SECTION 5
FINS Communications

This section provides information on communicating on Ethernet Systems and interconnected networks using FINS commands. The information provided in the section deals only with FINS communications in reference to Ethernet Units.

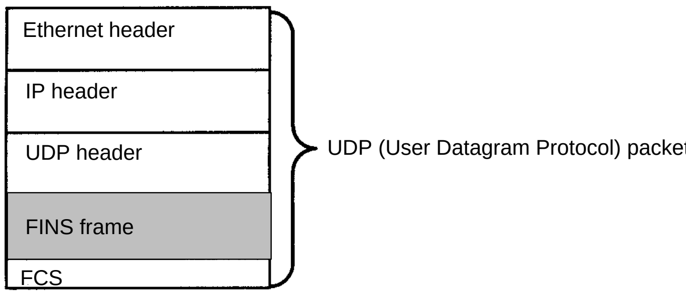
FINS commands issued from a PC are sent via the SEND(090), RECV(098), and CMND(490) instructions programmed into the user ladder-diagram program. Although an outline of these instructions is provided in this section, refer to the *CS1-series CS1G/H-CPU□□-E Programmable Controllers Programming Manual (W340)* for further details on programming these instructions.

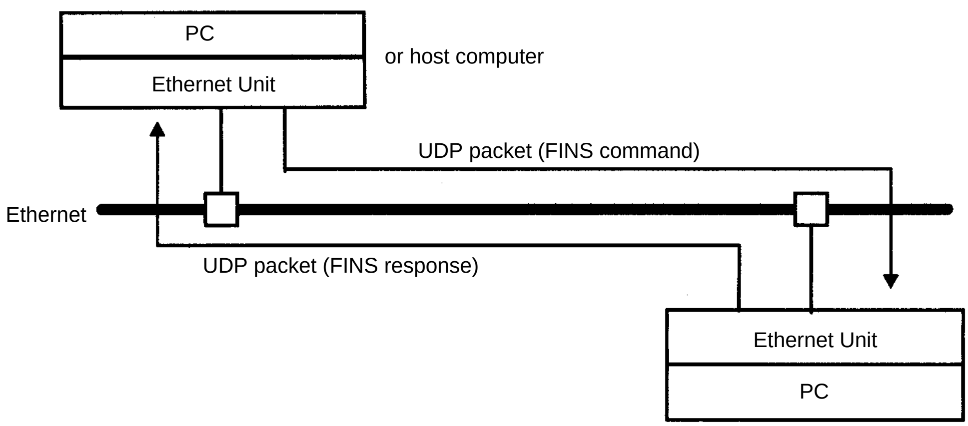
5-1 Overview of FINS Communications

The FINS communications service enables client control of operations such as reading or writing server PC memory area data without the need to program these operations into the server PC user program. The Ethernet Unit uses a dedicated UDP/IP port to execute the FINS communications service. (Refer to *FINS UDP Port Number* under *4-2-1 Settings*.)

5-1-1 Communications On an Ethernet Network

Data is sent and received as UDP packets on an Ethernet network. The FINS port number (default value: 9600) set in the CPU Bus Unit System Setup is used for FINS communications. (Refer to *4-2 CPU Bus Unit System Setup*.)





When a FINS command is issued from the Ethernet Unit, the IP address is found from the FINS address specified by CMND(490). For details, refer to *3-1 Before Operation*.

Note The UDP/IP protocol does not provide communications control to ensure communications reliability. Consequently, the FINS communications services using the UDP/IP protocols cannot guarantee that any message arrived safely and unaltered at the destination. Methods to ensure reliability, such as retries and appropriate processing of FINS responses, must be programmed into the user application.

5-1-2 FINS Communications Service Features

The FINS communications service is a function for controlling operations such as sending and receiving data, changing modes, and so on, between nodes on OMRON factory automation networks. It provides the following features.

- Communications instructions are executed in the user program.
- Writing data, changing modes, reading detailed information about Units, and so on, can be executed without any particular knowledge of communications procedures or network classification.

- Units and Boards that support FINS commands return responses automatically, so there is no need for a program at the receiving end.
- The FINS communications service is mainly used between OMRON CPU Bus Units, CPU Units, and Support Boards for FA Computers. By correctly setting information such as headers, however, it can also be used from ordinary Ethernet communications devices.

The FINS communications service can be used from a PC with either of the following three instructions:

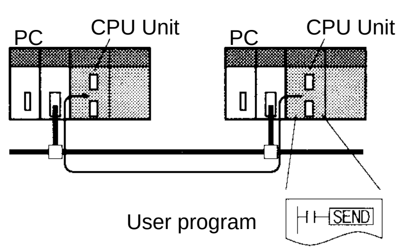
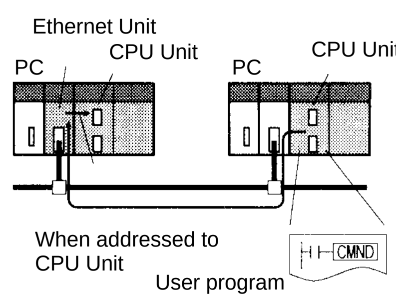
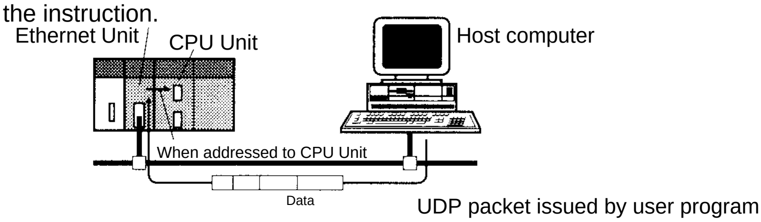
- **SEND(090)/RECV(098)**

  SEND(090) and RECV(098) are used to send and receive data (area reading and writing).

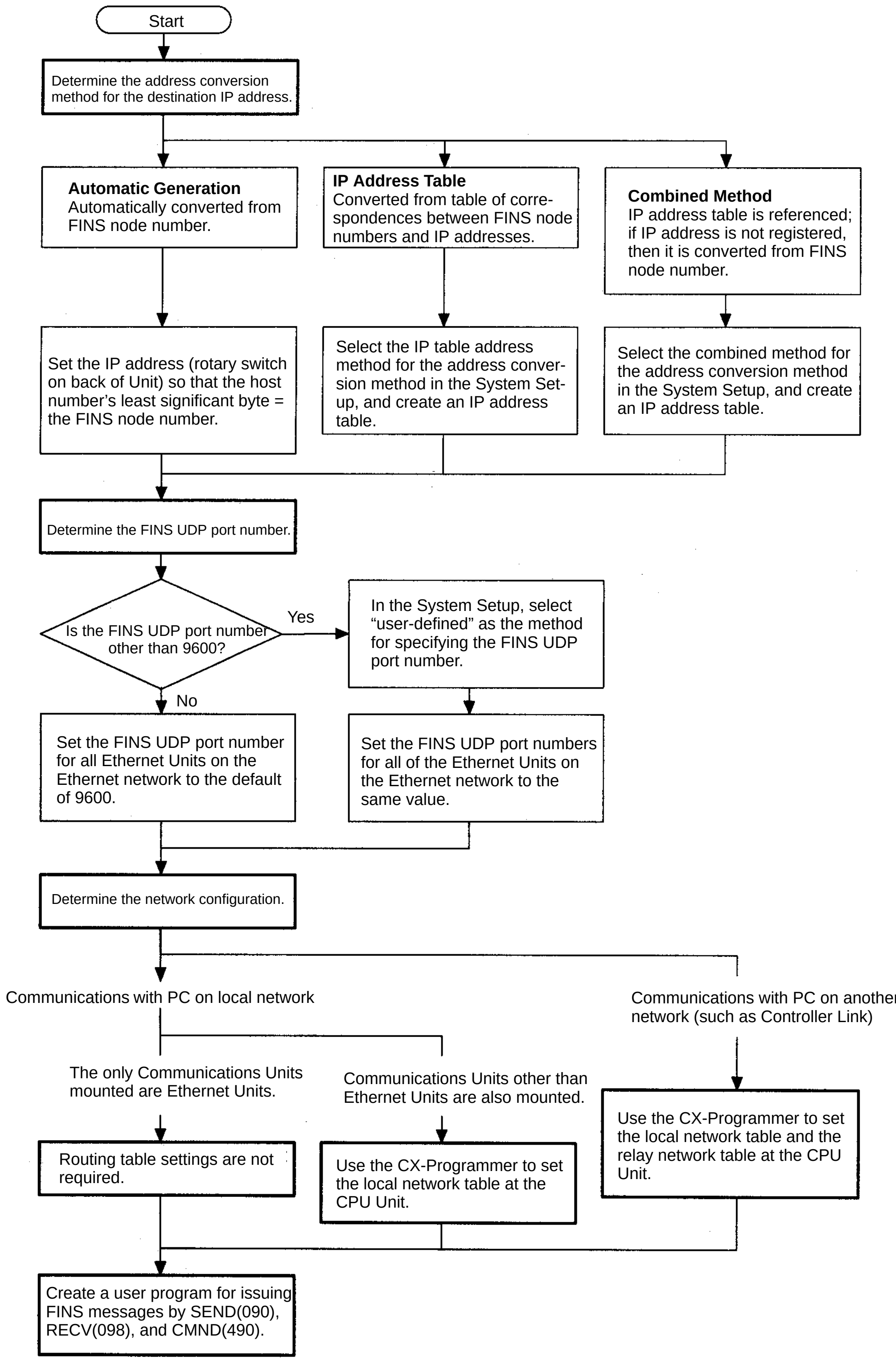
- **CMND(490)**

  CMND(490) is used to send FINS commands. The particular FINS commands that are supported vary depending of the type of Unit or Board. For details on FINS commands addressed to Ethernet Units, refer to *Section 11 FINS Commands Addressed to Ethernet Units*. For details regarding FINS commands addressed to CS1-series CPU Units, refer to the *CS1-series CS1G/H-CPU□□-E Programmable Controllers Communications Commands Reference Manual (W342)*.

The following table shows how the FINS communications service is used by the communications source and destination, for PC to PC and host computer to PC.

Local node to remote node	SEND(090)/RECV(098)	CMND(490) (FINS commands)
PC to PC	• When the PC executes SEND(090) or RECV(098), a program is not required for receiving a response. • When the PC receives a SEND(090) or RECV(098) instruction, a program is not required for processing the instruction. 	• When the PC executes CMND(490), a program is not required for receiving a response. • When the PC receives a CMND(490) instruction, a program is not required for processing the instruction. 
Host computer to PC	• The host computer issues the data for the SEND(090)/RECV(098) or CMND(490) instruction as a UDP datagram. Programs are required for sending data and for receiving and processing responses. • If the PC receives a SEND(090) or RECV(098) instruction, a program is not required for processing the instruction. 	

5-2 Procedure Before Using FINS Communications



5-3 Sending Commands From a PC

FINS commands can be sent from the user's ladder-diagram program in the PC by using the SEND(090), RECV(098), and CMND(490) instructions.

SEND(090): Writes I/O data from the local node to another node.
RECV(098): Reads I/O data from another node to the local node.
CMND(490): Issues FINS commands for controlling operations such as sending and receiving I/O memory data to and from other nodes, reading information regarding other nodes, and so on.

5-3-1 Communications Specifications

The following table shows the specifications for PC communications using the SEND(090), RECV(098), and CMND(490) instructions.

Item	Specifications
Destination	1:1 SEND(090), RECV(098), CMND(490) instructions 1:N SEND(090), CMND(490) instructions (broadcasting)
Data length	SEND(090): 990 words (1,980 bytes) max.; broadcasting: 727 words (1,454 bytes) RECV(098): 990 words (1,980 bytes) max. CMND(490): 1,990 bytes max.; broadcasting: 1,462 bytes (after FINS command code)
Data contents	The following data is sent and received with the execution of each instruction. SEND(090): Sends request for remote node to receive data, and receives response data. RECV(098): Sends request for remote node to send data, and receives response data. CMND(490): Sends any FINS command and receives response data.
Communications port number	Ports 0 to 7 (Eight transmissions can occur simultaneously.)
Response monitor time	0000: 2 s (default) 0001 to FFFF: 0.1 to 6,553.5 s in 0.1-s increments (specified by user)
Number of retries	0 to 15 retries

Note
1. The maximum data length is limited to 512 bytes for data exchange between the PC and SYSMAC LINK Systems or the PC and SYSMAC BUS/2 Remote I/O Systems.
2. When broadcasting, do not require a response.

PC Communications Data Areas

The following table shows the I/O data areas involved when SEND(090) and RECV(098) are used.

Area	Range
CIO Area	CIO 0000 to CIO 6143
Work Area	W000 to W511
Holding Area	H000 to H511
Auxiliary Area	A000 to A959 (See note 1.)
Timer Area	TIM0000 to 4095
Counter Area	CNT0000 to 4095
DM Area	D00000 to D32767
EM Area	E00000 to E32767 (See note 2.)

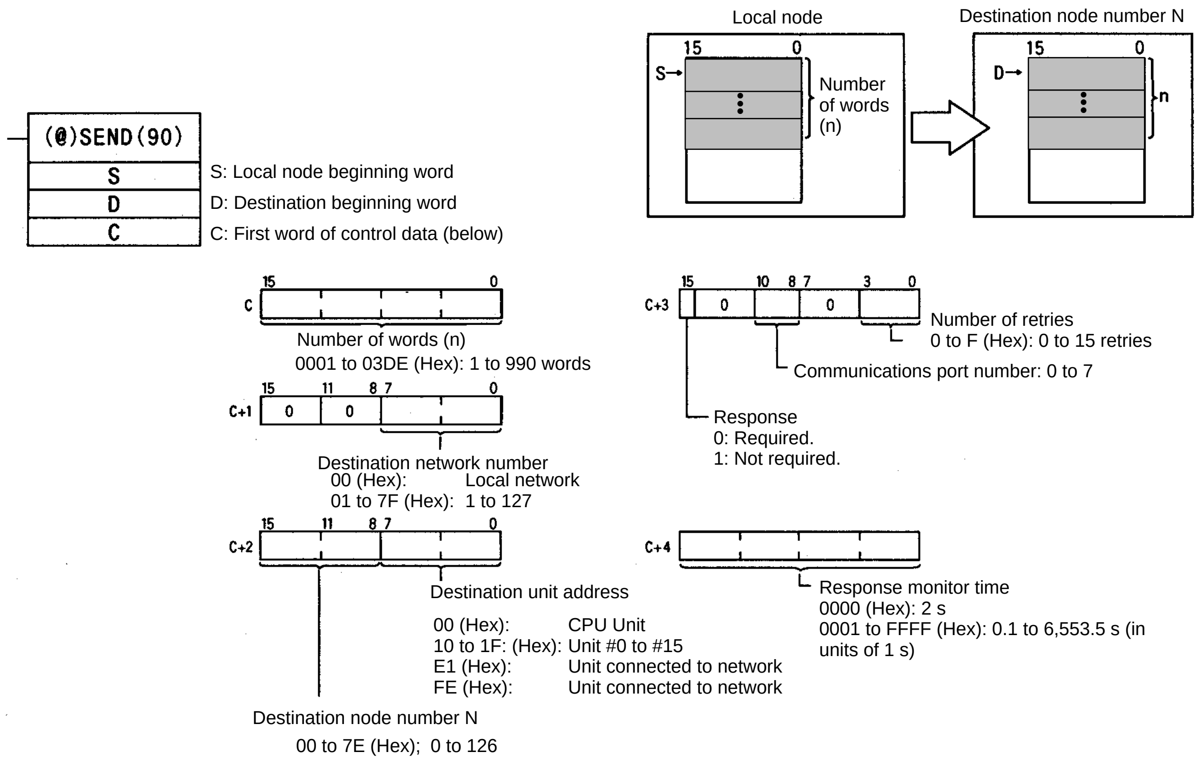
Note
1. Data cannot be written to words A000 to A447 in the Auxiliary Area.
2. A maximum of 13 banks in the EM Area can be used. For details regarding the EM Area, refer to the operation manual for the PC that is used.

Using SEND(090), RECV(098), and CMND(490)

Make the settings shown below when using the SEND(090), RECV(098), and CMND(490) instructions in the user's ladder-diagram program in the PC.

SEND(090)

The SEND(090) instruction sends the data in n number of words, starting from the beginning word S at the local node, to the words starting from the beginning word D at the remote destination node (node number N).



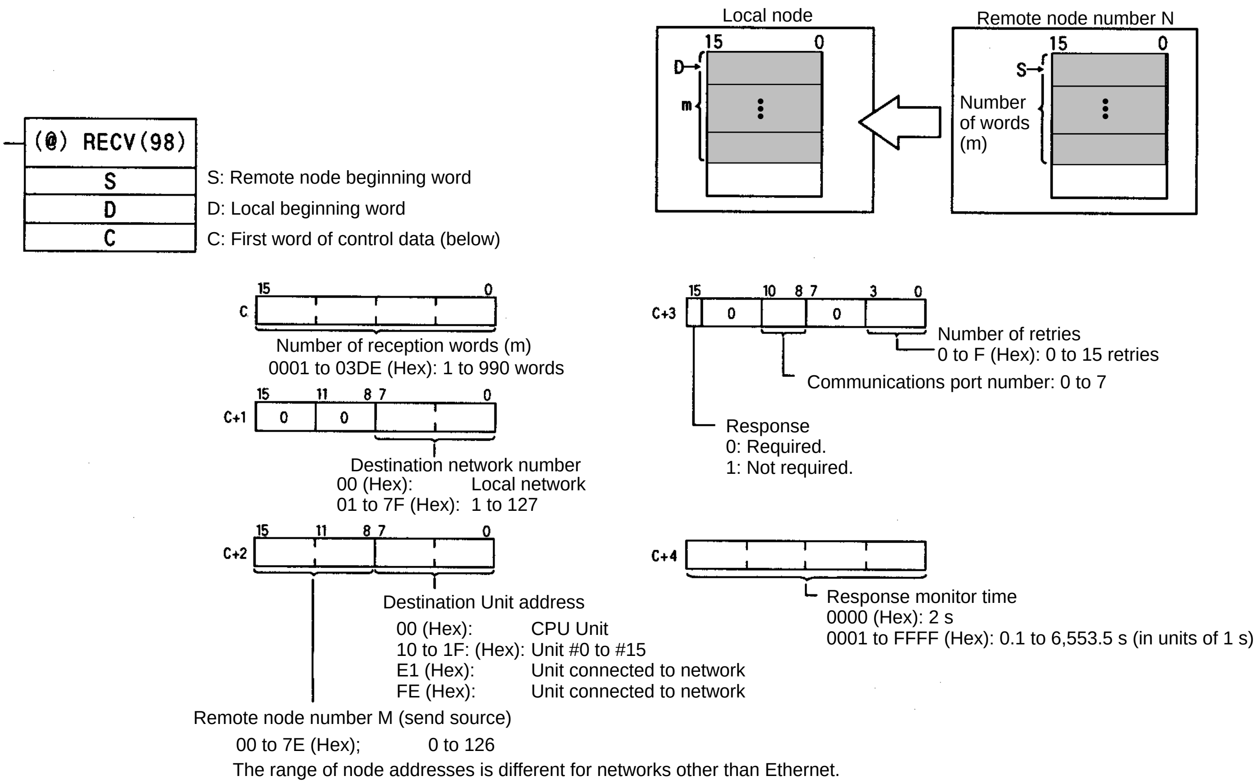
The same data can be broadcast to all nodes on the network by setting the destination node number to FF (Hex).

The range of node addresses is different for networks other than Ethernet.

Note The message service does not guarantee that a message will reach the destination node. A message may be lost during transmission due to factors such as noise. To prevent this from occurring when using message services, it is common to set up retry processing at the node from which instructions are issued. With the SEND(090), RECV(098), and CMND(490) instructions, retry processing is executed automatically by specifying the number of retries, so specify a number other than 0.

RECV(098)

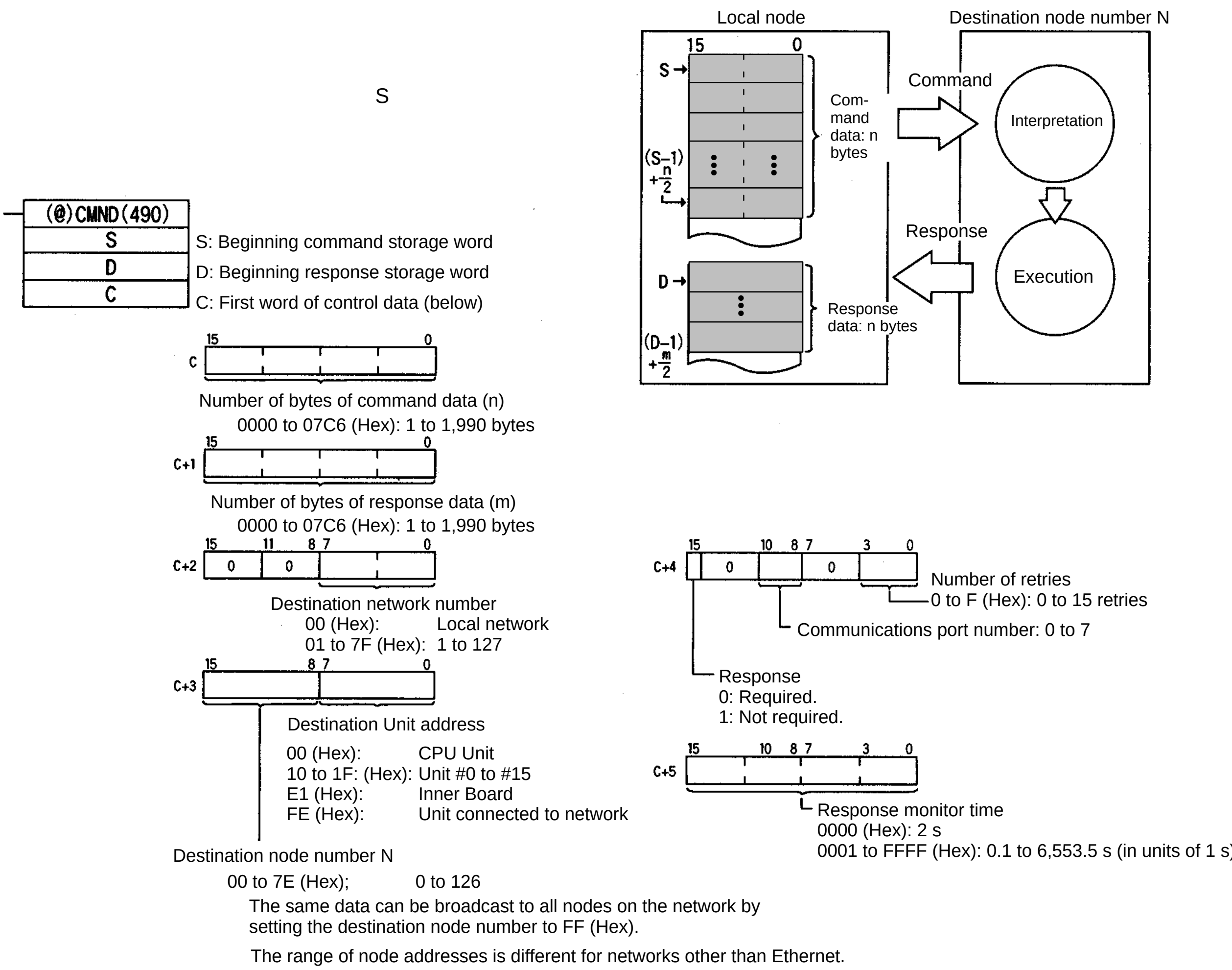
With the RECV(098) instruction, the data in m number of words, starting from the beginning word S at the remote node (node number M) is received at the words starting from the beginning word D at the local node.



Note The message services function does not guarantee that a message will reach the destination node. A message may be lost during transmission due to factors such as noise. In order to prevent this from occurring when using message services, it is common to set up retry processing at the node from which instructions are issued. With the SEND(090), RECV(098), and CMND(490) instructions, retry processing is executed automatically by specifying the number of retries, so specify a number other than 0.

CMND(049)

The CMND(049) instruction sends n bytes of command data, starting from the beginning word S at the local node, to the node at node number N. the data in m number of words, starting from the beginning word S at the remote node (node number M) is received at the words starting from the beginning word D at the local node.



Note The message services function does not guarantee that a message will reach the destination node. A message may be lost during transmission due to factors such as noise. In order to prevent this from occurring when using message services, it is common to set up retry processing at the node from which instructions are issued. With the SEND(090), RECV(098), and CMND(490) instructions, retry processing is executed automatically by specifying the number of retries, so specify a number other than 0.

Commands Addressed to CS1-series CPU Units

The following table provides a list of FINS commands that can be processed by a CS1-series CPU Unit. For details, refer to the *CS1-series CS1G/H-CPU□□-E Programmable Controllers Communications Commands Reference Manual (W342)*.

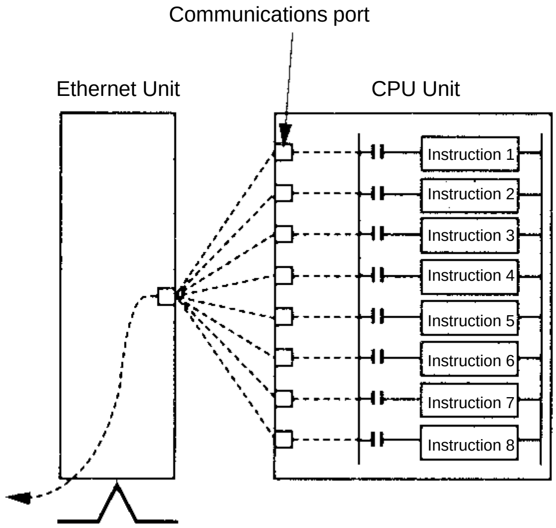
For details on FINS commands that can be processed by the Ethernet Unit, refer to *Section 11 FINS Commands Addressed to Ethernet Units*.

Usage	Command code		Name	Function
	MR	SR		
I/O memory area access	01	01	MEMORY AREA READ	Reads the contents of consecutive I/O memory area words.
	01	02	MEMORY AREA WRITE	Writes the contents of consecutive I/O memory area words.
	01	03	MEMORY AREA FILL	Writes the same data to the specified range of I/O memory area words.
	01	04	MULTIPLE MEMORY AREA READ	Reads the contents of specified non-consecutive I/O memory area words.
	01	05	MEMORY AREA TRANSFER	Copies the contents of consecutive I/O memory area words to another I/O memory area.
Parameter area access	02	01	PARAMETER AREA READ	Reads the contents of consecutive parameter area words.
	02	02	PARAMETER AREA WRITE	Writes the contents of consecutive parameter area words.
	02	03	PARAMETER AREA FILL (CLEAR)	Writes the same data to the specified range of parameter area words.
Program area access	03	06	PROGRAM AREA READ	Reads the UM (User Memory) area.
	03	07	PROGRAM AREA WRITE	Writes to the UM (User Memory) area.
	03	08	PROGRAM AREA CLEAR	Clears the UM (User Memory) area.
Operating mode changes	04	01	RUN	Changes the CPU Unit's operating mode to RUN or MONITOR.
	04	02	STOP	Changes the CPU Unit's operating mode to PROGRAM.
Machine configuration reading	05	01	CPU UNIT DATA READ	Reads CPU Unit data.
	05	02	CONNECTION DATA READ	Reads the model numbers of the device corresponding to addresses.
Status reading	06	01	CPU UNIT STATUS READ	Reads the status of the CPU Unit.
	06	20	CYCLE TIME READ	Reads the maximum, minimum, and average cycle time.
Time data access	07	01	CLOCK READ	Reads the present year, month, date, minute, second, and day of the week.
	07	02	CLOCK WRITE	Changes the present year, month, date, minute, second, or day of the week.
Message display	09	20	MESSAGE READ/CLEAR	Reads and clears messages, and reads FAL/FALS messages.
Access rights	0C	01	ACCESS RIGHT ACQUIRE	Acquires the access right as long as no other device holds it.
	0C	02	ACCESS RIGHT FORCED ACQUIRE	Acquires the access right even if another device already holds it.
	0C	03	ACCESS RIGHT RELEASE	Releases the access right that has been acquired.

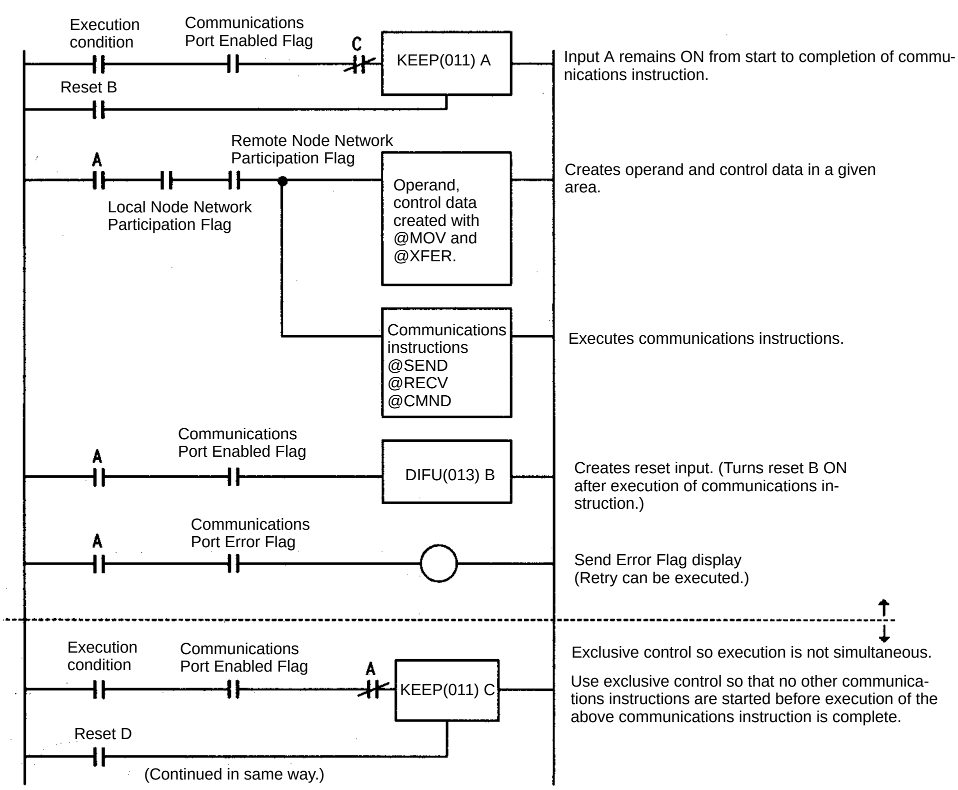
Usage	Command code		Name	Function
	MR	SR		
Error log	21	01	ERROR CLEAR	Clears errors or error messages.
	21	02	ERROR LOG READ	Reads the error log.
	21	03	ERROR LOG POINTER CLEAR	Clears the error log pointer.
File memory	22	01	FILE NAME READ	Reads file memory data.
	22	02	SINGLE FILE READ	Reads a specified length of file data from a specified position within a single file.
	22	03	SINGLE FILE WRITE	Writes a specified length of file data from a specified position within a single file.
	22	04	FILE MEMORY FORMAT	Formats (initializes) the file memory.
	22	05	FILE DELETE	Deletes specified files stored in the file memory.
	22	07	FILE COPY	Copies files from one file memory to another file memory in the same system.
	22	08	FILE NAME CHANGE	Changes a file name.
	22	0A	MEMORY AREA–FILE TRANSFER	Transfers or compares data between the I/O memory area and the file memory.
	22	0B	PARAMETER AREA–FILE TRANSFER	Transfers or compares data between the parameter area and the file memory.
	22	0C	PROGRAM AREA–FILE TRANSFER	Transfers or compares data between the UM (User Memory) area and the file memory.
	22	15	CREATE/DELETE DIRECTORY	Creates or deletes a directory.
Debugging	23	01	FORCED SET/RESET	Force-sets or force-resets bits, or releases force-set status.
	23	02	FORCED SET/RESET CANCEL	Cancels all bits that have been force-set or force-reset.

Writing Programs

Programs incorporating the SEND(090), RECV(098), and CMND(490) instructions are generally created using the Communications Port Enabled Flag and the Communications Port Error Flag as input conditions. CS1-series CPU Units have eight communications ports. Only one instruction can be executed at any given port at one time, however, so the program must not overlap the use of any of the ports. A program example is provided below.



There are eight communications ports, so up to eight communications instructions can be executed at a time. The number of messages that can be sent or received with a single CPU Bus Unit service, though, is not more than two each for the CPU Unit to the Ethernet Unit and for the Ethernet Unit to the CPU Unit.



Communications Flags

The execution status of the SEND(090), RECV(098), and CMND(490) instructions is always reflected by the communications flags (i.e., the Communications Port Enabled Flag and the Communications Port Error Flag). The CS1-series CPU Unit's communications flags are allocated in the Auxiliary Area as shown in the following table.

Flag name	Address		Contents
	Word	Bits	
Communications Port Enabled Flag	A202	Bit 7: Port 7 Bit 6: Port 6 Bit 5: Port 5 Bit 4: Port 4 Bit 3: Port 3 Bit 2: Port 2 Bit 1: Port 1 Bit 0: Port 0	OFF: Execution enabled (being executed) ON: Execution disabled (not being executed)
Communications Port Error Flag	A219	Bit 7: Port 7 Bit 6: Port 6 Bit 5: Port 5 Bit 4: Port 4 Bit 3: Port 3 Bit 2: Port 2 Bit 1: Port 1 Bit 0: Port 0	OFF: Normal completion ON: Abnormal completion

Note With CS1-series PCs, communications ports 0 to 7 are also used for executing the PCMR(260) (PROTOCOL MACRO) instruction, so these flags are used in common for SEND(090), RECV(098), CMND(490), and PCMR(260). While PCMR(260) is being executed, SEND(090), RECV(098), and CMND(490) cannot be executed at the same communications port.

Communications Port Completion Codes

The status of a SEND(090), RECV(098), and CMND(490) instruction after execution is reflected as a communications port completion code, in one word (two bytes) of data as shown in the following table. (The value is 0000 during instruction execution.) The recorded status is saved until execution of the next instruction.

Word	Contents
A203	Communications Port 0 Completion Code
A204	Communications Port 1 Completion Code
A205	Communications Port 2 Completion Code
A206	Communications Port 3 Completion Code
A207	Communications Port 4 Completion Code
A208	Communications Port 5 Completion Code
A209	Communications Port 6 Completion Code
A210	Communications Port 7 Completion Code

The meanings of the communications port completion codes are the same as those for FINS commands and responses. When CMND(490) is used, however, even if a FINS command has an abnormal completion, it will not be reflected in the communications port completion code. For details, refer to *Communications Port Error Flag and Completion Codes for CMND(490)* below.

Bits 08 to 15 in the communications port completion code correspond to the first byte of the response code, and bits 00 to 07 correspond to the second byte. For details, refer to *10-6 Troubleshooting With Response Codes*.

Communications Port Error Flag and Completion Codes CMND(490)

Errors that occur when CMND(490) is used generate a Communications Port Error Flag and are recorded in a communications port completion code only in the following cases:

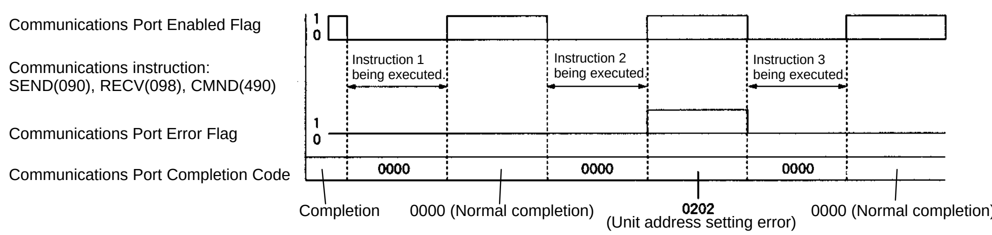
- When a response timeout error has occurred.
- When the number of communications data bytes exceeds the maximum value for the Unit (i.e., 2,000 bytes for the Ethernet Unit).
- When the actual number of response bytes is greater than the number of reception bytes that has been set. (The response is not stored in this case.)

Errors other than these are recorded in the response codes of the responses stored from the beginning response storage word onwards. Be careful of these, because there are no Communications Port Error Flags and they are not recorded in a communications port completion code.

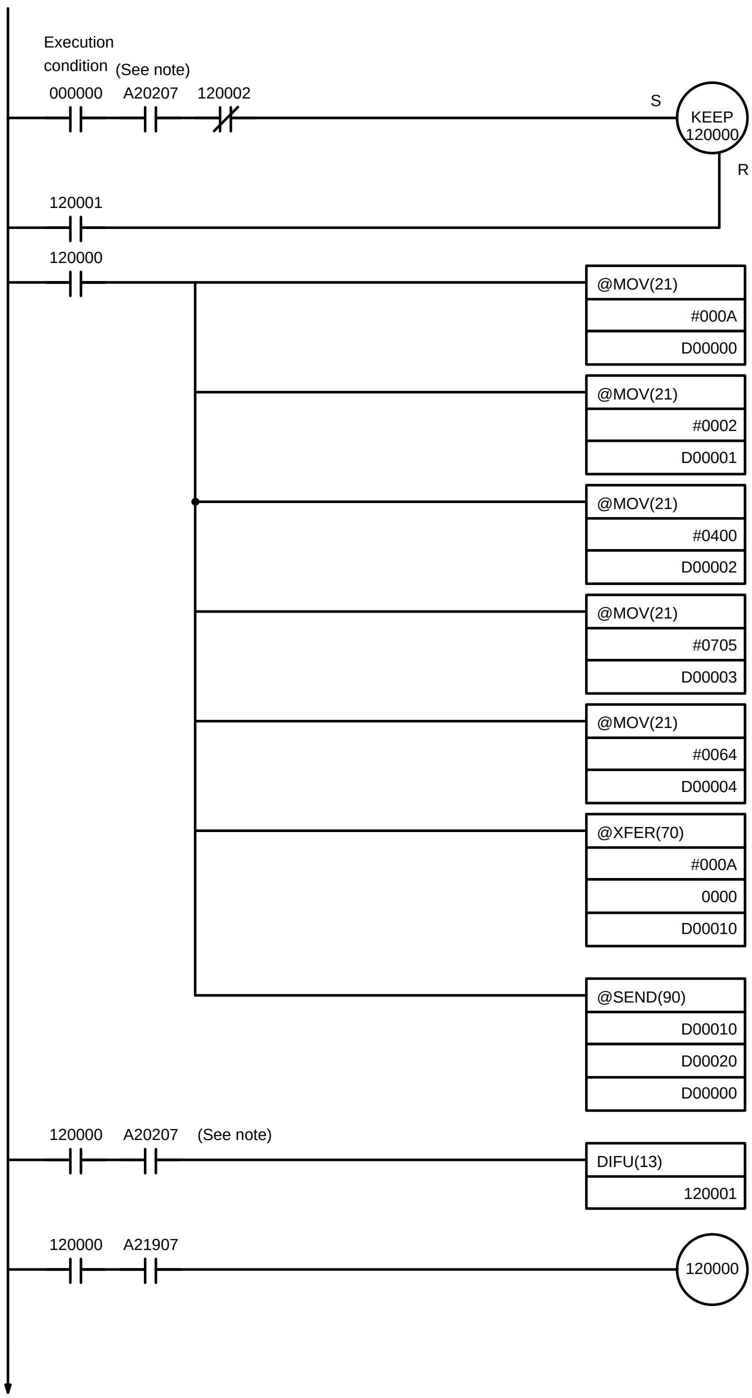
Timing of Communications Flag Changes

- The Communications Port Enabled Flag remains OFF during communications and turns ON when they are completed (regardless of whether or not an error occurs).
- The Communications Port Error Flag retains its status until the next transmission or reception.
- The Communications Port Error Flag turns OFF with the execution of the next communications instruction even if there was an abnormal completion.

Example



5-3-2 Program Example



When the Communications Port Enabled Flag for port 7 is ON, and RECV(098) is not being executed, the send execution program will start when execution condition CIO 000000 turns ON.

Input CIO 120000 remains ON from the start of SEND(090) execution until completion.

Control Data Creation

Word	Contents		Meaning
D0000	00	0A	Number of send words = 10
D0001	00	02	Destination network number = 2
D0002	04	00	Destination node number = 4 Destination unit address = 0
D0003	07	05	Response required. Communications port No. used = 7 Number of retries = 5
D0004	00	64	Response monitor time = 10 s

Send Data Creation

Ten words of data from word CIO 0000 is stored from D00010 onwards.

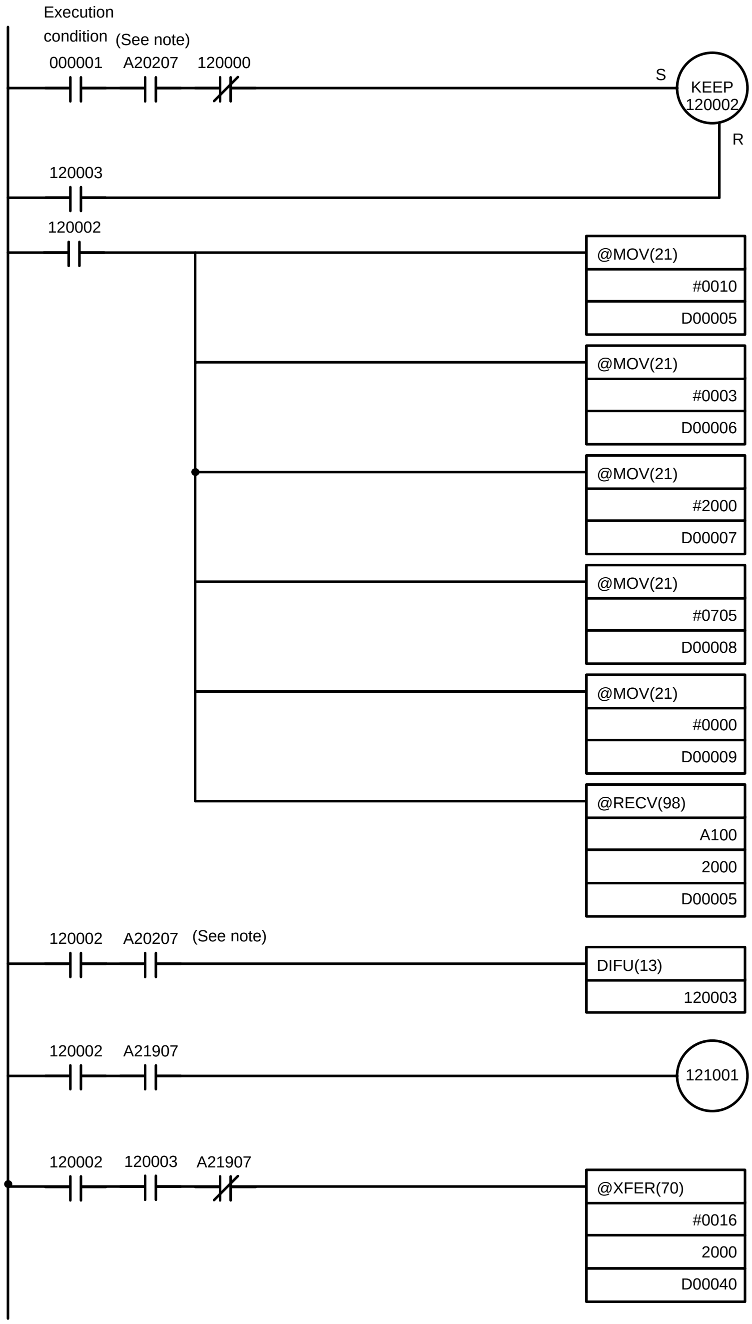
Ten words of data from D00010 at the local node is sent to D00020 onwards at network number 2, node number 4, unit address 0 (the PC).

Reset Input Creation

Send Error Display

(Continued on next page.)

(Continued from previous page.)



When the Communications Port Enabled Flag for port 7 is ON, and SEND(090) is not being executed, the transmission execution program will start when execution condition CIO 000001 turns ON.

Input CIO 120002 remains ON from the start of RECV(098) execution until completion.

Control Data Creation

Word	Contents		Meaning
D0005	00	10	Number of reception words = 16
D0006	00	03	Destination network number = 3
D0007	20	00	Destination node number = 32 Destination unit address = 0
D0008	07	05	Response required. Communications port No. used = 7 Number of retries = 5
D0009	00	00	Response monitor time = Default

A total of 16 words of data beginning from word A100 at network number 3, node number 32, unit address 0 (the PC) is received at word CIO 2000 onwards of the local node.

Reset Input Creation

Reception Error Display

Reception Data Processing

If there is no reception processing completion error, the 16 words of data received from word CIO 2000 onwards is stored at D00040 onwards.

Note With CS1-series PCs, the Communications Port Enabled Flags at bits 0 to 7 in word A202 turn OFF even when the PCMR(260) instruction is being executed using the ports corresponding to those flags.

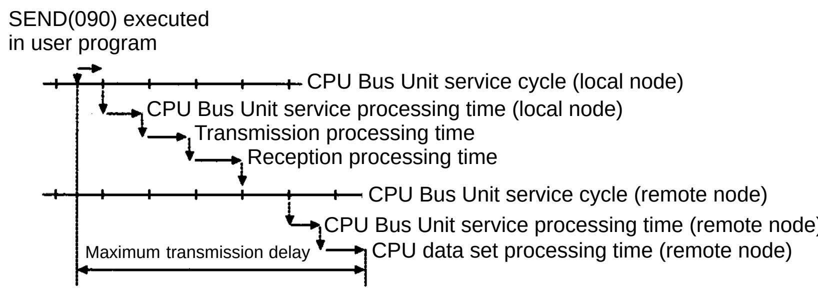
5-3-3 Transmission Delays

The methods of computing the maximum time required from execution of the SEND(090), RECV(098), and CMND(490) instructions until processing is completed are described in this section. These times, however, do not take transmission delay times on the network into account, so they may be increased depending on the conditions under which the instructions are executed.

SEND(090)

The transmission delay for the SEND(090) instruction can be calculated using the following equation, which is illustrated in the following diagram.

Max. delay = Local node service cycle + local node service processing time + transmission processing time + reception processing time + remote node service cycle + remote node service processing time + CPU data set processing time (remote node)



CPU Bus Unit Service Cycle

The CPU Bus Unit service cycle is a single PC cycle.

CPU Bus Unit Service Processing Time

This is the time required to process CPU Bus Units and is approximately 1 ms for Ethernet Units.

Transmission Processing Time

Number of words transferred x 0.011 + 3 ms

Reception Processing Time

Number of words transferred x 0.011 + 3 ms

CPU Data Set Processing Time

Number of words transferred x 0.02 + 20 ms

Note

1. The actual operating environment can cause transmission delays larger than those calculated with the methods given here. Among the causes of longer delays are the following: traffic on the network, window sizes at network nodes, traffic through the Ethernet Unit (e.g., socket servicing, FTP server communications, etc.), and the system configuration.
2. The CPU data set processing time is the standard when the peripheral service time is set in the CPU Unit System Setup to the default of 4%. As this value is increased, the processing time is shortened.

Example Calculations

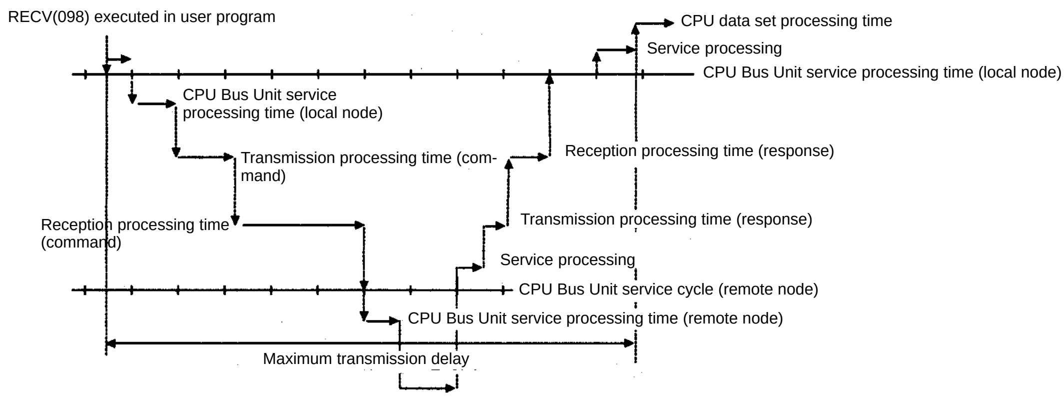
The following example shows calculations for sending 256 words between two PC nodes using SEND(090). The local node's CPU cycle time is 10 ms, and the remote node's CPU cycle time is 5 ms. Calculations are shown in the following table.

Item	Calculation
CPU Bus Unit service cycle (local node)	PC cycle time = 10 ms
CPU Bus Unit service processing time (local node)	1 ms
Transmission processing time	256 x 0.011 + 3 = 5.816 6 ms
Reception processing time	256 x 0.011 + 3 = 5.816 6 ms
CPU Bus Unit service cycle (remote node)	5 ms
CPU Bus Unit service processing time (remote node)	1 ms
CPU data set processing time (remote node)	256 x 0.02 + 20 = 25.12 25 ms
Total	10 + 1 +6 + 6 + 5 + 1 + 25= **54 ms**

RECV(098)

The transmission delay for the RECV(098) instruction can be calculated using the following equation, which is illustrated in the following diagram.

Max. delay = Local node service cycle + local node service processing time + transmission processing time (command) + reception processing time (command) + remote node service cycle + remote node service processing time + CPU data read processing time (remote node) + remote node service processing time + transmission processing time (response) + reception processing time (response) + local node service cycle + local node service processing time + CPU data set processing time (local node)



CPU Bus Unit Service Cycle

The CPU Bus Unit service cycle is a single PC cycle.

CPU Bus Unit Service Processing Time

This is the time required to process CPU Bus Units and is approximately 1 ms for Ethernet Units.

Transmission and Reception Processing Times

Command: 3 ms
Response: Number of words transferred x 0.011 + 3 ms

CPU Data Read Processing Time

The CPU data read processing time is the integer portion of the minimum cycle time, which satisfies the following formula.

Number of words transferred x 0.02 + 20 ms < Integer portion of remote node cycle time

CPU Data Set Processing Time

Number of words transferred x 0.02 + 20 ms

Example Calculations

The following example shows calculations for receiving 256 words between two PC nodes using RECV(098). The local node's CPU cycle time is 10 ms, and the remote node's CPU cycle time is 15 ms. Calculations are shown in the following table.

Item	Calculation
CPU Bus Unit service cycle (local node)	PC cycle time = 10 ms
CPU Bus Unit service processing time (local node)	1 ms
Transmission processing time (command)	3 ms
Reception processing time (command) + Remote node service cycle + CPU Bus Unit service processing time (remote node)	3 + 15 + 1 + (256 x 0.011 + 3) = 24.816 25 ms
CPU data read processing time (remote node)	256 x 0.02 + 20 = 25.12 30 ms
Service processing	1 ms
Transmission processing time (response)	256 x 0.011 + 3 = 5.816 6 ms
Reception processing time (response) + Local node service cycle + CPU Bus Unit service processing time (local node)	(256 x 0.011 + 3) + 10 + 1 = 16.816 17 ms
CPU data set processing (local node)	256 x 0.02 + 20 = 25.12 25 ms
Total	10 + 1 + 3 + 25 + 30 + 1 + 6 + 17 + 25 = **118 ms**

5-4 Sending Commands From a Host Computer

Commands and responses sent from host computers must be in the formats described in this section and must provide the proper FINS header information. These formats can also be used to decode commands and responses received from other network nodes.

5-4-1 Designating Remote Addresses

UDP sockets are used when sending FINS commands from a host computer to a PC. This section provides examples of addressing remote PCs from the host computer for communications.

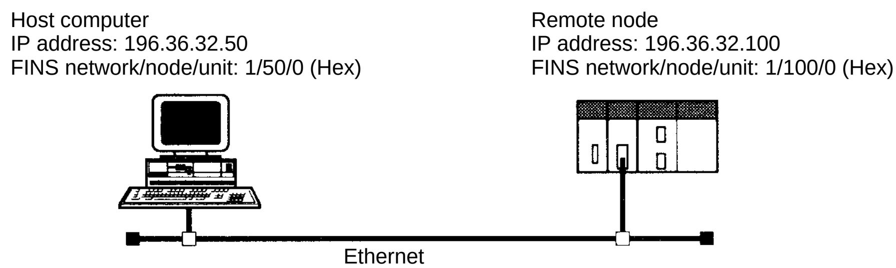
Note

1. The FINS UDP port number at the Ethernet Unit is set to the default of 9600. It can be changed in the CPU Bus Unit System Setup, but the same FINS UDP port number must be set for all of the Ethernet Units on the same Ethernet network.
2. Even if the Ethernet network is comprised of multiple segments, set the same value for the FINS network number.

Example 1: Intranetwork Addressing

In this example, the host computer and the remote node (Ethernet Unit) are on the same network. The communications parameters specified from the host computer would be as follows:

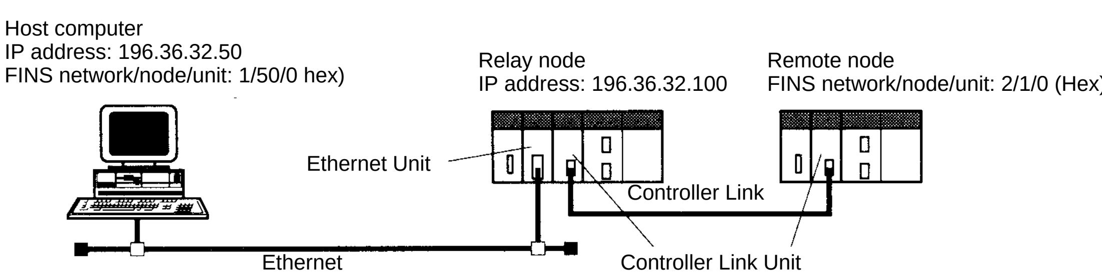
Destination IP Address: 196.36.32.100 (Ethernet Unit of remote node)
UDP port number: FINS UDP port No. (Ethernet Unit of remote node)
FINS addresses (Remote node CPU Unit):
  Network number: 1
  Node number: 100
  Unit number: 0
FINS addresses (Host computer):
  Network number: 1
  Node number: 50
  Unit number: 0



Example 2: Internetwork Addressing

In this example, the host computer and the remote node (Ethernet Unit) are on different networks, connected by a relay node. The communications parameters specified from the host computer would be as follows:

Destination IP Address: 196.36.32.100 (Ethernet Unit of relay node)
UDP port number: FINS UDP port number (Ethernet Unit of relay node)
FINS addresses (Remote node CPU Unit):
  Network number: 2
  Node number: 1
  Unit number: 0
FINS addresses (Host computer):
  Network number: 1
  Node number: 50
  Unit number: 0



5-4-2 FINS Frames

The FINS communications service is carried out through the exchange of FINS command frames and their corresponding response frames. (There are also commands with no responses.)

Both command frames and response frames are comprised of a FINS header for storing transfer control information, a FINS command field for storing a command, and a FINS parameter/data field for storing command parameters and transmission/response data.

FINS Command Frame Configuration

	Field	Size (bytes)	Contents
FINS header	ICF	1	Displays frame information
	RSV	1	Reserved by system.
	GCT	1	Permissible number of gateways
	DNA	1	Destination network address
	DA1	1	Destination node address
	DA2	1	Destination unit address
	SNA	1	Source network address
	SA1	1	Source node address
	SA2	1	Source unit address
	SID	1	Service ID
FINS command	MRC	1	Main request code
	SRC	1	Sub-request code
FINS parameter/data	Parameter/ data field	2000 max.	Command parameters and send data The data length depends on the MRC and SRC.

FINS Response Frame Configuration

The response code (one byte each for MRES and SRES) for the command is added at the beginning of the FINS parameter/data field in the response frame.

	Field	Size (bytes)	Contents
	FINS header	10	Same as in command frame.
	FINS command	2	Same as in command frame.
FINS parameter/data field	MRES	1	Main response code
	SRES	1	Sub-response code
	Data	1998 max.	Response data There are some frames in which the data length is 0.

FINS Header Information

ICF (Information Control Field)

Bit	7	6	5	4	3	2	1	0
	1		0	0	0	0	0	

Bit 0: Response Request Bit
(0: Response required; 1: Response not required)

Bit 6: Kind of data (0: command; 1: response)

RSV (Reserved by system)
Set to 00 (Hex).

GCT (Permissible Number of Gateways)
Set to 02 (Hex).

DNA (Destination Network Address)

Specifies the number of the network where the destination node is located.
00 (Hex): Local network
01 to 7F (Hex): Destination network number (1 to 127)

DA1 (Destination Node Address)

Specifies the number of the node where the command is being sent. This node number is the address used for FINS, and is different from the IP address used for Ethernet.

00 (Hex):	Local PC Unit
01 to 7E (Hex):	Destination node number (1 to 126)
FF (Hex):	Broadcasting

When multiple Communications Units are mounted, DA1 specifies the node number of the Unit connected to the network specified by DNA.

DA2 (Destination Unit Address)

Specifies the number of the Unit at the destination node.

00 (Hex):	PC (CPU Unit)
10 to 1F (Hex):	CPU Bus Unit #0 to #15 (16 to 31)
E1 (Hex):	Inner Board
FE (Hex):	Unit connected to network.

SNA (Source Network Address)

Specifies the number of the network where the source node is located.

00 (Hex):	Local network
01 to 7F (Hex):	Source network number (1 to 127)

SA1 (Source Node Address)

Specifies the local node number. The ranges of numbers that can be specified are the same as for DA1.

SA2 (Source Node Address)
Specifies the number of the Unit at the source node. The ranges of numbers that can be specified are the same as for DA2.

SID (Service ID)
The SID is used to identify the process that data is sent from. Set any desired number from 00 to FF for the SID. The same number will be returned in the response, allowing you to match commands and responses in your application.

5-4-3 Sample Program

Operation Overview

This program reads 150 words of the PC memory beginning at D00100 by sending an FINS command (MEMORY AREA READ, command code 0101) from a UNIX workstation to the PC on the Ethernet network. If no response is received within two seconds of sending the FINS command, the command will be resent.

Settings

The Ethernet Unit IP address is 196.36.32.100, and the FINS node number is 100. IP address conversion is set to automatic generation.

The workstation’s IP address is 196.36.32.50 and its FINS node number is 50.

The FINS UDP port number is 9600 (default).

Sample Program

```
#include <errno.h>
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <signal.h>

#define FINS_UDP_PORT 9600
#define SERV_IP_ADDR ”196.36.32.100”     /*Ethernet Unit IP ADDRESS*/
#define MAX_MSG      2010
#define RESP_TIMEOUT 2


/*
* FINS COMMUNICATIONS SAMPLE PROGRAM
```

```
*/
main(argc,argv)
int   argc;
char  *argv[];
{
     int    sockfd;
     struct sockaddr_in , ws_addr,cv_addr;
     char   fins_cmnd[MAX_MSG],fins_resp[MAX_MSG];
     int    sendlen,recvlen,addrlen;
     char   sid=0;
     extern recv_fail();

     /*GENERATE UDP SOCKET*/
     if((sockfd=socket(AF_INET,SOCK_DGRAM,0))<0)
          err_exit("can't open datagram socket");

     /*ALLOCATE IP ADDRESS AND PORT # TO SOCKET*/
     bzero((char*)&ws_addr,sizeof(ws_addr));
     ws_addr.sin_family=AF_INET;
     ws_addr.sin_addr.s_addr=htonl(INADDR_ANY);
     ws_addr.sin_port=htons(FINS_UDP_PORT);
     if(bind(sockfd,(struct sockaddr*)&ws_addr,sizeof(ws_addr))<0)
          err_exit("can't bind local address");

     /*
     *GENERATE MEMORY AREA READ COMMAND
     *(READ 150 WORDS FROM D00100.)
     /*
     fins_cmnd[0]=0x80;  /*ICF*/
     fins_cmnd[1]=0x00;  /*RSV*/
     fins_cmnd[2]=0x02;  /*GCT*/
     fins_cmnd[3]=0x01;  /*DNA*/
     fins_cmnd[4]=0x64;  /*DA1*/      /*Ethernet Unit FINS NODE NUMBER*/
     fins_cmnd[5]=0x00;  /*DA2*/
     fins_cmnd[6]=0x01;  /*SNA*/
     fins_cmnd[7]=0x32;  /*SA1*/      /*WS FINS NODE NUMBER*/
     fins_cmnd[8]=0x00;  /*SA2*/
     fins_cmnd[9]=++sid; /*SID*/
     fins_cmnd[10]=0x01; /*MRC*/
     fins_cmnd[11]=0x01; /*SRC*/
     fins_cmnd[12]=0x82; /*VARIABLE TYPE: DM*/
     fins_cmnd[13]=0x00; /*READ START ADDRESS: 100*/
     fins_cmnd[14]=0x64;
     fins_cmnd[15]=0x00;
     fins_cmnd[16]=0x00; /*WORDS READ: 150*/
     fins_cmnd[17]=0x96;


     /*SEND FINS COMMAND*/
     bzero((char*)&cv_addr,sizeof(cv_addr));
     cv_addr.sin_family=AF_INET;
     cv_addr.sin_addr.s_addr=inet_addr(SERV_IP_ADDR);
     cv_addr.sin_port=htons(FINS_UDP_PORT);

     singnal((SIGALRM,recv_fail);

CMND_SEND:
     sendlen = 18;
     if(sendto(sockfd,fins_cmnd,sendlen,0,&cv_addr,sizeof(cv_addr))
     ==sendlen){
           alarm(RESP_TIMEOUT); /*START RESPONSE MONITOR TIMER*/
           printf("send length %d¥n",sendlen);
     {
     else{
           err_exit("send error");
     }

     /*RECEIVE FINS RESPONSE*/
```

```
        if((recvlen = recvfrom(sockfd,fins_resp,MAX_MSG,0,&cv_addr,&addrlen))
        <0){
             if(errno == EINTR)
                  goto CMND_SEND; /*RE-SEND FINS COMMAND*/
             err_exit(”receive error”);
        }
        else{
             alarm(0); /*STOP RESPONSE MONITOR TIMER*/
             printf(”recv length %d¥n”,recvlen);
             if(recvlen<14) /*ILLEGAL RESPONSE LENGTH CHECK*/
                  err_exit(”FINS length error”);
             if((fins_cmnd[3]!=fins_resp[6])||(fins_cmnd[4]!=fins_resp[7])
                  ||(fins_cmnd[5]!=fins_resp[8])){ /*DESTINATION ADDRESS CHECK*/
                  err_exit(”illegal source address error”);
             }
             if(fins_cmnd[9]!=fins_resp[9]) /*SID CHECK*/
                  err_exit(”illegal SID error”);
        }


        /*CLOSE SOCKET*/
        close(sockfd);
    }
    /*
    * ERROR PROCESSING FUNCTIONS
    */
    err_exit(err_msg)
    char   *err_msg;
    {
        printf(”client: %s %x¥n”,err_msg,errno);
        exit(1);
    }

    /*
    *SIGNAL CAPTURE FUNCTIONS
    */
    recv_fail()
    {
        printf(”response timeout error ¥n”);
    }
```

5-4-4 Delays for Accessing PC Memory

The time for the response to be received after a remote node on the Ethernet network sends a memory area read or write command to a PC can be calculated using the following formula. This time does not take network transmission delays into account, so it may be extended under some operating conditions.

Write command delay time (ms) =
Remote node communications processing time + 4 + (0.011 x number of words written) + CPU cycle time + CPU data set processing time

Read command delay time (ms) =
Remote node communications processing time + 8 + (0.011 x number of words read) + CPU cycle time + CPU data read processing time

Note

1. The transfer time may exceed the calculated value due to the actual operating environment. Factors affecting the transfer time are network traffic, the window size of each node, Ethernet Unit traffic (e.g., socket services, FTP server communications, etc.), and the system configuration.
2. The CPU data set processing time is the standard when the peripheral service time is set in the CPU Unit System Setup to the default of 4%. As this value is increased, the processing time is shortened.

Example

This example shows the calculations for a host computer sending a write command for 256 words to the PC. The standard for the maximum transfer delay time is as follows when the PC’s CPU cycle time is 10 ms.

Maximum transfer delay time
= Host computer communications processing time + (0.011 x 256) + 10
+ (256 x 0.02 + 20) = host computer communications processing time + 42 ms

5-5 FINS Server

The following table shows the FINS commands that can be addressed to the Ethernet Units. The Ethernet Units automatically send responses for these commands. For details, refer to *Section 11 FINS Commands Addressed to Ethernet Units*.

MRC	SRC	Name
04	03	RESET
05	01	CONTROLLER DATA READ
06	01	CONTROLLER STATUS READ
08	01	INTERNODE ECHO TEST
	02	BROADCAST TEST RESULTS READ
	03	BROADCAST DATA SEND
21	02	ERROR LOG READ
	03	ERROR LOG CLEAR
27	01	UDP OPEN REQUEST
	02	UDP RECEIVE REQUEST
	03	UDP SEND REQUEST
	04	UDP CLOSE REQUEST
	10	PASSIVE TCP OPEN REQUEST
	11	ACTIVE TCP OPEN REQUEST
	12	TCP RECEIVE REQUEST
	13	TCP SEND REQUEST
	14	TCP CLOSE REQUEST
	20	PING
	50	IP ADDRESS TABLE WRITE
	60	IP ADDRESS TABLE READ
	61	IP ROUTER TABLE READ
	62	PROTOCOL STATUS READ
	63	MEMORY STATUS READ
	64	SOCKET STATUS READ
	65	ADDRESS INFORMATION READ