

FactoryFloor 3.0 Ethernet Peer-to-Peer

INTRODUCTION

This example demonstrates an Ethernet peer-to-peer connection using the Ethernet adapter card. This example also outlines how to transition Ethernet peer-to-peer strategies from earlier versions of FactoryFloor to FactoryFloor 3.0. **Note:** The term “peer” does not imply that two controllers are connected; it may also describe a connection to a controller from a PC or other connected Ethernet hosts.

SIMPLE COMMUNICATION EXAMPLE

The terms *host*, *client*, and *server* should first be clarified. The term *host* refers to an Ethernet-connected computer. The *client* is the host that opens a TCP/IP session to the server. The *server* supplies data to the client. In many peer-to-peer configurations, it is very difficult to determine who is the client and who is the server; it depends on the communication configuration, or on data sharing requirements between hosts. Many systems do not have an “explicit” client. Rather, each host has features that would name it both client and server.

The following brief outline describes the TCP/IP session opening example:

- The server starts to listen for connection requests using a loop with the command “Accept Session On TCP Port”.
- The client opens a session using the “Open Ethernet Session” command with a valid TCP configuration string on controller port 9.
- The server repeatedly checks the accept status using the “Accept Session On TCP Port” command. If the return value is greater than or equal to zero, then a valid connection has been accepted.
- The client receives an acceptance from the server (behind the scenes) and completes the “Open Ethernet

Session” with a successful opening flag and a valid session number. Had the opening failed, the error number –70 would have been returned.

- The session is now available for both hosts to begin communications.

Below is a simple algorithm to exchange data. This algorithm assumes that a single data type is transmitted to the peer host. In this example, a string is passed back and forth between the two hosts.

- The server assumes that a specific packet will be received (for example, string, integer, float, integer table, or float table).
- The client sends the string using the command “Transmit String via Ethernet” (for tables, use the command “Transmit Table via Ethernet”).
- The server checks for data received by using the command “Get Number Of Characters Waiting On Ethernet Session.” If the number of characters is greater than zero, then data is waiting. The data is received by the server using the command “Receive N Characters Via Ethernet” (or for tables, the command “Receive Table Via Ethernet”).
- The server may either send data back to the client or send back an acknowledgement.

When the data exchange via the Ethernet session is completed, this Ethernet session must be closed using the command “Close Ethernet Session.”

The simple peer-to-peer strategy implements a timeout function to monitor whether the session remains active. When any transmission or reception error occurs, the chart will either attempt to reconnect (if it is the client) or resume listening (if it is the server).

OPTOCONTROL ETHERNET COMMAND SUMMARY

For complete command information, see OptoControl online command help or the *OptoControl Command Reference* (Opto 22 form #725). To purchase the *OptoControl Command Reference*, contact Opto 22 Inside Sales.

Accept Session on TCP Port	Accepts an incoming session.
Close Ethernet Session	Closes the Ethernet session passed to this command.
Open Ethernet Session	Opens an Ethernet session to a host. The session number returned should be inspected for a successful opening.
Ethernet Session Open?	Conditional command returns information regarding the status of an Ethernet session.
Get Ethernet Session Name	Returns the connect string to "Open Ethernet Session." Performing this command on the server will only return a "T:" for a TCP session.
Get Number of Characters Waiting on Ethernet Session	Returns the number of characters waiting on the specific sessions buffer. The number of characters returned should also be inspected for session closure and/or other errors (errors are flagged when the characters waiting are less than zero).
Receive N Characters Via Ethernet	Receive a maximum of N characters via the specified Ethernet session. Inspect the receive code for any session closures or other errors.
Receive String Via Ethernet	Receives a string up to the first carriage return (ASCII 13). Use only for ASCII strings. This function blocks until the first carriage return is found. This function uses the "Configure Port Timeout" and a timeout error must also be inspected after this command is completed. Consider using Receive N Characters Via Ethernet as this function does not block with a timeout and will work with binary data, too.
Receive Table Via Ethernet	Receive a table via Ethernet. Tables are received in a 32-element chunk. Inspect the receive code for any session closures or other errors.
Transmit String Via Ethernet	Transmit a string via Ethernet. Inspect the transmit code for any session closures or other errors.
Transmit Table Via Ethernet	Transmits a table chunk via Ethernet. Up to 32 elements are transmitted and if less, the remainder is padded with bytes to make a 128-byte packet. Inspect the transmit code for any session closures or other errors.
Transmit/Receive String Via Ethernet	Transmit/Receive String Via Ethernet. This is a blocked command and uses the timeout configured from "Configure Port Timeout."

The following notes apply to Ethernet peer-to-peer configurations.

Session Limits	<p>The current firmware for the M4SENET-100 Ethernet card permits up to 128 concurrent sessions. Of these sessions, 32 may be used for host or peer communications.</p> <p>It is the strategy's responsibility to limit the number of accepted Ethernet sessions. A count should be maintained of active sessions, and excessive sessions should be closed by the server.</p>
Session Numbers Returned	<p>Session numbers returned by the commands "Open Ethernet Session" and "Accept Session on TCP Port" are allocated on a first-available, first-come basis. A strategy should not determine the next session number based on the previous value. Rather, use the session number returned from the Accept Session on TCP Port or Open Ethernet Session commands.</p>
Peer Client TCP Port Numbers	<p>The M4SENET-100 will use the first reasonable available TCP port number when a controller requests to open a session. This TCP port number will be constant over the life of this session but is not guaranteed to remain the same upon creation of the next similar connection.</p>
Stopping the Strategy	<p>Stopping the strategy closes all Ethernet peer sessions.</p>
Pausing the Peer Chart	<p>Pausing the peer charts may cause inadvertent timeouts with the other host. Both hosts implement a timeout and may cause an ISDISCONNECTED session on the M4SENET-100. Care should always be taken when pausing the peer chart. If all 32 sessions are filled, OptoControl and SCADA packages may be "locked out" of accessing the controller.</p>
OptoControl Session Requirements	<p>OptoControl uses two Ethernet host sessions on the M4SENET-100. Other SCADA packages vary depending on the session architecture.</p>

HISTORIC ISSUES FOR FACTORYFLOOR ETHERNET PEER-TO-PEER CONNECTIVITY

The following notes are important for users who have FactoryFloor 2.2 Ethernet peer-to-peer strategies and are considering upgrading them to FactoryFloor 3.0 or later.

Topic	Notes
Get Session Number command is now obsolete	<p>The historic FactoryFloor peer server model had a single buffer that received data. The session number that received this data had to be explicitly retrieved to identify the sender of the data.</p> <p>The FactoryFloor 3.0 peer server now has independent buffers based on "accepted" connections. See "Accept Session on TCP Port" in the <i>OptoControl Command Reference</i>.</p> <p>When a FactoryFloor 2.2 strategy is opened in version 3.0, this command will be commented out.</p>
New command: Accept Session on TCP Port	This command allows the peer server to accept a connection request from a TCP host.
New command: Get Number Of Characters Waiting On Ethernet Session	This command retrieves the number of characters waiting on an open TCP session.
Open Ethernet Session command has been changed	The connection string now supports TCP connections only. The connection string (for example, T:10.192.54.41:2002) must now contain the protocol type, the IP address, and the TCP port number.
Close Ethernet Session command has been changed	Every Ethernet session must be closed to release the session back to the M4SENET-100's operating systems.
Rapidly opening and closing sessions is not recommended	Ethernet sessions are designed for long-term use. Do not rapidly open and close Ethernet sessions. Socket resources are consumed faster than they are reallocated.