. In particular, the module has the following features:

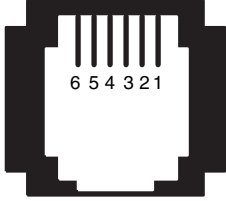

- **Compatible with TCP/IP and UDP protocols** - The 2717 uses TCP/IP and UDP and can easily fit into your current networking strategy. Both these protocols are universal and widely accepted on the plant floor. The module can handle numerous connections of either protocol at one time. It can even transmit unsolicited messages for use in event-based network configurations.
- **ModBus read/write capability** - The 2717 supports multiple reads and multiple writes of registers in both client and server configurations.
- **Ethernet connectivity** - The module supports both 10Base-T and 100Base-T. Auto-sensing circuitry automatically detects the correct speed on your network when a cable is plugged into the 2717's Ethernet port. You can support existing hardware investments and enable future expansion to faster networks.
- **Peer-to-Peer networking** - The 2717 acts as a conduit for peer-to-peer communications between CTC controllers. In addition, register information can be transferred in blocks through the use of arrays for Quickstep program access.
- **Internet/Intranet accessibility** - Real-time plant floor data can be accessed by any authorized user or system through remote access, VPNs, and the Internet. Tools for PC and HMI software, export utilities, and monitoring are now available.
- **Flash file system** - The 2717's flash memory uses FTP for programming and updates. You can transfer information over secure connections to and from the controller.
- **RS-232 and RS-485 ports** - The module has two RS-232 ports for serial communications. One of the ports (COMM1) is configurable for half-duplex, RS-485 communications.
- **Compatible with 2600/2700 series controllers** - The 2717 runs in existing 2600 and 2700 series controllers and will work with future generations of CTC controllers.

2717 Description



Connectors and Pinout Diagrams

Table 1–1. RS-232 / RS-485 Connectors

RS-232 (RS-485 ^{1 2}) Connector	Pin #	Signal
 	1	Not Used (NC)
	2	TxD Outbound (NC)
	3	Common (T/R+)
	4	Common (T/R-)
	5	RxD Outbound (NC)
	6	Common (NC)

1. RS-485 signals are enclosed in parentheses.
2. The RS-485 connector (COMM1 only) is configured for half-duplex operation and becomes active when the W4 jumper is in the 1-2 position as shown in the illustration below.

W4 jumper is in the RS-485 position

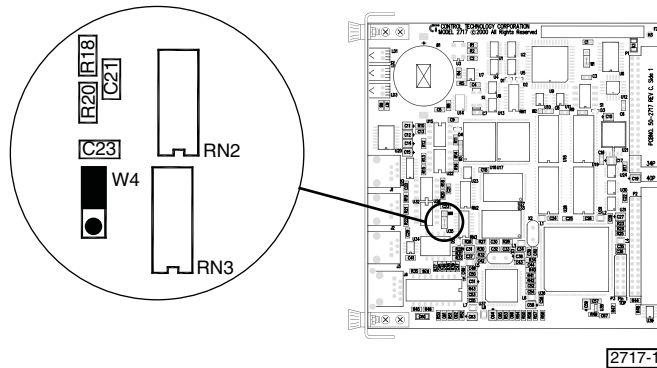
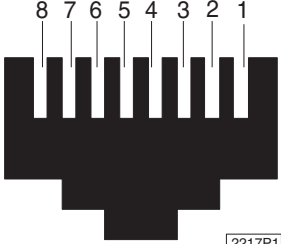



Table 1–2. Ethernet Connector

10/100Base-T Connector	Pin #	Signal
 	1	TX0+
	2	TX0-
	3	RX1+
	4	NC
	5	NC
	6	RX1-
	7	NC
	8	NC

Specifications



Note

All specifications are at 25°C unless otherwise specified.

Table 1–3. General Specifications

Description	Min.	Typical	Max.	Units
Absolute Maximum Ratings				
Ambient Temperature				
Operating	0		+50	°C
Storage	-20		+80	°C
Operating Characteristics ¹				
RS-232 Transmitters	± 3	± 5	± 10	VDC
RS-232 Receivers	3		12	VDC
Common Mode Voltage Range	-10		+10	VDC
RS-485 Common Mode Rejection	-7		+12	VDC
RS-485 Hysteresis		70		mVDC
Ethernet Transceivers (10/100 Megabits/sec) ²			1.5	VAC PP
Power Supply Requirements (from controller)				
Logic Supply (5 V)		370.0	410.0	mA
Auxiliary Supply (24 V)		0	0	mA
Flash Memory				
Storage space			32	MB
1. These values are derived with high communications priority active or when one task is running.				
2. This conforms to IEEE Standard 802.3.				

Table 1–4. Performance Specifications

Description	CTNet	UDP	TCP/IP	Modbus	Units
Performance Specifications ¹					
Host Communications					
Single-Register Transaction from 2717	1-2	2-4	3.5-4	6-8	msec
Single-Register Transaction from 2703AP	3-5	5-8	7-10	10-12	msec
16-Register Read from 2703AP	6-7	9-11	10-12	12-14	msec
50-Register Read from 2703AP	8-9	10-12	11-13	16-17	msec
1. These values are derived with high communications priority active or when one task is running.					

Table 1–5. Hardware / Firmware Revision Levels

Model Numbers	Hardware Revision Level	Firmware Revision Level
2717	Rev. C or greater	1.05I or greater ¹
2700 Series	Rev. C or greater	2.10 or greater ^{2 3}
2600 Series	Rev. 0 or greater	1.00 or greater ^{2 3}

1. You can retrieve the firmware revision level by setting up an FTP connection and issuing the “remote help” command.

2. You can confirm firmware revision levels by doing a register read in Quickstep's monitor program. Use register 13003 to confirm the firmware revision in a 2600/2700 series controller.

3. Firmware revision levels are not equivalent to standard decimal numbers. For example, firmware revision level 2.10 translates to:

Major Revision Level 2
Minor Revision Level 10

If this value changes to 2.20, it translates to:

Major Revision Level 2
Minor Revision Level 20 (not revision level 2)

Board Handling Precautions

The module's printed circuit board contains electrostatic discharge sensitive (ESD) devices. Improper board handling could result in damage to the board. The following precautions are recommended when handling the board or before inserting it into the controller:

- Make sure you are grounded electrically by using a wrist strap connected to an electrically grounded workstation or by physically touching the controller case or something electrically connected to the controller case.
- Avoid touching the leads or contacts of the circuit board and handle the board by its edges only.
- Transport circuit boards in protective, anti-static bags, bins, or totes. Do not insert boards into materials such as plastic, polystyrene foam, clear plastic bags, bubble wrap, or plastic trays.

Port Addressing

Each communications port is designed to function independently and is automatically serviced on an interrupt basis. Quickstep program activity will not affect data integrity. You can use CTC's communications protocols (DLLs) on any port in the controller from a host computer or intelligent host terminal.

Host Operation with the RS-232 and RS-485 Ports

Technical Note No. 30, ASCII Transmitting with CTC Controllers, contains details on how to send messages through these ports. You can obtain a copy of this document from the Customer Support area of our web site at www.ctc-control.com.

Computer - Controller Communications

The 2717's RS-232 ports provide a way to download Quickstep programs and also support data communications.

RS-232 Communications

The 2717's RS-232 ports allow the following activities:

- **Direct communications between a PC and the 2717's RS-232 ports** - This feature enables you to directly interact with all the controller's resources such as registers, inputs, outputs, and flags without modifying the controller's program.
- **Monitoring** - You can monitor a controller's activity through an RS-232 port with CTCMON.
- **Host configuration** - The 2717 is configurable as a host that can support communications with other external peripherals such as operator interface terminals, bar-code readers, printers, and other controllers. Refer to *Technote No. 30, ASCII Message Transmitting with CTC Controllers*, which is available in the Support section of our web site at www.ctc-control.com.



Note

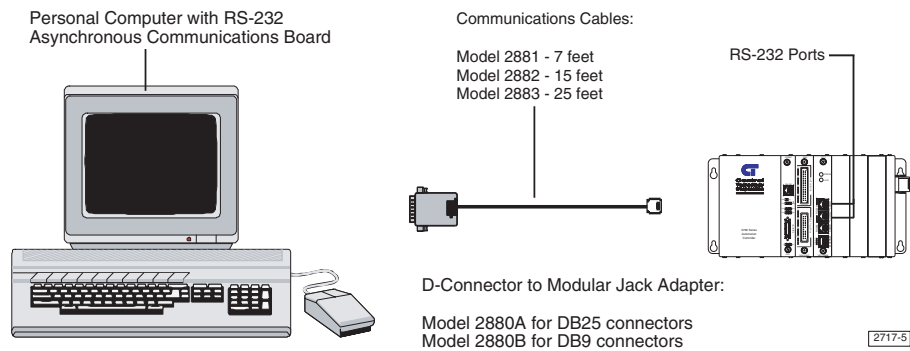
For more information on data communications and the DLL functions required to communicate with the 2717, refer to the *CTC 32-Bit Data Communications Functions Reference Guide*, which is available in the Customer Support area of our web site at www.ctc-control.com.

RS-232 Connections

Connect to the RS-232 ports through one of the modular jacks (labeled COMM1 and COMM2) on the 2717's front panel. These jacks carry the receive signal, two commons (ground), and the transmit signal for the communications channel (only the center four conductors of a six or eight conductor jack are used). Refer to Table 1-1 on page 9 for details on how this jack is wired.

Standard Control Technology cables are available for connecting to this jack (see Figure 1-2 for more information). As an alternative, many commonly available telephone cables may be substituted.

Figure 1–2. Communication Cables and Connectors



Connecting to a D Connector

RS-232 ports on computers are usually configured through 25-pin (DB25) or 9-pin (DB9) D-type connectors. Most PC manufacturers use standard wiring on these connector types.

Control Technology has adapters available that connect directly to a male DB25 (Model 2880A) or DB9 (Model 2880B) connector. These adapters have a modular jack that is wired for compatibility with the COMM port. To ensure full compatibility with these adapters, you should wire the computer's communications port as a DTE (Data Terminating Equipment) device.



Note

Do not connect the 2717 to a telephone line.

Figures 1–3 and 1–4 show computer-controller connections using an RS-232 connection and DB25 and DB9 connectors.

Figure 1–3. DB9 Connections

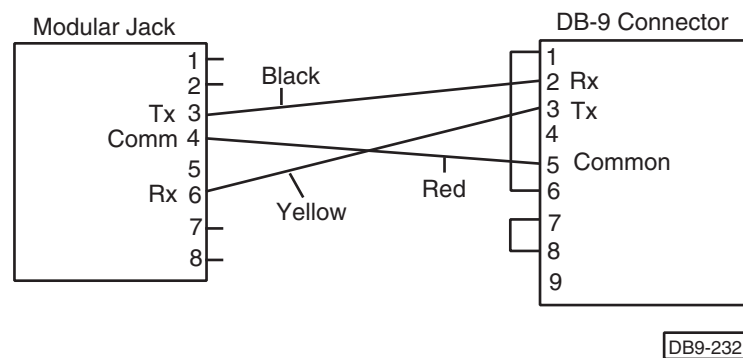
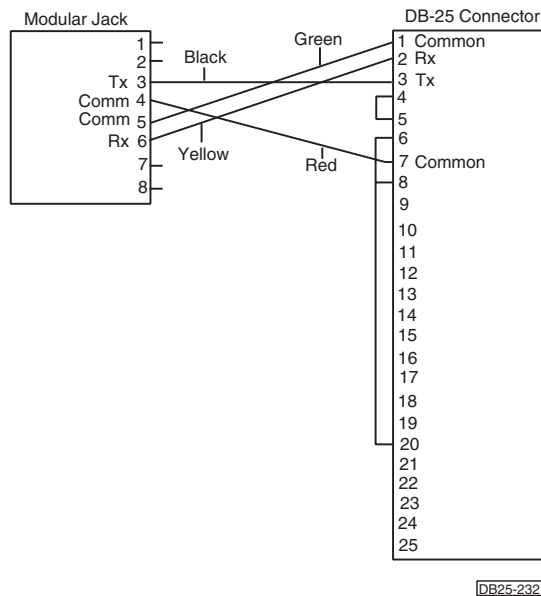


Figure 1–4. DB25 Connections



RS-485 Connections

Connect to the module's RS-485 port through the modular jack (labeled COMM1) on the module's front panel. Make sure that the W4 jumper is set as shown in Table 1–1. The RS-485 jack carries the receive signal and the transmit signal for the communications channel and operates in half-duplex mode (only the center two connectors of a six or eight conductor jack are used. Refer to Table 1–1 for details on how this jack is wired.

Ethernet Connections

The 2717 has a 10/100Base-T connector that conforms to IEEE standard 802.3. Auto-sensing circuitry automatically detects the correct speed on the Ethernet network when you plug a cable into the Ethernet connector. Wiring information for the Ethernet connector is listed in Table 1–2 and performance specifications are listed in Table 1–3. For more information on the Ethernet network protocol, refer to *Chapter 3, Network Protocols*.

This page is intentionally left blank.

Configuring the 2717 Module

File Transfer Protocol (FTP)	21
Updating Flash Memory with FTP	23
Modifying IP information	30
Dip Switch Settings	31

This page is intentionally left blank.

File Transfer Protocol (FTP)

This section discusses file transfer to the 2717 and describes how to program and format the 2717's flash memory.

The 2717 supports a basic file system that can store configuration files and web pages on a flash disk with a base capacity of slightly less than 1 Megabyte. The file system interface uses FTP, which is the industry standard file transfer protocol. You can download, delete, or transfer files to and from the controller and have the support of a directory architecture that resembles the architecture found on typical computer systems. For more information on FTP, refer to *Chapter 3, Network Protocols*.

Supported FTP Commands

The following FTP commands are supported by the 2717:

- **cd**- Change to a different directory.
- **delete**- Delete a file.
- **dir** - Displays a directory and the amount of flash memory that is available, used up, or deleted.
- **get** - Copies a file from the 2717 to a local directory on your PC.
Example: get filename.ext
- **ls** - Lists files in the current directory. "ls -l" is equivalent to the "dir" command.
- **mget** - Copies multiple files from the 2717 to a local directory on your PC. Wild cards are limited to files with the same file extension. For example, mget*.html copies all .html files from the 2717's directory to your local directory.
- **mkdir** - Create a new directory inside the current 2717 directory.
- **mput** - Copies multiple files to the 2717 from a local directory on your PC. Wild cards are limited to files with the same file extension. For example, mput*.html copies all .html files from your local directory to the 2717's directory.
- **put** - Copies a file from a local directory on your PC to the 2717.
Example: put filename.ext
- **pwd** - Displays the current directory on the 2717.
- **quit** - Quit the FTP session.
- **recv** - This command functions the same as "get".
- **remote help** - Displays the 2717's firmware version and the standard list of FTP commands.
- **rmdir** - Deletes a directory from the 2717's current directory.

- **send** - This command functions the same as "put".

Special Reserved Commands

The following special commands are supported by the 2717:

- **Rmdir _FORMATFLASH** - Formats the file system and removes all files and directories. Because this operation may take several minutes, you should only perform it when the 2717 is idle.
- **Rmdir _PROGRAMFLASH** - Writes new pgmimage.bin and loaderimage.bin files to the program flash. These files must reside on the flash disk within the /flash sub-directory before invoking this command. Because this operation may take several minutes, you should only perform it when the 2717 is idle.

Caveats

Flash memory is derived from EEPROM technology. It is non-volatile and retains its contents when power is turned OFF. However, although it can store files like a standard hard drive, it does not release storage space when you overwrite or delete existing files. This space is eventually depleted and is only regained by reformatting the flash memory. This is a potentially slow process, so make sure the 2717 is available during this operation and is not tied up with other tasks like performing peer-to-peer operations.

Updating Flash Memory with FTP

The special FTP commands mentioned above let you update flash memory in the field. You must load two files (loaderimage.bin and pgmimage.bin) to the /flash sub-directory on the flash disk. Pgmimage.bin (~600K bytes) is the new software revision and loaderimage.bin (~130K bytes) is a program that you load into RAM for execution during the flash rewrite process. Because there is only a single flash device for executing programs, you must not write to the flash chip while it is being programmed.

Because there is not enough space on the flash disk to accommodate both the image files and normal operating files (such as web pages), you must format the flash memory before loading files and after updating it in order to make space. As mentioned above, make sure that the 2717 and the controller are idle before running the rmdir_PROGRAMFLASH command.

Reformat the 2717's flash memory as follows:



Note

Reformatting the flash destroys all data in both the flash disk and program flash. Back up all files in the flash disk with the "mget" or "receive" FTP commands before starting this process.

1. rmdir _FORMATFLASH (this may take 15-30 seconds)
2. mkdir flash (create a directory called flash)
3. cd flash (change to the flash directory)
4. send loaderimage.bin loaderimage.bin (this takes approximately 16 seconds)
5. send pgmimage.bin pgmimage.bin (this takes approximately 72 seconds)
6. dir (make sure the correct files are listed)
7. cd ..(return to the root directory)
8. dir (make sure everything looks correct)
9. rmdir _PROGRAMFLASH (make sure the controller is idle)

The Status LED is solidly lit for 15 seconds as the program flash is erased.

The Status LED then starts to blink rapidly for 30 seconds as the new program images are programmed into flash.

10. Quit the FTP session and log in again when the LED stops blinking. The program is now re-flashed. Check in the /flash sub-directory for the file FlashLoaded.txt. If this file is present, then the operation was successful. You can also do a "remote help" FTP command to check the flash software revision.



Note

If you need to load a new .ini file or web pages, then do a `rmdir _FORMATFLASH` to make disk space available and restore the necessary files and directory structure.

2717.INI Initialization File

Configure the 2717 by placing the 2717.ini file into the root directory of its flash file system. This file lets you set various parameters such as initializing local registers. When a new .ini file is loaded, its parameters generally take effect immediately. Some parameters (such as changing IP information) requires that you cycle power to the controller. Each group and its associated parameters is discussed below. A sample .ini file is also provided.



Notes

1. Each definition belongs within a special group that is designated by the [...] brackets. There are currently three groups: NETWORK, SECURITY, and PEER_INITIALIZATION. It is important that these groups are initialized in the order shown in this example.
 2. Each group has parameters that you can modify. If parameters are not modified, a default setting is used.
 3. Pound signs appear before comment fields.
-

[NETWORK]

The Network Group defines communication address parameters such as IP addresses, ports, and protocols you can enable. The following parameters appear in this group:

- **IP_ADDRESS** - This address identifies the 2717 board on the network. Enter each of 4 octets and separate them with a dot. Changes are only recognized during the power-up sequence.
- **SUBNET_MASK** - This 32-bit value identifies the 2717's network ID. Enter each of 4 octets and separate them with a dot. Changes are only recognized during the power-up sequence.
- **GATEWAY_ADDRESS** - Packets are sent to this address (usually a router) if they don't reside on the 2717's network. The factory default setting is 0.0.0.0, which means there is no gateway. Changes are only recognized during the power-up sequence.
- **ETHERNET_MAC_ADDRESS** - A unique, 6-byte address that identifies the 2717 and is transmitted in the header of every network packet. It is assigned by Control Technology Corporation. All CTC addresses begin with 00.C0.CB and end in three bytes that are only assigned to that particular 2717 module. By default, the E²PROM is programmed at the factory with the proper ID and this address is not required except when non-volatile storage is re-initialized. Changes are only recognized during the power-up sequence.

- **CTNET_DEVICENODE** - A unique, single-byte address that identifies the 2717 when the CTNet protocol is used on the network. The factory default setting is 0, which disables the CTNet protocol and conserves CPU resources.
- **BINARYUDP_SERVER_PORT** - The IP port that the 2717 uses to listen for CTC Binary Protocol UDP request packets. The default factory setting is port 3000. Changes are only recognized during the power-up sequence. A zero ("0") disables the protocol and conserves CPU resources.
- **BINARYTCP_SERVER_PORT** - The IP port that the 2717 uses to listen for CTC Binary Protocol TCP request packets. The default factory setting is port 6000. Changes are only recognized during the power-up sequence. A zero ("0") disables the protocol and conserves CPU resources.
- **PEERUDP_SERVER_PORT** - The IP port that the 2717 uses to listen for CTC peer-to-peer request packets. The default factory setting is port 4500. Changes are only recognized during the power-up sequence. A zero ("0") disables the protocol and conserves CPU resources. Refer to [PEER_INITIALIZATION] for details on accessing and initializing peer-to-peer operation.
- **MODBUSTCP_SERVER_PORT** - The IP port that the 2717 uses to listen for ModBus TCP register command packets. The default factory setting is port 502. Changes are only recognized during the power-up sequence. A zero ("0") disables the protocol and conserves CPU resources. Refer to *Chapter 3, Network Protocols*, for more information on the ModBus Server protocol.
- **MODBUSTCP_CLIENT_PORT** - The IP port that the 2717 uses to initiate and register ModBus TCP command packets. In Client mode, we are the client and expect to communicate with another host or controller's server process. The default factory setting is port 502. Changes are only recognized during the power-up sequence. A zero ("0") disables the protocol and conserves CPU resources. Refer to *Chapter 3, Network Protocols*, for more information on the ModBus Client protocol.
- **SERIALREDIRECTION_SERVER_PORT** - You can configure the 2717 to constantly listen for connections that allow the controller to send information to that connection as though it was a serial port once the connection is established. A TCP server runs at this port once the parameter is defined. Changes are only recognized during the power-up sequence. A zero ("0") disables the protocol and conserves CPU resources.

[SECURITY]

The SECURITY Group restricts access to the 2717 module. You can toggle certain communication protocols ON/OFF and can enter ranges of IP addresses for the different services that are available. As a security measure, these IP addresses are checked before access is granted to a resource. The standard method of requesting a username and password is not required. Multiple entries are allowed and each is checked until permission is granted or no more entries exist.

An entry consists of an IP address (or range of IP addresses) followed by a list of what is allowed (a blank entry indicates that everything is allowed). The default setting allows all accesses when security is not defined.

Each IP Address has the following options that you can enable or disable:

- **FTPREAD** - Allows FTP file read-only access.
- **FTPRW** - Allows FTP read and write access; does not allow formatting or loading programs.
- **PEERUDP** - Allows peer-to-peer communications using UDP.
- **BINARYUDP** - Allows binary protocol UDP communications.
- **BINARYTCP** - Allows binary protocol TCP communications.
- **MODBUSTCP** - Allows Modbus Server protocol TCP communications.
- **CTNET** - Allows certain nodes to have access using the CNet protocol. It is defined by setting the first three octets to 0 and the last octet to the allowable address (0.0.0.#). Place at the beginning of the list to speed up access. The default setting is that all nodes are allowed.

The following examples show how these options work:

0.0.0.10 0.0.0.20 CNet	# Allows CNet access only on nodes 10 through 20.
208.164.187.27	# Allows all access with no restrictions from this IP Address.
208.164.187.25 FTPREAD	# Allows FTP access from this IP address.
208.164.187.240 208.164.187.250 PEERUDP	# Only peer-to-peer communications allowed in this range.
2.40.53.001 12.40.53.255	# Allows full access to this range of IP addresses.

[PEER_INITIALIZATION]

The Peer_Initialization Group lets you directly set local 2717 registers to any desired value. Because the registers are set sequentially in the order given, you must ensure they are in the proper sequence. This applies to the Peer Register blocks and virtually any local registers. It provides a convenient way to initialize register values without requiring a Quickstep program.

For example, to set register 20000 to a CNet node of 50, enter: **20000 = 50**. This functions the same as CNET_DEVICENODE (part of the NETWORK Group). Refer to the sample 2717.ini file below for more examples.



Note

Make sure that each line of your 2717.ini file is terminated with a carriage return (0x0d) and a line feed (0x0a).

Sample 2717.ini File

[NETWORK]

```
IP_ADDRESS = 12.40.53.219
SUBNET_MASK = 255.255.255.0
GATEWAY_ADDRESS = 12.40.53.204 # 0.0.0.0 to disable
# 48-bit Ethernet MAC Address is entered in dot notation. The first 3 are common to CTC;
# the remaining are unique to each board.
ETHERNET_MAC_ADDRESS = 0.192.203.0.9.53 # Hex = 00c0cb000935
CTNET_DEVICENODE = 35 # CTTNET Protocol device ID for us
# Setting any of the PORTs below to 0 disables the comm. protocol.
#BINARYUDP_SERVER_PORT = 3000 # Binary Protocol UDP Port
#BINARYTCP_SERVER_PORT = 6000 #Binary Protocol TCP Port
#PEERUDP_SERVER_PORT = 4000 # Peer-to-Peer Communications TCP Port
#MODBUSTCP_SERVER_PORT = 502 # Modbus/TCP Comm.TCP Port
#MODBUSTCP_CLIENT_PORT = 502 # Modbus/TCP Comm TCP Port
#SERIALREDIRECTION_SERVER_PORT = 4250 # Serial Redirection TCP Port
# The time-outs below are based in milliseconds
BACKPLANE_DEFAULT_TIMEOUT = 250 # Dualport Backplane packet time-out
PEER_DEFAULT_TIMEOUTS = 500 # Single request Peer packet time-out
```

[SECURITY]

```
#If the parameters below are left undefined, then all IP addresses are allowed with all
#permissions. Otherwise, you can limit each address to access permissions.
#You can add any of the following parameters to the IP address. If the parameter is left blank, then full access
#is allowed. If something is specified for that parameter, that is all that is allowed.
# FTPREAD - Allows FTP read-only access.
# FTPRW - Allows FTP read / write access; does not allow formatting or loading programs.
# PEERUDP - Allows Peer-to-Peer communications with UDP.
# BINARYUDP - Allows binary protocol UDP communications.
# BINARYTCP - Allows binary protocol TCP communications.
# MODBUSTCP - Allows Modbus protocol TCP communications.
#208.164.187.27 #This allows all access (supervisor; no restrictions)
#208.164.187.25 FTPREAD #FTP read-only access.
#208.164.187.240 208.164.187.250 PEERUDP #Only peer comm. is allowed.
#208.164.187.27 # Unrestricted access is available to this address.
#12.40.53.001 12.40.53.255 # This range of IP addresses allows general access.
```

[PEER_INITIALIZATION]

```
# Registers are initialized in the order given. Values are in decimal notation.
# Any block that is not monitored is filled in with 0's. Add other peer blocks as required.
# Controller 1
21005 = 4 #Sets the number of registers in the block for allocation.
21000 = 12 #Sets the peer IP address.
21001 = 40
21002 = 53
21003 = 27
21008 = 1003
21009 = 2 #Sets the Modbus Client protocol.
21008 = 0 #Sets this register to its previous value.
```

```
21004 = 8001    # Sets the register to start monitoring.
21006 = 50      # Sets the number of milliseconds between updates.
# Controller 2
21015 = 4       #Sets the number of registers in the block for allocation.
21010 = 12      #Sets the peer IP address.
21011 = 40
21012 = 53
21013 = 245
21018 = 1003
21019 = 2       #Sets the Modbus Client protocol.
21018 = 0       #Sets this register to its previous value.
21014 = 8001    # Sets the register to start monitoring.
21016 = 50      # Sets the number of milliseconds between updates.
# It is even possible to monitor our own registers and have them available for both local
# and remote reference. A number of I/O points are monitored in the example below. The
# register list is also required for the special Binary Protocol TCP System I/O command 0x53.
#Analog Out
21025 = 128     # Sets the number of registers in the block for allocation.
21020 = 12      # Sets the peer IP address to the local IP address.
21021 = 40
21022 = 53
21023 = 219
21024 = 8001    # Sets the register to start monitoring.
21026 = 50      # Sets the number of milliseconds between updates. Writing to this register starts the
#data collection poll.
#Analog In
21035 = 128     # Sets the number of registers in the block for allocation.
21030 = 12      # Sets the peer IP address to the local IP address.
21031 = 40
21032 = 53
21033 = 219
21034 = 8501    # Sets the register to start monitoring.
21036 = 50      # Sets the number of milliseconds between updates. Writing to this register starts the
#data collection poll.
#Digital Out
21045 = 10      # Sets the number of registers in the block for allocation.
21040 = 12      # Sets the peer IP address to the local IP address.
21041 = 40
21042 = 53
21043 = 219
21044 = 10001   # Sets the register to start monitoring.
21046 = 50      # Sets the number of milliseconds between updates. Writing to this register starts the
#data collection poll.
#Digital In
21055 = 10      # Sets the number of registers in the block for allocation.
21050 = 12      # Sets the peer IP address to the local IP address.
21051 = 40
21052 = 53
21053 = 219
21054 = 11001   # Sets the register to start monitoring.
21056 = 50      # Sets the number of milliseconds between updates. Writing to this register starts the
#data collection poll.
#Regs 901 to 1000
21065 = 100     # Sets the number of registers in the block for allocation.
21060 = 12      # Sets the peer IP address to the local IP address.
```

21061 = 40

21062 = 53

21063 = 219

21064 = 901 # Sets the register to start monitoring.

21066 = 50 # Sets the number of milliseconds between updates. Writing to this register starts the
#data collection poll.

Modifying IP information

This section describes how to modify IP information with CTC Monitor. You can also modify IP information by downloading a new 2717.ini file. For more information, refer to *Updating Flash Memory with FTP*.

Using CTC Monitor to set an IP address

1. Use CTC Monitor to set the IP address over a serial connection with the following registers:

IP Address: A.B.C.D

Register 20048 = A

Register 20049 = B

Register 20050 = C

Register 20051 = D

Subnet Mask: E.F.G.H

Register 20064 = E

Register 20065 = F

Register 20066 = G

Register 20067 = H

Gateway: I.J.K.L

Register 20080 = I

Register 20081 = J

Register 20082 = K

Register 20083 = L

2. Write a 1 to register 20096 to initiate a Serial E²PROM update. This action writes new values to the flash memory on the 2717 module and deletes the current copy of the 2717.ini file to avoid overriding the settings made by CTC Monitor.
3. Cycle the power on the controller to activate these changes.

Dip Switch Settings

The 2717 printed circuit board has four dip switches (S1). The factory default settings are shown in Figure 2–1. The function of each switch is listed in Table 2–1.

Figure 2–1. 2717 Board with Dip Switches

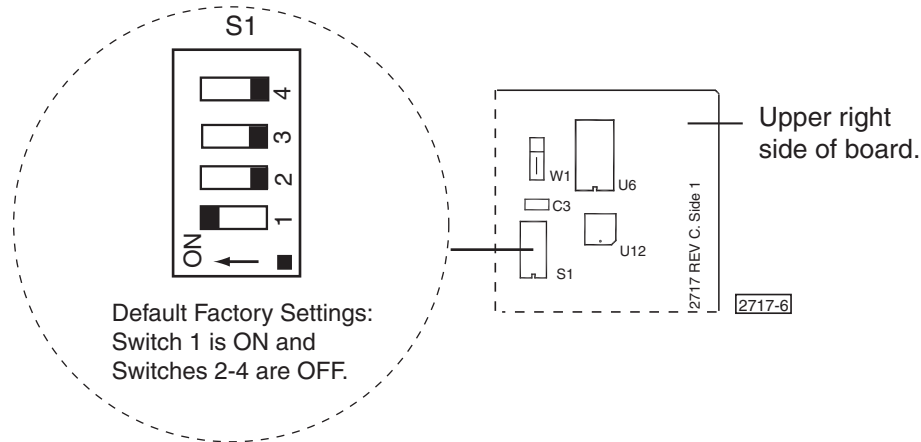


Table 2–1. Dip Switch Settings

Switch #	Description
1	Enables watchdog timer; default setting is ON.
2	Sets the following addresses and node numbers (default setting is OFF; Serial E2PROM initialized as listed below with flash disk erased and formatted):
	IP Address 12.40.53.219
	Gateway 0.0.0.0
	Subnet Mask 255.255.255.0
	MAC Address 0.192.203.0.9.52 (Hex = 0x00c0cb000934)
	CTNet Node 25
3	Not used; default setting is OFF.
4	Not used; default setting is OFF.

This page is intentionally left blank.

Network Protocols

Supported Protocols	35
CTCNET Binary Protocol (Server)	35
Peer-to-Peer Client/Server Protocol	36
Ethernet Protocol	36
File Transfer Protocol (FTP)	38
ModBus Protocol	39
Transmission Control Protocol/Internet Protocol (TCP/IP)	41
User Datagram Protocol (UDP)	50
Client/Server Computing	52
What is a Client?	52
What is a Server?	52
Client/Server Model Defined	52
Internet Applications	52
Pros and Cons of Client/Server Computing	53
Setting up an Intranet with CTC Controllers	55
Virtual Private Networks	58
Remote Access Service (RAS)	60
Point-to-Point Tunneling Protocol (PPTP)	60
Layer Two Tunneling Protocol (L2TP)	61

This page is intentionally left blank.

Supported Protocols

The 2717 supports numerous communication protocols:

- CTNet Binary Protocol Server
- Peer-to-Peer Client/Server
- Ethernet Protocol
- File Transfer Protocol (FTP)
- ModBus Client/Server Protocol
- Transmission Control Protocol/Internet Protocol (TCP/IP)
- User Datagram Protocol (UDP)

Most of these protocols interact with a CTC controller using the Peer-to-Peer Protocol Registers. Refer to *Chapter 4, Special Registers*, for details on these registers.

CTNet Binary Protocol (Server)

The CTNet binary protocol is a high-speed, non-routable protocol that has checksum and error reporting capabilities. It is used in cases where data integrity, response time, and processing time are the major criteria. Data transmission is fast for the following reasons:

- Both the commands and data are represented in binary form instead of ASCII. The information density is higher and fewer characters are transmitted during large data transfers.
- The controller can use the data “as is” and does not have to perform binary to ASCII conversion. This results in shorter execution times.
- The binary protocol is non-routable. Non-routable protocols do not contain a networking layer (IP stack), so they cannot cross a router and are limited to local subnets or intranets. However, lack of an IP stack reduces overhead by at least 20 bytes/packet. A smaller packet size increases the transmission rate, which is ideal for industrial controllers. Routable protocols such as TCP/IP are discussed in the *TCP/IP* section of this chapter.

A node number is used in place of an IP address. This node number is defined by writing to Register 20000, which is not updated until information is written to Register 20096. You can also determine the node number by reading the value in Register 20000. Set this value within the 2717.ini file by defining the CTNET_DEVICENODE parameter.

For more information on the CTNet binary protocol, refer to the *CTC Serial Data Communications Guide*.

Peer-to-Peer Client/Server Protocol

Peer-to-peer networking is a type of network in which each controller has equivalent capabilities and responsibilities. Controllers are configured to share resources and can automatically gather register information from other CTC controllers without requiring a dedicated server. The Peer-to-Peer Registers (21000-21999) are used for this function. You can point each register block to another controller and can automatically update a group of registers. This allows Quickstep to reference local memory instead of requesting information over the network.

Refer to *Chapter 4, Special Registers*, for more information on these registers.

Ethernet Protocol

Ethernet is defined by the IEEE 802.3 standard and is the most widely used local area network (LAN) access method. Data packets are transmitted over coaxial cable using the carrier sense multiple access with collision detection (CSMA/CD) algorithm until they arrive at their destination without any collisions. Ethernet nodes on a segment share the bandwidth, which is 10 MBps (Ethernet), 100 MBps (Fast Ethernet), or 1000 MBps (Gigabit Ethernet). The 2717 module has an Ethernet port that allows it to communicate over an Ethernet network using 10Base-T or 100Base-T connections. Auto-sensing circuitry automatically detects the transmission rate on the network when a cable is plugged into the port.

10Base-T

This connection type uses unshielded twisted pair (UTP) cabling and standard RJ-45 connectors. 10Base-T uses Category 3 (or higher) cables. Higher category numbers provide greater protection from outside electrical interference. CTC recommends using Category 5 UTP cable.

100Base-T (Fast Ethernet)

Fast Ethernet (IEEE 802.3u) is traditional CSMA/CD at 100 MBps over UTP cables. Because its design is based on the 10Base-T standard, it can be easily incorporated into existing networks. Three media types are supported:

- **100Base-TX** - 2 pairs of Category 5 UTP cable and an RJ-45 connector.
- **100Base-T4** - 4 pairs of Category 3-5 UTP cable.
- **100Base-FX** - multimode fiber-optic cable; primarily used on backbones.

Because 100Base-TX resembles 10Base-T so closely, it is the most popular type of Fast Ethernet connection.

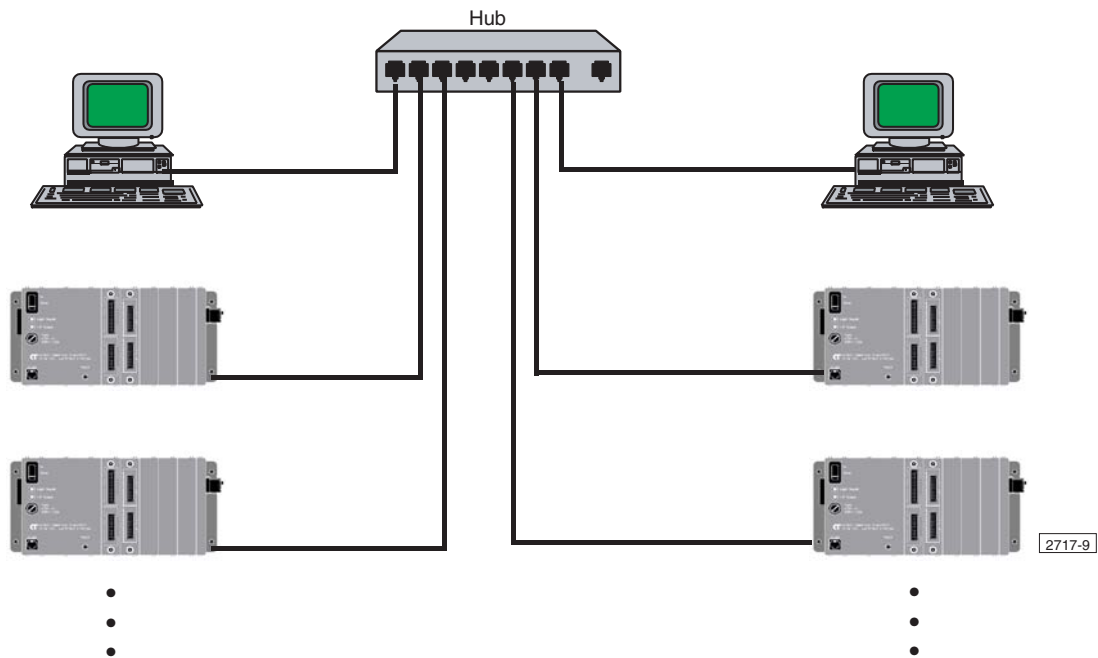
Network Specifications

Node and cable specifications for 10Base-T and 100Base-T connections are listed below. Termination for 10/100Base-T is provided by a hub. The total nodes per hub are determined by the hub size.

Total number of nodes supported:	32767
Maximum number of nodes per segment:	1024
Maximum cable length per segment:	100 meters
Maximum cable length per network:	500 meters (10Base-T) 200-250 meters (100Base-T)

Figure 3–1 shows computer-controller connections using an Ethernet network. It represents one segment out of 5 possible segments on the network. The total cable length between all devices and the hub must not exceed 100 meters or the rule is violated.

Figure 3–1. Ethernet Network with one Segment



Controllers and other devices can be added to this segment provided that you don't exceed 100 meters of cable.

File Transfer Protocol (FTP)

FTP is a client/server protocol that allows a user on one computer to transfer files to and from another computer over a TCP/IP network. It includes functions to log onto the network, list directories, and copy files. Perform FTP operations by:

- typing commands at the command prompt
- using an FTP client such as WS_FTP
- initializing transfers from within a web browser by entering **ftp://** followed by the URL

FTP is designed to handle binary files directly; decoding and encoding transferred data is not required. In addition, users from one system can interact with users from other systems without regard for the operating system in use. Because it works across so many platforms, users need to take care when they transfer files that are not native to their operating system.

Anonymous FTP

Many FTP sites use minimal security because they provide public access to their files. These sites are known as anonymous FTP sites. Anyone can log onto such a site and download files. However, uploads are generally not allowed.

FTP Connections

The FTP client handles most of the commands. It interprets commands and then sends a request to the FTP server using the FTP protocol. For example, imagine that a user wants to communicate with the 2717 module. They want to transfer files from their PC to the 2717 module, so they run the FTP client on their computer and the 2717 acts as the FTP server.

Commands and data are sent across two different connections. When a user starts FTP and connects with an FTP server, a connection to the server is opened that remains open until a close command is received. When a user requests a file transfer, the file is transferred over a different connection that closes when the transfer is complete. Therefore, a typical FTP session may involve several open connections if multiple files are being transferred.

Refer to *Chapter 2, Configuring the 2717 Module*, for information on the FTP command set and how it is used with the 2717 module.

ModBus Protocol

The ModBus protocol is a messaging structure that establishes connections between intelligent devices. For example, the 2717 module is able to communicate with ModBus-equipped devices.

There are two ways to transmit ModBus data:

- **ASCII transmission mode** - Each 8-bit byte is sent as two ASCII characters; allows time intervals of up to 1 second to occur between characters without causing an error.
- **RTU transmission mode** - Each 8-bit byte is sent as two, 4-bit hexadecimal characters; higher transfer rate than ASCII mode.

ModBus message frames are constructed as shown in Figure 3–2. Each frame section is described in the list below.

Figure 3–2. ModBus Message Frame

Address	Function	Data	Check Sum
---------	----------	------	-----------

2717-7

- **Address** - the network address of the slave or client (2717); contains either 2 characters (ASCII) or 8 bits (RTU); valid device addresses range from 0-247 decimal.
- **Function or command** - describes the action to be performed; contains commands such as “read register” or “write register”; contains either 2 characters (ASCII) or 8 bits (RTU); valid codes range from 1-255 decimal.
- **Data** - consists of 2 sets of hexadecimal digits ranging from 00-FF; includes information such as register addresses.
- **Checksum** - LRC or CRC methods of error checking; method used depends on how the message is transmitted.

Because this protocol is a messaging structure, it does not rely on the underlying physical layer. It is generally implemented on a serial interface such as RS-232 or RS-485 over different types of media (fiber, radio, etc.).

ModBus TCP/IP

Another way to transmit data is ModBus TCP/IP. This variant uses TCP/IP and Ethernet to carry a ModBus message and allows users to send messages over an intranet or the Internet. By combining Ethernet with a universal networking standard (TCP/IP) and a vendor-neutral data representation, this protocol provides a truly open and accessible network for exchanging process data on a local intranet or from one remote location to another.

ModBus Client Protocol

The ModBus Client protocol lets a CTC controller poll a remote controller that supports the ModBus protocol. Only multiple/single register read and write transactions are allowed. The peer-to-peer block registers perform this function and allow the controller to reference values locally while they are dynamically updated over the network in the background. The example below shows how a ModBus Client peer-to-peer block is initialized.

```
21005 = 4      # Set the number of registers in the block for allocation.
21000 = 12     # Set the peer IP address.
21001 = 40
21002 = 53
21003 = 27
21008 = 1003
21009 = 2      #Set the ModBus Client protocol.
21008 = 0      #Set this register to its previous value.
21004 = 8001   #Set the register to start monitoring.
21006 = 50     # Set the number of milliseconds between updates.
```

This program sample shows an attempt at making a TCP connection to the initialized ModBus Server Port, which is normally 502. Once this connection is made, the first 4 registers starting at 8001 are read and stored locally and are updated every 50 milliseconds. A ModBus multiple read command request is then made of the remote controller. You can make write requests to the remote controller with the standard peer-to-peer register access methods.

Certain index register values (1004-1006) within the Peer-to-Peer Registers apply to the ModBus Client Protocol. For more information, refer to the *21XX8 - Index Offset Register* listing in *Chapter 4, Special Registers*.

ModBus Server Protocol

The ModBus Server Protocol lets a CTC controller transmit register information and updates over a ModBus connection. Set up a peer-to-peer register block (pointing to the 2717's IP address) that starts with the register that will be queried by a remote controller.

This block also has the following stipulations:

- Block size can exceed the size of a retrieved data block but cannot exceed the size of a requested data block.
- Requests for data must be on odd register boundaries that contain a minimum of 2 registers. This requires a multiple register read or write transaction (4001, 4003, 8005, and so forth) because the ModBus protocol defines registers as 16-bit and CTC's registers are 32-bit registers.

Transmission Control Protocol/Internet Protocol (TCP/IP)

TCP/IP is a universal protocol for computer communications that is capable of interconnecting computers regardless of platform or operating system differences. It “carries” data across a network and governs how data is moved. It is regarded as the industry standard for open networking. On a TCP/IP network, you can send/receive e-mail, transfer files, log onto remote computers, and browse the World Wide Web.

Packet Transmission and Routable Protocols

TCP/IP tells network devices how to handle data when it is transmitted across a network. When e-mail is sent to someone, it must first be broken down into smaller chunks, or packets. Each packet contains the data to transmit and control information that tells the network what to do with the packet. Since packets may not arrive in the correct order, TCP/IP makes sure that data is re-assembled correctly at the receiving end.

It accomplishes these tasks with the help of routers. Routers are internetworking devices that connect similar and heterogeneous network segments into internetworks. They move packets by accessing the network layer address in the packet and then deciding where to move them. Since TCP/IP contains a network address layer, it is called a routable protocol.

Routable protocols have a layering structure that resembles the Network Layer of the OSI Reference Model. They are primarily used to move data beyond the boundaries of a single LAN and can tabulate network connections before selecting the most efficient path for transmitting data between devices. They can also dynamically select an alternative path if a designated path fails. The price to pay for all this functionality is a loss in speed. Larger data packets add to system overhead and lead to slower transmission rates.

TCP/IP Services

A network service is a special function such as e-mail which is available to the network and its computers. TCP/IP has three categories of services which operate at specific levels of the network hierarchy. These are:

- **Connection Services** - operate at the lowest level of the TCP/IP stack and determine how data moves from one computer onto the network cable and then how that data moves from the network cable to the next computer. These services do not guarantee that data will arrive in the correct order or even that it will arrive at all.
- **Transport Services** - operate at the middle level of the TCP/IP stack and enhance connection services to provide reliable communications between computers. Packets are numbered to ensure that data is ultimately placed in the right order. Computers then error-check the data to make sure it is not lost or damaged.
- **Application Services** - operate at the highest level of the TCP/IP stack and let an application on one computer talk to a similar application on another computer in order to perform a task such as copying files. These services depend on the other services to guarantee reliable, efficient communication.

Client/Server Computing and TCP/IP

Client/server computing combines traditional styles of computing into a different, distributed way of working. These styles include mainframe computing, non-networked desktop PCs, enterprise-wide databases, and so forth. Client/server is defined by software rather than hardware. A client application on one computer requests services from a computer running server software.

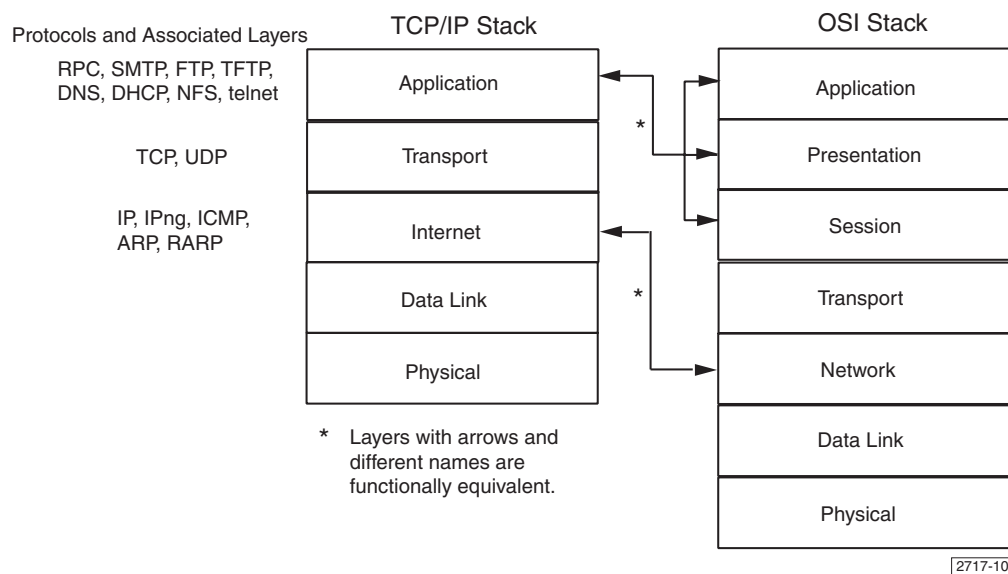
This concept is at the heart of how a web server and browser work together. A web “server” accepts requests from browser “clients” to deliver web objects such as home pages and other documents. The browser “client” receives services from a web server and then lets users navigate through the files and associated information.

TCP/IP is one of the major enablers of client/server computing as well as one of the biggest users of it. Its layered and modular design make it easy to design and implement new network services. For more information on client/server systems, refer to the *Client/Server Computing* section in this chapter.

TCP/IP Layers

TCP/IP follows the OSI (Open Systems Interconnect) example of modular layers and clean interfaces. The seven-layer OSI Reference Model is out of scope for this discussion, but it’s mentioned because the TCP/IP model, with its five-layer stack, closely resembles the OSI model. Figure 3–3 compares the two model types.

Figure 3–3. TCP/IP and OSI Stack Comparison



Each layer depends on the layer below it. In other words, each layer provides services to the layer above it. When two computers are communicating with each other, each computer has its own set of layers. When an e-mail message is sent to someone on the network, it starts at the top layer and travels down through all layers to the bottom of the stack, then travels to the computer of the intended recipient. There, it starts at the bottom layer and moves up the stack to the top (application) layer.

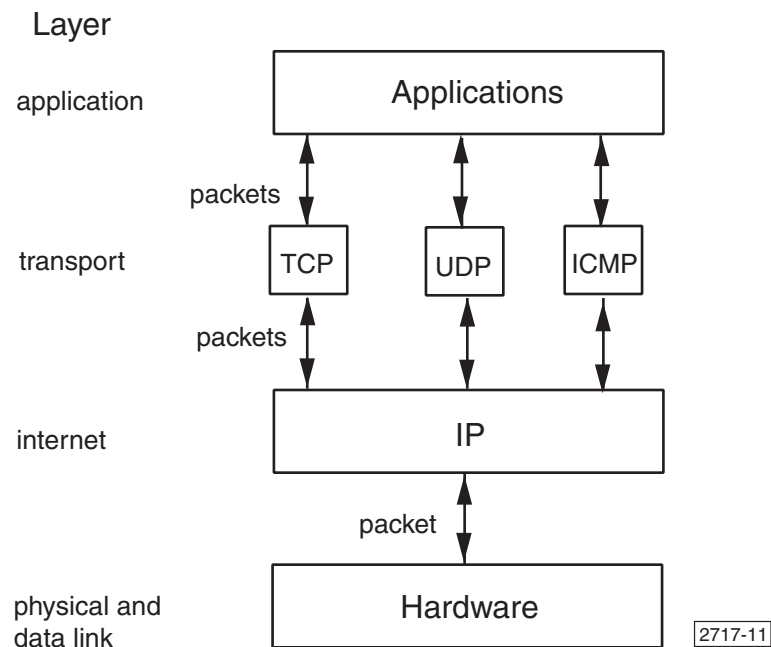
TCP/IP Protocol Suite

This section describes the most common members of the TCP/IP Protocol Suite. These are:

- **IP (Internet Protocol)** - IP is responsible for basic network connectivity. Its core works with Internet addresses. Every computer on a TCP/IP network requires a numeric address. The IP on a user's computer knows how and where to send messages to these addresses. IP has no way of knowing whether a packet gets lost and doesn't arrive at its destination.
- **TCP (Transmission Control Protocol)** - TCP makes sure that data isn't lost and arrives safely at its destination. It checks packets for errors and numbers them in sequence.
- **UDP (User Datagram Protocol)** - UDP uses IP to deliver packets to upper layer applications but does not check for errors or number data packets. Data is not re-sent in case of error.
- **ICMP (Internet Control Message Protocol)** - This protocol reports problems and relays other network-specific information such as error status from a network device.
- **FTP (File Transfer Protocol)** - This protocol helps users to copy files between two computers. Users can either "get" (download) the files from a remote PC or "put" (upload) files onto a remote computer.
- **SMTP (Simple Mail Transfer Protocol)** - This is the protocol for Internet e-mail. It transfers e-mail messages between computers.
- **HTTP (HyperText Transfer Protocol)** - This protocol transfers HTML and other components from the servers out on the network and World Wide Web and sends it back to a browser client.
- **DHCP (Dynamic Host Configuration Protocol)** - This protocol is a client/server solution for sharing numeric IP addresses. A DHCP server maintains a pool of shared addresses which are recycled. When a DHCP computer wants to use a TCP/IP application, the client must request an IP address from the DHCP server. The server then checks the shared supply and if all addresses are in use, the server sends a "busy signal" to the client and tells it to try again later. This approach works in areas where computers don't use TCP/IP applications all the time or when there aren't enough addresses available for the computers that need them. Some ISPs (Internet Service Providers) also use this approach. For example, when a user signs up with an ISP, they might be given either a static (permanent) address or get a lower rate because of dynamic (temporary) addressing. Static addresses ensure (given the reliability of the Internet) a connection every time, but dynamic addresses make no such promise. If all available addresses are currently in use when users attempt to log on, they are told to try again later.

Figure 3–4 shows the relationship between IP, TCP, and UDP and the applications at the upper layer.

Figure 3–4. IP, TCP, and UDP Relationships



TCP/IP Addressing

An IP address is a set of numbers separated by dots. It represents one network interface on a host. Every network interface (a host may have more than one) requires a unique IP address. Each address is a 32-bit number that is divided into two sections: the network number and the host number. The entire address is actually a binary number that is broken down into 4 fields of 8 bits each which are then separated by dots. Each field can be a number ranging from 0 to 255.

To connect a network to the Internet, users need an official block of addresses. The overseer of IP addresses is the InterNIC. Local ISPs may also assign official addresses when their services are used.

In the first paragraph in this section, it was stated that each address has four fields of 8 bits each. The meaning of each field depends on the network class. There are currently four network classes, but only three are widely used. The *Network Classes* section discusses these classes in more detail.

For example, suppose that 192.6.132.0, a class C network, is assigned to a company. Class C networks reserve the first three fields for a network number, so all interfaces on this network should share this prefix. The last zero is replaced by a number from 0-254 for the node address. Therefore, nodes on the network have addresses like 192.6.132.1, 192.6.132.2, and so forth.

Network Classes

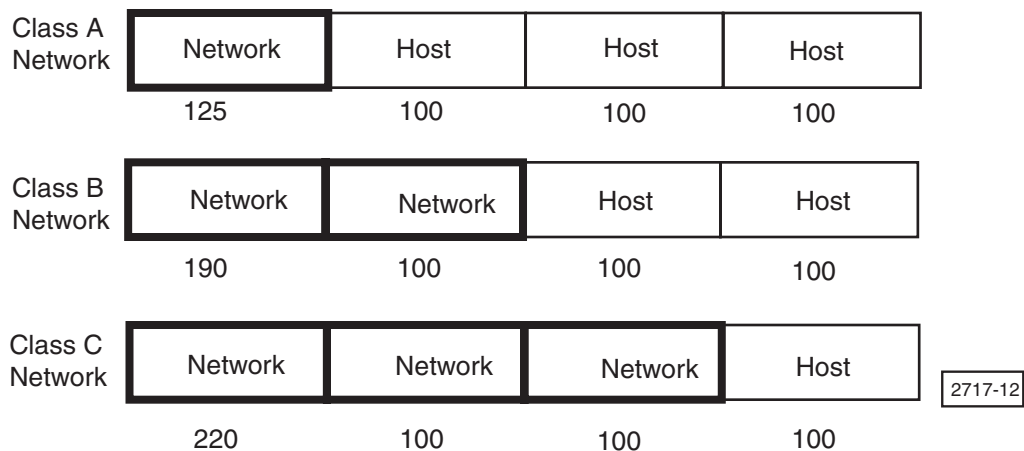
IP-based networks are separated by size and are divided into three primary classes:

- **Class A** - These are enormous networks with up to approximately 17 million nodes (or hosts). Network addresses range from 1.0.0.0 to 126.0.0.0. The zeros are ultimately replaced with node addresses. Class A networks reserve the first field for the network number and the last three fields for node numbers.
- **Class B** - These are large networks with up to 65,000 nodes. Network addresses range from 128.0.0.0 to 191.0.0.0 where the last two zeros are replaced with node addresses. Class B networks reserve the first two fields for the network number and the last two fields for the node number.
- **Class C** - These are small networks with a maximum of 254 nodes. Network addresses range from 192.0.0.0 to 223.0.0.0 where the last zero is replaced by a node address. Class C networks reserve the first three fields for the network number and the last field for node numbers.

Figure 3–5 shows how the fields of an IP address correspond to a class of network.

Figure 3–5. IP Addresses and Network Classes

Note: The boxed-in area with the heavy lines represents the network number of the IP address and the remainder represents the host number.




For a given network address, the last node address is the broadcast address. For example, for a class C network with address 192.168.1.0, the address 192.168.1.255 is the broadcast address, which is used to broadcast to all nodes on the network. Therefore, this address and the one that ends in zero should not be used as node addresses.

Most addresses now available through the InterNIC are Class C addresses. Class D and Class E networks are primarily used for experimental purposes.

Constructing Addresses in Binary Notation

IP addresses are generally expressed as decimal numbers, but as stated previously, they are actually binary numbers. A computer sees the number as a sequence of zeros and ones, because computers do everything in binary. For example, Figure 3–6 pulls the number 127 apart to show how it is constructed in binary notation. The place value columns have 1s, 2s, 4s, and so forth, instead of the familiar 1s, 10s, 100s, and so forth, from the decimal system.

Figure 3–6. Binary Notation for Decimal Number 127

Class A Network							
128	64	32	16	4	2	1	Place Value Columns
0	1	1	1	1	1	1	Bit Values (either 0 or 1)
High order bit						Low order bit	
							<div>2717-13</div>

If each and every bit of the class A network is set to 0 or 1, a higher number than the 127 allowed by the Internet is achieved. TCP/IP requires that the higher-order bit for a class A network always be 0. By following this rule and adding up the bits as shown in Figure 3–6, the number of class A networks allowed by a 32-bit address is achieved.

For class B networks, the first two higher-order bits must be 1 and 0, and for class C networks, the first two higher-order bits must be 1 and 1.

Subnets

On a regular IP network, every interface on the same physical network sees all the packets sent out on the network. As the network grows, the volume of network traffic grows along with it and performance starts to degrade. In cases like this, users might want to divide a network into multiple subnetworks, or subnets.

Subnets have the following advantages over large networks:

- Smaller networks are easier to manage and troubleshoot, even though there are more pieces.
- Network traffic is reduced and performance usually improves because most traffic stays local to a subnet.
- Network security is more easily administered at the interconnections between subnets.

Subnet Masks

When subnets are created, users need to borrow bits from the host section of the main network address. These bits allow each subnet to have its own network address. TCP/IP needs to know the particular host bits that are being appropriated for use in the network address. Subnet masks are used to borrow the host bits. A subnet mask is 32-bits long; the bits for the network address are set to 1 and the bits for the host address are all set to 0.

Before defining the mask, you need to determine how many subnets they want to create and how many hosts will be on each subnet. This determines how many bits should be set to 1.

Despite the fact that some networks don't have any subnets, they still have subnet masks. Figure 3–7 shows the default subnet mask for each class of network.

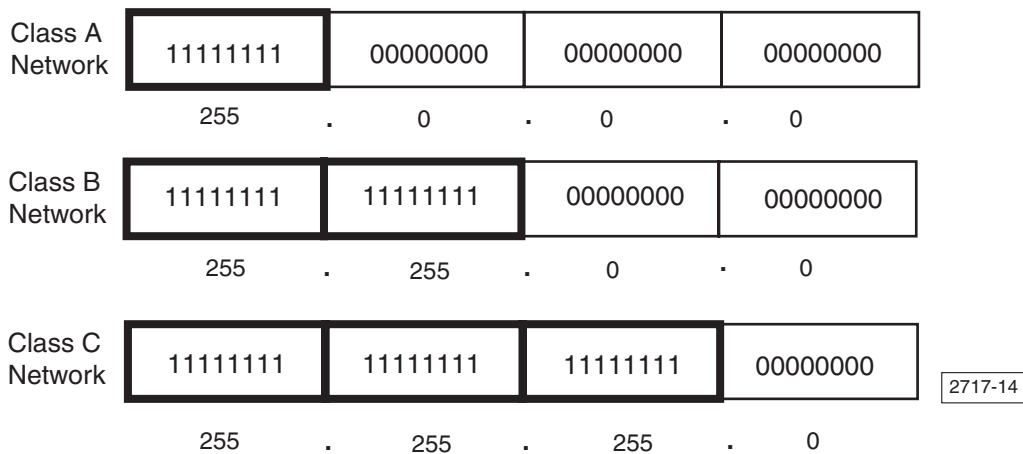


Note

The mask must be the same for each interface on the network.

Figure 3–7. Default Subnet Masks for Different Network Classes

Note: The boxed-in area with the heavy lines represents the network number of the subnet mask and the remainder represents the host number. Network address bits are set to 1 and host address bits are set to 0.



After deciding on the number of subnets and hosts, you can then apply the subnet mask to the IP address in every message coming over the network in order to separate the network number and the host number.

Applying Subnet Masks To A Network

After choosing a subnet mask, you should perform a logical AND operation between the mask and their network addresses. You have the option of using a default mask or a custom mask in this operation.

When TCP/IP was first developed, only a limited number of computer systems existed and IP addresses were plentiful. It was assumed that there would always be enough addresses to go around. Because of this, Classic networking rules were developed. The Classic rules for networking and subnetting made lavish use of IP addresses and frequently wasted large numbers of addresses.

Over time, as growth on the Internet exploded, the Classic rules were replaced with a newer set of rules called Classless Inter-Domain Routing, or CIDR (pronounced cider). CIDR rules make better use of available IP addresses, but unfortunately, there are still pieces of equipment and applications that use classic IP rules.

If a network has only CIDR-compliant equipment and applications, you should use the CIDR subnetting rules. If older equipment or applications are employed (particularly older Ethernet-TCP/IP printer cards or UNIX systems), then use the Classic subnetting rules.

The following examples illustrate the use of default and custom masks using both Classic and CIDR subnetting rules.

Default Masks

In this example, the following Class C network addresses are used:

204.123.16.4 204.123.17.4

The binary representation of each address is:

204.123.16.4 11001100 01111011 00010000 00000100
204.123.17.4 11001100 01111011 00010001 00000100

The default subnet mask for a Class C network is 255.255.255.0, which is represented in binary as 24 ones followed by 8 zeroes. This masks the left 24 bits of both addresses and identifies those bits as the network addresses. This gives a network address of 204.123.16.0 for the first address and 204.123.17.0 for the second address as illustrated in Figure 3–8.

Figure 3–8. Binary Representation of Default Masks

	204.123.16.4	11001100	01111011	00010000	00000100
AND	255.255.255.0	11111111	11111111	11111111	00000000
	204.123.16.0	11001100	01111011	00010000	00000000
	204.123.17.4	11001100	01111011	00010001	00000100
AND	255.255.255.0	11111111	11111111	11111111	00000000
	204.123.17.0	11001100	01111011	00010001	00000000

2717-15

Since these network addresses are clearly different, the default mask defines that these addresses belong to two different networks.

Custom Masks - Classic Rules

A common situation on a Class C network is the need to break a single network into two subnets.

Under Classic rules, the subnet mask most commonly used is 255.255.255.192, which results in 2 subnets of 62 usable addresses each. This may cause you to wonder what happened to all the other addresses. Under both classic and CIDR IP rules, within a subnet, if the host portion of the address is all zeroes or all ones, then it has a special meaning.

A host address with all ones is the broadcast address for the subnet. In other words, packets addressed to the broadcast address are received by all devices on the subnet. A host address with all zeroes is the same as the network address and cannot be used as an actual host address.

With Classic rules and a subnet mask of 192, you have 4 possible subnets:

0	00 000000
64	01 000000
128	10 000000
192	11 000000

With Classic rules, it is recommended that you drop the first and last possible subnets because the first one has a subnet ID that is all zeroes and the last has a subnet ID that is all ones. This leaves the second (64) and third (128) choices for subnets, and in each case, we lose the addresses whose host sections result in all zeroes or all ones. This is where we derive the result of 2 subnets of 62 addresses each.

Therefore, for Classic IP rules, a Class C network subnetted with the mask 255.255.255.192 results in two recommended usable networks. The first has a network address of 64, a broadcast address of 128, and 62 available addresses in the range of 65-126. The second subnet has a network address of 128, a broadcast address of 192, and 62 available addresses in the range of 129-190.

A summary of the recommended net masks for a Class C network using Classic IP rules is shown in Table 3–1.

Table 3–1. Recommended Subnet Masks - Classic IP Rules

Class C # of Bits	Mask	Available Subnets	Available Addresses
2	255.255.255.192	2	62
3	255.255.255.224	6	30
4	255.255.255.240	14	14
5	255.255.255.248	30	6
6	255.255.255.252	62	2

Custom Masks - CIDR Rules

Under CIDR rules, the mask that is used to break a Class C network into 2 subnets is 255.255.255.128, which results in 2 subnets of 126 usable addresses each. As with Classic IP rules, within a subnet, the addresses with host portions that are all zeroes or all ones are not allowed.

With a subnet mask of 128, there are two possible subnets:

0	00 000000
128	10 000000

Therefore, under CIDR rules, a Class C network with a mask of 255.255.255.128 results in two usable subnets. The first has a network address of 0, a broadcast address of 127, and 126 available addresses in the range of 1-126. The second subnet has a network address of 128, a broadcast address of 255, and 126 available addresses in the range of 129-254.

A summary of the recommended net masks for a Class C network using CIDR rules is shown in Table 3–2.

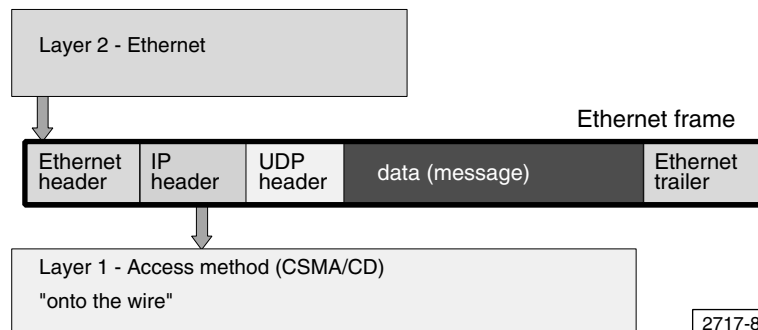
Table 3–2. Recommended Subnet Masks - CIDR Rules

Class C # of Bits	Mask	Available Subnets	Available Addresses
1	255.255.255.128	2	126
2	255.255.255.192	4	62
3	255.255.255.224	8	30
4	255.255.255.240	16	14
56	255.255.255.248	32	6
6	255.255.255.252	64	2

User Datagram Protocol (UDP)

UDP, which resides in the transport layer of the TCP/IP stack, is a connectionless communications protocol that is used on packet-switched, communications networks. Reliable delivery is not required. The UDP packet structure is shown in Figure 3–9.

Figure 3–9. UDP Packet Structure



Datagrams

A datagram, or packet, consists of one single unit of binary data. The first 8 bytes contain the header information and the remaining bytes contain the data.

Headers

A header has 4 fields of two bytes each. The fields are:

- **Source Port Number** - The sending application sends datagrams through the source port. Different applications can maintain their own data channels, which allows simultaneous transmissions. Valid port numbers range from 0-65535.
- **Destination Port Number** - The location where the packet recipient accepts datagrams. Valid port numbers range from 0-65535.

- **Datagram Size** - The number of bytes contained in the header and data sections. The maximum size depends on the operating environment. With a 2 byte field, the theoretical maximum size is 65535 bytes. Some applications restrict the datagram to a smaller number.
- **Checksum**- An optional safety feature that represents encoded datagram data that is calculated first by the sender and then by the receiver. If a datagram is tampered with or is corrupted during transmission, the calculations made at each end will not match and UDP detects an error. Turning off this feature squeezes extra performance out of a system.

Services

UDP provides two services that are not offered by the IP layer:

1. A direct interface with IP and the ability to address a particular application through a port number (address where an application is available on a particular host) This helps to distinguish between different user requests and does not require connection sessions. An entire transmission can be sent in one or two datagrams (packets).
2. An optional checksum capability to verify that data has arrived intact.

UDP does not provide the service of dividing a message into packets and reassembling it at the other end. Because of this limitation, UDP is often used for video and audio multicasts that deliver real-time data from end station to end station across the Internet or an intranet. Because the multicasts are live, the services offered by TCP are unnecessary and add too much overhead. If a packet is lost, it is not practical to retransmit information that is out of sync with the current audio and video being received by the destination.

Client/Server Computing

This section describes the client/server computing model.

What is a Client?

A client is either a device or user on a network that takes advantage of the services offered by a server. A client is often loosely defined as a computer on the network, but it also refers to a user that runs the client side of a client/server application.

What is a Server?

A server is a network-connected computer system that provides services to network users. Computers can act as file servers, application servers, database servers, e-mail gateways, and communications servers. These systems run network operating systems such as Novell Netware, Windows NT/2000 Server, and UNIX.

Client/Server Model Defined

Client/Server is a computational architecture that involves client processes requesting service from server processes. A server process running on one system waits for a client's request. The request may be for a file or to process a sophisticated transaction that takes place over multiple servers.

Client/Server networking focuses primarily on the applications and not the hardware. The same devices may function as both client and server. For example, web server hardware functions as both client and server when local browser sessions are run there.

In industrial control applications, a CTC controller or module can act as either a client or a server. This is determined by the program residing in the controller and how it interacts with other devices. These devices include other controllers or modules on a network and may also include a PC that is communicating with the controller.

Internet Applications

Some of the most popular Internet applications use the client/server model:

- E-mail clients
- FTP clients
- Web browsers

These programs present a user interface (graphic or text-based) in a client process that lets you connect to servers. For e-mail and FTP, computer names (or IP addresses) are entered in the interface to set up future connections to the server process.

For example, when you first configure your e-mail client, you need to specify a configuration name such as smtp.ISP_Name.net before you can send messages over the Internet. In general, this only needs to be done once since the server information rarely changes.

For FTP, a different server name is usually entered each time you use the FTP client. For Web browsers, the server's name or address appears in the URL of each request.

You may start surfing at one site and access a bunch of different servers as you click on links. Server information is provided by the Web developer.

Pros and Cons of Client/Server Computing

There are many benefits and a few drawbacks to the client/server computing model. This section discusses both aspects of this model.

Benefits

Some of the benefits of using this model include:

- **Improved scalability** - Connections are made as needed instead of being hard-wired.
- **Modular applications** - In two-tier and three-tier types of client/server systems, software applications are separated into modular pieces that are installed on hardware specialized for that system.
- **Enterprise-wide communication** - The client/server model helps companies down-size from mainframes/minicomputers to networks that provide an enterprise-wide data communications platform.
- **Parallel processing** - Multiple systems can join together for parallel processing, where they work cooperatively to complete a processing task.
- **Local storage** - Data is stored close to the services that act on that data, which minimizes network traffic.
- **Caching** - A large amount of information is cached once into the server's memory rather than the memory of every workstation that needs it.
- **Reduction in network traffic** - Servers only provide the information that is requested instead of large chunks of data that the workstation must process.
- **Offloaded data** - Large servers can offload applications that are better handled by personal workstations.
- **Secure data warehousing** - Information is safe and secure in one location. Data warehousing makes specific data available at intermediate servers while maintaining control of the data.
- **Centralized data** - Administrators can apply security controls to restrict data access and use tracking mechanisms to monitor data access.

Drawbacks

There are a few drawbacks to the client/server model:

- **Difficult system management** - Applications are often distributed across a network, which makes it challenging to keep configuration information up-to-date and consistent among all devices on a network.
- **Managing upgrades** - Upgrades to a newer version of a client/server application are sometimes difficult to synchronize or stage appropriately.
- **Network Reliability** - Client/Server systems rely heavily on a network's reliability. Redundancy and fail-safe features are sometimes expensive to implement.

In summary, the client/server approach is a cost-effective way to share data between small or large numbers of clients. Usually, the client and server are separate entities on a network, but client/server systems work especially well on wide-area networks such as the Internet.

Setting up an Intranet with CTC Controllers

This section provides a broad overview of the decisions you need to make when setting up a manufacturing intranet. General networking information is outlined along with the recommended approach to networking CTC's controllers. CTC controllers have the following network capabilities:

- **Data rates of 10 Mbps (10 million bits per second) or 100 Mbps (100 million bits per second)**- Information is transmitted in packets of no more than 1500 bytes. For example, a 150K file would be split into 100 packets.
- **Built-in protocols** - DOS and Windows drivers are available that allow computer-controller communications.
- **Asynchronous operation** - The ability to start the next I/O operation before the current one is completed.
- **Host communications** - Individual controller read/write capability, the ability to download/upload Quickstep programs through the controller's RS-232 port with programs such as CTC Utilities, and access to all controller resources.
- **Peer-to-Peer communications with indirect node access** - Refer to the *Peer-to-Peer Client/Server Protocol* section for more information.
- **Built-in controller security from the network.**
- **Built-in error checking.**
- **Network access from any controller's RS-232 port** - A host computer can interrogate the network area continuously while local computers or operator interface terminals can simultaneously access the network port using conventional communications protocols from any controller's RS-232 port. For fast data retrieval, the controller supports block area transfer from a single command request both locally and over the network.

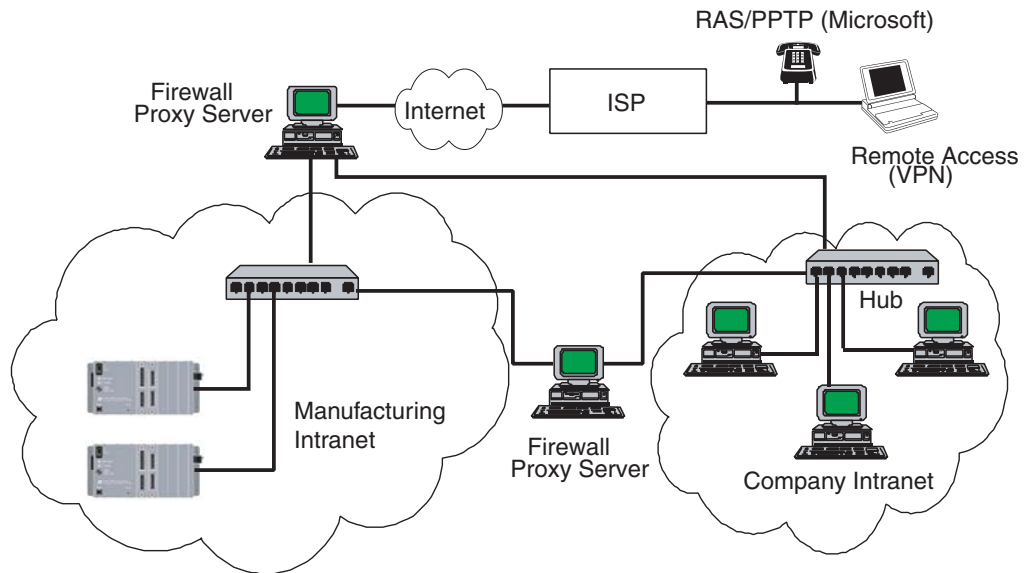
There are many considerations to make in the design of a manufacturing intranet. The two most important issues are a company's networking policy and its network's capabilities. An intranet needs to conform to a company's policy. If it deviates in any way, then network administrators should assist with the installation process. It's also important to determine a network's capabilities before doing any installation. If the intranet is completely isolated, then the design process is easier. However, in all likelihood, an intranet's design must allow end users to communicate between the company intranet and the manufacturing intranet. Authorized remote users that require access to the manufacturing intranet over a secure telephone line must also be considered.

Before implementing an intranet design, users also need to consider the following items:

1. **Each networked controller requires a unique IP address.** There are two ways to provide IP addresses to the controller. The first method, which is used here at CTC, is to code the address into each controller. The second method is to use a DHCP (Dynamic Host Configuration Protocol) server, which is not currently supported on CTC controllers. This protocol automatically assigns and keeps track of IP addresses. When the controller comes on-line, it requests an available address from the server and dynamically configures itself to use the requested address. Make sure that the pool of available addresses can accommodate all available controllers.
2. **The manufacturing intranet is a real-time system.** A controller must not interrupt its current task to process incoming requests from outside the manufacturing intranet. Therefore, CTC recommends using a proxy server to buffer the manufacturing intranet from the company intranet as well as remote users. Proxy servers sit between client applications (a Web browser) and a real server (on your manufacturing intranet). It intercepts all requests to the real server to see if it can fulfill them itself by referring to previously cached information.
3. **The manufacturing intranet must be free of viruses and all unnecessary software.** CTC recommends using firewalls to shield the manufacturing intranet from the company intranet and from remote users. Firewalls are systems or groups of systems that enforce an access control policy between two networks (or intranets). This can be accomplished in several different ways, but the firewall usually acts to block traffic or to permit traffic between two different networks. This prevents unauthorized users from accessing resources on the manufacturing intranet. Although a proxy server is not a firewall, it is generally a recommended component of a firewall system.
4. **Intranet design must consider the needs of remote users.** TCP/IP is an open networking standard that allows users, regardless of platform or location, to use e-mail, transfer files (FTP), log onto remote computers (telnet), or browse the Web. Because of its wide acceptance, it is incorporated into many applications and allows someone like a debugging engineer to monitor the factory floor from another building, another state, or even another continent. Because of an intranet's specific security needs, two remote access methods are viable:
 - The first access method is to have your engineer dial into a private bank of modems. The disadvantage here is the cost of a long-distance phone call.
 - A second, less expensive method is to create a VPN (Virtual Private Network). An engineer can dial into their Internet Service Provider with a local call and is then able to access the manufacturing intranet over the Internet. CTC recommends using Microsoft RAS (remote access server, which is an optional component of Windows 2000) and the built-in option for Microsoft's PPTP (point-to-point tunneling protocol). PPTP works in conjunction with a VPN to "tunnel" through the Internet and create a channel between an intranet and the outside world. Refer to the *Virtual Private Networks* section for more information on implementing a VPN.

Figure 3–10 shows a typical setup for an intranet.

Figure 3–10. Typical Intranet Setup



Notes: All communications are two-way.
 Controllers and other devices can be added to these
 intranets provided that you follow the IEEE 802.3
 Ethernet standard for 10Base-T or 100Base-T networks.

2717-16

Virtual Private Networks

A Virtual Private Network (VPN) is a connection between a remote site and a company's intranet or extranet. Although VPNs mimic a dedicated line, VPN connections are carried over the Internet. Using a technique called tunneling, data packets are wrapped in an IP sandwich and are transmitted across a publicly routed network (the Internet) in a tunnel that simulates a point-to-point connection. Tunnels can be opened with PPTP, Microsoft's built-in VPN component, or an emerging standard such as Layer Two Tunneling Protocol (L2TP). Windows 2000 also supports digital certificates, which further enhances VPN security.

A VPN connection simulates a dedicated link and has many of its advantages without its inherently high cost. Companies can extend their corporate network out to distant offices, telecommuters, business partners, and customers by using worldwide IP network services such as local service provider backbones. For the cost of a local phone call to an ISP, a user gets many of the benefits of a leased line without the toll charges.

Other advantages include:

- **Scalability** - You can easily expand the capacity and reach of your network by making arrangements with your network service provider. This flexibility allows you to rapidly respond to organizational change and market demands.
- **Equipment** - VPNs help you to save on equipment installation, maintenance, and obsolescence. Existing modem pools can be reduced in favor of dial-up traffic over an existing or augmented Internet connection.
- **Tunnel Switching** - This feature allows multiple users to enable tunnels that are terminated at different locations. Traffic from partners and customers can be separated from remote employee traffic.
- **Full Control** - You may decide to out source some of your overhead (facilities, administration, and other services) to your network service provider, but you can still maintain full control of user authentication, access privileges, network addressing, and management of network changes.
- **Transparency** - Remote users can easily access your network over a secure connection. Connections are so transparent that users feel like they're directly connected to the network.

Before implementing a VPN, you must determine the connectivity requirements between corporate offices, telecommuters, and traveling employees. Your VPN may allow a remote user access to all of your network resources or it may limit them to only one or two resources at corporate headquarters.

Requirements for network connectivity often include:

- Corporate Security Policy
- Business Models
- Intranet Server Access
- Application Requirements

- Data Sharing
- Application Server Access

Because VPNs are platform-independent, any computer that is configured to run on an IP network can be easily incorporated onto a VPN. No modifications are required except the installation of remote access software.

Depending on your requirements, your VPN may require a great deal of computational power. If so, a hardware-based VPN product delivers the best performance. Hardware also offers tighter physical and logical security and is designed to be scalable so it can accommodate an increasing bandwidth demand, more VPNs, and higher speed lines. However, hardware solutions generate a large amount of operational overhead and installation will generally require the services of a qualified technician.

Software solutions are best suited for lower volume connections at small to medium-sized companies with minimal security requirements. Software costs less than hardware, but it also offers less performance. Encryption and decryption chew up CPU cycle time and bog down the system. Software is also less secure and is more vulnerable to hack attacks. Reliability is also an issue, because software depends on complex computer systems that may cause glitches. As with hardware, installation and management of the software is difficult to set up and maintain.

Your implementation plan must also consider whether you have the staff to maintain a VPN. If your resources are inadequate, then you may need to out source some of the VPN services to a network service provider.

Before implementing any program or deciding who will ultimately be responsible for maintaining the VPN, you should institute a pilot program. This allows you to scope the management burden along with any economic benefits. PPTP and L2TP are good candidates for this program and you should also explore private-key cryptosystems.

VPN implementation involves a number of trade-offs. Although you can virtually guarantee that you'll save money by eliminating leased lines and most long-distance phone calls, you also need to consider these other factors:

- **Internet connections and leased lines** - It may be less expensive to run your backbone over the Internet, but you cannot count on the Internet's reliability. Internet glitches could result in delays or loss of connectivity. Although leased lines are expensive, they also offer unparalleled safe and reliable data transport. Before entrusting the Internet with any sensitive data, you must institute the proper protocols so that only authorized users can access your network.
- **System administration** - VPN implementation requires a high level of knowledge to properly install any software or hardware and manage the security and reliability of an Internet connection. Part of this complexity involves data encryption, user authentication, and access rights. While your organization may have someone with the required skills, it may not be prudent to tax them with this additional burden in cases where resources are limited. In situations like this, companies may decide to off load such duties to a network service provider.
- **Speed** - VPN performance is determined by the speed of transmissions over the Internet and the efficiency of VPN processing at each end of the connection. When you

encapsulate data in an IP packet, you add information to that packet and increase its size. This degrades performance, because this extra padding increases the chance that an Internet router will decide that a packet is oversized and will end up fragmenting it. If your data is also encrypted, then this can reduce dial-in system performance to unacceptable levels. Data compression helps to solve this problem, but the combination of compression and encapsulation requires computational power that goes beyond what is required for security.

Remote Access Service (RAS)

RAS allows remote users to dial into a Windows RAS server and use the network's resources as if they were directly connected. The remote user logs onto their PC and checks a box in their log-in window that establishes the RAS connection and authenticates the session.

RAS uses the standard single-logon technique. RAS can also be configured so it automatically calls back a specific number for each account, which ensures that a user's remote access privileges only originate from a specific phone number. RAS requires that all communications be encrypted with a 40-bit or 128-bit cipher.

Point-to-Point Tunneling Protocol (PPTP)

Microsoft's PPTP is bundled with various versions of the Windows operating system and is currently the most widely used protocol for VPNs. PPTP is part of Remote Access Service (RAS), a service that combines remote access and VPN connectivity. It extends RAS use to the Internet and allows a remote RAS client to dial a local Internet service provider and establish an Internet link to the resources of the RAS server itself. Configuration is generally straightforward, especially if the network has basic needs. However, the interface is a bit awkward and the protocol is not as secure as it should be.

PPTP has adequate features and provides enough security for businesses whose data isn't very sensitive. Because it's directly integrated within each OS, user administration is fairly easy. PPTP uses Microsoft Point-to-Point Encryption (MPPE), which adds integrated data encryption into standard Microsoft dial-up networking. PPP packets on the client workstation are encrypted before entering a PPP tunnel. When the workstation negotiates PPP with a tunnel terminator, an encryption session is initiated. However, because the authentication session happens before encryption is active and MPPE encryption does not offer data authentication and integrity services, PPTP is not as secure as L2TP used in conjunction with Internet Protocol Security (IPSec).

If you have a small network and all you want to do is replace your leased lines and existing dial-up access with a VPN, then PPTP over MPPE works well. If you want certificates, extra-nets, or interoperability with other VPNs, then this protocol is not the best solution.

Layer Two Tunneling Protocol (L2TP)

L2TP is based on PPTP and the Layer Two Forwarding Protocol (L2F). L2F is a more robust protocol that supports the encapsulation of more protocols than PPTP. It can help you create multiple secure and reliable communications channels between two endpoints such as a client and server. Other benefits include:

- **Prioritization of Network Traffic** - Depending on the needs of your application, one tunnel can receive a higher quality of service than another.
- **Improved scalability** - The MultiLink Point-to-Point Protocol (MPPP) is supported. This allows a Windows 2000 server to load-share across multiple ISP connections and for a destination Windows 2000 server to reassemble and reorder data as required.

L2TP uses IPSec authentication to protect data on the connection. IPSec is an emerging standard that was developed by the Internet Engineering Task Force (IETF). It is a do-everything, tunneling and security protocol for IP. It consists of a set of IP-level protocols for setting up an “agreement” between two IP workstations about the encryption and digital signature methods that will be used. It is more robust than MPPE and encompasses user authorization, privacy, and data integrity. It can also extend beyond a tunnel terminator to a destination host workstation. Windows 2000 supports IPSec, which is tightly integrated with system policy management to enforce encryption between systems and make it transparent to end users.

This page is intentionally left blank.

Special Registers

Network and Communications Registers	65
Serial Port Redirection TCP Protocol (Server) - Register 20120 - R/W	67
Peer-to-Peer Protocol Registers	67

This page is intentionally left blank.

Special Purpose Registers

The 2717 module uses certain registers within the 20000 block for networking and communications. Registers 21000-21999 are used for peer-to-peer communications. Other registers mentioned in this section reside on the controller and interact with the 2717 module. Refer to *Chapter 3, Network Protocols*, for information on network protocols and the *Register Reference Guide* for details on the controller registers.

Network and Communications Registers

These registers are used to set node numbers, MAC addresses, IP addresses, and information on subnets and gateways.

20000 - CNet/Ethernet Device Node Number Register - R/W

Store a CNet device node number between 1 to 255 to this register. Ethernet node numbers range from 1 to 32767. All node numbers must be unique.

20005 - MAC Address Register - Upper Four Bytes - R/W

Store the upper four bytes of your CTC MAC address in this register. The last two bytes are significant and are stored as long integers.

20006 - MAC Address Register - Lower Four Bytes - R/W

Store the lower four bytes of your CTC MAC address in this register. These bytes are stored as long integers.

20010, 20014 - Serial Port Baud Rate - R/W

Registers 20010 and 20014 set the baud rates for the 2717's serial ports. 20010 sets the baud rate for the first port and 20014 sets the baud rate for the second port. Available baud rates are 300, 600, 1200, 2400, 4800, 9600, 19200, and 38400 BPS. Storing a value to these registers allows you to change the baud rate of the selected port.

20011, 20015 - Serial Port Data Length - R/W

Not supported — 8 bit data length only.

20012, 20016 - Serial Port Parity - R/W

Not supported — no parity only (NONE).

20048-20051 - IP Address Registers - R/W

These four registers store the IP Address in octets where 20048 is the first octet and 20051 is the last octet (least significant). For example, if 12.40.53.219 is the IP address, then 12 is stored in register 20048 and 219 is stored in register 20051.

20064-20067 - Subnet Mask Registers - R/W

These four registers store the subnet mask in octets where 20064 is the first octet and 20067 is the last octet (least significant). For example, if 255.255.255.0 is the IP address, then 255 is stored in register 20064 and 0 is stored in register 20067.

20080-20083 - Gateway Address Registers - R/W

These four registers store the gateway address in octets where 20080 is the first octet and 20083 is the last octet (least significant). For example, if 12.40.53.204 is the gateway address, then 12 is stored in register 20080 and 204 is stored in register 20083.

A subnet mask value of 0.0.0.0 disables the gateway.

20096 - Serial E2PROM Update Register - R/W

Write a 1 to this register store IP information in nonvolatile E2PROM. Refer to *Modifying IP Information* in *Chapter 2* for more information.

20102 - On-board Millisecond Timer - Read-only

Register 20102 is automatically incremented by one after every millisecond regardless of program operation. It is typically used to time functions on a machine or to keep track of system up-time. This register continuously counts up to the maximum 32-bit value (2,147,483,647) and then rolls over to a negative number (-2,147,483,648).

Serial Port Redirection TCP Protocol (Server) - Register 20120 - R/W

This register stores the number of active TCP connections slated to receive data that is sent through serial port channel 0xFF. Channel selection is made with register 12000 and writing information from the data table is achieved with register 12001. Before transmitting any data, you must check the status of register 12000 to determine whether the controller is idle (status = 0) or is busy transmitting a message (status = 1).

You can configure the 2717 to constantly listen for connections that allow a controller to send information to that connection as though it was a serial port. A TCP server is run at the port defined within the 2717.ini file by the parameter SERIALREDIRECTION_SERVER_PORT.

Peer-to-Peer Protocol Registers



Note

The 2717 can only perform peer-to-peer operations with other 2717 modules. It is not compatible with the 2217 communications module.

The 2717's peer-to-peer registers let it communicate directly with other 2717 modules without requiring a dedicated server. It can also gather register information locally for different network protocols.

Registers 21000-21999 are read/write registers that are reserved for peer-to-peer networks. Each block of 10 sequential registers is assigned to a designated peer node and defines the peer environment for that connection. You can retrieve data from and automatically update up to 100 sequential registers with a single request. This register block is used for many functions by different network protocols (Peer-to-Peer, Modbus, etc.) that all interface with the registers in the same manner. Registers 21000-21999 are defined below.

21XX0 - First Octet IP Address Register (Most Significant) - R/W

This is the first octet of the IP address (XXX.000.000.000) that is used to make peer requests.

21XX1 - Second Octet IP Address Register - R/W

This is the second octet of the IP address (000.XXX.000.000) that is used to make peer requests.

21XX2 - Third Octet IP Address Register - R/W

This is the third octet of the IP address (000.000.XXX.000) that is used to make peer requests.

21XX3 - Fourth Octet IP Address Register (Least Significant) - R/W

This is the fourth octet of the IP address (000.000.000.XXX) that is used to make peer requests.



Note

Once a peer connection is attempted, you cannot change the IP octet register settings.

21XX4 - Start Register - R/W

This register stores the starting register address in the controller for peer-to-peer communications. You can change this register number after a peer connection is attempted, but the number of sequential registers must stay the same (see *Register 21XX5* for more information).

21XX5 - Sequential Number Register - R/W

This register stores the number of sequential registers (starting with Register 21XX4) you want to read during a peer-to-peer session. The value 1 represents a single register and the maximum number of registers allowed is 100. Configure this register before setting up any other registers. Do not change this value during a peer-to-peer transaction or all data will be lost and new values will have to be entered. If you modify this register, it lets you reset the peer connection.

21XX6 - Poll Timer Register - R/W

Set this register to 0 for a single read request. Specify a value (in units of ms/count) if this register is going to receive periodic updates from the server controller (the controller sending information to the register). The minimum value allowed is 26 ms. For example, the value 500 would refresh the data registers with new peer data every 1/2 second.

You can write to this register at any time. Writing a 0 to this register while actively conducting a peer-to-peer session cancels the periodic update and causes a new single read transaction to occur. A time-out (Status Flag Register 21XX7 = 0) occurs if the server has not refreshed peer data in a time equal to 2-1/2 multiplied by the poll timer value. You can access this register at any time once you have initialized the Sequential Number Register (Register 21XX5).



Note

Data registers are mentioned in numerous places throughout the listings below. These registers are represented by Register 21XX9, which is a phantom register. For more information, refer to the 21XX9 listing in this section.

21XX7 - Status Flag Register - Read-Only

This register reflects the current status of the data registers. Its value is based on any requested operations. Typically, you initiate an operation and then wait for a status of 1. Possible values are:

- 0 - Offline; no connection is present.
- 1 - Last request is successful and completed. Data is available in the data registers if requested.
- -1 - Requested operation has failed.
- -2 - Busy; connecting to the desired host.
- -3 - Busy; reading data.
- -4 - Busy; writing data.

- -10 - Aborted operation; out of local memory or resources.

21XX8 - Index Offset Register - R/W

This register lets you access each of the requested sequential data registers. It works in conjunction with Register 21XX9 and acts as its pointer. You can store the number of a general or special purpose register in 21XX8 and 21XX9 can then access the resource contained in the pointer. The first register (with an index of 0) is the Start Register (Register 21XX4). 1 is the next register, and so forth. Once Register 21XX5 (the Sequential Number Register) is initialized, you can change this register's value at any time. For more information on how pointer registers function, refer to the *Register Reference Guide*.

The index register also has a few special features when you set it to 1000 or above. Modifications are made by writing to the data register and setting the index register appropriately as described below:

- **1000 - Peer Request Time-Out Register** - The timer starts when a peer node request is initiated and stops (times out) if no response is received within the time specified by this register. Retrys only occur if automatic updates are active (Register 21XX6 is set to a value other than 0). Defaults are 500 ms for single register reads and time-out value*2.5 for automatically updated register read transactions.
- **1001 - Peer Request Failed Index Register** - This register indicates when a peer transaction fails and an error occurs. The Status Flag Register (21XX7) is set to a value other than 1. Any data that was read or written when the error occurred has an offset value that is stored in 1001. If you read the data register, it returns the offset failure value. Data written before this offset value is valid. For example, if your process continuously updates 50 registers and the register returns a value of 25, it means the process failed while trying to write the 25th element of data. All data written before this element was written correctly.
- **1002 - Peer Request Retry Counter Index Register** - This debugging register points the data register to the retry counter. Quickstep can set this register to any value. The register is incremented by 1 when a time-out occurs because of waiting for data from a peer node.
- **1003 - Peer Request Protocol Index Register** - This register tells the data register what protocol to use for setting the peer block registers. You must set this register before setting the Start Register (21XX4). Default mode is 0 for CTC Peer-to-Peer protocol. 2 is used for ModBus TCP Client mode.
- **1004 - Peer Request TCP Client Port Index Register** - This register points the data register to the destination TCP Port address for your connection. You must set this register before setting the Start Register (21XX4). 1004 is currently used for ModBus TCP Client mode with a default port number of 502 (the industry standard).
- **1005 - Peer Request ModBus Client Unit ID Index Register** - This register points the data register to the Unit ID field value used in the Modbus Client request packet. The default ID is 00 but you can set it to any desired value. This ID affects all subsequent transmissions and allows multiplexed nodes to be addressed in a ModBus environment.

- **1006 - Peer Request ModBus Client Exception Index Register** - This register tells the data register where the last Modbus Exception error code is stored from a previously received message. Referencing this register helps to interpret failure types.
- **1999 - Peer Request Initiate Write Block Index Register** - This register writes a block of registers (beginning with the Start Register) to a destination at the other end of the connection. The number of registers written is defined by Register 21XX5. To write to a single register, set the index to the desired offset and then write to the data register. Do not attempt to access the data register during a block write operation.
- **2000 - 2099 Peer Request Write Block Index Registers** - This register points the data register to a temporary storage array and makes 2000 equivalent to an index offset of 0. Because data is written locally, you can write multiple values before initiating the write sequence by setting this register to 1999. Automatic polled updates cannot overwrite data values until the register is set to a non-2XXX value.

21XX9 - Data Registers/Peer Request Time-Out Register - R/W

This phantom register contains peer data that is read or written in a peer transaction. It is a “window” into a register array in the controller. The array size is set by Register 21XX5 and the offset is specified by Register 21XX8. Data integrity is indicated in Register 21XX7. For more information on how phantom registers function, refer to the *Register Reference Guide*.



Note

For more information on peer-to-peer networking, refer to *Technical Note No. 32, Setting Up Peer-to-peer Connections Between Two Controllers*.

Glossary

A

Access Control Methods, such as login passwords and time and computer restrictions, for controlling user access to network resources.

Access Control List Database that describes the type of access each user has to a service.

Access Permissions A rule associated with an object (usually a directory, file, or printer) to regulate which users can have access to the object and in what manner.

Application Layer Layer 7, the highest layer of the OSI Reference Model, defines the way applications interact with the network. Implemented by various network applications, including electronic mail, file transfer, and terminal emulation.

Authentication Validation of a user's login information. When a user logs onto an account on a computer running Windows NT Workstation, the authentication is performed by that workstation. When a user logs on to an account on a Windows NT Server domain, authentication may be performed by any server of that domain.

B

Backbone A LAN or WAN that interconnects intermediate systems (bridges, switches, and/or routers).

Bastion Host A machine placed on the perimeter network to provide publicly available services. Although secured against attack, it is assumed to be compromised because it is exposed to the Internet.

Binary A base-2 number system, in which values are expressed as combinations of two digits, 0 and 1.

Broadcast Message A network message sent from a single computer that is distributed to all other

devices on the same segment of the network as the sending computer.

C

Caching In DNS name resolution, caching refers to a local cache where information about the DNS domain name space is kept. Whenever a resolver request arrives, the local name server checks both its static information and the cache for the name to IP address mapping.

Circuit-Level Gateway A specialized function that relays TCP connections without performing any additional packet processing or filtering.

Classless Interdomain Routing Routing technique that allows routers to group routes together in order to cut down on the quantity of routing information carried by the core routers.

Client/Server A distributed system model of computing that brings computing power to the desktop, where users access resources from servers.

CRC Acronym for Cyclical Redundancy Checking. An error checking technique that ensures the accuracy of transmitted digital data. Transmitted messages are divided into predetermined lengths which, used as dividends, are divided by a fixed divisor. The remainder of the calculation is appended onto and sent with the message. At the receiving end, the computer recalculates the remainder. If it does not match the transmitted remainder, an error is detected.

CSMA/CD Acronym for Carrier Sense Multiple Access/Collision Detection. The LAN access method used in Ethernet. When a device wants to gain access to a network, it checks to see if the network is free. If not, it waits a random amount of time before retrying. If the network is free and two devices access the line simultaneously, their signals collide. When a collision is detected, they both back off and wait a random amount of time before retrying.

D

Datagram A packet of data and other delivery information that is routed through a packet-switched network or transmitted on a local area network.

Data Link Layer Layer 2 of the OSI Reference Model. Defines the rules for sending and receiving data across the physical connection between two systems.

Decryption

The restoration of data to its original form after it has been encrypted by applying a specific algorithm that altered the data's appearance in order to prevent other devices from reading the information. Decryption is accomplished by applying the encrypting algorithm in reverse.

DES Acronym for Data Encryption Standard, a type of encryption (the U.S. government standard) designed to protect against password discovery and playback. Microsoft RAS uses DES encryption when both the client and the server are using RAS.

DHCP Acronym for Dynamic Host Configuration Protocol, which offers dynamic configuration of IP addresses and related information. DHCP provides safe, reliable, and simple TCP/IP network configuration, prevents address conflicts, and helps conserve the use of IP addresses through centralized management of address allocation.

E

Encryption The process of making information indecipherable to protect it from unauthorized viewing or use, especially during transmission or when it is stored on a transportable magnetic medium.

Ethernet IEEE standard network protocol that specifies how data is placed on and retrieved from a common transmission medium. Has a transfer rate of 10 Mbps. Forms the underlying transport vehicle used by several upper-level protocols, including TCP/IP and XNS.

F

File Transfer Protocol (FTP) A service supporting file transfers between local and remote systems that support this protocol. FTP supports several commands that allow bidirectional transfer of binary and ASCII files between systems. The FTP Server service is part of the Internet Information Server (IIS). The FTP client is installed with TCP/IP connectivity utilities.

Firewall A system or group of systems that enforces an access control policy between an organization's network and the Internet for purposes of security.

H

Host Any device that is attached to the network and uses TCP/IP.

Host ID The portion of the IP address that identifies a computer within a particular network ID.

Host Name The name of a device on a network. For a device on a Windows or Windows NT network, this can be the same as the computer name, but it may not be. The host name must be in the host table or be known by a DNS server for that host to be found by another computer attempting to communicate with it.

HTML Acronym for Hypertext Markup Language, a simple markup language used to create hypertext documents that are portable from one platform to another. HTML files are simple ASCII text files with codes embedded (indicated by markup tags) to indicate formatting and hypertext links. HTML is used for formatting documents on the World Wide Web.

HTTP Acronym for HyperText Transfer Protocol, which is the protocol used to transmit and receive all data over the World Wide Web. When you type a URL into your browser, you're actually sending an HTTP request to a Web server for a page of information (that's why URLs all begin with "http://").

I

Internet Control Message Protocol (ICMP)

A maintenance protocol in the TCP/IP suite, required in every TCP/IP implementation, that allows two nodes on an IP network to share IP status and error information. ICMP is used by the ping utility to determine the readability of a remote system.

Internet Protocol (IP) The messenger protocol of TCP/IP, responsible for addressing and sending TCP packets over the network. IP provides a best-effort, connectionless delivery system that does not guarantee that packets arrive at their destination or that they are received in the sequence in which they were sent.

IP Address Used to identify a node on a network and to specify routing information. Each node on the network must be assigned a unique IP address, which is made up of the network ID, plus a unique host ID assigned by the network administrator. This address is typically represented in dotted-decimal notation, with the decimal value of each octet separated by a period (for example, 138.57.7.27).

IP Spoofing The use of a forged IP source address to circumvent a firewall. The packet appears to have come from inside the protected network and to be eligible for forwarding into the network.

ISP Acronym for internet service provider, a company or educational institution that enables remote users to access the Internet by providing dial-up connections or installing leased lines.

L

LAN Acronym for Local Area Network. A communications network that serves users within a confined geographical area. It is made up of servers, workstations, a network operating system, and a communications link.

LRC Acronym for Longitudinal Redundancy Check. An error checking method that generates a parity bit from a specified string of bits on a longitudinal track. In a row and column format, such as on magnetic tape, LRC is often used with VRC (Vertical Redundancy Check, which creates a parity bit for each character.

M

Multihomed Computer A system that has multiple network cards, or that has been configured with multiple IP addresses for a single network interface card.

N

Network Interface Card A board installed in a computer system, usually a PC, to provide network communication capabilities to and from that computer system. Also called an adapter.

Network Layer Layer 3 of the OSI Reference Model. Defines protocols for routing data by opening and maintaining a path on the network between systems to ensure that data arrives at the correct destination node.

Network Sniffer A hardware and software diagnostic tool that can also be used to decipher passwords, which may result in unauthorized access to network accounts. Clear text passwords are susceptible to network sniffers.

Node In the PC environment, a node is any device that is attached to the network and uses TCP/IP. A node can also be referred to as a host.

O

OSI Acronym for Open Systems Interconnection model. TCP/IP protocols map to a five-layered conceptual model consisting of Application, Transport, Internet, Data Link, and Physical Interface. Each layer in this TCP/IP model corresponds to one or more lay-

ers of the International Standards Organization (ISO) seven-layer OSI model consisting of Application, Presentation, Session, Transport, Network, Data Link, and Physical.

P

Packet A transmission unit of fixed maximum size that consists of binary information representing both data and a header containing an ID number, source and destination addresses, and error-control data.

Packet Header The part of a packet that contains an identification number, source and destination addresses, and—sometimes—error-control data.

Physical Layer Layer 1 of the OSI Reference Model that governs hardware connections and byte-stream encoding for transmission. It is the only layer that involves a physical transfer of information between network nodes.

Point-to-Point Protocol (PPP) A set of industry-standard framing and authentication protocols that is part of Windows NT RAS to ensure interoperability with third-party remote access software. PPP negotiates configuration parameters for multiple layers of the OSI model.

Point-to-Point Tunneling Protocol (PPTP) PPTP is a new networking technology that supports multiprotocol virtual private networks (VPNs), enabling remote users to access corporate networks securely across the Internet by dialing into an internet service provider (ISP) or by connecting directly to the Internet.

Presentation Layer Layer 6 of the OSI Reference Model; includes protocols that are part of the operating system, and defines how information is formatted for display or printing and how data is encrypted, and translation of other character sets.

Protocol A set of rules and conventions for sending information over a network. These rules govern the content, format, timing, sequencing, and error control of messages exchanged among network devices.

Protocol Stack The implementation of a specific protocol family in a computer or other node on the network.

Proxy Server A computer that listens to name query broadcasts and responds for those names not on the local subnet. The proxy communicates with the name server to resolve names and then caches them for a time period.

R

Remote Access Service (RAS) A service that provides remote networking for telecommuters, mobile workers, and system administrators who monitor and manage servers at multiple branch offices. Users with RAS on a Windows NT computer can dial in to remotely access their networks for services such as file and printer sharing, electronic mail, scheduling, and SQL database access.

S

Serial Line Internet Protocol (SLIP) An older industry standard that is part of Windows NT RAS to ensure interoperability with third-party remote access software.

Session Layer Layer 5 of the OSI Reference Model coordinates communication between systems, maintaining sessions for as long as needed and performing security, logging, and administrative functions.

Simple Network Management Protocol (SNMP) A protocol used by SNMP consoles and agents to communicate. In Windows NT, the SNMP service is used to get and set status information about a host on a TCP/IP network.

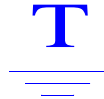
Single User Logon Windows NT network users can connect to multiple servers, domains, and applications with a single network logon.

Simple Mail Transfer Protocol (SMTP) A member of the TCP/IP suite of protocols that governs the exchange of electronic mail between message transfer agents.

SQL Acronym for structured query language, a database programming language used for accessing, querying, and otherwise managing information in a relational database system.

Subnet A portion of a network, which may be a physically independent network segment, which shares a network address with other portions of the network and is distinguished by a subnet number. A subnet is to a network what a network is to an internet.

Subnet Mask A 32-bit value that allows the recipient of IP packets to distinguish the network ID portion of the IP address from the host ID.



Telnet Standard application, supported by almost every TCP/IP implementation, that allows users (clients) to log on to many different hosts (servers) from a single virtual terminal running at their desktop. Telnet servers list on well-known TCP port 23, and Telnet clients use TCP port numbers greater than 1023. The connection is always initiated in the client-to-server direction.

Transmission Control Protocol (TCP)

A connection-based Internet protocol responsible for breaking data into packets, which the IP protocol sends over the network. This protocol provides a reliable, sequenced communication stream for network communication.

Transmission Control Protocol/Internet Protocol (TCP/IP)

A set of networking protocols that provide communications across interconnected networks made up of computers with diverse hardware architectures and various operating systems. TCP/IP includes standards for how computers communicate and conventions for connecting networks and routing traffic.

Transport Layer Layer 4 of the OSI Reference Model which controls the movement of data between systems, defines protocols for structuring messages, and supervises the validity of transmissions by performing error checking.



User Datagram Protocol (UDP) A TCP complement that offers a connectionless datagram service that guarantees neither delivery nor correct sequencing of delivered packets (much like IP).



VPN Acronym for virtual private network, a remote LAN that can be accessed through the Internet using the new PPTP. This connection has the appearance and many of the advantages of a dedicated link but occurs over a shared network. Using a technique called “tunneling,” data packets are transmitted across a public routed network in a private “tunnel” that simulates a point-to-point connection and allows network protocols to traverse incompatible infrastructures.



World Wide Web (WWW) The software, protocols, conventions, and information that enable hypertext and multimedia publishing of resources on different computers around the world.

This page is intentionally left blank.

Bibliography

- Ascend Communications. (1997). *Virtual Private Networks Resource Guide: A Resource Guide for Senior Telecommunications Engineers and Department Managers of Network Services Worldwide*.
- Check Point Software Technologies. (1998). *Virtual Private Network Security Components*.
- Freedman, A. (1999). *The Computer Desktop Encyclopedia* (2nd ed.). New York: Amacom.
- EZine Publications. (1998). *How Does IP Subnetting Work?* <http://www.ezine.com>.
- Hansen, B. (1999). *The Dictionary of Computing and Digital Media*. Wilsonville, Oregon: ABF Content.
- Sheldon, T. (1998). *Encyclopedia of Networking* (Electronic Ed.). Berkeley, California: Osborne/McGraw Hill.

This page is intentionally left blank.

Index

Numerics

10/100Base-T, description, 36

A

addresses
 constructing, binary notation, 46
 TCP/IP, 44

C

cables and connectors
 communication, 16
classes
 network, TCP/IP, 45
client, definition, 52
client/server
 computing, 52–54
 model
 benefits and drawbacks, 53–54
 definition, 52
connections
 computer-controller, setting, 15
 DB9 and DB25, 16
 Ethernet, 17
 RS-485, 17
connectors
 10/100-Base-T pinout diagram, 9
 RS-232 pinout diagram, 9
 RS-485 pinout diagram, 9
controllers, CTC
 intranet setup, 55
 network capabilities, 55

D

D connectors, configuration, 16
datagrams, 50–51
DB9 and DB 25 connections, 16
DHCP
 description, TCP/IP, 43
dip switch settings, 31

E

ESD devices
 board handling precautions, 12
Ethernet
 10/100Base-T, 36
 connections, 17

F

features, 2717 module, 7
flash memory
 reformatting procedure, 23
 updating, 23
FTP
 description, TCP/IP, 43
 overview, 38
 supported commands, 21
 updating flash memory, 23

H

hardware/firmware
 revision levels, 11
host operation
 RS-232 and RS-485 ports, 14
HTTP
 description, TCP/IP, 43

I

ICMP
 description, TCP/IP, 43
initialization file
 2717.ini, 24–29
 sample, 27
installation procedures
 module, 13
internet applications
 client/server model, 52
intranet, setting up with CTC controllers, 55–57

IP

- addresses
 - network classes, 45
 - options, 26
- description, TCP/IP, 43
- information
 - modifying, 30

L

- L2TP, description, 61
- layers, TCP/IP, 42

M

- masks
 - custom
 - CIDR rules, 49
 - classic rules, 48
 - subnets, 46
- ModBus protocol, 39–40

N

- network
 - classes, 45–46
 - layer
 - OSI reference model, 41
 - specifications
 - Ethernet, 37
- Network group, parameters, 24–25
- non-routable protocols, 35

O

- OSI reference model
 - comparison to TCP/IP stack, 42

P

- packet
 - structure, UDP, 50
 - transmission, 41
- port addressing, 14
- PPTP, description, 60
- precautions
 - board handling, 12
- protocols
 - CTNet binary, 35
 - DHCP, 43
 - Ethernet, 36–37
 - FTP, 43

- HTTP, 43
- ICMP, 43
- IP, 43
- ModBus, 39
- peer-to-peer, 36
- routable, 41
- SMTP, 43
- supported by 2717, 35
- TCP/IP, 41, 43
- UDP, 43, 50

R

- RAS, description, 60
- registers
 - network and communications, 65–66
 - peer-to-peer protocol, 67–70
 - serial redirection, 67
 - special purpose, 65–70
- revision levels, hardware/firmware, 11
- routable protocols, 41
- routers, 41
- RS-232 connections, 15
- RS-485 connections, 17

S

- Security group, parameters, 25–26
- server, definition, 52
- services
 - TCP/IP, 41
 - UDP, 51
- SMTP
 - description, TCP/IP, 43
- specifications
 - general, 10
 - performance, 10
- subnet masks
 - custom
 - CIDR, 49
 - classic, 48
 - description, 46
 - network classes, default settings, 47
- subnets, TCP/IP, 46–50
- switches, dip
 - setting, 31
- system
 - description, 8
 - features, 7

T

TCP/IP

- addressing, 44
- client/server computing, 42
- comparison to OSI Stack, 42
- layers, 42
- network classes, 45
- protocol, 41–50
- services, 41

U

UDP

- description, TCP/IP, 43
- overview, 50–51

V

VPNs

- advantages, 58
- L2TP, 61
- overview, 58–61
- PPTP, 60
- RAS, 60

This page is intentionally left blank.