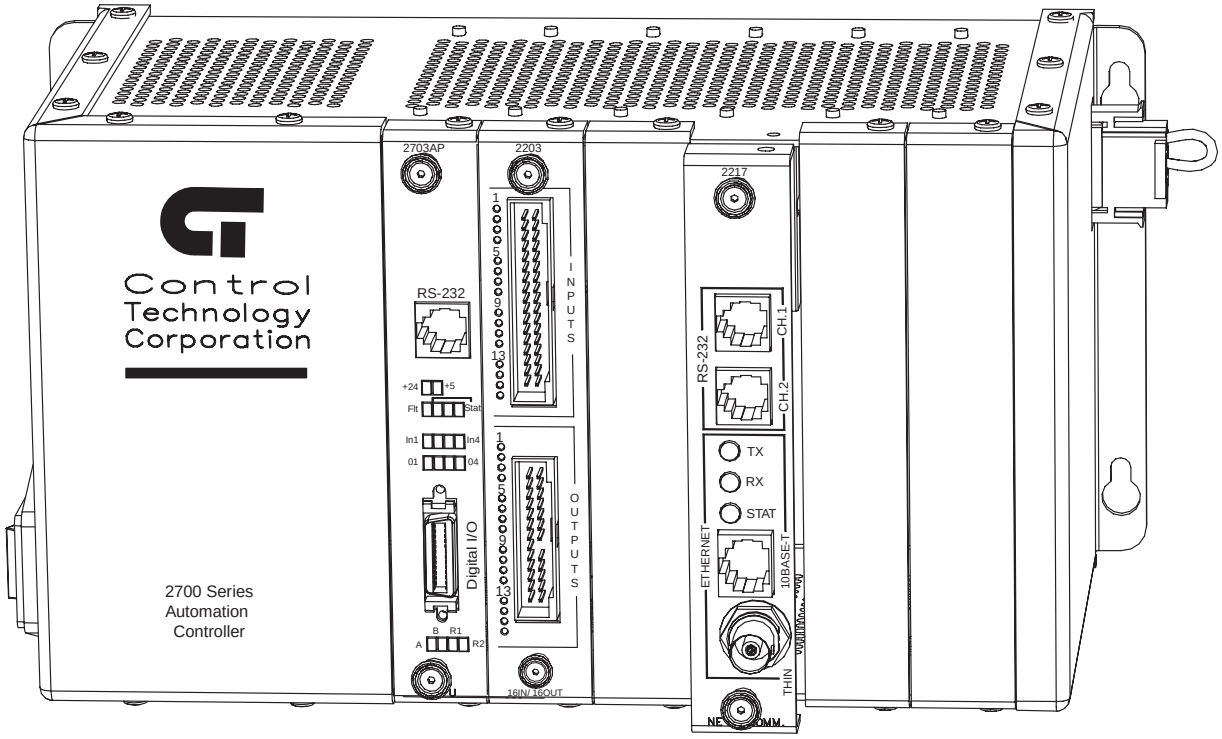
Model 2217 EtherNetworking Module Installation Guide



Doc. No. 2217IG
Revision B
January 2002

The following are trademarks of Control Technology Corporation:

- Quickstep
- CTC Monitor
- CTC Utilities

Windows is a trademark of Microsoft Corporation.

Contents

Notes to Readers

The *Model 2217 Installation Guide* provides the following information:

- **System Overview** — describes the 2217's basic features.
- **Description** — an overview of the 2217's basic functions.
- **Connectors and Pinout Diagrams** — wiring diagrams for all connectors.
- **Specifications** — general and performance specifications.
- **Hardware/Firmware Revision Levels** — lists the hardware and firmware revisions for the 2217 module and various controllers.
- **Board Handling Precautions** — contains general guidelines on handling printed circuit boards with ESD devices.
- **Installation** — describes how to install the 2217 module in a CTC controller.
- **Port Addressing and Computer – Controller Communications** — describes how to configure a port; provides a general overview of RS-232 communications.
- **Ethernet Connections** — describes 10Base-2 and 10Base-T protocols.
- **Peer-to-Peer Protocol** — describes peer-to-peer networking and provides a sample Quickstep program.
- **Special Purpose Registers** — how to use the 2217's on-board registers in special applications; setting serial port parameters.

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.
Italics	Indicates a word requiring an appropriate substitution. For example, replace *filename* 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

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.

System Overview

The Model 2217 EtherNetworking Module has useful features that are essential in any industrial control application. The 2217 supports several 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:

- **Ethernet connectivity** - The module supports the IEEE 802.3 protocols, 10Base-2 and 10Base-T at data rates up to 10 Mbps. You can support existing hardware investments and enable future expansion to faster networks.

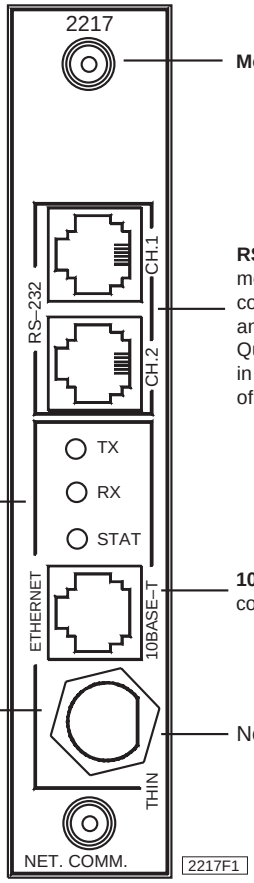
- **Peer-to-Peer networking** - The 2217 acts as a conduit for peer-to-peer communications between CTC controllers. Peer-to-peer communications also offers indirect node access, built-in controller security, and built-in error checking. In addition, register information can be transferred in blocks through the use of arrays for Quickstep program access.

- **Host communications** – The module supports global and group broadcasts, individual node interrogation (read/write), application program transfers (download/upload), and access to all controller resources.

- **RS-232 ports** - The module has two RS-232 ports for serial communications.

- **Compatible with 2600/2700/2700AP series controllers** - The 2217 runs in existing controllers and will work with future generations of CTC controllers.

2217 Description



Module Extraction Handle (2 places) - Do not detach!

RS-232 Connectors - Channels 1 and 2 use the same modular jack as the controller's on-board port. They are compatible with CTC's 2881-2883 communications cables and also work with the cable that is supplied with your Quickstep software. Refer to the RS-232 pinout diagram in the Connectors and Pinout Diagrams section of this guide.

Status LEDs - These LEDs indicate the controller's communications status. The RX LED is ON when the 2217 receives a message and the TX LED is OFF when the 2217 transmits a message. Under normal conditions, the STAT LED flashes once on system startup and then remains OFF. If an error occurs, it flashes continuously.

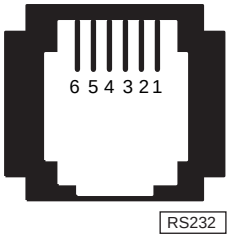
10/Base-T Connector - Provides Ethernet communications through a 10Base-T connector.

10Base-2 Connector - Provides Ethernet communications through a 10Base-2 connector. Use Model 2297 Ethernet cables, Model 2299 Ethernet terminator connectors, and Model 2298 Ethernet tee connectors to set up communications.

Note: The 2217T does not have this connector.

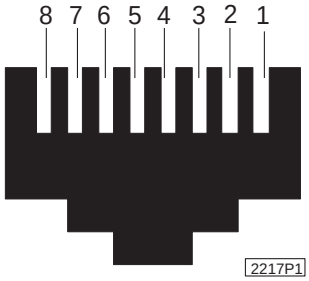
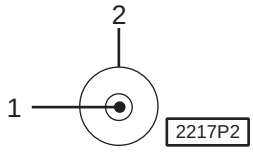
Connectors and Pinout Diagrams

Table 1. RS-232 Connector

	Pin #	Signal
	1	Not Used (+5 VDC) [1]
	2	TxD Outbound
	3	Common
	4	Common
	5	RxD Outbound
	6	Common (+5 VDC Return) [1]

1. The signals in parentheses apply to Channel 1 only.

Table 2. Ethernet Connectors

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

1. The Model 2217T does not have this connector.

Specifications

Table 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
Ethernet Transceivers (10 Megabits/sec) [2]			1.5	VAC PP
CTNET Performance Specifications [1]				
Host Communications				
Single register transaction from 2217		1-2		ms
Single register transaction from 2703AP		3-5		ms
16-register read from 2703AP		6-7		ms
50-register read from 2703AP		8-9		ms
Peer-to-Peer Communications				
Single register transaction from 2217		9-11		ms
Single register transaction from 2703AP		10-13		ms
Power Supply Requirements (from controller)				
Logic Supply (5 V)		630.0	750.0	mA
Auxiliary Supply (24 V)		53.0	170.0	mA

1. This value is derived with high communications priority active or when one task is running.
2. This conforms to IEEE Standard 802.3.
3. The specifications listed above are at 25°C, unless otherwise specified.

Table 4. Hardware / Firmware Revision Levels

Model Numbers	Hardware Revision Level	Firmware Revision Level [1] [2]
2217	B	3.72
2700	Rev. C or greater	2.10 or greater
2600	Rev. 0 or greater	1.00 or greater
2700AP	A or greater	N/A
2703AP	A or greater	3.05

1. 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.
2. 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.

Installing the 2217 Module

The module fits into one of the slots of your automation controller (Figure 1). You can insert any combination of modules into the controller (subject to system limits) and can install them in any order. This is possible because the controller's CPU dynamically assigns such items as motor numbers, input numbers, and output numbers each time power is re-applied to the controller. These numbers are assigned from left-to-right across the controller.

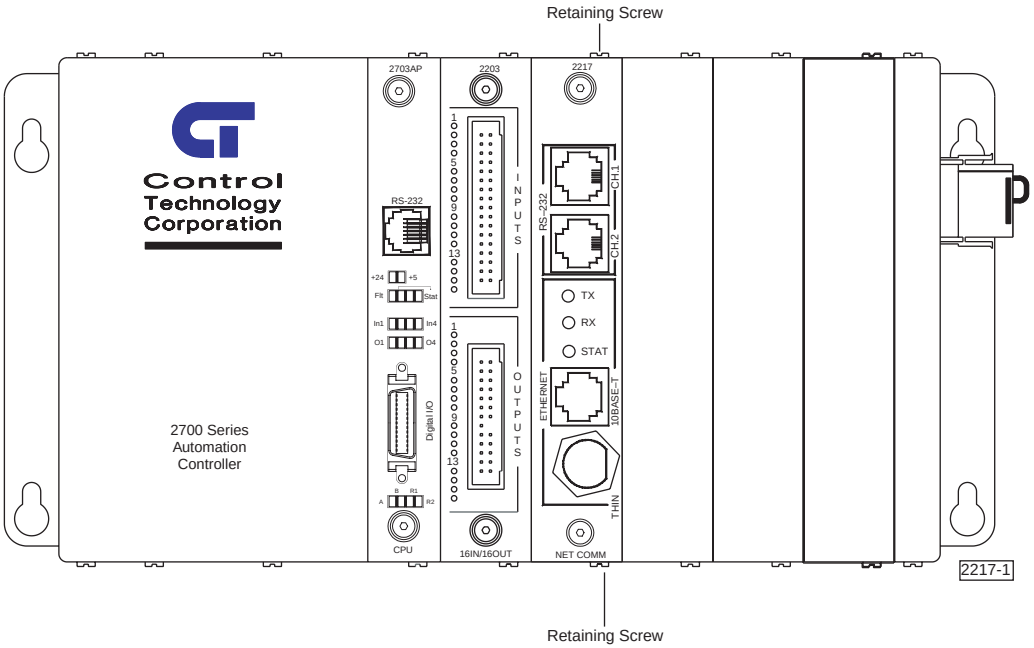
To install a module into the automation controller:

Note

Retain all hardware removed during this procedure.

1. Remove all AC and DC power, including any external supplies connected to the controller.
2. Locate an unused slot and remove two retaining screws from the top and bottom of its cover plate.
3. Slide the module into the slot and make sure that the circuit board slides into the nylon guides at the top and bottom of the controller case. Make sure that the card is oriented properly so that its labels are right-side-up.
4. Press the module firmly into the controller. Make sure that the module's faceplate is flush with the adjacent sheet metal surface.
5. Re-install two retaining screws in the top and bottom of the new module.

Figure 1. 2700AP Series Controller with the 2217 module installed in the right-hand slot



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 (ASCII and Binary) on any port in the controller from a host computer or intelligent host terminal.

Configuring the RS-232 Ports

Both RS-232 ports may be configured to act as a host using currently supported message transmitting and receiving conventions. To select a port for host communications, you must store the port number to register 12000 before accessing any of the communications registers. The 2703AP's on-board RS232 ports are Ports 0 and 1. The ports in the controller's rack begin with 2 and 3. The example below configures one of the ports.

```
[1] Store_Port_Number, Transmit _Row_10, Test_Port
   ;;; Select the second communications port in the controller.
   ;;; Send Row 10 of the data table out of the communications port.
   ;;; Test to see if port is busy.
  ------------------------------------------------------------------------
   <NO CHANGE IN DIGITAL OUTPUTS>
  ------------------------------------------------------------------------
   store 2 to Reg_12000
   store 10 to Reg_12001
   if Reg_12000 = 0 goto Next
```

Notes

1. Register 12000 has different meanings when reading and writing to it.

2. For more information on registers, refer to the *Quick Register Reference Guide* in the Customer Support area of our web site at www.ctc-control.com

Computer - Controller Communications

The 2217's RS-232 ports provide a way to download Quickstep programs and also support data communications.

RS-232 Communications

The 2217's RS-232 ports allow the following activities:

- **Direct communications between a PC and the 2217'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 2217 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

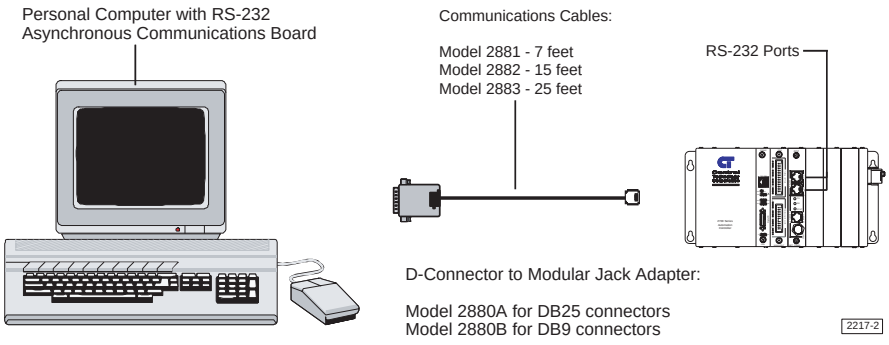
For more information on data communications and the DLL functions required to communicate with the 2217, refer to the *CTC 32-Bit Data Communications Functions Reference Guide,* which is available in the Customer Support area of our web site at www.ctc-control.com.

RS-232 Connections

Connect to the RS-232 ports through one of the modular jacks (labeled CH1 and CH2) on the 2217'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 on page 7 for details on how this jack is wired.

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

Figure 1. 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 2217 to a telephone line.

Figures 2 and 3 show computer-controller connections using an RS-232 connection and DB25 and DB9 connectors.

Figure 2. DB9 Connections

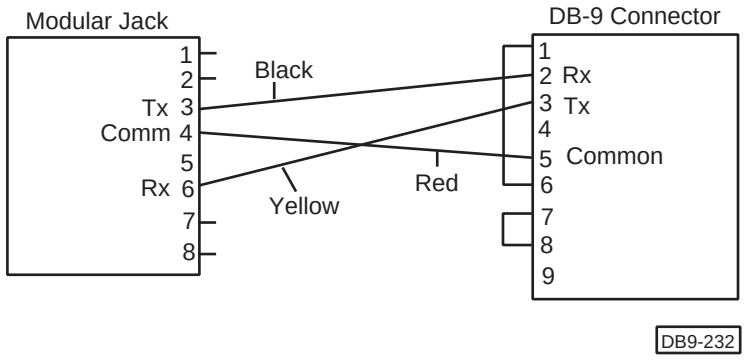
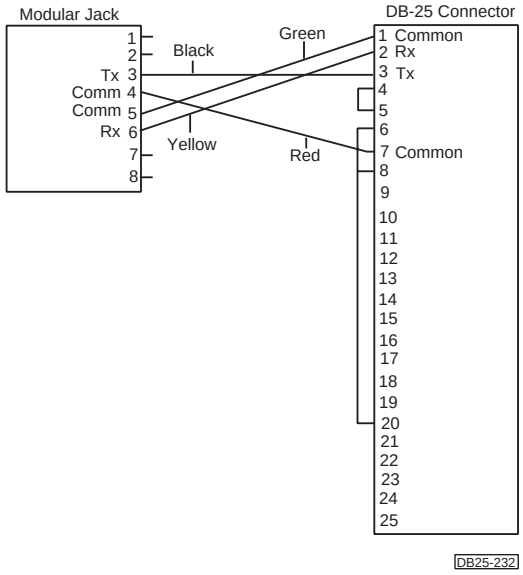


Figure 3. DB25 Connections



Ethernet Connections

The 2217 has 10Base-2 and 10Base-T connectors that conform to IEEE standard 802.3. This section discusses the Ethernet protocol and illustrates a typical network connection diagram. Wiring information for the Ethernet connector is listed in Table 2 and performance specifications are listed in Table 3.

Ethernet Protocol

Ethernet 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). The 2217 has an Ethernet port that allows it to communicate over an Ethernet network using a 10Base-T connections.

10Base-2 (Thin Ethernet)

Thin Ethernet (10Base-2), also called ThinNet and CheaperNet, uses a thin, inexpensive coax cable that is easy to connect but has a limitation of 607 feet per segment. ThinNet uses T-type BNC connectors and the transceivers are built into the adapter cards.

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, connectors, and wiring techniques.

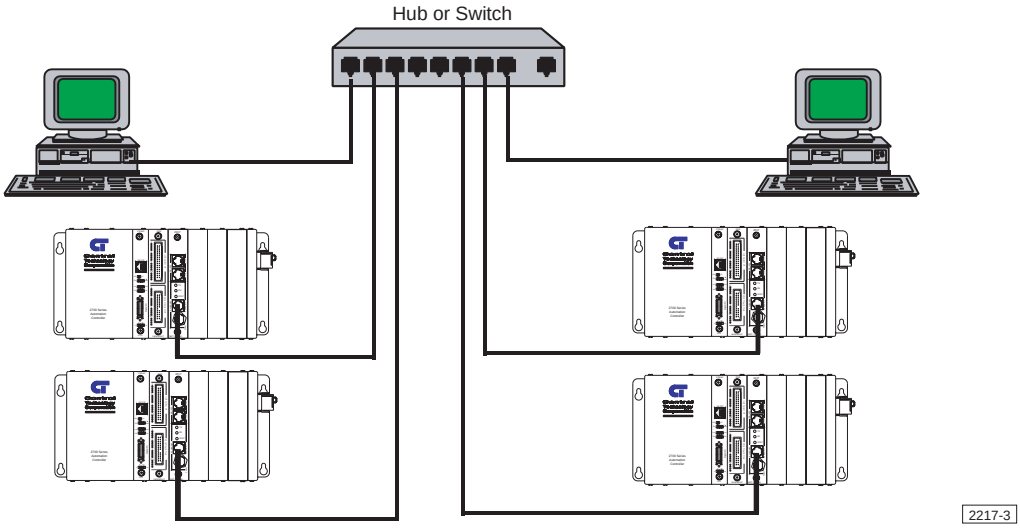
Network Specifications

Node and cable specifications for 10Base-T connections are listed below. Termination for 10Base-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)

Figure 4 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 4. 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.

Host Communications

The 2217's networking capabilities includes host communications, peer-to-peer communications offering indirect node access, built-in error checking, and network access from any controller's RS-232 port. A host computer can interrogate the area network continuously while local computers or operator interface terminals can access the network port using conventional communications protocols from any controller's RS-232 port. For fast data retrieval, the controller supports both block area transfer from a single command request both locally and over the network.

Peer-to-Peer Communications

This section has an example that shows peer-to-peer networking with a Quickstep program. It also lists some limits on using peer-to-peer networks in your application.

Peer-to-Peer Programs

When you design peer-to-peer programs, the following conditions apply:

1. All nodes, including PCs, must have unique node numbers in order to place these values in the peer-to-peer registers (21000-21999).

2. You must check the transaction status registers (24000-24999) before using the data in the peer access registers (23000-23999) or after sending data to another controller.

The following sample Quickstep program demonstrates peer-to-peer networking:

```
[1] Begin
    ------------------------------------------------------------------------------
    <NO CHANGE IN DIGITAL OUTPUTS>
    ------------------------------------------------------------------------------
    store 2 to reg_20000               /*Set my node
    store 3 to reg_21000               /*Set Node #3
    store 700 to reg_22000             /*Point to register #700 in node #3
    store 4 to reg_21001               /*Set Node #4
    store 701 to reg_22001             /*Point to register #701 in node #4
    goto next

[2] INIT
    ------------------------------------------------------------------------------
    <NO CHANGE IN DIGITAL OUTPUTS>
    ------------------------------------------------------------------------------
    store 0 to reg_10                  /*Initialize retry counter
    goto next

[3] GET_NODE_3
    ------------------------------------------------------------------------------
    <NO CHANGE IN DIGITAL OUTPUTS>
    ------------------------------------------------------------------------------
    store reg_23000 to reg_11          /*Read reg#700 from node#3
    if reg_24000 = 1 goto next         /*Is packet OK?
    store reg_10 + 1 to reg_10         /*Increment retry counter
    if reg_10 >= 200 goto NODE_DOWN    /*Retry 200 times
    delay 50 ms goto GET_NODE_3        /*Dwell between retries

[4]
    ------------------------------------------------------------------------------
    <NO CHANGE IN DIGITAL OUTPUTS>
    ------------------------------------------------------------------------------
    store 0 to reg_10                  /*Initialize retry counter
    goto next

[5] LOAD_NODE_4
    ------------------------------------------------------------------------------
    <NO CHANGE IN DIGITAL OUTPUTS>
    ------------------------------------------------------------------------------
    store reg_11 + 2000 to reg_12      /*do some math
    store reg_12 * 45 to reg_12        /*do some math
    store reg_12 to reg_23001          /*Load result into reg#701 of node#4
    if reg_24001 = 1 goto next         /*Verify that transaction took place
    store reg_10 + 1 to reg_10         /*Increment retry counter
    if reg_10 > 200 goto NODE_DOWN     /*Retry 200 times
    delay 50 ms goto LOAD_NODE_4       /*Dwell between retries

[6]
    ------------------------------------------------------------------------------
    <NO CHANGE IN DIGITAL OUTPUTS>
    ------------------------------------------------------------------------------
    delay 1 sec goto INIT              /*Dwell then go back for more
```

Caveats on using Peer-to-Peer Communications

Be aware of the following caveats of peer-to-peer networks:

1. A node may not always be operational. Retry a transaction with the appropriate action.

2. Limit the number of hosts on your network or nobody will be listening.

3. Do not bog down the network with unnecessary traffic.

2217 Registers

This section discusses the 2217's special registers and provides examples of how the registers are used.

Peer-to-Peer Registers

These registers are used to support peer-to-peer communications between multiple controllers on a network. Peer-to-peer communications requires the establishment of a lookup table of remote connection information. This information includes the network node number and register number for each data point. Several examples show how these registers interact.

Registers 21000 – 21999: Peer Node (R/W)

Registers 21000 - 21999 specify the network node number of the controller you want to communicate with.

Registers 22000 – 22999: Peer Resource (R/W)

Registers 22000 - 22999 specify the register address of the register resource of the controller you want to communicate with.

Registers 23000 – 23999: Peer Access (R/W)

Registers 23000 - 23999 cause the peer-to-peer transactions to execute. The transaction result is returned in the transaction status registers (24000 - 24999). Check the transaction status before using the result returned by the transaction.

Registers 24000 – 24999: Transaction Status (R/W)

Registers 24000 - 24999 return the status of the peer-to-peer transaction. A return value of 1 means the transaction was good; 0 means the transaction was bad.

Examples

The peer-to-peer registers are used together to send and receive messages between controllers. The following example sends the value 10 to register 501 in remote controller node 2.

```
store 2 to reg_21000
store 501 to reg_22000
store 10 to reg_23000
if reg_24000 = 1 goto Next
```

The following example requests the value of register 501 in controller node 2 and stores it in register 10 of the local controller.

```
store 2 to reg_21000
store 501 to reg_22000
store reg_23000 to reg_10
if reg_24000 = 1 goto Next
```

Other 2217 Registers

This section describes the 2217's other special registers.

Register 20000 - CTCNET/Ethernet Device Node Number (R/W)

Register 20000 specifies the CTCNET or Ethernet node number for the controller. Each controller must have a unique node number. Store a CTCNET device node number between 1 to 255 to this register. Ethernet node numbers range from 1 to 32767.

Register 20005 - Ethernet Address Register - Upper Four Bytes (R/W)

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

Register 20006 - Ethernet Address Register - Lower Four Bytes (R/W)

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

Register 20007 - Ethernet Connection Type Register (R/W)

Register 20007 specifies the Ethernet connection used. Zero (0) specifies a 10Base-2 connection; one (1) specifies a 10Base-T connection. If you change your Ethernet connection type, you must cycle the power in the controller.

Register 20102 - On-board Millisecond Timer (Read-only)

Register 20102 is automatically incremented by 1 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).

Setting the Serial Port Parameters

You can adjust the 2217 EtherNetworking Module's serial ports to establish communications with a wide range of RS-232C compatible devices. The default parameters are 9600 baud, NO Parity, 8 data bits, 1 stop bit, and NO handshaking. Special purpose registers let you change the baud rate, parity, and data length. Each parameter is individually selectable on each of the 2217's serial ports.

Selecting a Serial Port

Change parameters for a given serial port by first selecting the port. Select a port by storing the assigned port number to special purpose register 12000. The controller's base port is assigned a value of 0. The first port has an assigned value of 1 and the second port has a value of 2.

Changing the Baud Rate

Change the baud rate of the selected port by storing the appropriate value to special purpose register 12301. Table 5 lists the available baud rates and corresponding register setting.

Table 5. Baud Rate and Register Settings

Baud Rate	Register 12301 Setting	Baud Rate	Register 12301 Setting
300	0	4800	4
600	1	9600	5
1200	2	19200	6
2400	3	38400	7

Changing the Parity

Change a selected port's parity by storing the desired value to the appropriate special purpose register. Register 20012 stores parity for the first port and Register 20016 stores parity for the second port. Table 6 lists the available parity values. Available parity values are limited by the choice of data length.

Table 6. Data Length and Available Parity Values

Data Length	Parity	Register Value
9	NONE	20048
8	EVEN	20549
8	NONE	20048
8	ODD	20559
7	EVEN	20549
7	NONE	20048
7	ODD	20559

Changing the Data Length

Change the data length of a selected port by storing the desired value to the appropriate special purpose register. Use Register 20011 to store the data length for the first port and Register 20015 for the second port. Available data lengths are 7, 8, and 9 bits.