April 22, 2002

OPENNET CONTROLLER
USER'S MANUAL

COMMUNICATION PROTOCOL

idec

ONC COMMUNICATION PROTOCOL

Communication Command List

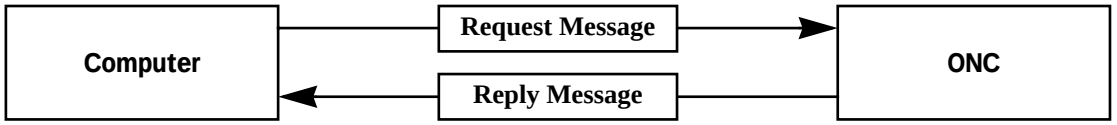
All communication commands available for the OpenNet Controller (ONC) are listed in the table below. Some of the commands are the same as for the MICRO^3 and MICRO^{3}C, with increased operands and operand number ranges.

Note: When timer/counter preset or current values are read out from the ONC using the read N bytes command, the result is different from that read from the MICRO^3 because the ONC has different timer and counter internal codes to enable 16-bit timers and counters. The ONC has new commands for timers and counters; Read Timer Information and Read Counter Information.

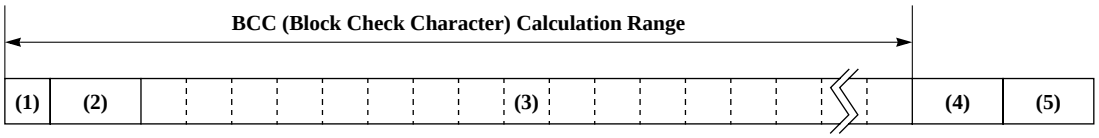
Communication Procedure

The computer and ONC CPU module communicate data by sending and receiving communication messages, which consist of request messages and reply messages. The request message is sent from the computer to write data to, read or clear data from the ONC. The reply message is sent from the ONC in response to the request message from the computer.

Communication is always initiated by the computer by sending a request message to the ONC, which then returns a reply message to the computer. The ONC cannot initiate communication in the computer link system. The ONC can initiate communication using the user communication function.



Message Format

<table>
<tr><td>Communi-cation Message</td><td colspan="5"> 
</td></tr>
<tr><td rowspan="2">(1)</td><td rowspan="2">Communication control character (1 byte)</td><td rowspan="2">Message start character</td><td>ENQ (05h)</td><td>Enquiry</td><td>Request message</td></tr>
<tr><td>ACK (06h)
NAK (15h)</td><td>Acknowledge
Negative acknowledge</td><td>Reply message</td></tr>
<tr><td rowspan="3">(2)</td><td rowspan="3">Communication device number (2 bytes)</td><td rowspan="2">Device number to send request to</td><td>00 (0) through 1F (31)</td><td colspan="2">Designates ONC device number (hardware switch, FUN16, or FUN17) to which the computer sends a request message in the 1:N communication computer link system.</td></tr>
<tr><td>FF (255)</td><td colspan="2">Used in the 1:1 communication computer link system. ONC of any device number receives request message.</td></tr>
<tr><td>Device number to send reply from</td><td>00 (0) through 1F (31)</td><td colspan="2">Indicates the device number (hardware switch, FUN16, or FUN17) of ONC which returns the reply message.</td></tr>
<tr><td>(3)</td><td>Data (variable length)</td><td>Communication command, data type, etc.</td><td colspan="3">Depends on each command.
See “Request Messages” on page ONC Protocol-3.
See “Reply Messages” on page ONC Protocol-5.</td></tr>
<tr><td>(4)</td><td>BCC (2 bytes)</td><td>Block check character</td><td colspan="3">Exclusive OR (XOR) of the BCC calculation range.</td></tr>
<tr><td rowspan="2">(5)</td><td rowspan="2">Terminator (1 or 2 bytes)</td><td rowspan="2">Message end code</td><td>CR (0Dh)</td><td colspan="2">Default</td></tr>
<tr><td>CR (0Dh) + LF (0Ah)</td><td colspan="2">Selected using FUN15 (data link port communication mode setting), FUN16 or FUN17 (RS232C port communication mode setting)</td></tr>
</table>

Request Messages

Like the MICRO^3, request messages are available in request message 1 and request message 2 with different data structures.

Request Message 1

Request message 1 is a command message to be sent from the computer to the ONC, containing a command. The data type code included in the request message determines the function. The data structure of request message 1 is shown below. In addition to data types used for the MICRO^3 and MICRO^{3}C, L (link register), O and o (link relay) are available.

Request Message 1	ENQ	05h	Device	(1)	(2)	(3)	(4)	BCC	Terminator

(1)	Continuation (1 byte)	0 (30h)	Discontinued (no message follows)					
		1 (31h)	Continued (another message follows)					
(2)	Command (1 byte)	W (57h)	Write data to ONC					
		R (52h)	Read data from ONC					
		C (43h)	Clear data from ONC					
(3)	Data type (1 byte)	X (58h)	Input	N-byte designation	t (74h)	Timer (current value)	N-byte designation	
		Y (59h)	Output		c (63h)	Counter (current value)		
		M (4Dh)	Internal relay		O (4Fh)	Link relay		
		R (52h)	Shift register		x (78h)	Input	1-bit designation	
		T (54h)	Timer (preset value)		y (79h)	Output		
		C (43h)	Counter (preset value)		m (6Dh)	Internal relay		
		D (44h)	Data register		r (72h)	Shift register		
		L (4Ch)	Link register		o (6Fh)	Link relay		
		A (41h)	User communication status 2 (read)					
		B (42h)	User communication status 1 (read)					
		E (45h)	Error code (read/clear)					
		F (46h)	FUN area settings (read)					
		G (47h)	User communication receive buffer 1 (read)					
		H (48h)	Communication mode (read)					
		I (49h)	User communication receive buffer 2 (read) Link formatting sequence (clear)					
		K (4Bh)	Scan time (read)					
		N (4Eh)	PLC system program version (read)					
		P (50h)	User program in ASCII format (read/write)					
		S (53h)	PLC operating status (read)					
		U (55h)	High-speed counter preset and current values (read)					
		V (56h)	User program protection					
		W (57h)	Calendar/clock (read/write)					
		Z (5Ah)	System reset (clear)					
		[(5Bh)	Word operands for monitor (read/write)					
		] (5Dh)	Random words (read)					
		_ (5Fh)	Timer information (read)					
		` (60h)	Counter information (read)					
		a (61h)	Timer preset value change status (read)					
		b (62h)	Counter preset value change status (read)					
		g (67h)	User communication transmit buffer 1 (read)					
		i (69h)	User communication transmit buffer 2 (read)					
		p (70h)	User program in binary format (read/write)					
(4)	Data (variable length)	Data (depends on command and data type)						

(1) "Continued" is used in request message 1 for writing the user program to inform the ONC that another request message will be sent successively. In all other request messages, "discontinued" is used. When "continued" is specified, the computer sends a request message, receives a reply message, and sends another request message.

(2) The command code is available in three types; write data, read data, and clear data.

(3) The data type code selects an operand or function. Upper- and lower-case characters have different functions.

(4) The data specifies the operand number, the quantity of bytes of the data for reading or writing, etc. depending on the command and data type.

Request Message 2

Request message 2 is a command message used for writing and reading user programs. The data structure of request message 2 is shown below:

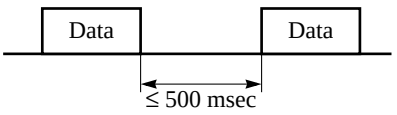
Request Message 2	**ENQ** 05h | Device | (1) | (2) | BCC | Terminator		
(1)	Continuation (1 byte)	0 (30h)	Discontinued (no message follows)
(2)	Data (variable length)	User program (write user program)	
	Data (1 byte)	R (52h)	Read user program

(1) "Discontinued" is used for both writing and reading user programs to inform the ONC that no request message will be sent successively.

(2) The data length is variable for writing user programs and is 1-byte long ("R") for reading user programs.

Receive Timeout

When a request message contains an interval of 500 msec or more between one-byte character data and the next one-byte character data, the ONC understands that the communication is canceled and does not return a reply message.



When the interval is 500 msec or more, extend the receive timeout value using FUN15, FUN16, and FUN17 (communication mode setting). The receive timeout can be selected between 10 and 2550 msec in 10-msec increments. To enable the optional communication mode, turn on the mode selection input designated by FUN15, FUN16, and FUN17.

Reply Messages

Reply messages are available in ACK reply message and NAK reply message with different data structures.

ACK Reply Message

The ACK reply message is a reply or response to the request message and is sent from the ONC to the computer when communication is completed normally.

<table>
<tr><td>ACK Reply Message</td><td colspan="5">ACK
06h | Device | (1) | (2) | BCC | Terminator</td></tr>
<tr><td rowspan="3">(1)</td><td rowspan="3">Command (1 byte)</td><td>0 (30h)</td><td>OK: Discontinued</td><td colspan="2">All communication is completed normally (end of processing).</td></tr>
<tr><td>1 (31h)</td><td>OK: Continued</td><td colspan="2">Communication in reply to request is completed normally and another reply message follows when reading a user program.</td></tr>
<tr><td>2 (32h)</td><td>NG: Error</td><td colspan="2">Communication device number, command, data type, data, or continuation code is not within the range supported by the ONC or does not match its status. When this error occurs, communication is halted without regard to the continuation code.</td></tr>
<tr><td rowspan="3">(2)</td><td rowspan="3">Data (variable length)</td><td rowspan="3">0 (30h) to 9 (39h)
or
A (41h) to F (46h)</td><td>OK reply</td><td>When request command is W or C</td><td>No data exists. (0 byte)</td></tr>
<tr><td></td><td>When request command is R</td><td>The data length depends on the request command (variable length).</td></tr>
<tr><td>NG reply</td><td colspan="2">NG code (2 bytes)
<table>
<tr><th>NG Code</th><th>Error</th><th>Cause</th></tr>
<tr><td>01</td><td>Program size error</td><td>Improper write/read program size</td></tr>
<tr><td>02</td><td>Protect error</td><td>Protected against write/read in ONC</td></tr>
<tr><td>03</td><td>RUN error</td><td>Writing user program is attempted while ONC is running</td></tr>
<tr><td>04</td><td>CRC error</td><td>User program CRC code does not match</td></tr>
<tr><td>05</td><td>Protect code error</td><td>Protect code in the request message does not match that set in FUN27. Attempt was made to enable protection on a protected user program.</td></tr>
<tr><td>06</td><td>Data range error</td><td>Invalid data range designated</td></tr>
<tr><td>07</td><td>Timer/counter preset value change error</td><td>Preset value change is attempted to timer or counter with preset value designated by data register</td></tr>
<tr><td>08</td><td>Calendar/clock data error</td><td>Invalid value written to calendar/clock</td></tr>
<tr><td>09</td><td>Data clear error</td><td>Designated data cannot be cleared</td></tr>
<tr><td>10</td><td>Data error</td><td>Invalid data other than 0 (30h) - 9 (39h) or A (41h) - F (46h)</td></tr>
<tr><td>11</td><td>Setting error</td><td>Incorrect setting for user communication</td></tr>
<tr><td>12</td><td>Operand write error</td><td>Too many write commands in one scan. Only three write commands are accepted in one scan.</td></tr>
</table></td></tr>
</table>

(1) The command code indicates whether the request command is completed normally or not and also whether another reply message will be sent successively.

When reading a user program from the ONC, reply message 1 is returned in response to request message 1 and reply message 2 is returned in response to request message 2. Reply message 1 contains command 1 (OK: continued) to inform the computer that another reply message follows. All other reply messages contain command 0 (OK: discontinued) to indicate that no reply message follows when communication is completed normally.

(2) When an OK reply is returned in response to request command R (read data), the read data is included in this place. When an NG reply is returned, the cause of error exists in the ONC. See page ONC Protocol-59.

NAK Reply Message

When an error is found during communication, a NAK reply message is sent from the ONC to the computer.

NAK Reply Message

NAK

15h	Device	(1)	(2)	BCC	Termi-nator

(1)	Command	0 (no meaning): dummy data for consistent communication format
(2)	Communication error code (2 bytes)	Depending on the communication error, an error code is set in this place.

Error Code	Error Type	Error Contents
00	BCC error	Appended BCC code does not match BCC calculated value of received data.
01	Frame error	Quantity of received bits differs from the preset value (stop bit is 0 for example).
02	Data send/receive error	Parity error or overrun error occurred.
03	Command error	Unsupported request message is received.
04	Procedure/data quantity error	Received request message does not match the expected data (including quantity of data).

(1) The command code in the NAK reply message is always 0.

(2) The next two bytes indicate the communication error code.

Write User Program in ASCII Format

The user program can be written from a computer to the ONC CPU module. When transferring a user program through modem, this command is recommended to transfer the user program in ASCII format because modems understand ASCII codes.

When writing a user program from a computer, two request messages must be sent to the ONC.
Send request message 1 first. After confirming that the returned reply message is an OK reply, send request message 2.

Request Messages (Write User Program in ASCII Format)

Request Message 1

05h	** **	31h	57h	50h	** ** ** **	** **	0Dh
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)

(1)	**Communication control character**	1 byte	ENQ (05h)	Enquiry
(2)	**Communication device number**	2 bytes	00 - 1F FF	Device number 0 through 31 Device number 255 (all devices)
(3)	**Continuation**	1 byte	1 (31h)	Continued
(4)	**Command**	1 byte	W (57h)	Write data
(5)	**Data type**	1 byte	P (50h)	User program in ASCII format
(6)	**Program capacity (Note)**	4 bytes	0AD0 12A0 : FD00	1K words (2000 + 768 bytes) 2K words (4000 + 768 bytes) : (1K-word increments) 32K words (64000 + 768 bytes)
(7)	**BCC**	2 bytes	00 - 7F	Block check character
(8)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

Note: User program includes data of rung comments and tag comments plus FUN area settings. The FUN area occupies 768 bytes.

Request Message 2

05h	** **	30h	** ** ** ** ** ** **	**	** **	0Dh
(1)	(2)	(3)	(4)		(5)	(6)

(1)	**Communication control character**	1 byte	ENQ (05h)	Enquiry
(2)	**Communication device number**	2 bytes	00 - 1F FF	Device number 0 through 31 Device number 255 (all devices)
(3)	**Continuation**	1 byte	0 (30h)	Discontinued
(4)	**User program**	Variable length 129,536 bytes max.	0 (30h) - 9 (39h) A (41h) - F (46h)	User program (ASCII code file)
(5)	**BCC**	2 bytes	00 - 7F	Block check character
(6)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

Note: The user program must be stored in a file of the ASCII code format. Ladder program files (.LDR) cannot be sent to the ONC using this request message.

Reply Messages (Write User Program in ASCII Format)

OK Reply (Reply to Request Messages 1 and 2)

06h	** **	30h	** **	0Dh
(1)	(2)	(3)	(4)	(5)

(1)	**Communication control character**	1 byte	ACK (06h)	Acknowledge
(2)	**Communication device number**	2 bytes	00 - 1F	Device number 0 through 31
(3)	**Command**	1 byte	0 (30h)	OK: Discontinued
(4)	**BCC**	2 bytes	00 - 7F	Block check character
(5)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

NG Reply (Reply to Request Message 1)

06h	** **	32h	30h 3*h	** **	0Dh
(1)	(2)	(3)	(4)	(5)	(6)

(1)	**Communication control character**	1 byte	ACK (06h)	Acknowledge
(2)	**Communication device number**	2 bytes	00 - 1F	Device number 0 through 31
(3)	**Command**	1 byte	2 (32h)	NG
(4)	**NG code**	2 bytes	01 (30h 31h) 02 (30h 32h) 03 (30h 33h) 04 (30h 34h)	Program capacity error Protect error RUN error CRC error
(5)	**BCC**	2 bytes	00 - 7F	Block check character
(6)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

Note: NG reply never occurs in response to reply message 2.

Write User Program in Binary Format

The user program can be written from a computer to the ONC CPU module. This command can send a user program faster than the Write User Program in ASCII format command.

When writing a user program from a computer, two request messages must be sent to the ONC.
Send request message 1 first. After confirming that the returned reply message is an OK reply, send request message 2.

Request Messages (Write User Program in Binary Format)

Request Message 1

05h	**	**	31h	57h	70h	**	**	**	**	**	**	0Dh
(1)	(2)		(3)	(4)	(5)	(6)				(7)		(8)

(1)	**Communication control character**	1 byte	ENQ (05h)	Enquiry
(2)	**Communication device number**	2 bytes	00 - 1F FF	Device number 0 through 31 Device number 255 (all devices)
(3)	**Continuation**	1 byte	1 (31h)	Continued
(4)	**Command**	1 byte	W (57h)	Write data
(5)	**Data type**	1 byte	p (70h)	User program in binary format
(6)	**Program capacity (Note)**	4 bytes	0AD0 12A0 : FD00	1K words (2000 + 768 bytes) 2K words (4000 + 768 bytes) : (1K-word increments) 32K words (64000 + 768 bytes)
(7)	**BCC**	2 bytes	00 - 7F	Block check character
(8)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

Note: User program includes data of rung comments and tag comments plus FUN area settings. The FUN area occupies 768 bytes.

Request Message 2

05h	**	**	30h	**	**	**	**	**	**	**	**	**	**	0Dh
(1)	(2)		(3)				(4)					(5)		(6)

(1)	**Communication control character**	1 byte	ENQ (05h)	Enquiry
(2)	**Communication device number**	2 bytes	00 - 1F FF	Device number 0 through 31 Device number 255 (all devices)
(3)	**Continuation**	1 byte	0 (30h)	Discontinued
(4)	**User program**	Variable length 64,768 bytes max.	(00h) - (FFh)	User program (binary code file)
(5)	**BCC**	2 bytes	00 - FF	Block check character
(6)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

Note: The user program must be stored in a file of the binary code format. Ladder program files (.LDR) cannot be sent to the ONC using this request message.

Reply Messages (Write User Program in Binary Format)

OK Reply (Reply to Request Messages 1 and 2)

06h	** **	30h	** **	0Dh
(1)	(2)	(3)	(4)	(5)

(1)	**Communication control character**	1 byte	ACK (06h)	Acknowledge
(2)	**Communication device number**	2 bytes	00 - 1F	Device number 0 through 31
(3)	**Command**	1 byte	0 (30h)	OK: Discontinued
(4)	**BCC**	2 bytes	00 - 7F	Block check character
(5)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

NG Reply (Reply to Request Message 1)

06h	** **	32h	30h 3*h	** **	0Dh
(1)	(2)	(3)	(4)	(5)	(6)

(1)	**Communication control character**	1 byte	ACK (06h)	Acknowledge
(2)	**Communication device number**	2 bytes	00 - 1F	Device number 0 through 31
(3)	**Command**	1 byte	2 (32h)	NG
(4)	**NG code**	2 bytes	01 (30h 31h) 02 (30h 32h) 03 (30h 33h) 04 (30h 34h)	Program capacity error Protect error RUN error CRC error
(5)	**BCC**	2 bytes	00 - 7F	Block check character
(6)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

Note: NG reply never occurs in response to reply message 2.

Read User Program in ASCII Format

The user program can be read from the ONC CPU module to a computer.

When reading a user program to a computer, two request messages must be sent from the computer to the ONC. Send request message 1 first. After confirming that the returned reply message is an OK reply, send request message 2.

Specify a value larger than the user program capacity selected in the ONC in place of the program capacity in request message 1. Reserve a buffer in the computer to store the data of the specified program capacity temporarily.

Request Messages (Read User Program in ASCII Format)

Request Message 1

05h	**	**	30h	52h	50h	**	**	**	**	**	**	0Dh
(1)	(2)		(3)	(4)	(5)		(6)			(7)		(8)

(1)	**Communication control character**	1 byte	ENQ (05h)	Enquiry
(2)	**Communication device number**	2 bytes	00 - 1F FF	Device number 0 through 31 Device number 255 (all devices)
(3)	**Continuation**	1 byte	0 (30h)	Dummy (no meaning)
(4)	**Command**	1 byte	R (52h)	Read data
(5)	**Data type**	1 byte	P (50h)	User program in ASCII format
(6)	**Program capacity**	4 bytes	0AD0 12A0 : FD00	1K words (2000 + 768 bytes) 2K words (4000 + 768 bytes) : (1K-word increments) 32K words (64000 + 768 bytes)
			FFFF	Entire user program in the ONC
(7)	**BCC**	2 bytes	00 - 7F	Block check character
(8)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

Request Message 2

05h	**	**	30h	52h	**	**	0Dh
(1)	(2)		(3)	(4)	(5)		(6)

(1)	**Communication control character**	1 byte	ENQ (05h)	Enquiry
(2)	**Communication device number**	2 bytes	00 - 1F FF	Device number 0 through 31 Device number 255 (all devices)
(3)	**Continuation**	1 byte	0 (30h)	Dummy (no meaning)
(4)	**Command**	1 byte	R (52h)	Read data
(5)	**BCC**	2 bytes	00 - 7F	Block check character
(6)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

Reply Messages (Read User Program in ASCII Format)

OK Reply

- **Reply Message 1**

06h	**	**	31h	**	**	**	**	**	**	0Dh
(1)	(2)		(3)		(4)			(5)		(6)

(1)	**Communication control character**	1 byte	ACK (06h)	Acknowledge
(2)	**Communication device number**	2 bytes	00 - 1F	Device number 0 through 31
(3)	**Command**	1 byte	1 (31h)	OK: Continued
(4)	**Program capacity**	4 bytes	0AD0 12A0 : FD00	1K words (2000 + 768 bytes) 2K words (4000 + 768 bytes) : (1K-word increments) 32K words (64000 + 768 bytes)
(5)	**BCC**	2 bytes	00 - 7F	Block check character
(6)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

Note: User program includes comments and FUN area settings. FUN area occupies 768 bytes.

- **Reply Message 2**

06h	**	**	30h	**	**	**	**	**	**	**	**	**	**	0Dh
(1)	(2)		(3)				(4)						(5)	(6)

(1)	**Communication control character**	1 byte	ACK (06h)	Acknowledge
(2)	**Communication device number**	2 bytes	00 - 1F	Device number 0 through 31
(3)	**Command**	1 byte	0 (30h)	OK: Discontinued
(4)	**User program**	Variable length 129,536 bytes max.	0 (30h) - 9 (39h) A (41h) - F (46h)	User program (ASCII code file)
(5)	**BCC**	2 bytes	00 - 7F	Block check character
(6)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

Note: The received user program is stored on the disk in the ASCII code format.

NG Reply (Reply to Request Message 1)

06h	**	**	32h	30h	3*h	**	**	0Dh
(1)	(2)		(3)	(4)		(5)		(6)

(1)	**Communication control character**	1 byte	ACK (06h)	Acknowledge
(2)	**Communication device number**	2 bytes	00 - 1F	Device number 0 through 31
(3)	**Command**	1 byte	2 (32h)	NG
(4)	**NG code**	2 bytes	01 (30h 31h) 02 (30h 32h)	Program capacity error Protect error
(5)	**BCC**	2 bytes	00 - 7F	Block check character
(6)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

Note: NG reply never occurs in response to reply message 2.

Read User Program in Binary Format

The user program can be read from the ONC CPU module to a computer.

When reading a user program to a computer, two request messages must be sent from the computer to the ONC. Send request message 1 first. After confirming that the returned reply message is an OK reply, send request message 2.

Specify a value larger than the user program capacity selected in the ONC in place of the program capacity in request message 1. Reserve a buffer in the computer to store the data of the specified program capacity temporarily.

Request Messages (Read User Program in Binary Format)

Request Message 1

05h	** **	30h	52h	70h	** ** ** **	** **	0Dh
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)

(1)	**Communication control character**	1 byte	ENQ (05h)	Enquiry
(2)	**Communication device number**	2 bytes	00 - 1F FF	Device number 0 through 31 Device number 255 (all devices)
(3)	**Continuation**	1 byte	0 (30h)	Dummy (no meaning)
(4)	**Command**	1 byte	R (52h)	Read data
(5)	**Data type**	1 byte	p (70h)	User program in binary format
(6)	**Program capacity**	4 bytes	0AD0 12A0 : FD00	1K words (2000 + 768 bytes) 2K words (4000 + 768 bytes) : (1K-word increments) 32K words (64000 + 768 bytes)
			FFFF	Entire user program in the ONC
(7)	**BCC**	2 bytes	00 - 7F	Block check character
(8)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

Request Message 2

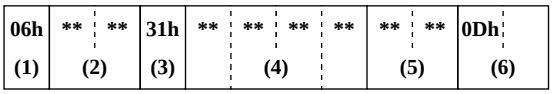
05h	** **	30h	52h	** **	0Dh
(1)	(2)	(3)	(4)	(5)	(6)

(1)	**Communication control character**	1 byte	ENQ (05h)	Enquiry
(2)	**Communication device number**	2 bytes	00 - 1F FF	Device number 0 through 31 Device number 255 (all devices)
(3)	**Continuation**	1 byte	0 (30h)	Dummy (no meaning)
(4)	**Command**	1 byte	R (52h)	Read data
(5)	**BCC**	2 bytes	00 - 7F	Block check character
(6)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

Reply Messages (Read User Program in Binary Format)

OK Reply

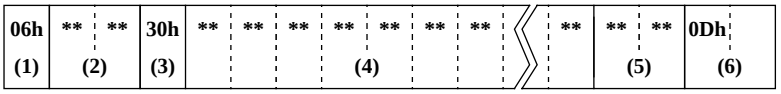
- **Reply Message 1**



(1)	**Communication control character**	1 byte	ACK (06h)	Acknowledge
(2)	**Communication device number**	2 bytes	00 - 1F	Device number 0 through 31
(3)	**Command**	1 byte	1 (31h)	OK: Continued
(4)	**Program capacity**	4 bytes	0AD0 12A0 : FD00	1K words (2000 + 768 bytes) 2K words (4000 + 768 bytes) : (1K-word increments) 32K words (64000 + 768 bytes)
(5)	**BCC**	2 bytes	00 - 7F	Block check character
(6)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

Note: User program includes comments and FUN area settings. FUN area occupies 768 bytes.

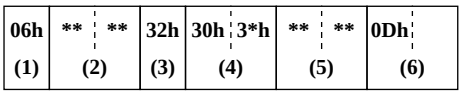
- **Reply Message 2**



(1)	**Communication control character**	1 byte	ACK (06h)	Acknowledge
(2)	**Communication device number**	2 bytes	00 - 1F	Device number 0 through 31
(3)	**Command**	1 byte	0 (30h)	OK: Discontinued
(4)	**User program**	Variable length 64,768 bytes max.	(00h) - (FFh)	User program (binary code file)
(5)	**BCC**	2 bytes	00 - 7F	Block check character
(6)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

Note: The received user program is stored on the disk in the binary code format.

NG Reply (Reply to Request Message 1)



(1)	**Communication control character**	1 byte	ACK (06h)	Acknowledge
(2)	**Communication device number**	2 bytes	00 - 1F	Device number 0 through 31
(3)	**Command**	1 byte	2 (32h)	NG
(4)	**NG code**	2 bytes	01 (30h 31h) 02 (30h 32h)	Program capacity error Protect error
(5)	**BCC**	2 bytes	00 - 7F	Block check character
(6)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

Note: NG reply never occurs in response to reply message 2.

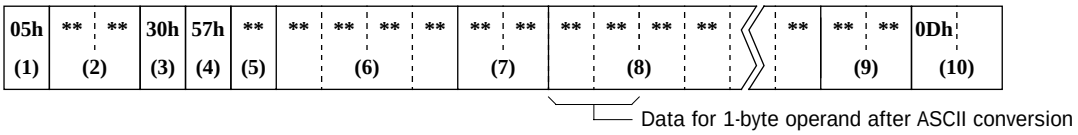
Write N Bytes

Data can be written into N-bytes of operands starting with the specified operand number in the ONC CPU module.

This command can be used to turn on or off bit operands such as inputs, outputs, internal relays, link relays, and shift register bits in units of 8 bits.

This command can also be used to change timer and counter preset values, enter data into data registers, and set calendar/clock data.

Request Message (Write N Bytes)



(1)	**Communication control character**	1 byte	ENQ (05h)	Enquiry
(2)	**Communication device number**	2 bytes	00 - 1F FF	Device number 0 through 31 Device number 255 (all devices)
(3)	**Continuation**	1 byte	0 (30h)	Discontinued
(4)	**Command**	1 byte	W (57h)	Write data
(5)	**Data type**	1 byte	See table below.	N-byte designation
(6)	**Operand number**	4 bytes	See table below.	First operand number to write to
(7)	**Data length (n)**	2 bytes	00 - C8	Byte count of data to write 200 (C8h) bytes maximum
(8)	**Data**	2n bytes $1 \leq n \leq 200$	0 (30h) - 9 (39h) A (41h) - F (46h)	Data to write
(9)	**BCC**	2 bytes	00 - 7F	Block check character
(10)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

(5) Data type code		(6) Operand number	Remarks
X (58h)	Input	0000 - 0597	The least significant digit of the operand number is an octal number (0-7). Upper digits are decimal numbers.
Y (59h)	Output	0000 - 0597	
M (4Dh)	Internal relay	0000 - 2557, 8000 - 8117	
O (4Fh)	Link relay	0000 - 8477	
R (52h)	Shift register	0000 - 0255	All four digits of the operand number are decimal numbers.
T (54h)	Timer (preset value)	0000 - 0255	
t (74h)	Timer (current value)	0000 - 0255	
C (43h)	Counter (preset value)	0000 - 0255	
c (63h)	Counter (current value)	0000 - 0255	
D (44h)	Data register	0000 - 8999	
W (57h)	Calendar/clock	0000 - 0006	
L (4Ch)	Link register	0100 - 0127, 0200 - 0227, 0300 - 0327, 0400 - 0427, 0500 - 0527, 0600 - 0627, 0700 - 0727, 1000 - 1317	The least significant digit of the operand number is an octal number (0-7). Upper digits are decimal numbers.

Operand numbers for calendar and clock are allocated as listed on the right:

Calendar/clock operand number	Data
0000	Year
0001	Month
0002	Day
0003	Day of week
0004	Hour
0005	Minute
0006	Second

When the range specified by the data type and data length is invalid, the ONC returns an NG reply.

When a data register is designated as a preset value for a timer or counter, data cannot be written into the preset value. To change the preset value, write data into the data register designated as a preset value.

Reply Messages (Write N Bytes)

OK Reply

06h	**	**	30h	**	**	0Dh
(1)	(2)		(3)	(4)		(5)

(1)	**Communication control character**	1 byte	ACK (06h)	Acknowledge
(2)	**Communication device number**	2 bytes	00 - 1F	Device number 0 through 31
(3)	**Command**	1 byte	0 (30h)	OK: Discontinued
(4)	**BCC**	2 bytes	00 - 7F	Block check character
(5)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

NG Reply

06h	**	**	32h	30h	3*h	**	**	0Dh
(1)	(2)		(3)	(4)		(5)		(6)

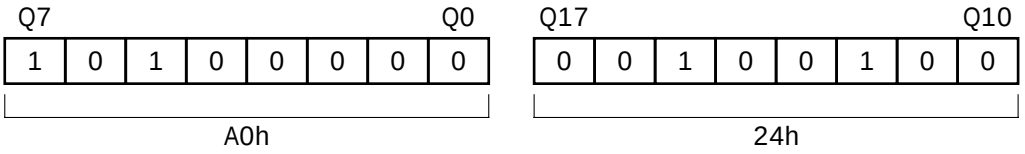
(1)	**Communication control character**	1 byte	ACK (06h)	Acknowledge
(2)	**Communication device number**	2 bytes	00 - 1F	Device number 0 through 31
(3)	**Command**	1 byte	2 (32h)	NG
(4)	**NG code**	2 bytes	06 (30h 36h) 07 (30h 37h) 08 (30h 38h)	Data range error Timer/counter preset value change error Calendar/clock data error
(5)	**BCC**	2 bytes	00 - 7F	Block check character
(6)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

Data Format in the Request Message (Write N Bytes)

X (Input), Y (Output), M (Internal Relay), O (Link Relay), and R (Shift Register)

To write ON/OFF statuses of bit operands such as inputs, outputs, internal relays, link relays, or shift registers, divide the operand numbers into 8-bit (1-byte) groups, and convert the 8-bit value into a hexadecimal value. Then, convert the hexadecimal value into ASCII codes. Include the ASCII codes in place of “Data” in the request message.

Example: To write data to outputs Q0 through Q17 to set Q5, Q7, Q12, and Q15 and reset other outputs.



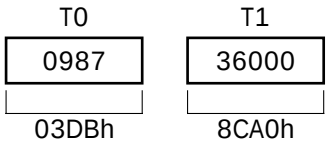
In this example, convert the hexadecimal value A024 into ASCII codes, and include the results (41h 30h 32h 34h) in the request message. Consequently, the data to send has a length of 4 bytes in the request message.

The data length to write in this example is 16 bits, or 2 (02h) bytes, of output points. So, include ASCII codes 30h 32h in place of “Data length” in the request message.

T (Timer Preset Value) and C (Counter Preset Value)

To write data into word operands such as timers and counters, convert the hexadecimal values into ASCII codes. Include the ASCII codes in place of “Data” in the request message.

Example: To write decimal 987 and 36000 to preset values for timers T0 and T1, respectively.



In this example, convert the decimal values into hexadecimal values and send data 03DB8CA0 (30h 33h 44h 42h 38h 43h 41h 30h).

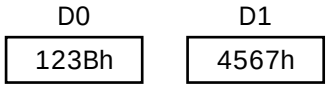
The data length of this example is 2 words, or 4 (04h) bytes. So, include ASCII codes 30h 34h in place of “Data length” in the request message.

Since the ONC uses separate memory areas for timers and counters, timer and counter preset values are written into the specified operand number in different memory areas.

D (Data Register)

To write data into word operands of data registers, convert the hexadecimal values into ASCII codes. Include the ASCII codes in place of “Data” in the request message.

Example: To write 123Bh and 4567h to data registers D0 and D1, respectively.



In this example, send data 123B4567 (31h 32h 33h 42h 34h 35h 36h 37h).

The data length of this example is 2 words, or 4 (04h) bytes. So, include ASCII codes 30h 34h in place of “Data length” in the request message.

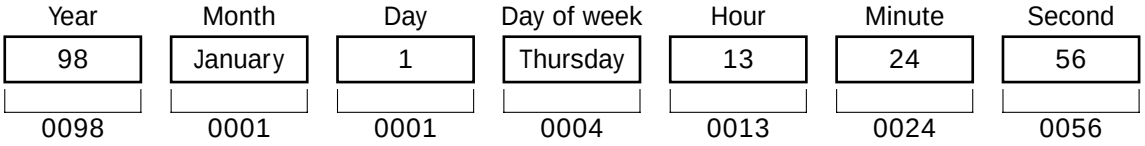
W (Calendar / Clock)

To send calendar/clock operands such as year, month, day, day of week, hour, minute, and second, write each one-word (2 bytes) data directly.

Day of week data format (0 through 6) is assigned as follows:

0	1	2	3	4	5	6
Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday

Example: To send calendar/clock data Thursday, January 1, 1998, 13 hour, 24 minutes, 56 seconds.



In this example, send data 00980001000100004001300240056 (30h 30h 39h 38h 30h 30h 30h 31h 30h 30h 30h 31h 30h 30h 30h 34h 30h 30h 31h 33h 30h 30h 32h 34h 30h 30h 35h 36h).

The data length of this example is 7 words, or 14 (0Eh) bytes. So, include ASCII codes 30h 3Eh in place of “Data length” in the request message.

Read N Bytes

Data can be read from N-bytes of operands starting with the specified operand number in the ONC CPU module.

This command can be used to monitor the ON/OFF statuses of bit operands such as inputs, outputs, internal relays, link relays, and shift register bits in units of 8 bits.

This command can also be used to monitor preset and current values of timers and counters, data of data registers, and read calendar/clock data.

Request Message (Read N Bytes)

05h	** **	30h	52h	**	** ** ** **	** **	** **	0Dh
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)

(1)	**Communication control character**	1 byte	ENQ (05h)	Enquiry
(2)	**Communication device number**	2 bytes	00 - 1F FF	Device number 0 through 31 Device number 255 (all devices)
(3)	**Continuation**	1 byte	0 (30h)	Discontinued
(4)	**Command**	1 byte	R (52h)	Read data
(5)	**Data type**	1 byte	See table below.	N-byte designation
(6)	**Operand number**	4 bytes	See table below.	First operand number to read
(7)	**Data length (n)**	2 bytes	00 - C8	Byte count of data to read 200 (C8h) bytes maximum
(8)	**BCC**	2 bytes	00 - 7F	Block check character
(9)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

(5) Data type code		(6) Operand number	Remarks
X (58h)	Input	0000 - 0597	The least significant digit of the operand number is an octal number (0-7). Upper digits are decimal numbers.
Y (59h)	Output	0000 - 0597	
M (4Dh)	Internal relay	0000 - 2557, 8000 - 8237	
O (4Fh)	Link relay	0000 - 8477	
R (52h)	Shift register	0000 - 0255	All four digits of the operand number are decimal numbers.
T (54h)	Timer (preset value)	0000 - 0255	
t (74h)	Timer (current value)	0000 - 0255	
C (43h)	Counter (preset value)	0000 - 0255	
c (63h)	Counter (current value)	0000 - 0255	
D (44h)	Data register	0000 - 8999	
W (57h)	Calendar/clock	0000 - 0006	
L (4Ch)	Link register	0100 - 0127, 0200 - 0227, 0300 - 0327, 0400 - 0427, 0500 - 0527, 0600 - 0627, 0700 - 0727, 1000 - 1317	The least significant digit of the operand number is an octal number (0-7). Upper digits are decimal numbers.

Operand numbers for calendar and clock are allocated as listed on the right:

Calendar/clock operand number	Data
0000	Year
0001	Month
0002	Day
0003	Day of week
0004	Hour
0005	Minute
0006	Second

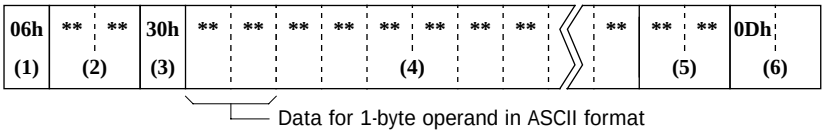
The internal relay memory area is divided into the ordinary internal relays and special internal relays. N-byte data cannot be read from the internal relay area continuing from the ordinary internal relays through special internal relays.

When the range specified by the data type and data length is invalid, the ONC returns an NG reply.

When a preset value is read from a timer or counter for which a data register is designated as a preset value, the preset value is returned as a reply, rather than the data register number.

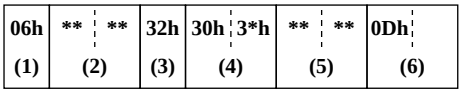
Reply Messages (Read N Bytes)

OK Reply



(1)	**Communication control character**	1 byte	ACK (06h)	Acknowledge
(2)	**Communication device number**	2 bytes	00 - 1F	Device number 0 through 31
(3)	**Command**	1 byte	0 (30h)	OK: Discontinued
(4)	**Data**	2n bytes 1 ≤ n ≤ 200	0 (30h) - 9 (39h) A (41h) - F (46h)	Read data
(5)	**BCC**	2 bytes	00 - 7F	Block check character
(6)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

NG Reply



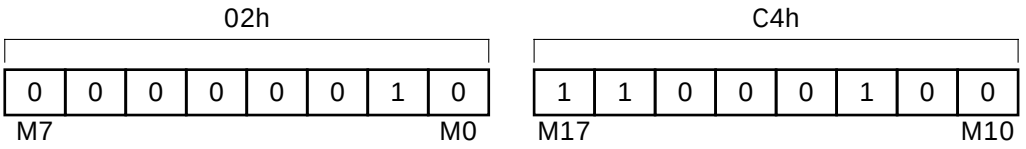
(1)	**Communication control character**	1 byte	ACK (06h)	Acknowledge
(2)	**Communication device number**	2 bytes	00 - 1F	Device number 0 through 31
(3)	**Command**	1 byte	2 (32h)	NG
(4)	**NG code**	2 bytes	06 (30h 36h) 08 (30h 38h)	Data range error Calendar/clock data error
(5)	**BCC**	2 bytes	00 - 7F	Block check character
(6)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

Data Format in the Reply Message (Read N Bytes)

X (Input), Y (Output), M (Internal Relay), O (Link Relay), and R (Shift Register)

When reading ON/OFF statuses of bit operands such as inputs, outputs, internal relays, link relays, or shift registers, the received data show the hexadecimal value of 8-bit groups.

Example: The read data is 02C4 (30h 32h 43h 34h) when reading 2 bytes starting with internal relay M0.

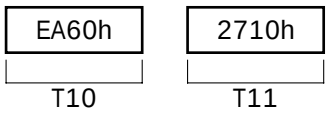


Divide the read data into one-byte (8-bit) groups. The bits where a 1 is stored are ON. In this example, internal relays M1, M12, M16, and M17 are on.

T (Timer Preset Value), t (Timer Current Value), C (Counter Preset Value), and c (Counter Current Value)

The Read N Bytes command can be used to read preset or current values of consecutive timers or counters. When reading timer/counter preset or current values, the received data show the hexadecimal values in four characters each.

Example: The read data is EA602710 (45h 41h 36h 30h 32h 37h 31h 30h) when reading 4 bytes of timer current values starting with timer T10.



Divide the received data into 4-character groups and convert the data into 4-digit hexadecimal values. In this example, the read data is shown below:

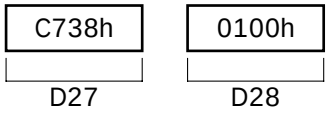
T10 = EA60h (60000 decimal)
T11 = 2710h (10000 decimal)

To read preset, current values, and timer status of timers, use the Read Timer Information command (see page ONC Protocol-48).
To read preset, current values, and counter status of counter, use the Read Counter Information command (see page ONC Protocol-50).

D (Data Register)

When reading data registers, the received data show the hexadecimal values in four characters each.

Example: The read data is C7380100 (43h 37h 33h 38h 30h 31h 30h 30h) when reading 4 bytes starting with data register D27.



Divide the received data into 4-character groups and convert the data into 4-digit hexadecimal values. In this example, the read data is shown below:

D27 = C738h (51000 decimal)
D28 = 100h (256 decimal)

W (Calendar / Clock)

Calendar/clock data are received in units of 2 bytes starting with the specified operand number 0000 (year) through 0006 (second). For operand numbers for the calendar and clock, see page ONC Protocol-18.

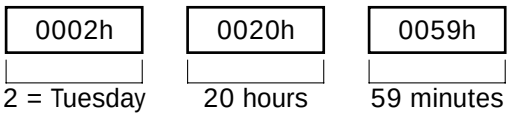
Day of week data format (0 through 6) is assigned as follows:

0	1	2	3	4	5	6
Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday

Example: The read data is 000200200059 (30h 30h 30h 32h 30h 30h 32h 30h 30h 30h 35h 39h) when reading 6 bytes (3 words) starting with operand number 0003 (day of week) for reading calendar/clock values.

Divide the received data into 4-character groups and convert the data into 4-digit hexadecimal values.

Data of three operands starting with 0003 (day of week) is read as shown on the right.



Write 1 Bit

Data can be written into 1 bit of the specified operand in the ONC CPU module, enabling to set (ON) or reset (OFF) the operand.

ONC operation can be started or stopped by setting or resetting start control special internal relay M8000 using this request message.

Request Message (Write 1 Bit)

05h	**	**	30h	57h	**	**	**	**	**	**	3*h	**	**	0Dh
(1)	(2)		(3)	(4)	(5)	(6)					(7)	(8)		(9)

(1)	**Communication control character**	1 byte	ENQ (05h)	Enquiry
(2)	**Communication device number**	2 bytes	00 - 1F FF	Device number 0 through 31 Device number 255 (all devices)
(3)	**Continuation**	1 byte	0 (30h)	Discontinued
(4)	**Command**	1 byte	W (57h)	Write data
(5)	**Data type**	1 byte	See table below.	1-bit designation
(6)	**Operand number**	4 bytes	See table below.	Operand number to write to
(7)	**ON / OFF status**	1 byte	0 (30h) 1 (31h)	OFF ON
(8)	**BCC**	2 bytes	00 - 7F	Block check character
(9)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

(5) Data type code		(6) Operand number	Remarks
x (78h)	Input	0000 - 0597	The least significant digit of the operand number is an octal number (0-7). Upper digits are decimal numbers.
y (79h)	Output	0000 - 0597	
m (6Dh)	Internal relay	0000 - 2557, 8000 - 8117	
o (6Fh)	Link relay	0000 - 8477	
r (72h)	Shift register	0000 - 0255	All four digits of the operand number are decimal numbers.

Reply Messages (Write 1 Bit)

OK Reply

06h	** **	30h	** **	0Dh
(1)	(2)	(3)	(4)	(5)

(1)	**Communication control character**	1 byte	ACK (06h)	Acknowledge
(2)	**Communication device number**	2 bytes	00 - 1F	Device number 0 through 31
(3)	**Command**	1 byte	0 (30h)	OK: Discontinued
(4)	**BCC**	2 bytes	00 - 7F	Block check character
(5)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

NG Reply

06h	** **	32h	30h 36h	** **	0Dh
(1)	(2)	(3)	(4)	(5)	(6)

(1)	**Communication control character**	1 byte	ACK (06h)	Acknowledge
(2)	**Communication device number**	2 bytes	00 - 1F	Device number 0 through 31
(3)	**Command**	1 byte	2 (32h)	NG
(4)	**NG code**	2 bytes	06 (30h 36h)	Data range error
(5)	**BCC**	2 bytes	00 - 7F	Block check character
(6)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

Read 1 Bit

Data can be read from 1 bit of the specified operand in the ONC CPU module to see if the operand is on or off.

The read 1 bit command can be used to monitor the ON/OFF status of a bit operand such as input, output, internal relay, link relay, or shift register bit.

Request Message (Read 1 Bit)

05h	**	**	30h	52h	**	**	**	**	**	**	**	0Dh
(1)	(2)		(3)	(4)	(5)	(6)				(7)		(8)

(1)	**Communication control character**	1 byte	ENQ (05h)	Enquiry
(2)	**Communication device number**	2 bytes	00 - 1F FF	Device number 0 through 31 Device number 255 (all devices)
(3)	**Continuation**	1 byte	0 (30h)	Discontinued
(4)	**Command**	1 byte	R (52h)	Read data
(5)	**Data type**	1 byte	See table below.	1-bit designation
(6)	**Operand number**	4 bytes	See table below.	Operand number to read from
(7)	**BCC**	2 bytes	00 - 7F	Block check character
(8)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

(5) Data type code		(6) Operand number	Remarks
x (78h)	Input	0000 - 0597	The least significant digit of the operand number is an octal number (0-7). Upper digits are decimal numbers.
y (79h)	Output	0000 - 0597	
m (6Dh)	Internal relay	0000 - 2557, 8000 - 8237	
o (6Fh)	Link relay	0000 - 8477	
r (72h)	Shift register	0000 - 0255	All four digits of the operand number are decimal numbers.

Reply Messages (Read 1 Bit)

OK Reply

06h	**	**	30h	3*h	**	**	0Dh
(1)	(2)		(3)	(4)	(5)		(6)

(1)	**Communication control character**	1 byte	ACK (06h)	Acknowledge
(2)	**Communication device number**	2 bytes	00 - 1F	Device number 0 through 31
(3)	**Command**	1 byte	0 (30h)	OK: Discontinued
(4)	**ON / OFF status**	1 byte	0 (30h) 1 (31h)	OFF ON
(5)	**BCC**	2 bytes	00 - 7F	Block check character
(6)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

NG Reply

06h	**	**	32h	30h	36h	**	**	0Dh
(1)	(2)		(3)	(4)		(5)		(6)

(1)	**Communication control character**	1 byte	ACK (06h)	Acknowledge
(2)	**Communication device number**	2 bytes	00 - 1F	Device number 0 through 31
(3)	**Command**	1 byte	2 (32h)	NG
(4)	**NG code**	2 bytes	06 (30h 36h)	Data range error
(5)	**BCC**	2 bytes	00 - 7F	Block check character
(6)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

Read High-speed Counter Preset and Current Values

Preset and current values of the high-speed counter can be read from the ONC base unit.

This command can be used to monitor the preset and current values of the high-speed counter.

The ONC high-speed counter is a 16-bit reversible counter, while the MICRO^3 has 32-bit high-speed counters. Consequently, high-speed counter number 0000 is used in this request message for the ONC.

In the reply message, upper 4 bytes of the preset and current values contain zeros.

Request Message (Read High-speed Counter Preset and Current Values)

05h	**	**	30h	52h	55h	30h	30h	30h	30h	**	**	0Dh
(1)	(2)		(3)	(4)	(5)	(6)				(7)		(8)

(1)	**Communication control character**	1 byte	ENQ (05h)	Enquiry
(2)	**Communication device number**	2 bytes	00 - 1F FF	Device number 0 through 31 Device number 255 (all devices)
(3)	**Continuation**	1 byte	0 (30h)	Discontinued
(4)	**Command**	1 byte	R (52h)	Read data
(5)	**Data type**	1 byte	U (55h)	High-speed counter preset and current values
(6)	**High-speed counter number**	4 bytes	0000	HSC
(7)	**BCC**	2 bytes	00 - 7F	Block check character
(8)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

Reply Messages (Read High-speed Counter Preset and Current Values)

OK Reply

06h	**	**	30h	30h	30h	30h	30h	30h	**	**	**	**	30h	30h	30h	30h	**	**	**	**	**	**	0Dh
(1)	(2)		(3)	(4)				(5)								(6)					(7)		(8)

(1)	**Communication control character**	1 byte	ACK (06h)	Acknowledge
(2)	**Communication device number**	2 bytes	00 - 1F	Device number 0 through 31
(3)	**Command**	1 byte	0 (30h)	OK: Discontinued
(4)	**Preset value operand type**	1 byte	0 (30h)	Data register (variable)
(5)	**Preset value data**	8 bytes	00000000 - 0000FFFF	HSC preset value stored in special data register D8047
(6)	**Current value data**	8 bytes	00000000 - 0000FFFF	HSC current value (Note)
(7)	**BCC**	2 bytes	00 - 7F	Block check character
(8)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

Since the ONC high-speed counter value ranges from 0 through 65535, the upper 4 bytes of “Preset value data” and “Current value data” always contain 0000 (30h 30h 30h 30h).

The preset value is read from special data register D8047 used for storing the high-speed counter preset value.

Note: The instantaneous current value of the high-speed counter is read by this command. The high-speed counter current value is stored in special data register D8045, and the data in this data register is updated at the END processing of the user program.

NG Reply

06h	**	**	32h	30h	36h	**	**	0Dh
(1)	(2)		(3)	(4)		(5)		(6)

(1)	**Communication control character**	1 byte	ACK (06h)	Acknowledge
(2)	**Communication device number**	2 bytes	00 - 1F	Device number 0 through 31
(3)	**Command**	1 byte	2 (32h)	NG
(4)	**NG code**	2 bytes	06 (30h 36h)	Data range error
(5)	**BCC**	2 bytes	00 - 7F	Block check character
(6)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

Read Error Code

Error codes can be read from the ONC CPU module.

Request Message (Read Error Code)

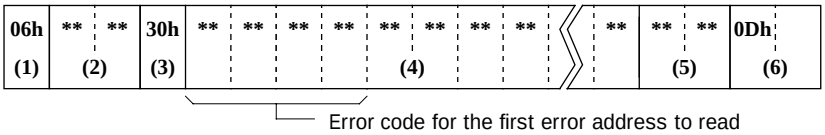
05h	**	**	30h	52h	45h	30h	30h	30h	3*h	30h	**	**	**	0Dh
(1)	(2)		(3)	(4)	(5)		(6)			(7)		(8)		(9)

(1)	**Communication control character**	1 byte	ENQ (05h)	Enquiry
(2)	**Communication device number**	2 bytes	00 - 1F FF	Device number 0 through 31 Device number 255 (all devices)
(3)	**Continuation**	1 byte	0 (30h)	Discontinued
(4)	**Command**	1 byte	R (52h)	Read data
(5)	**Data type**	1 byte	E (45h)	Error code
(6)	**Error address**	4 bytes	See table below.	First error address to read
(7)	**Data length (n)**	2 bytes	02 - 0C	2 bytes per error address 12 (0Ch) bytes maximum
(8)	**BCC**	2 bytes	00 - 7F	Block check character
(9)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

(6) Error address	Error details
0000	General error code
0001	User program syntax error: Type code
0002	User program syntax error: Address code
0003	Advanced instruction syntax error
0004	User program execution error
0005	Link communication error

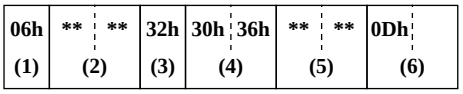
Reply Messages (Read Error Code)

OK Reply



(1)	**Communication control character**	1 byte	ACK (06h)	Acknowledge
(2)	**Communication device number**	2 bytes	00 - 1F	Device number 0 through 31
(3)	**Command**	1 byte	0 (30h)	OK: Discontinued
(4)	**Data**	4n bytes (1 ≤ n ≤ 6)	0 (30h) - 9 (39h) A (41h) - F (46h)	Error code (4 bytes per error address)
(5)	**BCC**	2 bytes	00 - 7F	Block check character
(6)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

NG Reply



(1)	**Communication control character**	1 byte	ACK (06h)	Acknowledge
(2)	**Communication device number**	2 bytes	00 - 1F	Device number 0 through 31
(3)	**Command**	1 byte	2 (32h)	NG
(4)	**NG code**	2 bytes	06 (30h 36h)	Data range error (error address)
(5)	**BCC**	2 bytes	00 - 7F	Block check character
(6)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

Data Format in the Reply Message (Read Error Code)

When reading error codes, the received data show the hexadecimal values in four characters each.

Example: The read data is 0080 0006 0004 0001 0000 0000 when reading 12 (0Ch) bytes starting with error address 0000.

Divide the received data into 4-character groups and convert the data into 4-digit hexadecimal values.

Error address:	0000	0001	0002	0003	0004	0005
	0080h	0006h	0004h	0001h	0000h	0000h

In this example, the read data is shown below:

0000	General error code	80h
0001	User program syntax error: Type code	6h
0002	User program syntax error: Address code	4h
0003	Advanced instruction syntax error	1h
0004	User program execution error	0h
0005	Link communication error	0h

The above data means that user program syntax error (error code 80h) is found.
The type code of the user program syntax error is 6h (invalid data for advanced instruction).
The address code of the user program syntax error is 4h (address 4).
The advanced instruction syntax error code is 1h which means that the internal allocation number of the operand is invalid.

For details of error codes, see the next page.

Error Codes

General Error Code (GeneralError)

Error Code	Bit Position	Error Status
1h	bit 0	Power failure
2h	bit 1	Watch dog timer error
4h	bit 2	Data link connection error
8h	bit 3	User program sum check error
10h	bit 4	Timer/counter preset value sum check error
20h	bit 5	User program sum check error
40h	bit 6	Keep data sum check error
80h	bit 7	User program syntax error
100h	bit 8	User program writing error
200h	bit 9	Protect output overload error
400h	bit 10	Calendar/clock error
800h	bit 11	I/O bus error (error indicator flashes)
1000h	bit 12	IBS master access error
——	bits 13 to 15	Unused

User Program Syntax Error: Type Code (CERCOD)

Type Code	Error Details
1	Invalid value for FUN settings
2	Invalid opcode for basic instruction
3	Invalid operand for basic instruction
4	Invalid data for TIM, CNT, CC, TC, DC, SFR, or SFRN
5	Invalid opcode (instruction number or data type) for advanced instruction
6	Invalid data (such as repeat designation) for advanced instruction other than opcodes
7	Invalid operand for advanced instruction
8	Invalid repeated usage of advanced instructions, such as DISP, DGRD, and CDISP
9	User program capacity over

User Program Syntax Error: Address Code (CERADR)

Type Code	Address Code	Address Code Description
1	**0 to 27 (0000 to 001Bh)**	FUN number of the invalid value
2 to 8	**0 to 32000 (0000 to 7D00h)**	Address of the incorrect program
9	**—**	—

Advanced Instruction Syntax Error (CERCODB)

When a user program syntax error (general error code 80h — error related with advanced instruction) is indicated with type code 5, 6, or 7, the detailed information can be viewed at this error address.

Error Code	Error Details
1	The internal allocation number of the operand is invalid.
2	Input or read-only special internal relay is designated as a destination.
3	Quantity of repeat cycles is set to 101 or more. Quantity of shift bits is set to 16 (data type W) or 32 (data type D) for shift/rotate instructions or more.
4	Advanced instruction (LABEL) which is not allowed for repeat usage is programmed more than once.

User Program Execution Error (UEJIK) — Special internal relay M8004 goes on

This error indicates that invalid data is found during execution of a user program. When this error occurs, special internal relay M8004 (user program execution error) is also turned on. The detailed information of this error can be viewed at this error address. When this error occurs, program operation and all output statuses are maintained.

User Program Execution Error Code (D8006)	Error Details
1	Source/destination operand is out of range
2	MUL result is out of data type range
3	DIV result is out of data type range, or division by 0
4	BCDLS has S1 or S1+1 exceeding 9999
5	HTOB(W) has S1 exceeding 9999 HTOB(D) has S1(S1+1) exceeding 99999999
6	BTOH has any digit of S1(S+1) exceeding 9
7	HTOA/ATOH/BTOA/ATOB has quantity of digits to convert out of range
8	ATOH/ATOB has non-ASCII data for S1 through S1+4
9	WKCMP has S1, S2, and S3 exceeding the valid range S1: 0 through 127 S2: Hour data 0 through 23, minute data 0 through 59 S3: 0 through 2
10	WKTBL has S1 through Sn out of range
11	DGRD data exceeds 65535 with BCD5 digits selected
12	CVXTY/CVYTX is executed without matching XYFS
13	CVXTY/CVYTX has S2 exceeding the value specified in XYFS
14	Label in LJMP/LCAL/DJNZ is not found
15	TXD/RXD is executed while the RS232C port 1 or 2 is *not* set to user communication mode

Data Link Communication Error (ExpansionComError) — Special internal relay M8005 goes on

This error indicates a communication error in the data link system. When this error occurs, special internal relay M8005 (data link communication error) is also turned on. The detailed information of this error can be viewed at this error address. When this error occurs, program operation and all output statuses are maintained.

Error Code	Bit Position	Error Details
1h	bit 0	Overrun error (Data is received when the receive data registers are full.)
2h	bit 1	Framing error (Failure to detect start or stop bit)
4h	bit 2	Parity error (An error was found by the parity check.)
8h	bit 3	Receive timeout (Line disconnection)
10h	bit 4	BCC (block check character) error (Disparity with data received up to BCC.)
20h	bit 5	Retry cycle over (Error occurred in all 3 trials of communication.)
40h	bit 6	I/O definition quantity error

When more than one error is detected in the data link system, the total of error codes is indicated. For example, when framing error (error code 2h) and BCC error (error code 10h) are found, error code 12 is displayed.

Clear Operand Data

All data of selected operand area or all operands can be cleared from the ONC CPU module.

Request Message (Clear Operand Data)

05h	**	**	30h	43h	**	**	**	0Dh
(1)	(2)		(3)	(4)	(5)	(6)		(7)

(1)	**Communication control character**	1 byte	ENQ (05h)	Enquiry
(2)	**Communication device number**	2 bytes	00 - 1F FF	Device number 0 through 31 Device number 255 (all devices)
(3)	**Continuation**	1 byte	0 (30h)	Discontinued
(4)	**Command**	1 byte	C (43h)	Clear data
(5)	**Data type**	1 byte	See table below.	
(6)	**BCC**	2 bytes	00 - 7F	Block check character
(7)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

(5) Data type	Data to clear	(5) Data type	Data to clear
X (58h)	Input	**C (43h)**	Counter (preset value)
Y (59h)	Output	**c (63h)**	Counter (current value)
M (4Dh)	Internal relay	**D (44h)**	Data register
O (4Fh)	Link relay	**L (4Ch)**	Link register
R (52h)	Shift register	**E (45h)**	Error code
T (54h)	Timer (preset value)	**Z (5Ah)**	System reset (all operands)
t (74h)	Timer (current value)	**I (49h)**	Link formatting sequence

When the timer preset value (T) or counter preset value (C) is cleared, the changed preset values in the ONC CPU module RAM are cleared and the original preset values are restored.

When the system reset is executed with Z (5Ah) specified for “Data type” in the request message, data is cleared from all operand areas of inputs (X), outputs (Y), internal relays (M), link relays (O), shift registers (R), timer current values (t), counter current values (c), data registers (D), and link registers (L).

When the link formatting sequence (I) is executed, the data link terminal connection data is updated. This function is the same as turning on special internal relay M8007 (data link communication initialize flag).

Reply Messages (Clear Operand Data)

OK Reply

06h	** **	30h	** **	0Dh
(1)	(2)	(3)	(4)	(5)

(1)	**Communication control character**	1 byte	ACK (06h)	Acknowledge
(2)	**Communication device number**	2 bytes	00 - 1F	Device number 0 through 31
(3)	**Command**	1 byte	0 (30h)	OK: Discontinued
(4)	**BCC**	2 bytes	00 - 7F	Block check character
(5)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

NG Reply

06h	** **	32h	30h 39h	** **	0Dh
(1)	(2)	(3)	(4)	(5)	(6)

(1)	**Communication control character**	1 byte	ACK (06h)	Acknowledge
(2)	**Communication device number**	2 bytes	00 - 1F	Device number 0 through 31
(3)	**Command**	1 byte	2 (32h)	NG
(4)	**NG code**	2 bytes	09 (30h 39h)	Data clear error
(5)	**BCC**	2 bytes	00 - 7F	Block check character
(6)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

Enable / Disable User Program Protection

The user program in the ONC CPU module can be protected from reading, writing, or both using FUN27 (user program protect mode and code). To enable user program protection, enter proper values for user program protect mode and user program protect code in FUN27, and transfer the user program from the PC to the ONC. Then, the user program is protected from reading, writing, or both depending on the user program protect mode specified in FUN27.

The user program protection can be temporarily canceled using a communication command. To disable the user protection, send this command including the correct protect code and protect option 0 (disable protection) to the ONC. The user program protection is disabled until the ONC is shut down or the user program protection is enabled again by this command including protect option 1 (enable protection).

Request Message (Enable / Disable User Program Protection)

05h	**	**	30h	57h	56h	**	**	**	**	**	**	**	**	**	3*h	**	**	0Dh
(1)	(2)		(3)	(4)	(5)					(6)					(7)	(8)		(9)

(1)	**Communication control character**	1 byte	ENQ (05h)	Enquiry
(2)	**Communication device number**	2 bytes	00 - 1F FF	Device number 0 through 31 Device number 255 (all devices)
(3)	**Continuation**	1 byte	0 (30h)	Discontinued
(4)	**Command**	1 byte	W (57h)	Write data
(5)	**Data type**	1 byte	V (56h)	User program protection
(6)	**Protect code**	32 bytes	0000 0000 0000 0000 0000 0000 0000 0000 to FFFF FFFF FFFF FFFF FFFF FFFF FFFF FFFF	
(7)	**Protect option**	1 bytes	0 (30h) 1 (31h)	Disable protection Enable protection
(8)	**BCC**	2 bytes	00 - 7F	Block check character
(9)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

(6) Protect code

The protect code consists of 16 ASCII characters (20h through 7Fh), and is converted to 32 bytes of ASCII codes in the request message.

(7) Protect option

Protect option 0 (disable protection): When the protect code in the request message matches that set in FUN27, the user program protection is canceled temporarily.

Protect option 1 (enable protection): When the protect code in the request message matches that set in FUN27, the user program protection is restored. The protect mode of read, write, or read/write protect depends on the setting in FUN27.

Unlike the MICRO^3 and MICRO^{3}C, this request command is primarily used for the ONC to disable the user program protection temporarily when the user program is protected by FUN27 (user program protect mode and code). In addition, this request command can also be used to restore the user program protection which has been disabled temporarily by this request command. Note that this request command cannot be used for the ONC to protect a user program which is not protected by FUN27.

Reply Messages (Enable / Disable User Program Protection)

OK Reply

06h	** **	30h	** **	0Dh
(1)	(2)	(3)	(4)	(5)

(1)	**Communication control character**	1 byte	ACK (06h)	Acknowledge
(2)	**Communication device number**	2 bytes	00 - 1F	Device number 0 through 31
(3)	**Command**	1 byte	0 (30h)	OK: Discontinued
(4)	**BCC**	2 bytes	00 - 7F	Block check character
(5)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

NG Reply

06h	** **	32h	30h 35h	** **	0Dh
(1)	(2)	(3)	(4)	(5)	(6)

(1)	**Communication control character**	1 byte	ACK (06h)	Acknowledge
(2)	**Communication device number**	2 bytes	00 - 1F	Device number 0 through 31
(3)	**Command**	1 byte	2 (32h)	NG
(4)	**NG code**	2 bytes	05 (30h 35h)	Protect code error
(5)	**BCC**	2 bytes	00 - 7F	Block check character
(6)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

(4) NG code

Protect code error: The protect code in the request message does not match that set in FUN27. Attempt was made to enable protection on a protected user program.

Read PLC Operating Status

This command can be used to read the operating status of the ONC CPU module to the computer. When this command is executed, the received data also indicates whether the timer/counter preset values have been changed, whether the user program in the ONC is protected, and the type of the ONC CPU module.

Request Message (Read PLC Operating Status)

05h	**	**	30h	52h	53h	**	**	0Dh
(1)	(2)		(3)	(4)	(5)	(6)		(7)

(1)	**Communication control character**	1 byte	ENQ (05h)	Enquiry
(2)	**Communication device number**	2 bytes	00 - 1F FF	Device number 0 through 31 Device number 255 (all devices)
(3)	**Continuation**	1 byte	0 (30h)	Discontinued
(4)	**Command**	1 byte	R (52h)	Read data
(5)	**Data type**	1 byte	S (53h)	PLC operating status
(6)	**BCC**	2 bytes	00 - 7F	Block check character
(7)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

Reply Message (Read PLC Operating Status)

OK Reply

06h	**	**	30h	3*h	3*h	3*h	33h	3*h	**	**	0Dh
(1)	(2)		(3)	(4)	(5)	(6)	(7)		(8)		(9)

(1)	**Communication control character**	1 byte	ACK (06h)	Acknowledge
(2)	**Communication device number**	2 bytes	00 - 1F	Device number 0 through 31
(3)	**Command**	1 byte	0 (30h)	OK: Discontinued
(4)	**PLC operating status**	1 byte	0 (30h) 1 (31h)	Run Stop
(5)	**Timer / counter preset value change**	1 byte	0 (30h) 1 (31h)	Not changed Changed
(6)	**User program protection**	1 byte	0 (30h) 1 (31h) 2 (32h) 3 (33h)	Not protected Write protect Read protect Read and write protect
(7)	**CPU module type code**	2 bytes	30 (33h 30h) 31 (33h 31h) 32 (33h 32h) 33 (33h 33h) 34 (33h 34h) 35 (33h 35h) 36 (33h 36h) 37 (33h 37h)	FC3A-CP2K FC3A-CP2S FC3A-CP1K FC3A-CP1S FC3A-CP2KM FC3A-CP2SM FC3A-CP1KM FC3A-CP1SM
(8)	**BCC**	2 bytes	00 - 7F	Block check character
(9)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

NG Reply

NG reply never occurs in response to the request message of reading the PLC operating status.

Read Scan Time

The scan time of the user program in operation can be read from the ONC CPU module. When this command is executed, the received data indicates the current and maximum values of the user program scan time.

Request Message (Read Scan Time)

05h	**	**	30h	52h	4Bh	**	**	0Dh
(1)	(2)		(3)	(4)	(5)	(6)		(7)

(1)	**Communication control character**	1 byte	ENQ (05h)	Enquiry
(2)	**Communication device number**	2 bytes	00 - 1F FF	Device number 0 through 31 Device number 255 (all devices)
(3)	**Continuation**	1 byte	0 (30h)	Discontinued
(4)	**Command**	1 byte	R (52h)	Read data
(5)	**Data type**	1 byte	K (4Bh)	Scan time
(6)	**BCC**	2 bytes	00 - 7F	Block check character
(7)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

Reply Message (Read Scan Time)

OK Reply

06h	**	**	30h	**	**	**	**	**	**	**	**	**	**	0Dh
(1)	(2)		(3)		(4)				(5)			(6)		(7)

(1)	**Communication control character**	1 byte	ACK (06h)	Acknowledge
(2)	**Communication device number**	2 bytes	00 - 1F	Device number 0 through 31
(3)	**Command**	1 byte	0 (30h)	OK: Discontinued
(4)	**Scan time (current value)**	4 bytes	0000 - FFFF	Current value of the scan time
(5)	**Scan time (maximum value)**	4 bytes	0000 - FFFF	Maximum value of the scan time
(6)	**BCC**	2 bytes	00 - 7F	Block check character
(7)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

(4) and (5) Data Format in the Reply Message (Read Scan Time)

The scan time is read in units of msec.

The current and maximum values of the scan time are displayed in the hexadecimal notation for the integer and in the octal notation for the fraction.

Example: The read data is 0145 when reading the scan time.

Divide the received data into upper 3 characters and the lowest 1 character.

014h	5o

The upper 3 digits indicate the integer part of the scan time in the hexadecimal notation.
The lowest digit indicates the fraction part of the scan time in the octal notation.

In this example, the scan time reads $1 \times 16 + 4 + 0.5 \times 1.25 = 20.625$ msec in the decimal notation.

NG Reply

NG reply never occurs in response to the request message of reading the scan time.

Read PLC System Program Version

The system program version of the ONC CPU module can be read to the computer.

Request Message (Read PLC System Program Version)

05h	**	**	30h	52h	4Eh	**	**	0Dh
(1)	(2)		(3)	(4)	(5)	(6)		(7)

(1)	**Communication control character**	1 byte	ENQ (05h)	Enquiry
(2)	**Communication device number**	2 bytes	00 - 1F FF	Device number 0 through 31 Device number 255 (all devices)
(3)	**Continuation**	1 byte	0 (30h)	Discontinued
(4)	**Command**	1 byte	R (52h)	Read data
(5)	**Data type**	1 byte	N (4Eh)	PLC system program version
(6)	**BCC**	2 bytes	00 - 7F	Block check character
(7)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

Reply Message (Read PLC System Program Version)

OK Reply

06h	**	**	30h	**	**	**	**	**	**	0Dh
(1)	(2)		(3)		(4)			(5)		(6)

(1)	**Communication control character**	1 byte	ACK (06h)	Acknowledge
(2)	**Communication device number**	2 bytes	00 - 1F	Device number 0 through 31
(3)	**Command**	1 byte	0 (30h)	OK: Discontinued
(4)	**PLC system program version**	4 bytes	0000 - FFFF	System program version of the PLC
(5)	**BCC**	2 bytes	00 - 7F	Block check character
(6)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

NG Reply

NG reply never occurs in response to the request message of reading the PLC system program version.

Data Format in the Reply Message (Read PLC System Program Version)

The PLC system program version is the decimal equivalent of the hexadecimal reply data.

Example: The read data is 0065 when reading the PLC system program version.

The PLC program version is 101.

Read User Communication Transmit / Receive Buffer

While user communication is performed through the RS232C-1 or RS232C-2 port on the ONC CPU module, the transmit and receive buffers store the data of the last communication. The data stored in the transmit and receive buffers can be read through the data link terminals using the read user communication transmit/receive buffer command.

Request Message (Read User Communication Transmit / Receive Buffer)

05h	**	**	30h	52h	**	30h	3*h	3*h	3*h	**	**	**	**	0Dh
(1)	(2)		(3)	(4)	(5)	(6)				(7)		(8)		(9)

(1)	**Communication control character**	1 byte	ENQ (05h)	Enquiry
(2)	**Communication device number**	2 bytes	00 - 1F FF	Device number 0 through 31 Device number 255 (all devices)
(3)	**Continuation**	1 byte	0 (30h)	Discontinued
(4)	**Command**	1 byte	R (52h)	Read data
(5)	**Data type**	1 byte	G (47h) g (67h) I (49h) i (69h)	User communication receive buffer (RS232C-1 port) User communication transmit buffer (RS232C-1 port) User communication receive buffer (RS232C-2 port) User communication transmit buffer (RS232C-2 port)
(6)	**Operand number**	4 bytes	0000 - 0199	First address to read data (Nth byte in the 200-byte buffer)
(7)	**Data length (n)**	2 bytes	01 - C9	Byte count of data to read 200 (C8h) bytes maximum + 1h
(8)	**BCC**	2 bytes	00 - 7F	Block check character
(9)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

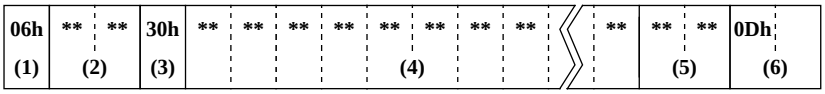
(7) Data length

Specify the byte count of the data to read plus one as a data length. Since a 2-byte ASCII code is attached to the beginning of the data codes in the reply message, one byte must be added to the data length in the request message. The additional 2-byte ASCII code represents the byte count of read data (see the next page).

The transmit/receive buffer has a capacity of 200 (C8h) bytes. When reading the entire data in the transmit/receive buffer, specify C9 (43h 39h) as a data length.

Reply Messages (Read User Communication Transmit / Receive Buffer)

OK Reply

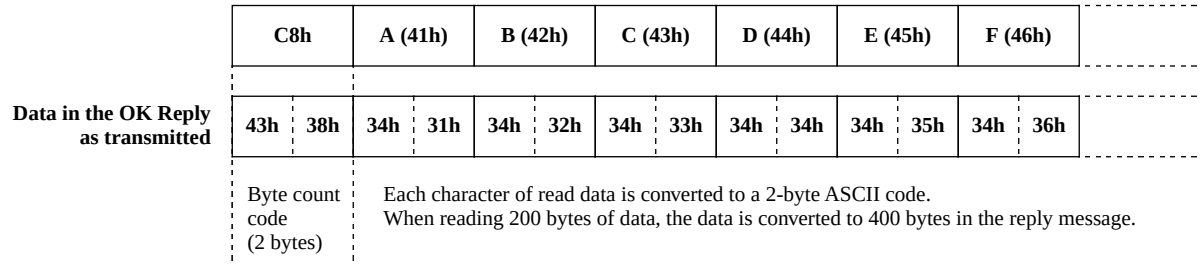


(1)	**Communication control character**	1 byte	ACK (06h)	Acknowledge
(2)	**Communication device number**	2 bytes	00 - 1F	Device number 0 through 31
(3)	**Command**	1 byte	0 (30h)	OK: Discontinued
(4)	**Data**	2(n+1) bytes 0 ≤ n ≤ 200	0 (30h) - 9 (39h) A (41h) - F (46h)	Read data byte count and read data
(5)	**BCC**	2 bytes	00 - 7F	Block check character
(6)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

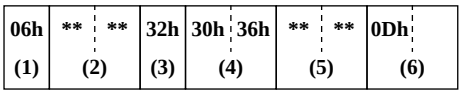
(4) Data

The byte count of data read from the transmit/receive buffer is stored in the first 2 bytes of the data codes, represented in ASCII code. The read data is stored starting at the third byte in the data codes, also represented in ASCII code.

Example: The byte count of data to read is C8h (maximum value of 200 bytes).
The data in the transmit/receive buffer is ABCEDF ..... (41h 42h 43h 44h 45h 46h .....)



NG Reply



(1)	**Communication control character**	1 byte	ACK (06h)	Acknowledge
(2)	**Communication device number**	2 bytes	00 - 1F	Device number 0 through 31
(3)	**Command**	1 byte	2 (32h)	NG
(4)	**NG code**	2 bytes	06 (30h 36h)	Data range error
(5)	**BCC**	2 bytes	00 - 7F	Block check character
(6)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

Read User Communication Status

Various information concerning user communication can be read out using this command, such as the PLC operating status, user communication mode, user communication receive error interrupt, and user communication transmit/receive instruction status. Communication parameter settings are also read out. For details, see the reply message format on the next page.

Request Message (Read User Communication Status)

05h	**	**	30h	52h	42h	**	**	0Dh
(1)	(2)		(3)	(4)	(5)	(6)		(7)

(1)	**Communication control character**	1 byte	ENQ (05h)	Enquiry
(2)	**Communication device number**	2 bytes	00 - 1F FF	Device number 0 through 31 Device number 255 (all devices)
(3)	**Continuation**	1 byte	0 (30h)	Discontinued
(4)	**Command**	1 byte	R (52h)	Read data
(5)	**Data type**	1 byte	B (42h) A (41h)	User communication status (RS232C-1 port) User communication status (RS232C-2 port)
(6)	**BCC**	2 bytes	00 - 7F	Block check character
(7)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

Reply Messages (Read User Communication Status)

OK Reply

06h	**	**	30h	3*h	3*h	3*h	3*h	3*h	3*h	3*h	3*h	3*h	**	**	**	**	0Dh
(1)	(2)		(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)		(14)		(15)

(1)	**Communication control character**	1 byte	ACK (06h)	Acknowledge
(2)	**Communication device number**	2 bytes	00 - 1F	Device number 0 through 31
(3)	**Command**	1 byte	0 (30h)	OK: Discontinued
(4)	**PLC operating status**	1 byte	0 (30h) 1 (31h)	Running Stopped
(5)	**User communication mode**	1 byte	0 (30h) 1 (31h)	User communication mode disabled User communication mode enabled
(6)	**User communication receive error interrupt**	1 byte	0 (30h) 1 (31h)	Interrupt not occurred Interrupt occurred
(7)	**User communication receive instruction status**	1 byte	0 (30h) 2 (32h) 3 (33h)	No valid RXD instruction in user program Receiving data Waiting for executing RXD instruction
(8)	**User communication transmit instruction status**	1 byte	0 (30h) 2 (32h) 4 (34h)	No valid TXD instruction in user program Transmitting data Waiting for executing TXD instruction
(9)	**Baud rate**	1 byte	0 (30h) 1 (31h) 2 (32h) 3 (33h) 4 (34h)	1200 bps 2400 bps 4800 bps 9600 bps 19200 bps
(10)	**Data length**	1 byte	0 (30h) 1 (31h)	7 bits 8 bits
(11)	**Parity**	1 byte	0 (30h) 1 (31h) 2 (32h)	None Odd Even
(12)	**Stop bits**	1 byte	0 (30h) 1 (31h)	1 bit 2 bits
(13)	**Receive timeout**	2 bytes	01 - FF	10 to 2550 msec
(14)	**BCC**	2 bytes	00 - 7F	Block check character
(15)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

NG Reply

NG reply never occurs in response to the request message of reading the user communication status.

Read Communication Mode

Various information concerning the ONC communication functions can be read out using this command, such as the protocol selector switch position, special mode selection, modem mode selection, PLC type code, and device number selector switch position. For details, see the reply message format on the next page.

Request Message (Read Communication Mode)

05h	**	**	30h	52h	48h	**	**	0Dh
(1)	(2)		(3)	(4)	(5)	(6)		(7)

(1)	**Communication control character**	1 byte	ENQ (05h)	Enquiry
(2)	**Communication device number**	2 bytes	00 - 1F FF	Device number 0 through 31 Device number 255 (all devices)
(3)	**Continuation**	1 byte	0 (30h)	Discontinued
(4)	**Command**	1 byte	R (52h)	Read data
(5)	**Data type**	1 byte	H (48h)	Communication mode
(6)	**BCC**	2 bytes	00 - 7F	Block check character
(7)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

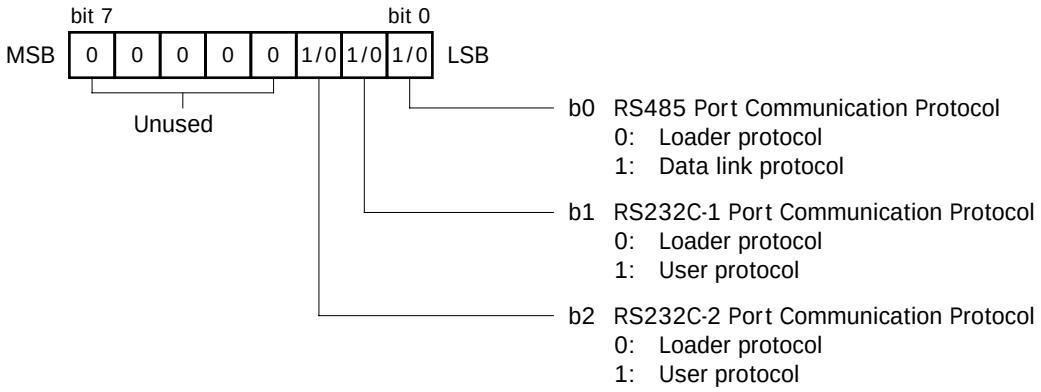
Reply Messages (Read Communication Mode)

OK Reply

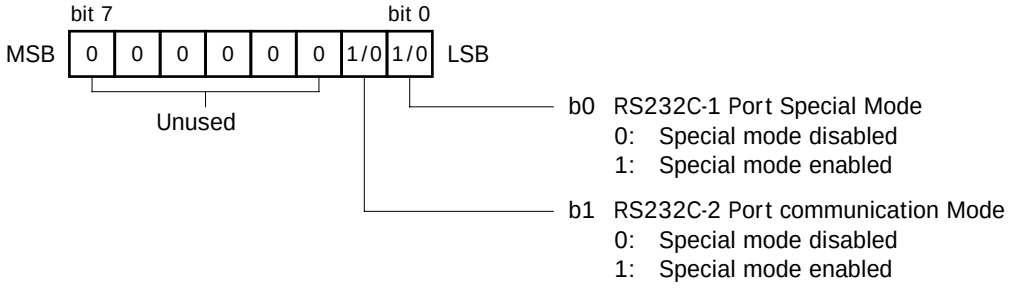
06h	**	**	30h	3*h	3*h	3*h	3*h	33h	3*h	3*h	3*h	**	**	0Dh
(1)	(2)		(3)	(4)	(5)	(6-1)	(6-2)	(7)		(8)		(9)		(10)

(1)	**Communication control character**	1 byte	ACK (06h)	Acknowledge
(2)	**Communication device number**	2 bytes	00 - 1F	Device number 0 through 31
(3)	**Command**	1 byte	0 (30h)	OK: Discontinued
(4)	**Protocol selector switch position**	1 byte	0 (30h) - 7 (37h)	0 to 7 (see table below)
(5)	**Special mode**	1 byte	0 (30h) - 3 (33h)	0 to 3 (see table below)
(6-1)	**RS232C-1 port modem mode**	2 bytes	0 (30h) 1 (31h)	Modem mode disabled Modem mode enabled
(6-2)	**RS232C-2 port modem mode**			
(7)	**CPU module type code**	2 bytes	30 (33h 30h) 31 (33h 31h) 32 (33h 32h) 33 (33h 33h) 34 (33h 34h) 35 (33h 35h) 36 (33h 36h) 37 (33h 37h)	FC3A-CP2K FC3A-CP2S FC3A-CP1K FC3A-CP1S FC3A-CP2KM FC3A-CP2SM FC3A-CP1KM FC3A-CP1SM
(8)	**Device number selector switch position**	2 bytes	00 to 31	Device number 0 to 31
(9)	**BCC**	2 bytes	00 - 7F	Block check character
(10)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

(4) Protocol selector switch position



(5) Special mode



NG Reply

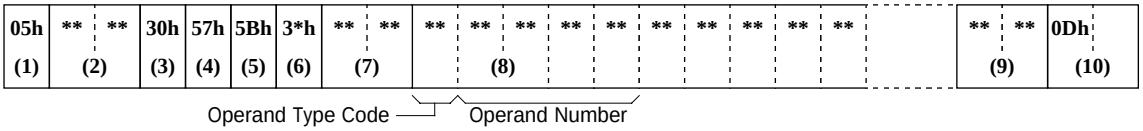
NG reply never occurs in response to the request message of reading the communication mode.

Select Word Operands for Monitor

This command is used to select operands for group monitoring. Inputs, outputs, internal relays, link relays, shift registers, timer/counter preset and current values, data registers, and link registers can be designated as operands for monitor in units of word. When a bit operand is designated, 16 bits are selected starting with the selected operand number.

One operand is designated by 5 bytes to represent the operand type and operand number. A maximum of 40 words of operands can be selected for group monitoring.

Request Message (Select Word Operands for Monitor)



(1)	**Communication control character**	1 byte	ENQ (05h)	Enquiry
(2)	**Communication device number**	2 bytes	00 - 1F FF	Device number 0 through 31 Device number 255 (all devices)
(3)	**Continuation**	1 byte	0 (30h)	Discontinued
(4)	**Command**	1 byte	W (57h)	Write data
(5)	**Data type**	1 byte	[(5Bh)	Word operands for monitor
(6)	**Table number**	1 byte	0 or 1	Select Table 0 or 1 to store data
(7)	**Quantity of word operands (n)**	2 bytes	01 - 28	Store 1 to 40 (28h) word operands
(8)	**Operand type code and number**	5n bytes (1 ≤ n ≤ 40)	0 (30h) - 9 (39h) A (41h) - F (46h)	1 byte for operand type 4 bytes for operand number
(9)	**BCC**	2 bytes	00 - 7F	Block check character
(10)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

The ONC has 2 areas (Tables 1 and 2) in the internal buffer to store the selected operands for group monitoring. These areas are cleared to zero when the ONC is powered up.

(8) Operand type code		(8) Operand number	Remarks
X (58h)	Input	0000 - 0597	The least significant digit of the operand number is an octal number (0-7). Upper digits are decimal numbers.
Y (59h)	Output	0000 - 0597	
M (4Dh)	Internal relay	0000 - 2557, 8000 - 8237	
O (4Fh)	Link relay	0000 - 8477	
R (52h)	Shift register	0000 - 0255	All four digits of the operand number are decimal numbers.
T (54h)	Timer (preset value)	0000 - 0255	
t (74h)	Timer (current value)	0000 - 0255	
C (43h)	Counter (preset value)	0000 - 0255	
c (63h)	Counter (current value)	0000 - 0255	
D (44h)	Data register	0000 - 8999	
W (57h)	Calendar/clock	0000 - 0006	
L (4Ch)	Link register	0100 - 0127, 0200 - 0227, 0300 - 0327, 0400 - 0427, 0500 - 0527, 0600 - 0627, 0700 - 0727, 1000 - 1317	The least significant digit of the operand number is an octal number (0-7). Upper digits are decimal numbers.

Operand numbers for calendar and clock are allocated as listed on the right:

Calendar/clock operand number	Data
0000	Year
0001	Month
0002	Day
0003	Day of week
0004	Hour
0005	Minute
0006	Second

When the range specified by the operand type and number is invalid, the ONC returns an NG reply.

When a preset value is read from a timer or counter for which a data register is designated as a preset value, the preset value is returned as a reply, rather than the data register number.

Reply Messages (Select Word Operands for Monitor)

OK Reply

06h	**	**	30h	**	**	0Dh
(1)	(2)		(3)	(4)		(5)

(1)	**Communication control character**	1 byte	ACK (06h)	Acknowledge
(2)	**Communication device number**	2 bytes	00 - 1F	Device number 0 through 31
(3)	**Command**	1 byte	0 (30h)	OK: Discontinued
(4)	**BCC**	2 bytes	00 - 7F	Block check character
(5)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

NG Reply

06h	**	**	32h	3*h	3*h	**	**	0Dh
(1)	(2)		(3)	(4)		(5)		(6)

(1)	**Communication control character**	1 byte	ACK (06h)	Acknowledge
(2)	**Communication device number**	2 bytes	00 - 1F	Device number 0 through 31
(3)	**Command**	1 byte	2 (32h)	NG
(4)	**NG code**	2 bytes	06 (30h 36h) 10 (31h 30h) 11 (31h 31h)	Data range error Data error Setting error (invalid table number)
(5)	**BCC**	2 bytes	00 - 7F	Block check character
(6)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

Monitor Selected Word Operands

This command is used to monitor the word operands which have been selected using the Select Word Operands for Monitor command (see page ONC Protocol-44).

Request Message (Monitor Selected Word Operands)

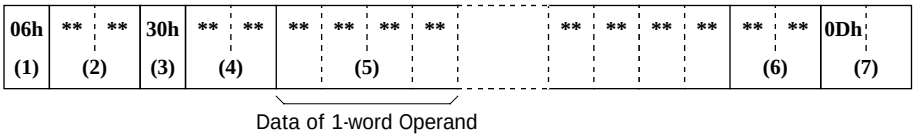
05h	**	**	30h	52h	5Bh	3*h	**	**	0Dh
(1)	(2)		(3)	(4)	(5)	(6)	(7)		(8)

(1)	**Communication control character**	1 byte	ENQ (05h)	Enquiry
(2)	**Communication device number**	2 bytes	00 - 1F FF	Device number 0 through 31 Device number 255 (all devices)
(3)	**Continuation**	1 byte	0 (30h)	Discontinued
(4)	**Command**	1 byte	R (52h)	Read data
(5)	**Data type**	1 byte	[(5Bh)	Word operands for monitor
(6)	**Table number**	1 byte	0 or 1	Select Table 0 or 1 to store data
(7)	**BCC**	2 bytes	00 - 7F	Block check character
(8)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

The ONC has 2 areas (Tables 1 and 2) in the internal buffer to store the selected operands for group monitoring. These areas are cleared to zero when the ONC is powered up.

Reply Messages (Monitor Selected Word Operands)

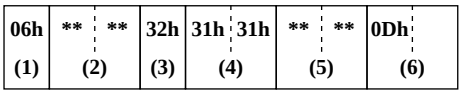
OK Reply



(1)	**Communication control character**	1 byte	ACK (06h)	Acknowledge
(2)	**Communication device number**	2 bytes	00 - 1F	Device number 0 through 31
(3)	**Command**	1 byte	0 (30h)	OK: Discontinued
(4)	**Quantity of word operands (n)**	2 bytes	01 - 28	Monitor 1 to 40 (28h) word operands
(5)	**Monitored data**	4n bytes (1 ≤ n ≤ 40)	0 (30h) - 9 (39h) A (41h) - F (46h)	4 bytes per word operand
(6)	**BCC**	2 bytes	00 - 7F	Block check character
(7)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

When a preset value is read from a timer or counter for which a data register is designated as a preset value, the preset value is returned as a reply, rather than the data register number.

NG Reply



(1)	**Communication control character**	1 byte	ACK (06h)	Acknowledge
(2)	**Communication device number**	2 bytes	00 - 1F	Device number 0 through 31
(3)	**Command**	1 byte	2 (32h)	NG
(4)	**NG code**	2 bytes	11 (31h 31h)	Setting error (table number not saved in CPU)
(5)	**BCC**	2 bytes	00 - 7F	Block check character
(6)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

Read Timer Information

Since the ONC has a timer preset range of 0 through 65535, the MSB of the timer current value data read using the Read N Byte command for the MICRO^3 cannot be used for timeout status flag. Consequently the timeout status information must be prepared by the system program separately. The ONC has a new data type to implement a command to read timer current value, preset value, timeout status, and preset value change status.

When a preset value is read from a timer for which a data register is designated as a preset value, the data register number is returned as a reply, rather than the preset value.

This command can read data from 48 timers at the maximum.

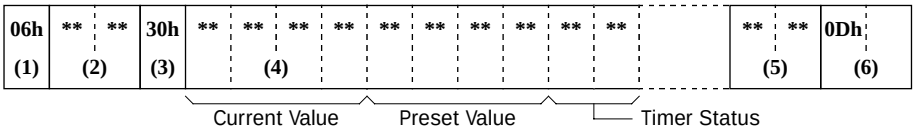
Request Message (Read Timer Information)

05h	**	**	30h	52h	5Fh	**	**	**	**	**	**	**	**	0Dh
(1)	(2)		(3)	(4)	(5)	(6)				(7)		(8)		(9)

(1)	**Communication control character**	1 byte	ENQ (05h)	Enquiry
(2)	**Communication device number**	2 bytes	00 - 1F FF	Device number 0 through 31 Device number 255 (all devices)
(3)	**Continuation**	1 byte	0 (30h)	Discontinued
(4)	**Command**	1 byte	R (52h)	Read data
(5)	**Data type**	1 byte	_ (5Fh)	Timer information
(6)	**First timer number**	4 bytes	0000 - 0255	First timer number to start reading
(7)	**Quantity of timers (n)**	2 bytes	01 - 30	Read information from 1 to 48 (30h) timers
(8)	**BCC**	2 bytes	00 - 7F	Block check character
(9)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

Reply Messages (Read Timer Information)

OK Reply

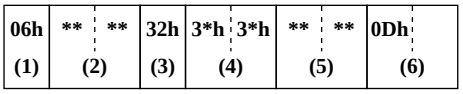


(1)	**Communication control character**	1 byte	ACK (06h)	Acknowledge
(2)	**Communication device number**	2 bytes	00 - 1F	Device number 0 through 31
(3)	**Command**	1 byte	0 (30h)	OK: Discontinued
(4)	**Data**	10n bytes (1 ≤ n ≤ 48)	0000 - FFFF 0000 - FFFF 00 - FF	Current value (4 bytes) Preset value (4 bytes) Timer status (2 bytes)
(5)	**BCC**	2 bytes	00 - 7F	Block check character
(6)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

(4) Data — Timer Status

bit 7	tim_use	Specified timer number programmed	0: Not programmed 1: Programmed
bit 6	tim_dr_const	Preset value designation	0: Constant 1: Data register
bit 5	tim_setting_value_change	Preset value change	0: Changed 1: Not changed
bits 4 and 3	tim_unit	Time base	00: 1 msec 10: 10 msec 01: 100 msec 11: 1 sec
bit 2	—	Unused	—
bit 1	tim_first-execute	Execution cycle	0: Subsequent timer cycle 1: First timer cycle
bit 0	tim_up	Timeout status	0: Not timeout 1: Timeout

NG Reply



(1)	**Communication control character**	1 byte	ACK (06h)	Acknowledge
(2)	**Communication device number**	2 bytes	00 - 1F	Device number 0 through 31
(3)	**Command**	1 byte	2 (32h)	NG
(4)	**NG code**	2 bytes	06 (30h 36h) 10 (31h 30h)	Data range error Data error
(5)	**BCC**	2 bytes	00 - 7F	Block check character
(6)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

Read Counter Information

Since the ONC has a counter preset range of 0 through 65535, the MSB of the counter current value data read using the Read N Byte command for the MICRO^3 cannot be used for countout status flag. Consequently the countout status information must be prepared by the system program separately. The ONC has a new data type to implement a command to read counter current value, preset value, countout status, and preset value change status.

When a preset value is read from a counter for which a data register is designated as a preset value, the data register number is returned as a reply, rather than the preset value.

This command can read data from 48 counters at the maximum.

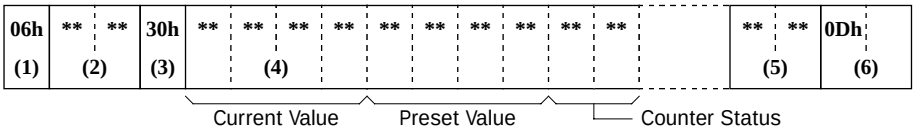
Request Message (Read Counter Information)

05h	** **	30h	52h	60h	** ** ** **	** **	** **	0Dh
(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)

(1)	**Communication control character**	1 byte	ENQ (05h)	Enquiry
(2)	**Communication device number**	2 bytes	00 - 1F FF	Device number 0 through 31 Device number 255 (all devices)
(3)	**Continuation**	1 byte	0 (30h)	Discontinued
(4)	**Command**	1 byte	R (52h)	Read data
(5)	**Data type**	1 byte	` (60h)	Counter information
(6)	**First counter number**	4 bytes	0000 - 0255	First counter number to start reading
(7)	**Quantity of counters (n)**	2 bytes	01 - 30	Read information from 1 to 48 (30h) counters
(8)	**BCC**	2 bytes	00 - 7F	Block check character
(9)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

Reply Messages (Read Counter Information)

OK Reply

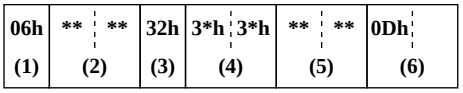


(1)	**Communication control character**	1 byte	ACK (06h)	Acknowledge
(2)	**Communication device number**	2 bytes	00 - 1F	Device number 0 through 31
(3)	**Command**	1 byte	0 (30h)	OK: Discontinued
(4)	**Data**	10n bytes (1 ≤ n ≤ 48)	0000 - FFFF 0000 - FFFF 00 - FF	Current value (4 bytes) Preset value (4 bytes) Counter status (2 bytes)
(5)	**BCC**	2 bytes	00 - 7F	Block check character
(6)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

(4) Data — Counter Status

bit 7	cnt_use	Specified counter number programmed	0: Not programmed 1: Programmed
bit 6	cnt_dr_const	Preset value designation	0: Constant 1: Data register
bit 5	cnt_setting_value_change	Preset value change	0: Changed 1: Not changed
bits 4 and 3	cnt_unit	Counter type	00: Adding 10: Up/down selection reversible 01: Dual pulse reversible
bit 2	Internal information	—	—
bit 1	Internal information	—	—
bit 0	cnt_up	Countout status	0: Not countout 1: Countout

NG Reply



(1)	**Communication control character**	1 byte	ACK (06h)	Acknowledge
(2)	**Communication device number**	2 bytes	00 - 1F	Device number 0 through 31
(3)	**Command**	1 byte	2 (32h)	NG
(4)	**NG code**	2 bytes	06 (30h 36h) 10 (31h 30h)	Data range error Data error
(5)	**BCC**	2 bytes	00 - 7F	Block check character
(6)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

Read Timer Preset Value Change Status

This command is used to check at once if preset values of all 256 timers are changed or not.

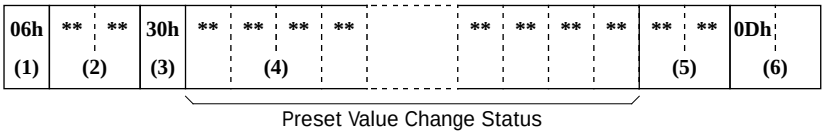
Request Message (Read Timer Preset Value Change Status)

05h	**	**	30h	52h	61h	**	**	0Dh
(1)	(2)		(3)	(4)	(5)	(6)		(7)

(1)	**Communication control character**	1 byte	ENQ (05h)	Enquiry
(2)	**Communication device number**	2 bytes	00 - 1F FF	Device number 0 through 31 Device number 255 (all devices)
(3)	**Continuation**	1 byte	0 (30h)	Discontinued
(4)	**Command**	1 byte	R (52h)	Read data
(5)	**Data type**	1 byte	a (61h)	Timer preset value change status
(6)	**BCC**	2 bytes	00 - 7F	Block check character
(7)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

Reply Messages (Read Timer Preset Value Change Status)

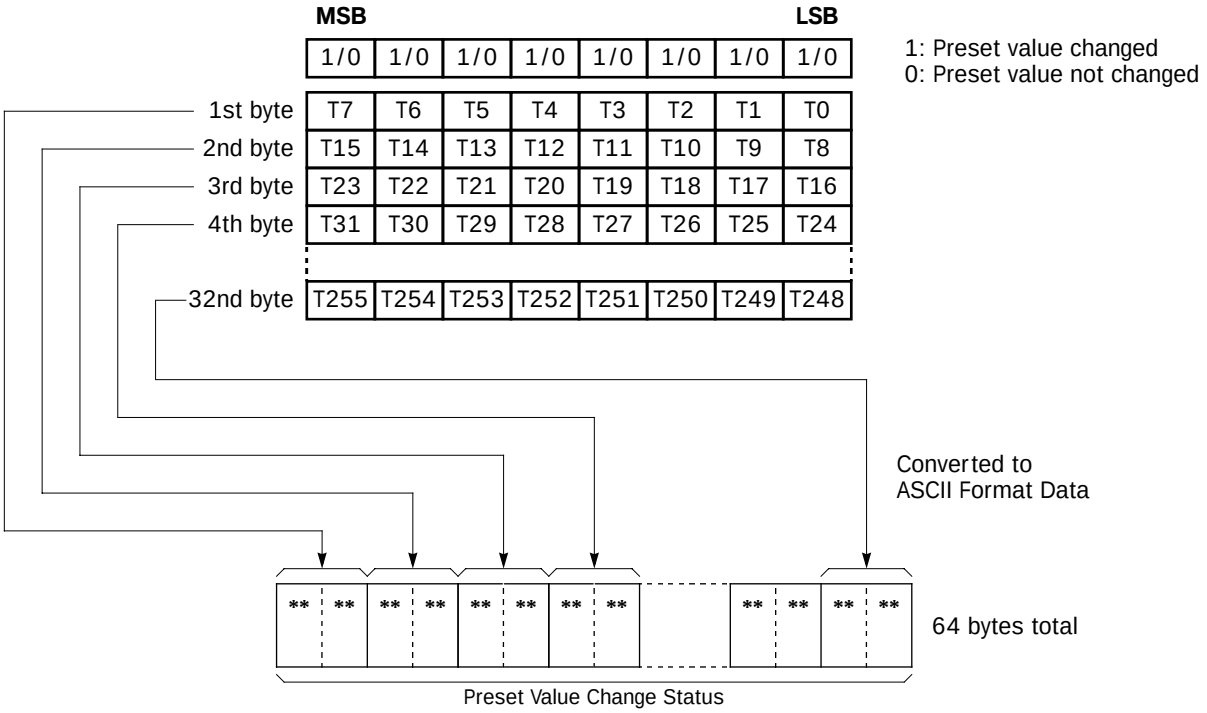
OK Reply



(1)	**Communication control character**	1 byte	ACK (06h)	Acknowledge
(2)	**Communication device number**	2 bytes	00 - 1F	Device number 0 through 31
(3)	**Command**	1 byte	0 (30h)	OK: Discontinued
(4)	**Data**	64 bytes	0 (30h) - 9 (39h) A (41h) - F (46h)	Preset value change status of all 256 timers
(5)	**BCC**	2 bytes	00 - 7F	Block check character
(6)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

(4) Data — Preset Value Change Status

In the 64 bytes of data, the statuses of 256 timers are allocated as shown below:



NG Reply

NG reply never occurs in response to the request message of reading the timer preset value change status.

Read Counter Preset Value Change Status

This command is used to check at once if preset values of all counters are changed or not.

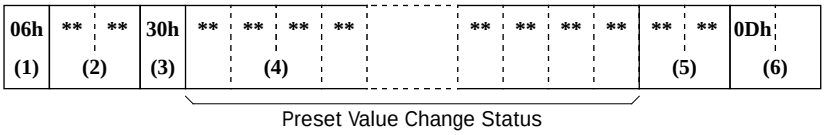
Request Message (Read Counter Preset Value Change Status)

05h	**	**	30h	52h	62h	**	**	0Dh
(1)	(2)		(3)	(4)	(5)	(6)		(7)

(1)	**Communication control character**	1 byte	ENQ (05h)	Enquiry
(2)	**Communication device number**	2 bytes	00 - 1F FF	Device number 0 through 31 Device number 255 (all devices)
(3)	**Continuation**	1 byte	0 (30h)	Discontinued
(4)	**Command**	1 byte	R (52h)	Read data
(5)	**Data type**	1 byte	b (62h)	Counter preset value change status
(6)	**BCC**	2 bytes	00 - 7F	Block check character
(7)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

Reply Messages (Read Counter Preset Value Change Status)

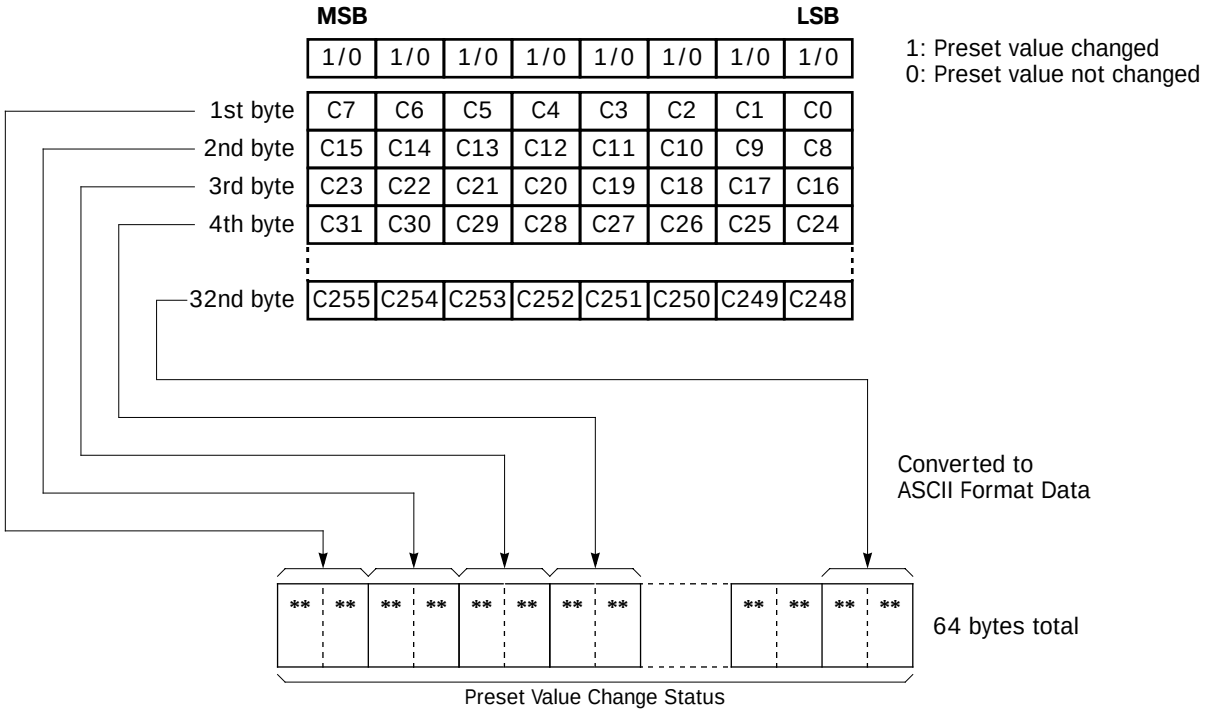
OK Reply



(1)	**Communication control character**	1 byte	ACK (06h)	Acknowledge
(2)	**Communication device number**	2 bytes	00 - 1F	Device number 0 through 31
(3)	**Command**	1 byte	0 (30h)	OK: Discontinued
(4)	**Data**	64 bytes	0 (30h) - 9 (39h) A (41h) - F (46h)	Preset value change status of all 256 counters
(5)	**BCC**	2 bytes	00 - 7F	Block check character
(6)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

(4) Data — Preset Value Change Status

In the 64 bytes of data, the statuses of 256 counters are allocated as shown below:



NG Reply

NG reply never occurs in response to the request message of reading the counter preset value change status.

Read FUN Area Settings

This command is used to read settings from specified FUN areas allocated in a user program.

Request Message (Read FUN Area Settings)

05h	**	**	30h	52h	46h	**	**	**	**	**	**	**	**	0Dh
(1)	(2)		(3)	(4)	(5)	(6)				(7)		(8)		(9)

(1)	**Communication control character**	1 byte	ENQ (05h)	Enquiry
(2)	**Communication device number**	2 bytes	00 - 1F FF	Device number 0 through 31 Device number 255 (all devices)
(3)	**Continuation**	1 byte	0 (30h)	Discontinued
(4)	**Command**	1 byte	R (52h)	Read data
(5)	**Data type**	1 byte	F (46h)	FUN area settings
(6)	**First FUN address**	4 bytes	0000 - 0300	First FUN address to start reading (0 to 768th byte)
(7)	**Quantity of FUN area bytes (n)**	2 bytes	01 - F0	Read information from 1 to 240 (F0h) bytes of FUN area settings
(8)	**BCC**	2 bytes	00 - 7F	Block check character
(9)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

Reply Messages (Read FUN Area Settings)

OK Reply

06h	**	**	30h	**	**	**	**		**	**	**	**	**	**	0Dh
(1)	(2)		(3)		(4)								(5)		(6)

FUN Area Data

(1)	**Communication control character**	1 byte	ACK (06h)	Acknowledge
(2)	**Communication device number**	2 bytes	00 - 1F	Device number 0 through 31
(3)	**Command**	1 byte	0 (30h)	OK: Discontinued
(4)	**Data**	2n bytes 1 ≤ n ≤ 240	0 (30h) - 9 (39h) A (41h) - F (46h)	ASCII format data of specified FUN areas
(5)	**BCC**	2 bytes	00 - 7F	Block check character
(6)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

NG Reply

06h	**	**	32h	3*h	3*h	**	**	0Dh
(1)	(2)		(3)	(4)		(5)		(6)

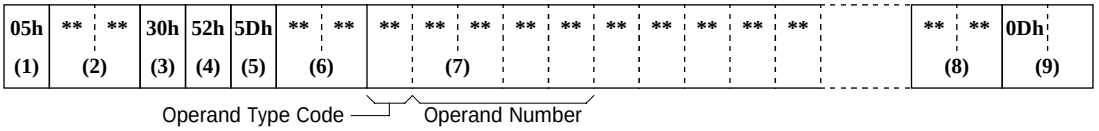
(1)	**Communication control character**	1 byte	ACK (06h)	Acknowledge
(2)	**Communication device number**	2 bytes	00 - 1F	Device number 0 through 31
(3)	**Command**	1 byte	2 (32h)	NG
(4)	**NG code**	2 bytes	06 (30h 36h) 10 (31h 30h)	Data range error Data error
(5)	**BCC**	2 bytes	00 - 7F	Block check character
(6)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

Read Random Words

This command is used to read word data from a maximum of 20 operands. Inputs, outputs, internal relays, link relays, shift registers, timer/counter preset and current values, data registers, and link registers can be designated as operands for reading in units of word. When a bit operand is designated, 16 bits are selected starting with the selected operand number.

One operand is designated by 5 bytes to represent the operand type and operand number. A maximum of 20 words of operands can be selected for reading random words.

Request Message (Read Random Words)



(1)	**Communication control character**	1 byte	ENQ (05h)	Enquiry
(2)	**Communication device number**	2 bytes	00 - 1F FF	Device number 0 through 31 Device number 255 (all devices)
(3)	**Continuation**	1 byte	0 (30h)	Discontinued
(4)	**Command**	1 byte	R (52h)	Read data
(5)	**Data type**	1 byte	] (5Dh)	Random words
(6)	**Quantity of word operands (n)**	2 bytes	01 - 14	Read 1 to 20 (14h) word operand data
(7)	**Operand type code and number**	5n bytes (1 ≤ n ≤ 20)	0 (30h) - 9 (39h) A (41h) - F (46h)	1 byte for operand type 4 bytes for operand number
(8)	**BCC**	2 bytes	00 - 7F	Block check character
(9)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

(7) Operand type code		(7) Operand number	Remarks
X (58h)	Input	0000 - 0597	The least significant digit of the operand number is an octal number (0-7). Upper digits are decimal numbers.
Y (59h)	Output	0000 - 0597	
M (4Dh)	Internal relay	0000 - 2557, 8000 - 8237	
O (4Fh)	Link relay	0000 - 8477	
R (52h)	Shift register	0000 - 0255	All four digits of the operand number are decimal numbers.
T (54h)	Timer (preset value)	0000 - 0255	
t (74h)	Timer (current value)	0000 - 0255	
C (43h)	Counter (preset value)	0000 - 0255	
c (63h)	Counter (current value)	0000 - 0255	
D (44h)	Data register	0000 - 8999	
W (57h)	Calendar/clock	0000 - 0006	
L (4Ch)	Link register	0100 - 0127, 0200 - 0227, 0300 - 0327, 0400 - 0427, 0500 - 0527, 0600 - 0627, 0700 - 0727, 1000 - 1317	The least significant digit of the operand number is an octal number (0-7). Upper digits are decimal numbers.

Operand numbers for calendar and clock are allocated as listed on the right:

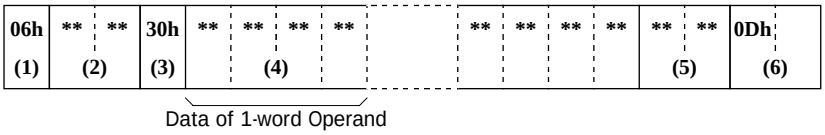
Calendar/clock operand number	Data
0000	Year
0001	Month
0002	Day
0003	Day of week
0004	Hour
0005	Minute
0006	Second

When the range specified by the operand type and number is invalid, the ONC returns an NG reply.

When a preset value is read from a timer or counter for which a data register is designated as a preset value, the preset value is returned as a reply, rather than the data register number.

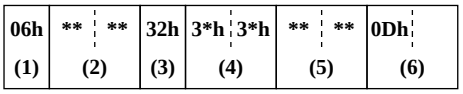
Reply Messages (Read Random Words)

OK Reply



(1)	**Communication control character**	1 byte	ACK (06h)	Acknowledge
(2)	**Communication device number**	2 bytes	00 - 1F	Device number 0 through 31
(3)	**Command**	1 byte	0 (30h)	OK: Discontinued
(4)	**Read data**	4n bytes (1 ≤ n ≤ 20)	0 (30h) - 9 (39h) A (41h) - F (46h)	Data of 1 word operand consists of 4 bytes
(5)	**BCC**	2 bytes	00 - 7F	Block check character
(6)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

NG Reply



(1)	**Communication control character**	1 byte	ACK (06h)	Acknowledge
(2)	**Communication device number**	2 bytes	00 - 1F	Device number 0 through 31
(3)	**Command**	1 byte	2 (32h)	NG
(4)	**NG code**	2 bytes	06 (30h 36h) 10 (31h 30h)	Data range error Data error
(5)	**BCC**	2 bytes	00 - 7F	Block check character
(6)	**Terminator**	1 byte 2 bytes	CR (0Dh) CR LF (0Dh 0Ah)	Message end code

NG Code and Action (Provisional)

When an NG reply is returned. The first character of the reply message is ACK (06h) and the command code is 2 (32h) which means NG (error).

Probable Cause and Action

The reply message signals an error. Check the NG code and take a corrective action shown in the table below:

NG Code	Cause	Action
01 (Program size error)	Improper write/read program size. When writing a user program to ONC, the user program capacity is larger than the program capacity selected in ONC. When reading a user program from ONC, the program capacity selected in ONC is larger than the user program receive buffer in the computer.	Check the program capacity setting in ONC using FUN2. Send a user program smaller than or equal to the program capacity setting. Increase the user program receive buffer capacity and send to ONC a request message including the capacity data.
02 (Protect error)	The user program in ONC is read and/or write protected.	Cancel the program protection.
03 (RUN error)	Writing user program attempted while ONC is running.	Stop ONC and try writing user program to ONC again.
04 (CRC error)	User program CRC code not matched. The user program to be written is broken.	Correct the user program and send the corrected user program to ONC.
05 (Protect code error)	Protect code in the request message does not match that set in FUN27. Attempt was made to enable protection on a protected user program.	Send a correct protect code to the PLC. Do not attempt to enable protection on a protected user program.
06 (Data range error)	Designated data range is invalid.	Make sure of the correct data range.
07 (Timer/counter preset value change error)	Preset value change attempted to timer or counter with preset value designated by data register.	Timer/counter preset values in ONC can not be changed when a data register is designated as a preset value. Check the user program in ONC to see that the timer/counter has a constant designated as a preset value. To change a timer or counter preset value designated by a data register, change the value of the data register.
08 (Calendar/clock data error)	Writing invalid value to calendar/clock attempted. The calendar/clock in ONC is broken.	Make sure of correct values for the calendar/clock.
09 (Data clear error)	Designated data cannot be cleared.	Check if an error has occurred in ONC. Correct the error and try again.
10 (Data error)	Invalid data other than 0 (30h) - 9 (39h) or A (41h) - F (46h) is included in the request message.	Check the request message and send a correct request message.
11 (Setting error)	Incorrect setting for user communication	Check the request message and send a correct request message.
12 (Operand write error)	Too many commands in one scan	Only three write commands are accepted in one scan. Make sure that more than three write commands do not enter in one scan.

Operand Allocation Numbers

Available I/O numbers depend on the type of the MicroSmart CPU module and the combination of I/O modules. I/O modules can be used with only the 24-I/O type CPU module among all-in-one type CPU modules. All slim type CPU modules can be used with I/O modules to expand the I/O points.

All-in-One Type CPU Modules

Operand	FC4A-C10R2		FC4A-C16R2		FC4A-C24R2	
	Allocation No.	Points	Allocation No.	Points	Allocation No.	Points
Input (I)	I0 - I5	6	I0 - I7 I10	9	I0 - I7 I10 - I15	14
Expansion Input (I)	—	—	—	—	I30 - I107	64 (78 total)
Output (Q)	Q0 - Q3	4	Q0 - Q6	7	Q0 - Q7 Q10 - Q11	10
Expansion Output (Q)	—	—	—	—	Q30 - Q107	64 (74 total)
Internal Relay (M)	M0 - M317	256	M0 - M1277	1024	M0 - M1277	1024
Special Internal Relay (M)	M8000 - M8157	128	M8000 - M8157	128	M8000 - M8157	128
Shift Register (R)	R0 - R63	64	R0 - R127	128	R0 - R127	128
Timer (T)	T0 - T31	32	T0 - T99	100	T0 - T99	100
Counter (C)	C0 - C31	32	C0 - C99	100	C0 - C99	100
Data Register (D)	D0 - D399	400	D0 - D1299	1300	D0 - D1299	1300
Special Data Register (D)	D8000 - D8099	100	D8000 - D8199	200	D8000 - D8199	200

Slim Type CPU Modules

Operand	FC4A-D20K3 FC4A-D20S3		FC4A-D20RK1 FC4A-D20RS1		FC4A-D40K3 FC4A-D40S3	
	Allocation No.	Points	Allocation No.	Points	Allocation No.	Points
Input (I)	I0 - I7 I10 - I13	12	I0 - I7 I10 - I13	12	I0 - I7 I10 - I17 I20 - I27	24
Expansion Input (I)	I30 - I187	128 (140 total)	I30 - I307	224 (236 total)	I30 - I307	224 (248 total)
Output (Q)	Q0 - Q7	8	Q0 - Q7	8	Q0 - Q7 Q10 - Q17	16
Expansion Output (Q)	Q30 - Q187	128 (136 total)	Q30 - Q307	224 (232 total)	Q30 - Q307	224 (240 total)
Internal Relay (M)	M0 - M1277	1024	M0 - M1277	1024	M0 - M1277	1024
Special Internal Relay (M)	M8000 - M8157	128	M8000 - M8157	128	M8000 - M8157	128
Shift Register (R)	R0 - R127	128	R0 - R127	128	R0 - R127	128
Timer (T)	T0 - T99	100	T0 - T99	100	T0 - T99	100
Counter (C)	C0 - C99	100	C0 - C99	100	C0 - C99	100
Data Register (D)	D0 - D1299	1300	D0 - D1299	1300	D0 - D1299	1300
Expansion Data Register (D)	—	—	D2000 - D7999	6000	D2000 - D7999	6000
Special Data Register (D)	D8000 - D8199	200	D8000 - D8199	200	D8000 - D8199	200