PCON, ACON, SCON, ERC2

Serial Communication [Modbus Version]

Operation Manual, Fifth Edition

IAI America, Inc.

Please Read Before Use

Thank you for purchasing our product.

This Operation Manual explains the handling methods, structure and maintenance of this product, among others, providing the information you need to know to use the product safely.

Before using the product, be sure to read this manual and fully understand the contents explained herein to ensure safe use of the product.
The CD/DVD that comes with the product contains operation manuals for IAI products.
When using the product, refer to the necessary portions of the applicable operation manual by printing them out or displaying them on a PC.

After reading the Operation Manual, keep it in a convenient place so that whoever is handling this product can reference it quickly when necessary.

[Important]

- The product cannot be operated in any way unless expressly specified in this Operation Manual. IAI shall assume no responsibility for the outcome of any operation not specified herein.
- Information contained in this Operation Manual is subject to change without notice for the purpose of product improvement.
- If you have any question or comment regarding the content of this manual, please contact the IAI sales office near you.
- Using or copying all or part of this Operation Manual without permission is prohibited.
- The company names, names of products and trademarks of each company shown in the sentences are registered trademarks.

Table of Contents

Safety Guide

When designing and manufacturing a robot system, ensure safety by following the safety guides provided below and taking the necessary measures.

Regulations and Standards Governing Industrial Robots

Safety measures on mechanical devices are generally classified into four categories under the International Industrial Standard ISO/DIS 12100, "Safety of machinery, " as follows:

Safety measures
- Inherent safety design
- Protective guards --- Safety fence, etc.
- Additional safety measures --- Emergency stop device, etc.
- Information on use --- Danger sign, warnings, operation manual

Based on this classification, various standards are established in a hierarchical manner under the International Standards ISO/IEC. The safety standards that apply to industrial robots are as follows:

Type C standards (individual safety standards) → ISO10218 (Manipulating industrial robots – Safety)

→ JIS B 8433 (Manipulating industrial robots – Safety)

Also, Japanese laws regulate the safety of industrial robots, as follows:

Industrial Safety and Health Law Article 59

Workers engaged in dangerous or harmful operations must receive special education.

Ordinance on Industrial Safety and Health

Article 36 --- Operations requiring special education
- No. 31 (Teaching, etc.) --- Teaching and other similar work involving industrial robots (exceptions apply)
- No. 32 (Inspection, etc.) --- Inspection, repair, adjustment and similar work involving industrial robots (exceptions apply)

Article 150 --- Measures to be taken by the user of an industrial robot

Requirements for Industrial Robots under Ordinance on Industrial Safety and Health

Work area	Work condition	Cutoff of drive source	Measure	Article
Outside movement range	During automatic operation	Not cut off	Signs for starting operation	Article 104
			Installation of railings, enclosures, etc.	Article 150-4
Inside movement range	During teaching, etc.	Cut off (including stopping of operation)	Sign, etc., indicating that work is in progress	Article 150-3
		Not cut off	Preparation of work rules	Article 150-3
			Measures to enable immediate stopping of operation	Article 150-3
			Sign, etc., indicating that work is in progress	Article 150-3
			Provision of special education	Article 36-31
			Checkup, etc., before commencement of work	Article 151
	During inspection, etc.	Cut off	To be performed after stopping the operation	Article 150-5
			Sign, etc., indicating that work is in progress	Article 150-5
		Not cut off (when inspection, etc., must be performed during operation)	Preparation of work rules	Article 150-5
			Measures to enable immediate stopping of operation	Article 150-5
			Sign, etc., indicating that work is in progress	Article 150-5
			Provision of special education (excluding cleaning and lubrication)	Article 36-32

Applicable Models of IAI's Industrial Robots

Machines meeting the following conditions are not classified as industrial robots according to Notice of Ministry of Labor No. 51 and Notice of Ministry of Labor/Labor Standards Office Director (Ki-Hatsu No. 340):

(1) Single-axis robot with a motor wattage of 80 W or less
(2) Combined multi-axis robot whose X, Y and Z-axes are 300 mm or shorter and whose rotating part, if any, has the maximum movement range of within 300 mm^3 including the end of the rotating part
(3) Multi-joint robot whose movable radius and Z-axis are within 300 mm

Among the products featured in our catalogs, the following models are classified as industrial robots:

1. Single-axis ROBO Cylinders
   RCS2/RCS2CR-SS8/RCS3 whose stroke exceeds 300 mm
2. Single-axis robots
   The following models whose stroke exceeds 300 mm and whose motor capacity also exceeds 80 W:
   ISA/ISB/ISPA/ISPB, SSPA,ISDA/ISDB/ISPDA/ISPDB, SSPDA, ISWA/ISPWA, IF, FS, NS
3. Linear servo actuators
   All models whose stroke exceeds 300 mm
4. Cartesian robots
   Any robot that uses at least one axis corresponding to one of the models specified in 1 to 3
5. IX SCARA robots
   IX-NNN (NNW, NNC) 3515 [H]
   IX-NNN (NNW, NNC) 50☐☐ [H] /60☐☐ [H] /70☐☐ [H] /80☐☐ [H]
   IX-NSN5016[H] /6016 [H]
   IX-TNN (UNN) 3015 [H]/3515 [H]
   IX-HNN (INN) 50☐☐ [H] /60☐☐ [H] /70☐☐ [H] /80☐☐ [H]

Notes on Safety of Our Products

Common items you should note when performing each task on any IAI robot are explained below.

No.	Task	Note
1	Model selection	● This product is not planned or designed for uses requiring high degrees of safety. Accordingly, it cannot be used to sustain or support life and must not be used in the following applications: [1] Medical devices relating to maintenance, management, etc., of life or health [2] Mechanisms or mechanical devices (vehicles, railway facilities, aircraft facilities, etc.) intended to move or transport people [3] Important safety parts in mechanical devices (safety devices, etc.) ● Do not use this product in the following environments: [1] Place subject to flammable gases, ignitable objects, flammables, explosives, etc. [2] Place that may be exposed to radiation [3] Place where the surrounding air temperature or relative humidity exceeds the specified range [4] Place subject to direct sunlight or radiated heat from large heat sources [5] Place subject to sudden temperature shift and condensation [6] Place subject to corrosive gases (sulfuric acid, hydrochloric acid, etc.) [7] Place subject to excessive dust, salt or iron powder [8] Place where the product receives direct vibration or impact ● Do not use this product outside the specified ranges. Doing so may significantly shorten the life of the product or result in product failure or facility stoppage.
2	Transportation	● When transporting the product, exercise due caution not to bump or drop the product. ● Use appropriate means for transportation. ● Do not step on the package. ● Do not place on the package any heavy article that may deform the package. ● When using a crane with a capacity of 1 ton or more, the crane must be operated by personnel qualified to operate cranes and perform slinging operations. ● When using a crane or other equipment, never use it to hoist any article exceeding the rated load of the applicable crane, etc. ● Use hoisting accessories suitable for the article to be hoisted. Select appropriate hoisting accessories by making sure there is an ample allowance for safety in their cutting load, etc. ● Do not climb onto the article being hoisted. ● Do not keep the article hoisted. ● Do not stand under the hoisted article.
3	Storage/ preservation	● The storage/preservation environment should conform to the installation environment. Among others, be careful not to cause condensation.
4	Installation/ startup	(1) Installing the robot, controller, etc. ● Be sure to firmly secure and affix the product (including its work part). If the product tips over, drops, malfunctions, etc., damage or injury may result. ● Do not step on the product or place any article on top. The product may tip over or the article may drop, resulting in injury, product damage, loss of/drop in product performance, shorter life, etc. ● If the product is used in any of the following places, provide sufficient shielding measures: [1] Place subject to electrical noise [2] Place subject to a strong electric or magnetic field [3] Place where power lines or drive lines are wired nearby [4] Place subject to splashed water, oil or chemicals

No.	Task	Note
4	Installation/ startup	(2) Wiring the cables ● Use IAI's genuine cables to connect the actuator and controller or connect a teaching tool, etc. ● Do not damage, forcibly bend, pull, loop round an object or pinch the cables or place heavy articles on top. Current leak or poor electrical continuity may occur, resulting in fire, electric shock or malfunction. ● Wire the product correctly after turning off the power. ● When wiring a DC power supply (+24 V), pay attention to the positive and negative polarities. Connecting the wires in wrong polarities may result in fire, product failure or malfunction. ● Securely connect the cables and connectors so that they will not be disconnected or come loose. Failing to do so may result in fire, electric shock or product malfunction. ● Do not cut and reconnect the cables of the product to extend or shorten the cables. Doing so may result in fire or product malfunction.
		(3) Grounding ● Be sure to provide class D (former class 3) grounding for the controller. Grounding is required to prevent electric shock and electrostatic charges, improve noise resistance and suppress unnecessary electromagnetic radiation.
		(4) Safety measures ● Implement safety measures (such as installing safety fences, etc.) to prevent entry into the movement range of the robot when the product is moving or can be moved. Contacting the moving robot may result in death or serious injury. ● Be sure to provide an emergency stop circuit so that the product can be stopped immediately in case of emergency during operation. ● Implement safety measures so that the product cannot be started only by turning on the power. If the product starts suddenly, injury or product damage may result. ● Implement safety measures so that the product will not start upon cancellation of an emergency stop or recovery of power following a power outage. Failure to do so may result in injury, equipment damage, etc. ● Put up a sign saying "WORK IN PROGRESS. DO NOT TURN ON POWER, " etc., during installation, adjustment, etc. If the power is accidently turned on, electric shock or injury may result. ● Implement measures to prevent the work part, etc., from dropping due to a power outage or emergency stop. ● Ensure safety by wearing protective gloves, protective goggles and/or safety shoes, as necessary. ● Do not insert fingers and objects into openings in the product. Doing so may result in injury, electric shock, product damage, fire, etc.
5	Teaching	● Whenever possible, perform teaching from outside the safety fences. If teaching must be performed inside the safety fences, prepare "work rules" and make sure the operator understands the procedures thoroughly. ● When working inside the safety fences, the operator should carry a handy emergency stop switch so that the operation can be stopped any time when an abnormality occurs. ● When working inside the safety fences, appoint a safety watcher in addition to the operator so that the operation can be stopped any time when an abnormality occurs. The safety watcher must also make sure the switches are not operated inadvertently by a third party. ● Put up a sign saying "WORK IN PROGRESS" in a conspicuous location. * Safety fences --- Indicate the movement range if safety fences are not provided.

No.	Task	Note
6	Confirmation operation	● After teaching or programming, carry out step-by-step confirmation operation before switching to automatic operation. ● When carrying out confirmation operation inside the safety fences, follow the specified work procedure just like during teaching. ● When confirming the program operation, use the safety speed. Failure to do so may result in an unexpected movement due to programming errors, etc., causing injury. ● Do not touch the terminal blocks and various setting switches while the power is supplied. Touching these parts may result in electric shock or malfunction.
7	Automatic operation	● Before commencing automatic operation, make sure no one is inside the safety fences. ● Before commencing automatic operation, make sure all related peripherals are ready to operate in the auto mode and no abnormalities are displayed or indicated. ● Be sure to start automatic operation from outside the safety fences. ● If the product generated abnormal heat, smoke, odor or noise, stop the product immediately and turn off the power switch. Failure to do so may result in fire or product damage. ● If a power outage occurred, turn off the power switch. Otherwise, the product may move suddenly when the power is restored, resulting in injury or product damage.
8	Maintenance/ inspection	● Whenever possible, work from outside the safety fences. If work must be performed inside the safety fences, prepare “work rules” and make sure the operator understands the procedures thoroughly. ● When working inside the safety fences, turn off the power switch, as a rule. ● When working inside the safety fences, the operator should carry a handy emergency stop switch so that the operation can be stopped any time when an abnormality occurs. ● When working inside the safety fences, appoint a safety watcher in addition to the operator so that the operation can be stopped any time when an abnormality occurs. The safety watcher must also make sure the switches are not operated inadvertently by a third party. ● Put up a sign saying “WORK IN PROGRESS” in a conspicuous location. ● Use appropriate grease for the guides and ball screws by checking the operation manual for each model. ● Do not perform a withstand voltage test. Conducting this test may result in product damage. * Safety fences --- Indicate the movement range if safety fences are not provided.
9	Modification	● The customer must not modify or disassemble/assemble the product or use maintenance parts not specified in the manual without first consulting IAI. ● Any damage or loss resulting from the above actions will be excluded from the scope of warranty.
10	Disposal	● When the product becomes no longer usable or necessary, dispose of it properly as an industrial waste. ● When disposing of the product, do not throw it into fire. The product may explode or generate toxic gases.

Indication of Cautionary Information

The operation manual for each model denotes safety guides under “Danger, ” “Warning, ” “Caution” and “Note, ” as specified below.

Level	Degree of danger/loss	Symbol
Danger	Failure to observe the instruction will result in an imminent danger leading to death or serious injury.	⚠ Danger
Warning	Failure to observe the instruction may result in death or serious injury.	⚠ Warning
Caution	Failure to observe the instruction may result in injury or property damage.	⚠ Caution
Note	The user should take heed of this information to ensure the proper use of the product, although failure to do so will not result in injury.	❗ Note

Handling Precautions

The explanations provided in this manual are limited to procedures of serial communication. Refer to the operation manual supplied with the ROBO Cylinder Controller (hereinafter referred to as RC controller) for other specifications, such as control, installation and connection.

Caution ⚠

(1) If any address or function not defined in this specification is sent to an RC controller, the controller may not operate properly or it may implement unintended movements. Do not send any function or address not specified herein.

(2) RC controllers are designed in such a way that once the controller detects a break (space) signal of 150 msec or longer via its SIO port, it will automatically switch the baud rate to 9600 bps.
On some PCs, the transmission line remains in the break (space) signal transmission mode while the communication port is closed. Exercise caution if one of these PCs is used as the host device, because the baud rate in your RC controller may have been changed to 9600 bps.

(3) Set the baud rate and other parameters using IAI's PC software or other dedicated teaching tool.

(4) If the controller is used in a place meeting any of the following conditions, provide sufficient shielding measures. If sufficient actions are not taken, the controller may malfunction:
[1] Where large current or high magnetic field generates
[2] Where arc discharge occurs due to welding, etc.
[3] Where noise generates due to electrostatic, etc.
[4] Where the controller may be exposed to radiation

(5) When performing wiring tasks and inserting/extracting connectors in/from sockets, make sure that the power supplies of the host and each RC controller are turned off. Carrying out such tasks with the power supplies turned on may result in electric shock and/or damage to parts.

(6) In order to prevent malfunctions due to noise, wire the communication cables such that the communication cables are isolated from power lines and other control wiring.

(7) In order to prevent malfunctions due to noise, make sure to take noise prevention measures on the electric equipment in the same power supply circuit or within the same device.

1 Overview

The ROBO Cylinder Controller (hereinafter referred to as RC controller) is equipped with a serial bus interface for asynchronous communication conforming to the EIA RS485 standard. This interface allows the RC controller to communicate with the host (host controller). In this way, it is possible to build an SIO link system that can connect and control up to 16 axes of slaves (RC controllers) (Note 1).
In addition to sending commands to each axis individually, it is also possible to broadcast the same command to all slaves at the same time.
Modbus Protocol is employed as the communication protocol, and it is possible to send commands from a host as well as read internal information.
Since the specifications of Modbus Protocol are disclosed globally, software development can be carried out easily.

(Note 1) Note that it is only possible to connect RC series devices on the same network; old RC series (protocol T) or other devices cannot be connected.

There are 2 types of serial transmission modes: ASCII mode (where 1-byte (8 bits) data is converted to ASCII code (2 characters) and sent) and RTU mode (where 1-byte (8 bits) data is sent as is). RC controllers identify the transmission mode on a packet-by-packet basis, thus making it possible to receive in both modes (Note 2).
Set the ROBONET RS485 to the SIO through mode. [Refer to the separate ROBONET Operation Manual.]

(Note 2) Make sure to use the same serial transmission mode for all devices on one network: it is not allowed to use both modes.

☆ Controllable controllers
- ERC2 (SE)
- PCON-C / CG / CF / CY / SE / PL / PO
- ACON-C / CG / CY / SE / PL / PO
- SCON-C / CA
- ROBONET_RS485 (SIO through mode)

1.1 Operation Manuals Relating to This Product You Find in the CD

No.	Name	Control No
1	Operation Manual for ERC2(PIO)	ME0158
2	Operation Manual for ERC2(SE:SIO)	ME0159
3	Operation Manual for PCON-C/CG/CF Controller	ME0170
4	Operation Manual for PCON-CY Controller	ME0156
5	Operation Manual for PCON-SE Controller	ME0163
6	Operation Manual for PCON-PL/PO Controller	ME0164
7	Operation Manual for ACON-C/CG Controller	ME0176
8	Operation Manual for ACON-CY Controller	ME0167
9	Operation Manual for ACON-SE Controller	ME0171
10	Operation Manual for ACON-PL/PO Controller	ME0166
11	Operation Manual for SCON-C Controller	ME0161
12	Operation Manual for SCON-CA Controller	ME0243
13	Operation Manual for ROBONET	ME0208

2 Specifications

Item	Method/condition
Interface	Conforming to EIA RS485
Communication method	Half-duplex communication
Maximum total extension distance	100 m
Synchronization method	Start-stop synchronization
Connection pattern	1-to-N unbalanced bus connection ($1 \leq N \leq 16$)
Transmission mode	RTU/ASCII (auto-detect) *
Baud rate (bps)	Selectable from the following speeds via parameter setting: 9600, 14400, 19200, 28800, 38400 57600, 76800, 115200, 230400
Bit length	8 bits
Stop bit	1 bit
Parity	None

2.1 Communication Mode

In the Modbus protocol, communication takes place in a single-master/multiple-slave configuration. In this communication, only the master (the PLC host in the example below) issues a query to a specified slave (the RC controller connected to axis C in the example below). When the specified slave receives this query, it executes the function specified in the query, and then returns a response message (one communication cycle is completed with this operation).

The query message format consists of the slave address (or broadcast), function code defining the content of request, data, and error check.

The response message format consists of the function code confirming the content of request, data, and error check. Following figure shows the query message format and response message configuration.

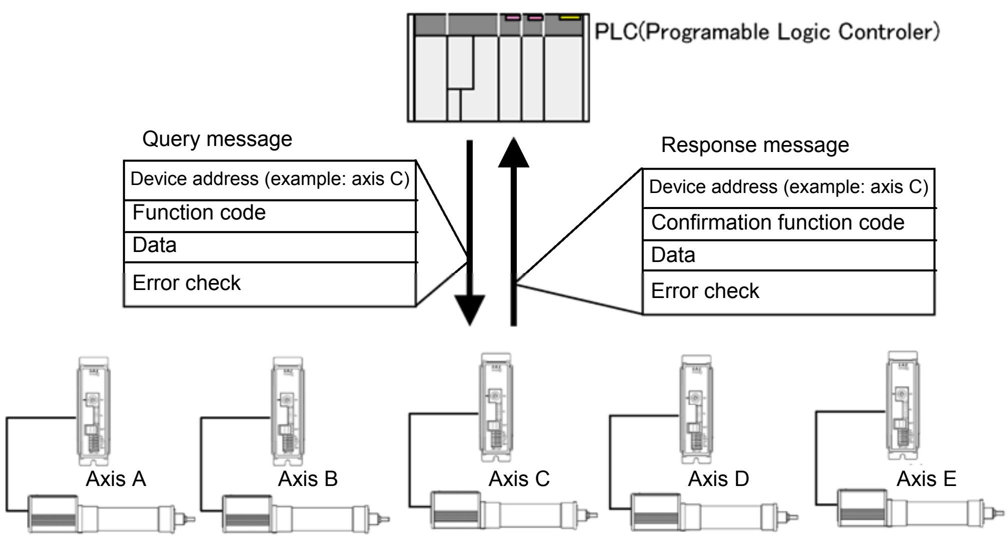
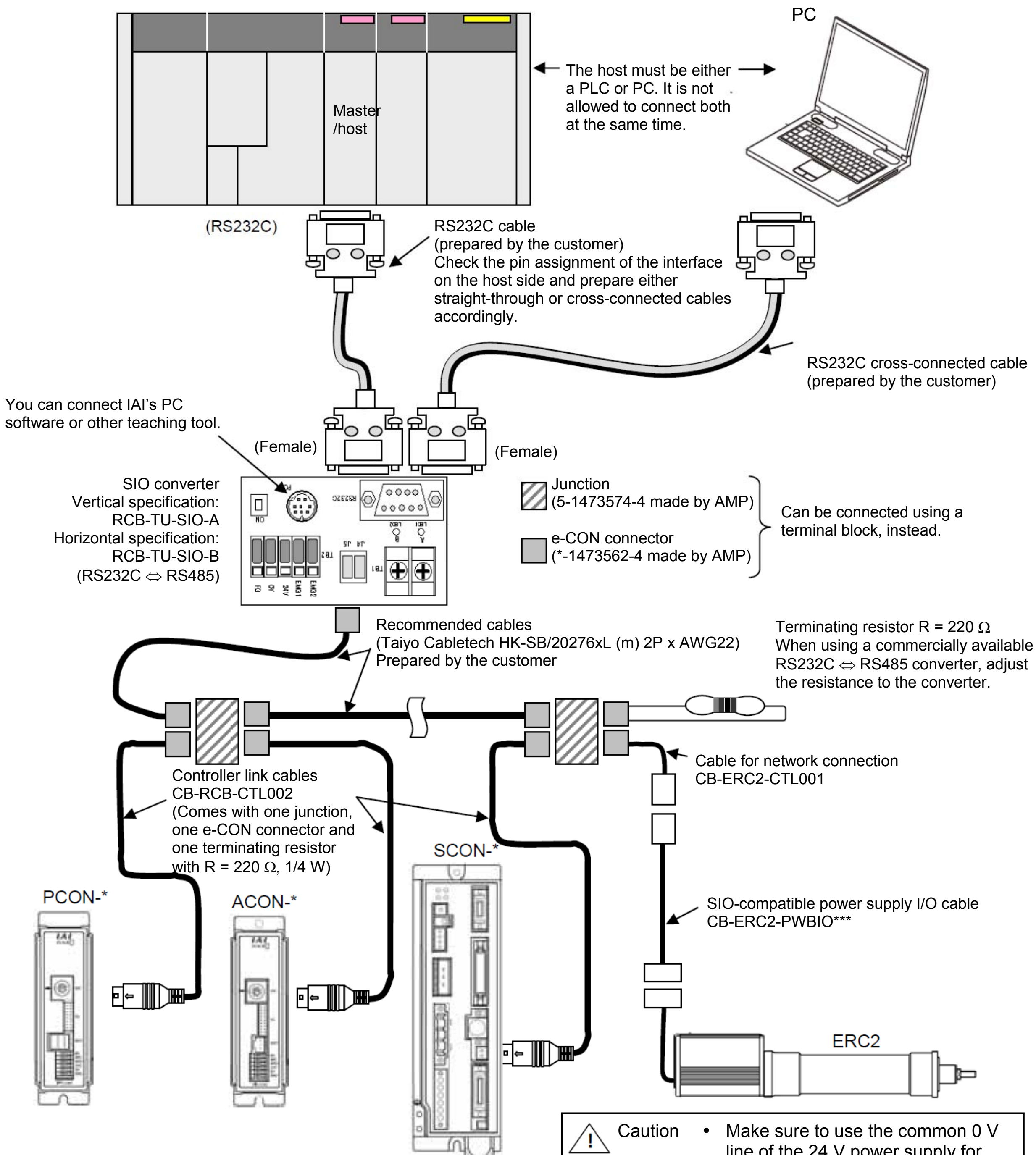


Fig. 2.1

3 Preparation for Communication

3.1 In Case the Host Uses RS232C Interface

(1) System configuration

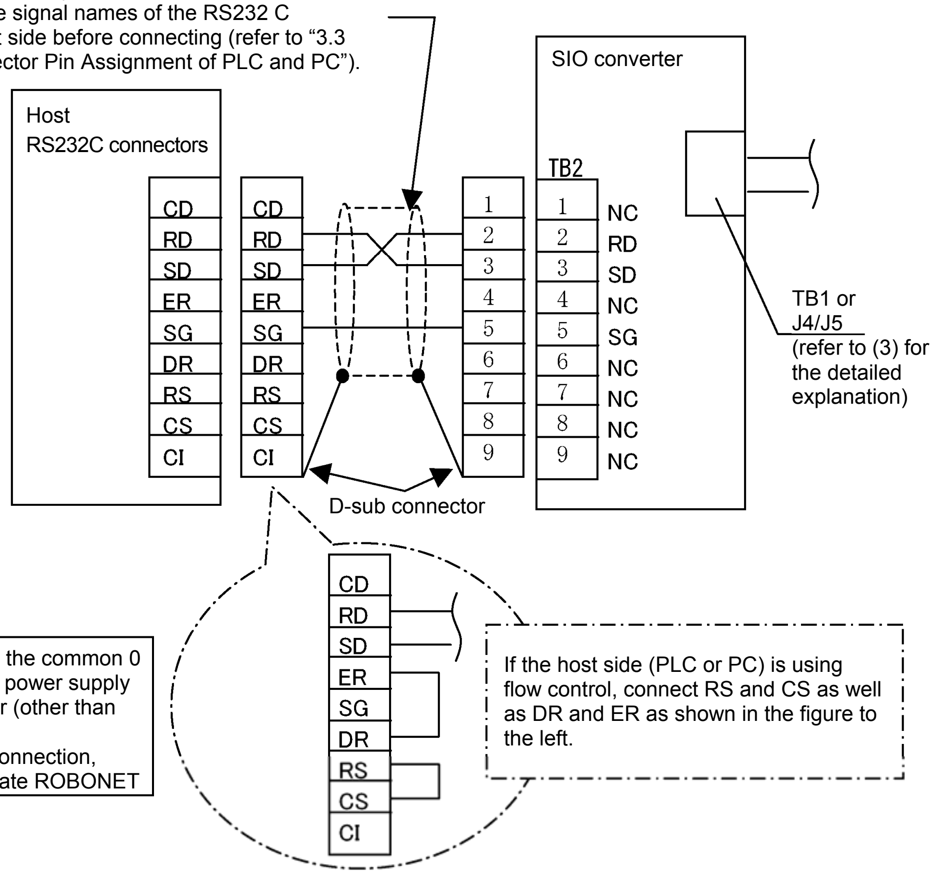


> ⚠ Caution
> - Make sure to use the common 0 V line of the 24 V power supply for each controller (other than SCON).
> - For ROBONET connection, refer to the separate ROBONET Operation Manual.

Fig. 3.1

(2) Wiring

RS232C cables (commercially available cables, etc.)
Make sure to check the signal names of the RS232 C connectors on the host side before connecting (refer to “3.3 Communication Connector Pin Assignment of PLC and PC”).



Caution: Make sure to use the common 0 V line of the 24 V power supply for each controller (other than SCON).
For ROBONET connection, refer to the separate ROBONET

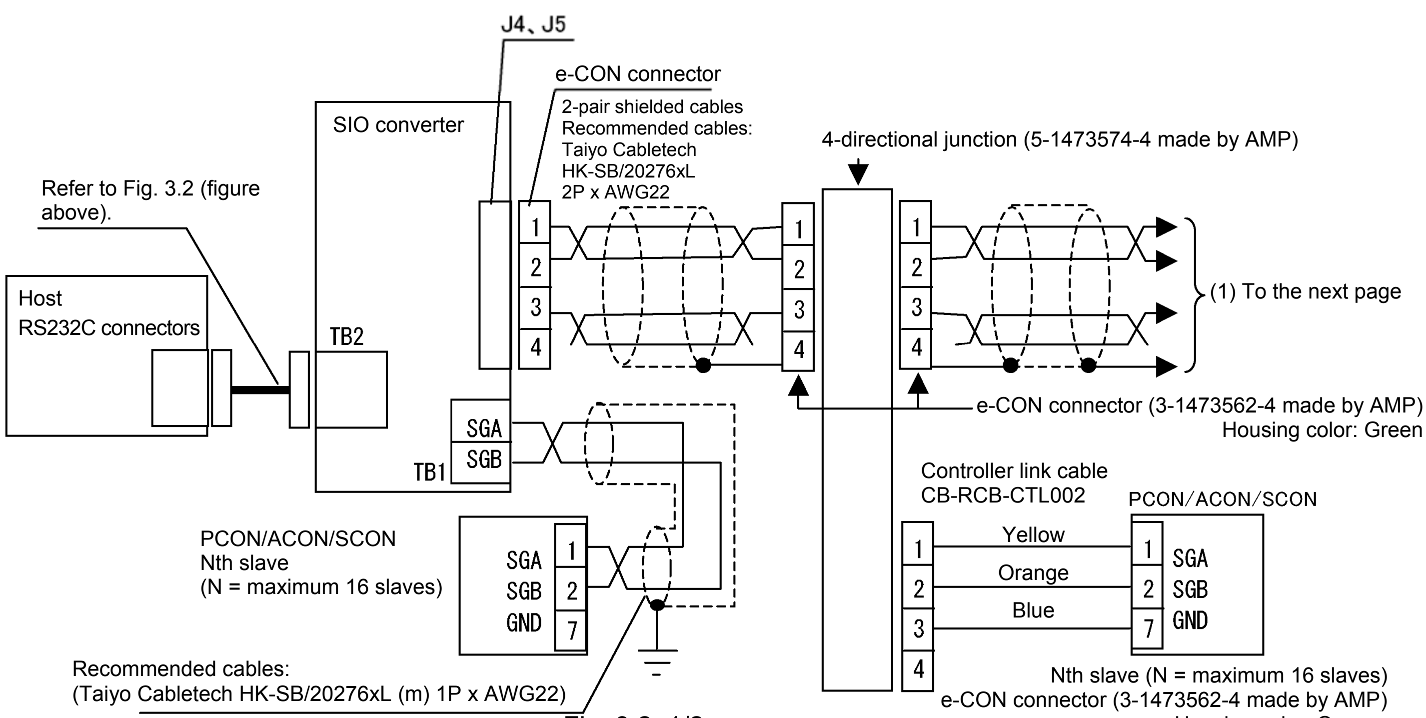


Fig. 3.2_1/2

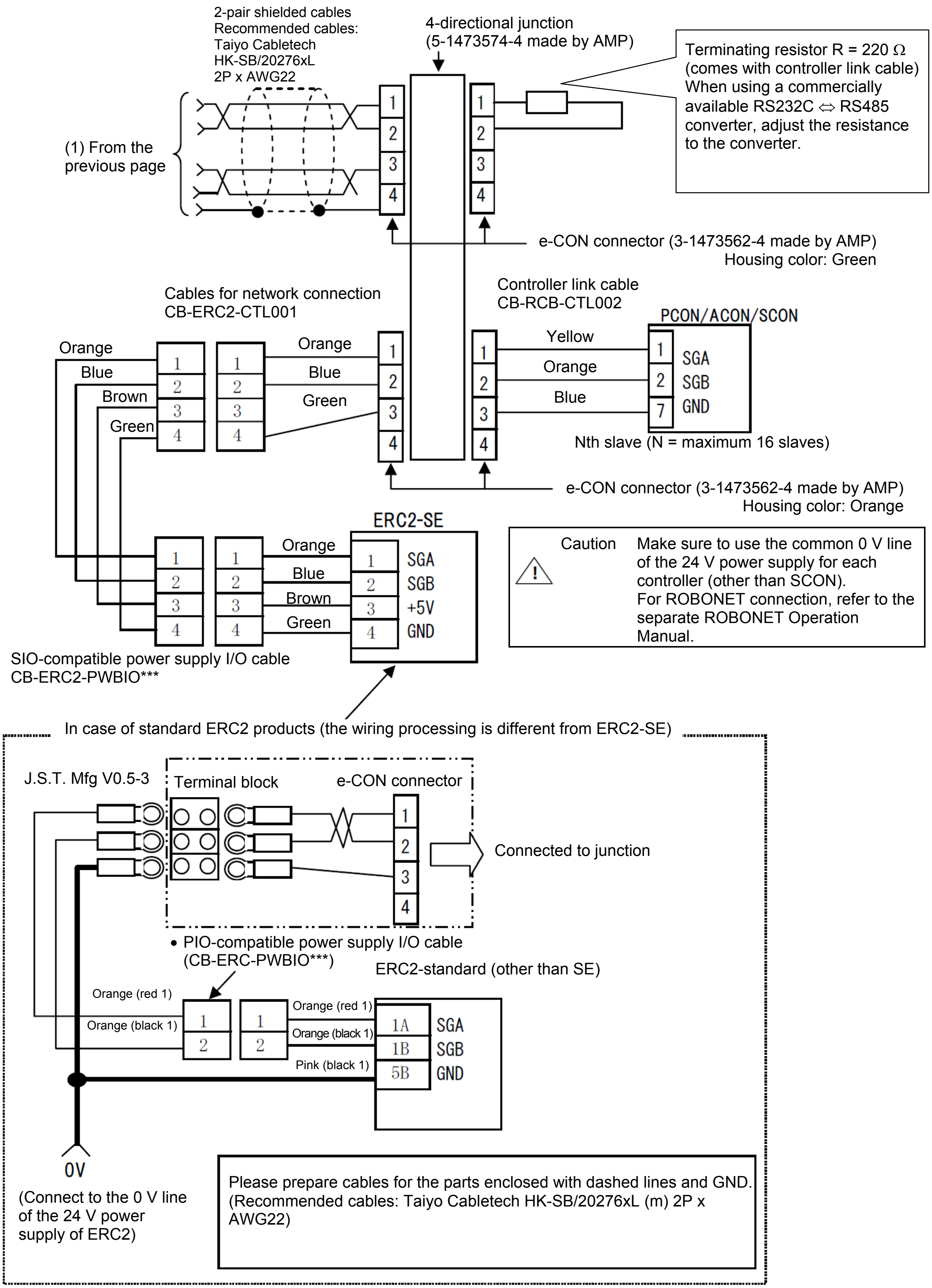

2-pair shielded cables
Recommended cables:
Taiyo Cabletech
HK-SB/20276xL
2P x AWG22
4-directional junction
(5-1473574-4 made by AMP)
Terminating resistor R = 220 Ω
(comes with controller link cable)
When using a commercially available RS232C ⇔ RS485 converter, adjust the resistance to the converter.
(1) From the previous page
e-CON connector (3-1473562-4 made by AMP)
Housing color: Green
Controller link cable
CB-RCB-CTL002
Cables for network connection
CB-ERC2-CTL001
PCON/ACON/SCON
Orange
Blue
Brown
Green
Yellow
SGA
SGB
GND
Nth slave (N = maximum 16 slaves)
e-CON connector (3-1473562-4 made by AMP)
Housing color: Orange
ERC2-SE
+5V
Caution
Make sure to use the common 0 V line of the 24 V power supply for each controller (other than SCON).
For ROBONET connection, refer to the separate ROBONET Operation Manual.
SIO-compatible power supply I/O cable
CB-ERC2-PWBIO***
In case of standard ERC2 products (the wiring processing is different from ERC2-SE)
J.S.T. Mfg V0.5-3
Terminal block
e-CON connector
Connected to junction
• PIO-compatible power supply I/O cable
(CB-ERC-PWBIO***)
ERC2-standard (other than SE)
Orange (red 1)
Orange (black 1)
Pink (black 1)
1A
1B
5B
0V
(Connect to the 0 V line of the 24 V power supply of ERC2)
Please prepare cables for the parts enclosed with dashed lines and GND.
(Recommended cables: Taiyo Cabletech HK-SB/20276xL (m) 2P x AWG22)

Fig. 3.2_2/2

(3) SIO converter (vertical specification: RCB-TU-SIO-A, horizontal specification: RCB-TU-SIO-B)

A RS232C ⇔ RS485 converter

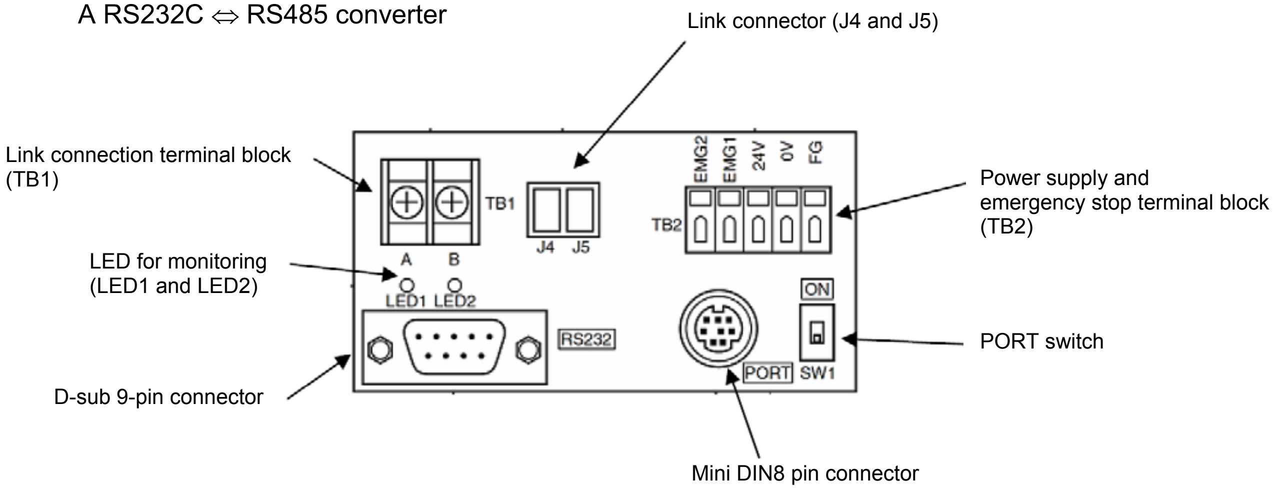


Fig. 3.3

⊙ Power supply and emergency stop terminal block (TB2)
- EMG1 and EMG2: Discrete outputs of the emergency stop switch of the teaching pendant
  EMG1 and EMG2 are connected to the emergency stop switch of the teaching pendant when the PORT switch is set to ON; EMG1 and EMG2 are short circuited when the switch is set to OFF.
- 24 V: Supply +24 V power (current consumption 0.1 A or less)
- 0 V: Supply 0 V power (use common 0 V for all 24 V DC-supplied controllers).
- FG: A terminal to which FG is connected

* Compatible wires: Single wire: ∅ 0.8 to 1.2 mm
  Twisted wire: AWG18 to 20 (strip length 10 mm)

⊙ Link connection terminal block (TB1)
A connector for link connection with an RC controller
- A: Connect to pin 1 (SGA) of the communication connector of the RC controller
- B: Connect to pin 2 (SGB) of the communication connector of the RC controller

⊙ D-sub 9 pin connector
A connector for connection with the master (host) side

⊙ Mini DIN8 pin connector
A connector for connection with teaching pendant or PC software

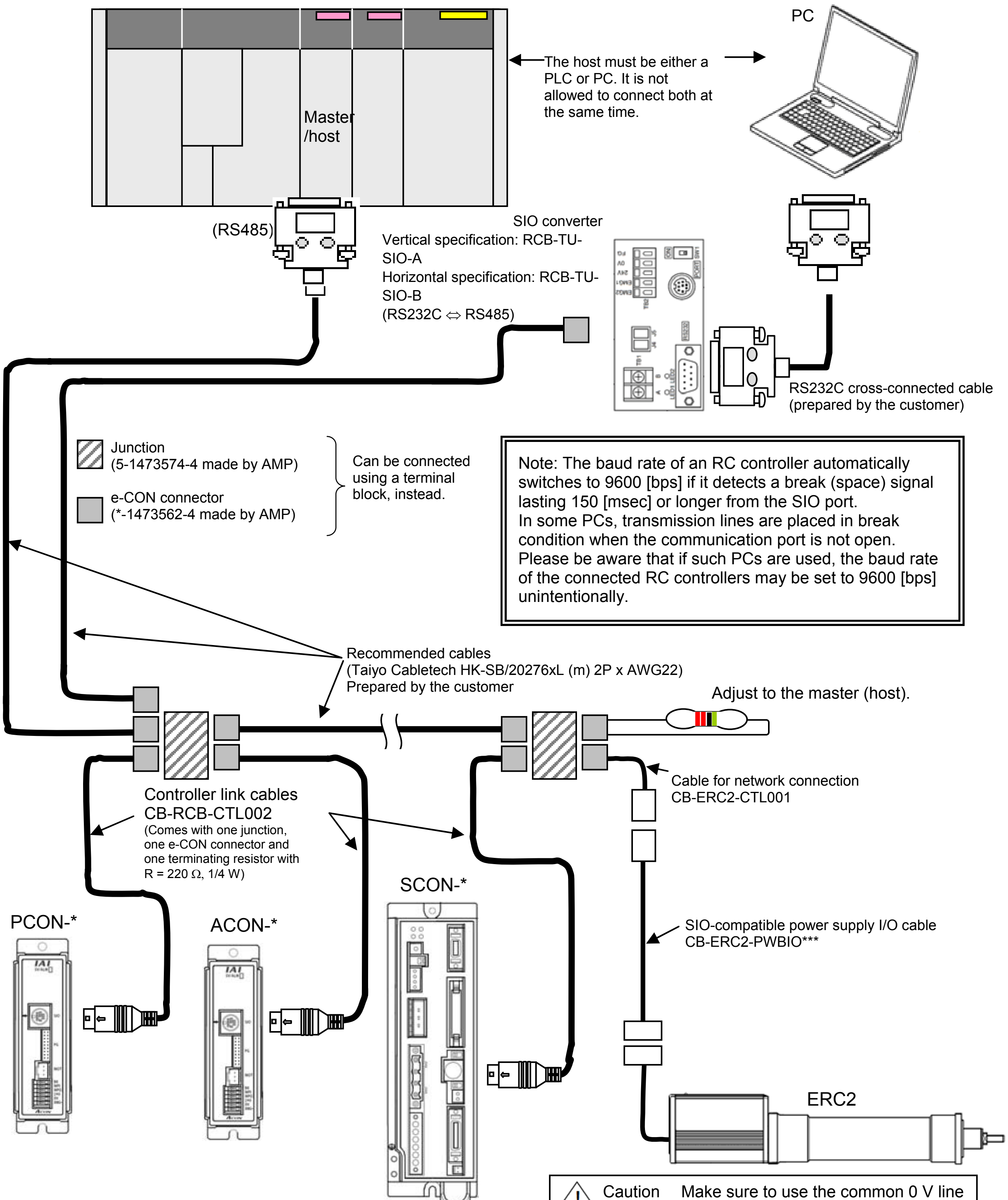
⊙ PORT switch
- ON: A teaching tool is used.
- OFF: A teaching tool is not used.

⊙ LED for monitoring (LED1 and LED2)
- LED1: Turns on/flashes when the RC controller is transmitting
- LED2: Turns on/flashes when the master (host) side is transmitting

⊙ Link connector (J4 and J5)
Connectors for link connection with an RC controller
An optional link cable (CB-RCB-CTL002) can be connected as is.

3.2 In Case the Host Uses RS485 Interface

(1) System configuration



Note: The baud rate of an RC controller automatically switches to 9600 [bps] if it detects a break (space) signal lasting 150 [msec] or longer from the SIO port.
In some PCs, transmission lines are placed in break condition when the communication port is not open.
Please be aware that if such PCs are used, the baud rate of the connected RC controllers may be set to 9600 [bps] unintentionally.

⚠ Caution	Make sure to use the common 0 V line of the 24 V power supply for each controller (other than SCON). For ROBONET connection, refer to the separate ROBONET Operation Manual.

Fig. 3.4

(2) Wiring

> ⚠ Caution: Make sure to use the common 0 V line of the 24 V power supply for each controller (other than SCON).
> For ROBONET connection, refer to the separate ROBONET Operation Manual.

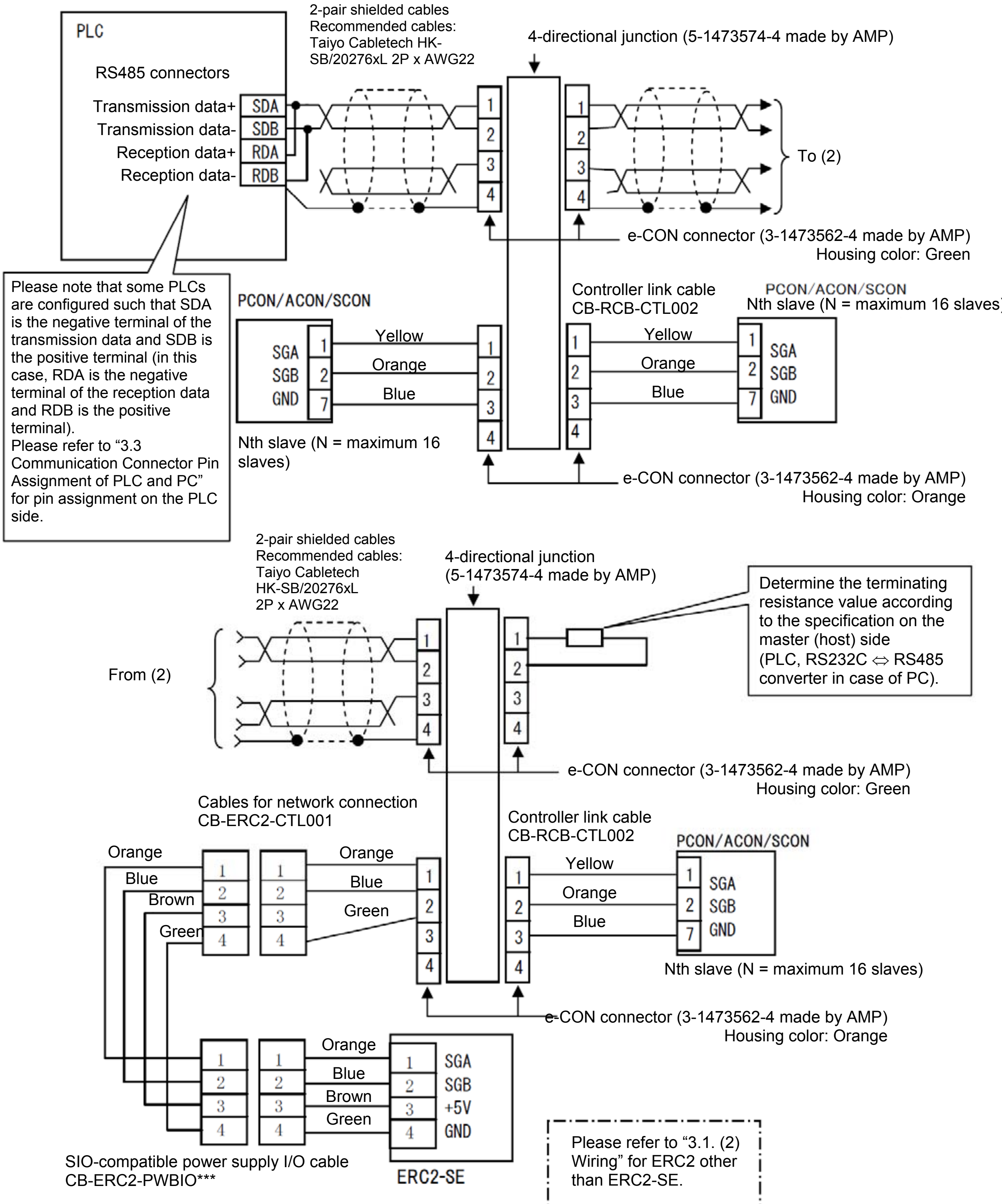
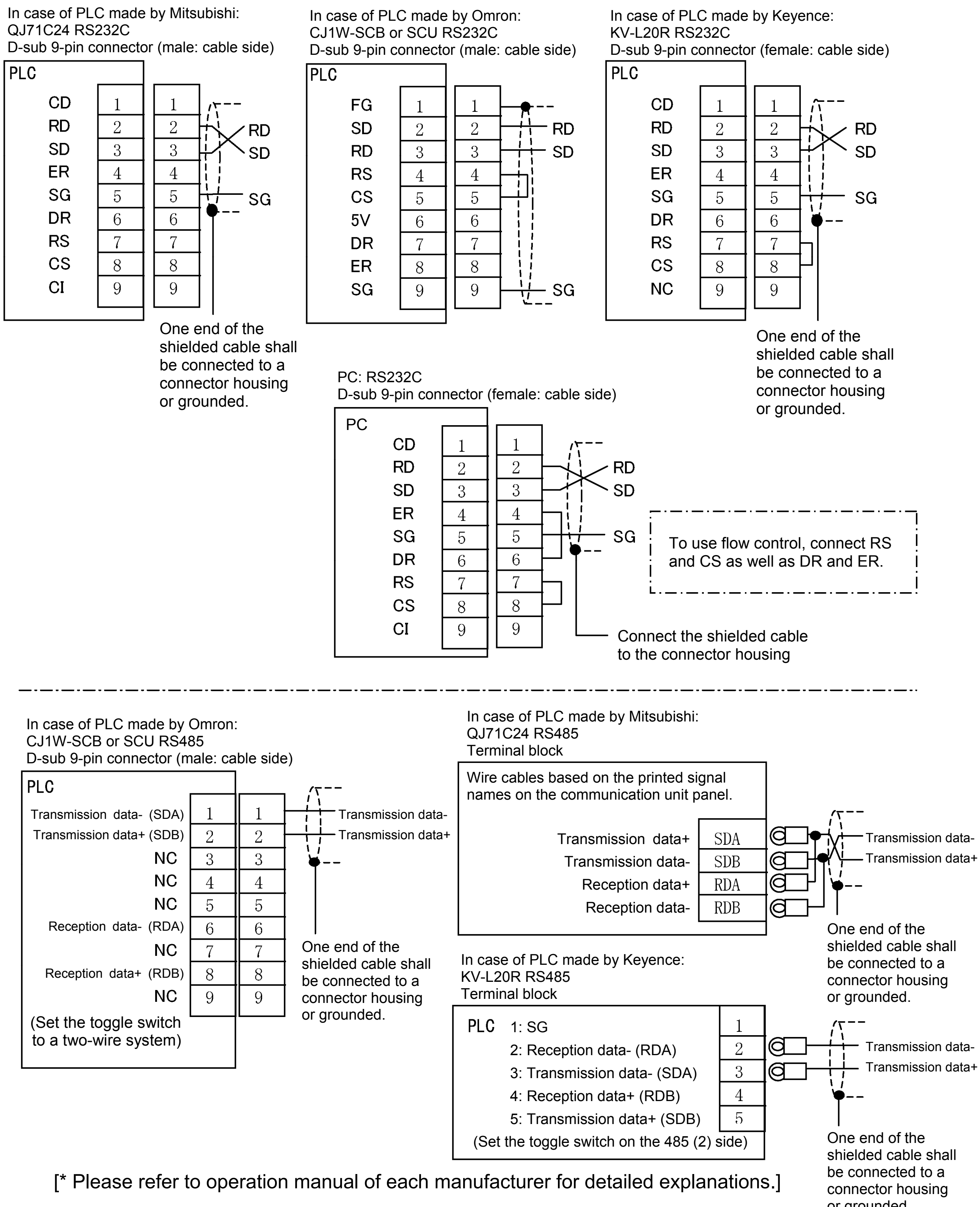


Fig. 3.5

3.3 Communication Connector Pin Assignment of PLC and PC (Reference)



[* Please refer to operation manual of each manufacturer for detailed explanations.]

Fig. 3.6

3.4 Various Setting before Starting Communication

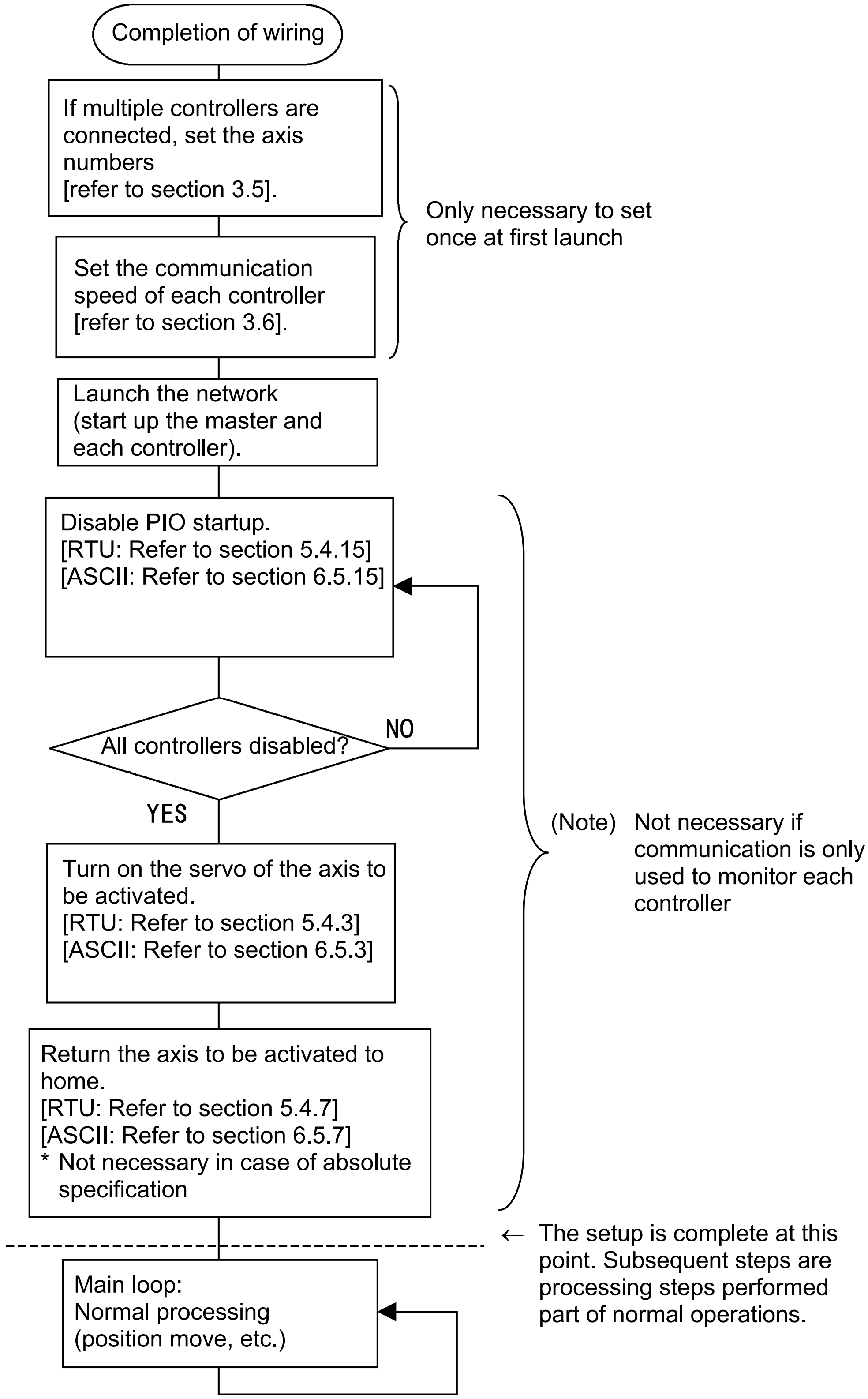


Fig. 3.7

3.5 Setting Axis Numbers

Set an axis number for each RC controller on the SIO link using hexadecimal digits from 0 to F_H, which is the number for the 16th axis.
If the panel surface of an RC controller has an axis number setting switch (ADRS) (PCON-C/CG/CF, ACON-C/CG, SCON-C/CA and ROBONET), adjust the arrow to point to the axis number using a flat bladed screwdriver (make sure that each axis number is unique).

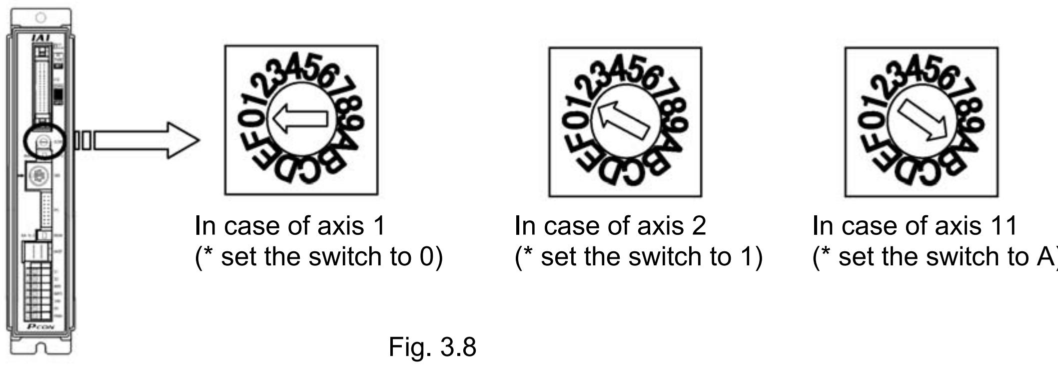


Fig. 3.8

On RC controllers having no axis number setting switches, use the PC software or other teaching pendant to set the axis number. In this example, how to set the axis number using the PC software is explained. [For information on how to set the axis number using your teaching pendant or teaching touch panel, refer to the operation manual for each tool (CON-T, CON-PT, RCM-E, RCM-T)].
Connect the PC to the SIO connector of the RC controller for which an axis number is to be set.

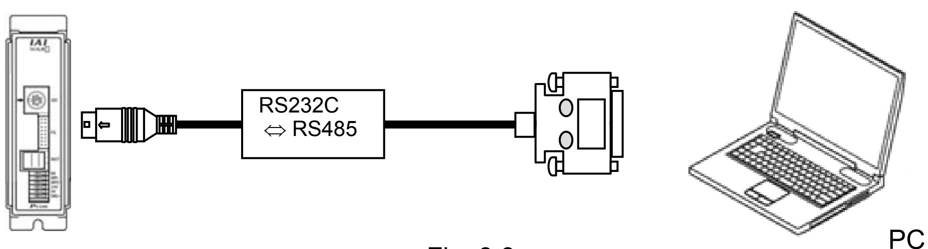


Fig. 3.9

Set the numbers using the following procedure.
[1] Start the RC connection software and select the [Setting (S)] menu.
[2] Select the [Controller Setting] menu item.
[3] Select the [Axis Number Assignment (N)] menu item.
[4] Enter an axis number (0 to 15) in the axis number table (*)
(make sure to enter a number that does not coincide with existing axis numbers).

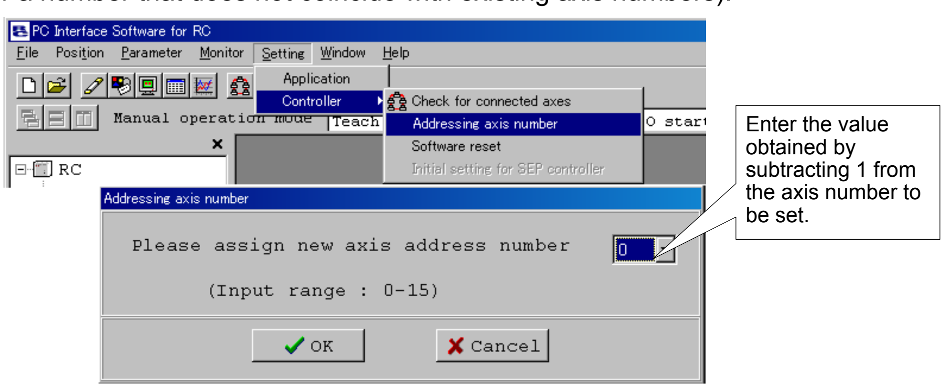


Fig. 3.10

3.6 Setting Controller Communication Speed

In order to perform communication, the communication speed of the PLC and each RC controller must match.
Set the communication speed according to the procedure explained in sections 3.6.1 and 3.6.2.
[For the settings on the host side, refer to the operation manual for your host equipment.]
Please be aware that the wiring is different depending on the system configuration.

3.6.1 Setting Wiring and Hardware for Each System

(1) In case of using a PC as the master (host) controller
It is possible to make settings without changing the current connection. In case of RC controllers equipped with a mode toggle switch (PCON-C/CG/CF, ACON-C/CG and SCON-C/CA), set the mode toggle switch to MANU before making the settings.

(2) In case a PLC is used as the master (host) controller connected via RS232C
Connect a PC as master (host) controller instead of the PLC (refer to Figure 3.1). At this point, disconnect the PLC from the SIO converter and connect the PC to the teaching port of the SIO converter [refer to section 3.1 (3)] using the cable supplied with the PC software. In case of RC controllers with a mode toggle switch (PCON-C/CG/CF, ACON-C/CG and SCON-C/CA), set the mode toggle switch to MANU.

(3) In case a PLC is used as the master (host) controller connected via RS485
Connect a PC directly to each RC controller in the same way as for setting axis numbers. In case of RC controllers with a mode toggle switch (PCON-C/CG/CF, ACON-C/CG and SCON-C/CA), set the mode toggle switch to MANU.

(4) When a ROBONET is connected
To set up your ROBONET, connect the cable supplied with your PC software to the teaching port on the GateWayR unit. Set the MODE selector switch on the GateWayR unit to "MANU."

3.6.2 Setting Communication Speed

Set the communication speed using the following procedure.
(Note) On ROBONET controllers, the baud rate is set using the ROBONET gateway parameter setting tool. [For details, refer to the separate ROBONET Operation Manual.]

[1] Start the RC connection software and select [Edit (E)] from the [Parameters (P)] menu.

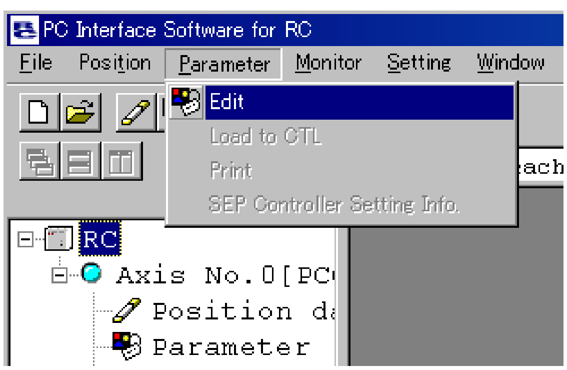


Fig. 3.11

[2] Select the axis number of the controller to be changed.

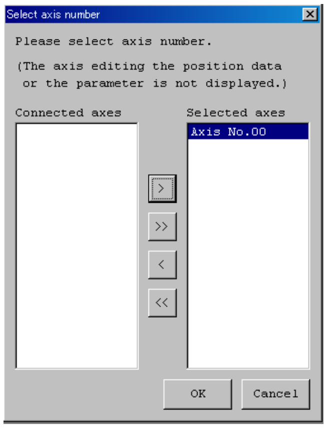
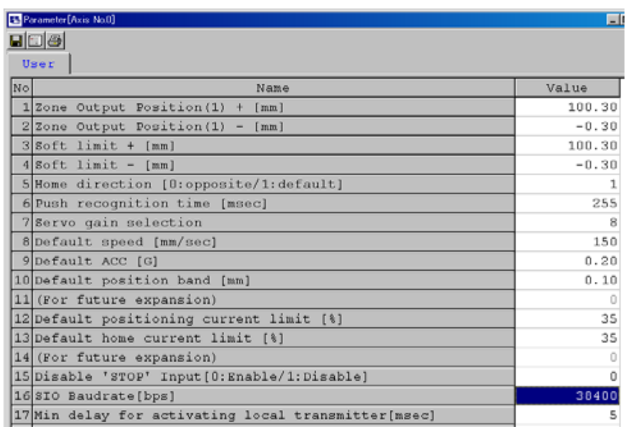


Fig. 3.12

[3] Set parameter No. 16, SIO communication speed.



No	Name	Value
1	Zone Output Position(1) + [mm]	100.30
2	Zone Output Position(1) - [mm]	-0.30
3	Soft limit + [mm]	100.30
4	Soft limit - [mm]	-0.30
5	Home direction [0:opposite/1:default]	1
6	Push recognition time [msec]	255
7	Servo gain selection	8
8	Default speed [mm/sec]	150
9	Default ACC [G]	0.20
10	Default position band [mm]	0.10
11	(For future expansion)	0
12	Default positioning current limit [%]	35
13	Default home current limit [%]	35
14	(For future expansion)	0
15	Disable 'STOP' Input[0:Enable/1:Disable]	0
16	SIO Baudrate[bps]	38400
17	Min delay for activating local transmitter[msec]	5

Fig. 3.13

4 Communication

4.1 Message Transmission Timing

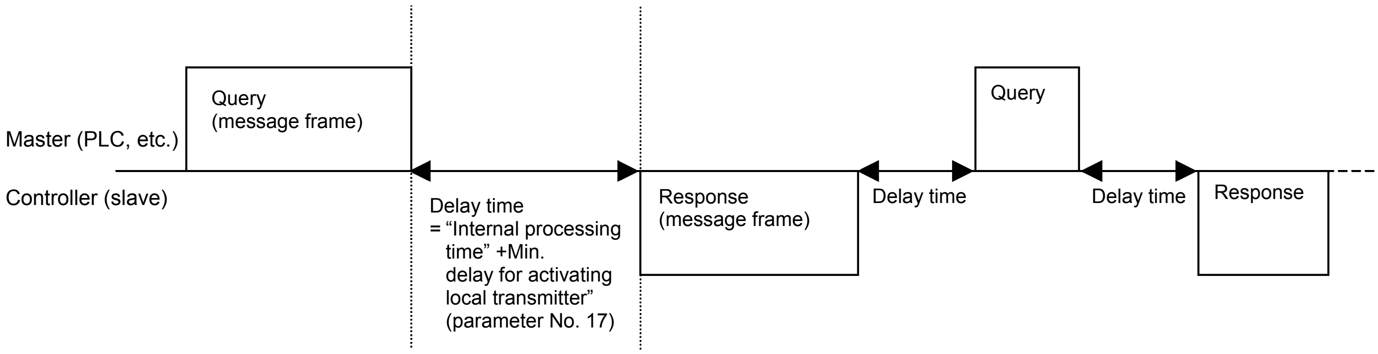


Fig. 4.1

The basic transmission control procedure consists of the master sending a query, and the RC controller that received the query sending a response, which are considered one unit.
The delay time after a query message is received until a response message is sent is calculated as the total sum of parameter No. 17 "Min. delay for activating local transmitter" (default value 5 ms) and the internal processing time (refer to the table below).
After receiving a query message, the RC controller waits for the "min. delay for activating local transmitter." Once this delay time elapses, the controller will activate the transmitter and start sending a response message. The master must enable the receive function of its own station within the aforementioned delay time after sending a query message.
After sending a response message, the RC controller immediately prepares to receive the next query.

Internal processing time (the processing time varies depending on the accessed category)

Item	Time
Read/write a register other than those in the low-speed memory area	1 msec max.
Position data (1 position) Read	4 msec max.
Position data (1 position) Write	15 msec max.
Position data (1 position) Read/write	18 msec max.
Position data (9 positions) Read	9 msec max.
Position data (9 positions) Write	90 msec max.
Position data (9 positions) Read/write	98 msec max.

4.2 Timeout and Retry

After sending a query, the host waits for a response from the controller (except when the query that has been sent is a broadcast query).
If the elapsed time after sending a command until a response is received exceeds the timeout value (Tout), the host may send the command again to reestablish communication. If the number of retries exceeds three times, it means that an irremediable communication error has occurred.
The method for calculating the timeout value (Tout) is explained below.

1. Timeout value (Tout)

   Tout = To + α + (10 x Bprt/Kbr) [ms]

   To: Internal processing time* x Safety factor (3)
   α: Min. delay for activating local transmitter [ms] (default value of parameter No. 17 is 5 ms)
   Kbr: Baud rate [kbps]
   Bprt Response message bytes + 8

⚠ Caution The internal processing time varies depending on the category of the register to be accessed. The processing time required for each action is listed in the table below.

Item	Maximum time [ms]
Read/write a register other than those in the low-speed memory area	1
Position data (1 position) Read	4
Position data (1 position) Write	15
Position data (1 position) Read/write	18
Position data (9 positions) Read	9
Position data (9 positions) Write	90
Position data (9 positions) Read/write	98

2. Number of Retries

   Nrt = 3 (note that setting of the number of retries is mandatory)

4.3 Internal Addresses and Data Structure of RC Controller

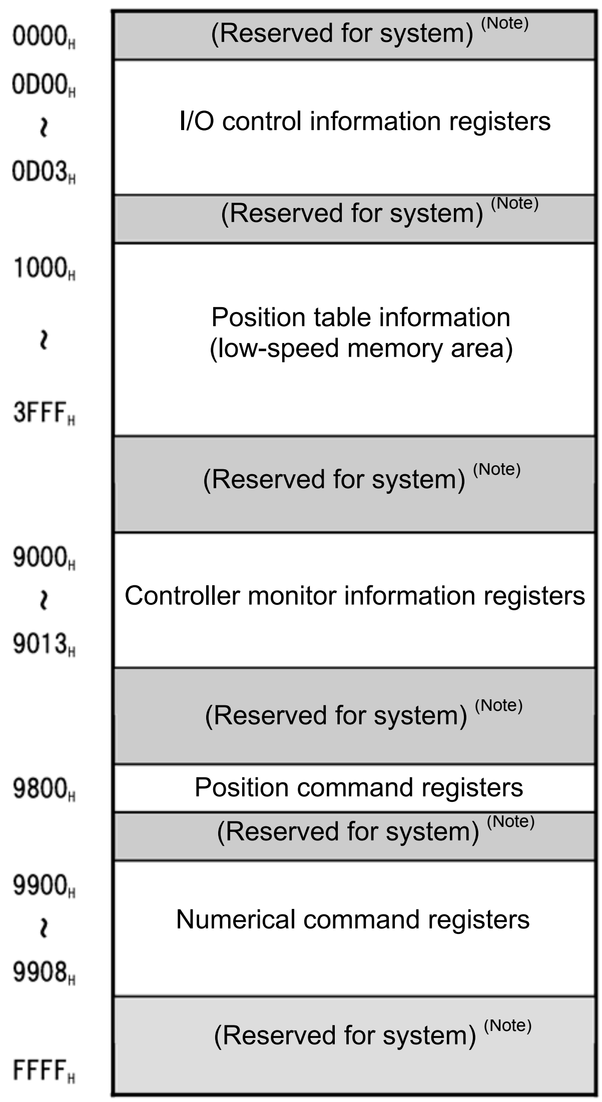
The memory area in your RC controller consists of the Modbus register area read/written in units of words and the Modbus status are written in units of bits (coils).

Memory area	Access unit	Address range	Function	
			Code* (Note)	Function
Modbus register [Refer to 4.3.1 and 4.3.2.]	Word	0D00~9908$_H$	03 $_H$	Read holding resisters
			06 $_H$	Write holding registers
			10 $_H$	Write multiple holding registers at the same time
Modbus status [Refer to 4.3.3 and 4.3.4.]	Bit	0100~043F$_H$	05 $_H$	Write coils

(Note) Function codes explained in this manual

4.3.1 Structure of Mudbus Registers

The layout of the Mudbus registers is shown below.



(Note) Areas reserved for the system cannot be used for communication.

4.3.2 Details of Mudbus Registers

Address [hex]	Area name	Description		Symbol	Reference page RTU		Reference page ASCII	
0000 to 0CFF	Reserved for system							
0D00	I/O control information category	Device control register 1		DRG1	123	29	234	29
0D01		Device control register 2		DRG2		30		30
0D03		Position number specification register		POSR		31		31
0D04 to 0FFF	Reserved for system							
1000 to 3FFF	Position table information (low-speed memory area)	Offset [hex]						
		$+0000_H$	Target position	PCMD	145	147	225	258
		$+0002_H$	Positioning band	INP				
		$+0004_H$	Speed command	VCMD				
		$+0006_H$	Individual zone boundary +	ZNMP				
		$+0008_H$	Individual zone boundary -	ZNLP				
* Detailed addresses can be calculated using the formula to the right. →		$+000A_H$	Acceleration command	ACMD				
		$+000B_H$	Deceleration command	DCMD		148		259
		$+000C_H$	Push-current limiting value	PPOW				
		$+000D_H$	Load current threshold	LPOW				
		$+000E_H$	Control flag specification	CTLF				
	* Address = 1000_H + (16 x position No.) + offset							
4000 to 8FFF	Reserved for system							
9000	Controller monitor information category	Current position register		PNOW	51	54	162	165
9002		Present alarm code register		ALMC		56		167
9003		Input port register		DIPM		58		169
9004		Output port register		DOPM		62		173
9005		Device status 1 register		DSS1		66		177
9006		Device status 2 register		DSS2		68		179
9007		Expansion device status register		DSSE		70		181
9008		System status register		STAT		72		183
900A		Current speed monitor register		VNOW		74		185
900C		Current ampere monitor register		CNOW		76		187
900E		Deviation monitor register		DEVI		78		189
9010		System timer register		STIM		80		191
9012		Special input port register		SIPM		82		193
9013		Zone status register		ZONS		84		195
9014		Position complete number status register		POSS		86		197
9015 to 901D	Reserved for system							
901E	Controller monitor information category	Force feedback data (SCON-CA only)		FBFC	51	88	162	199
9015 to 97FF	Reserved for system							
9800	Position command category	Position movement command register		POSR	123	31	234	31
9801 to 98FF	Reserved for system							
9900	Numerical value command category	Target position coordinate specification register		PCMD	127	129	238	240
9902		Positioning band specification register		INP				
9904		Speed specification register		VCMD				
9906		Acceleration/deceleration speed specification register		ACMD		130		241
9907		Push-current limiting value		PPOW				
9908		Control flag specification register		CTLF		131		242
9909 to FFFF	Reserved for system							

(1) Data of device control register 1 (Address = $0D00_H$) (DRG1)

Bit	Symbol	Name	Function
15	EMG	EMG operation specification	0: Emergency stop not actuated 1: Emergency stop actuated Changing this bit to 1 will switch the controller to the emergency stop mode. Take note that the drive source will not be cut off. (The ALM LED on the controller will not illuminate.)
14	SFTY	Safety speed command	0: Disable safety speed 1: Enable safety speed Changing this bit to 1 will limit the speeds of all movement commands to the speed specified by user parameter No. 35, “Safety speed.”
13	-	Cannot be used	
12	SON	Servo ON command	0: Servo OFF 1: Servo ON Changing this bit to 1 will turn the servo ON. However, the following conditions must be satisfied: • Device status register 1 (5.3.6 or 6.4.6): The EMG status bit is 0. • Device status register 1 (5.3.6 or 6.4.6): The major failure status is 0. • Device status register 2 (5.3.7 or 6.4.7): The enable status bit is 1. • System status register (5.3.9 or 6.4.9): The auto servo OFF status is 0.
11 to 9		Cannot be used	
8	ALRS	Alarm reset command	0: Normal 1: Alarm will reset Present alarms will be reset upon detection of a rising edge for this bit (this bit: 0 → 1). Note, however, that if any of the causes for the alarm has not been removed, the same alarm will be generated again. If a rising edge is detected for this bit (this bit: 0 → 1) during a pause, the remaining travel will be canceled.
7	BKRL	Brake forced-release command	0: Normal 1: Forcibly release brake You can forcibly release the brake by setting this bit to 1.
6	-	Cannot be used	
5	STP	Pause command	0: Normal 1: Pause command All motor movement is inhibited while this bit is 1. If this bit turns 1 while the actuator is moving, the actuator will decelerate to a stop. When the bit is set to 0 again thereafter, the actuator will resume the remaining travel. If this bit is turned 1 while the actuator is performing a home return, the movement command is held until the actuator reverses upon contact. When the bit turns 0 thereafter, the actuator will complete the remaining home return operation automatically. However, make sure you perform a home return again after the actuator reverses upon contact.
4	HOME	Home return command	0: Normal 1: Home return command Home return will start when a rising edge is detected for this bit (this bit: 0 → 1). Once the home return is completed, the HEND bit will become 1. You can input a home return command again even if the actuator has already completed a home return.
3	CSTR	Positioning start command	0: Normal 1: Position start command When a rising edge is detected for this bit (this bit: 0 → 1), the actuator will move to the target position of the position number stored in the position number specification register (POSR:$0D03_H$). If this bit remains 1, a position complete will not be output even when the actuator enters the positioning band (return to the normal status by writing 0 to this bit). If this command is executed before home return has been performed at least once after the power was turned on (the HEND bit is 0), the actuator will perform home return and then start moving to the target position. * Set the target position, speed, etc., in the position table of the controller beforehand.
2 to 0	-	Cannot be used	

(2) Data of device control register 2 (Address = $0D01_H$) (DRG2)

Bit	Symbol	Name	Function
15	-	Cannot be used	
14	JISL	Jog/inch switching	0: Jog 1: Inching When this bit is 0, the jog operation is selected. When this bit is 1, the inching operation is selected. If this bit turns 1 while the actuator is jogging, the actuator will accelerate to a stop. While the actuator is inching, turning this bit 0 will have no effect and the actuator will continue with the inching operation. The setting of this bit is not reflected in any jog/inching operation set from the teaching tool.
13	-	Cannot be used	
12	-	Cannot be used	
11	MOD	Teaching mode command	0: Normal operation mode 1: Teaching mode Changing this bit to 1 will switch the controller to the teaching mode.
10	TEAC	Position data load command	0: Normal 1: Position data load command The current position data will be written to the position number specified by the position number specification register if 1 is written to this bit while the 11th bit of the teach mode command is 1 (teaching mode). The current position data is loaded to the position data line specified by the position number specification register. If the position number under which the data is loaded is an empty position, meaning that no data is currently set, the data fields other than target position (such as positioning band, etc.) will be automatically populated by the default values of the respective parameters. Make sure that after this bit is set to 1, it will remain 1 for at least 20 ms.
9	JOG+	Jog+ command	0: Normal 1: Jog+ command • The actuator jogs in the direction opposite home as long as this bit is 1 if the 14th JISL bit is 0. The speed and acceleration/deceleration match the specifications in user parameter No. 26 “PIO jog speed” and rated acceleration/deceleration speed. If this bit is set to 0 or the 8th bit of the jog-command is changed to 1, the actuator will decelerate to a stop. • If a positive edge (this bit: 0 → 1) is detected for the jog+ command while the 14th JISL bit is 0, the actuator inches in the direction opposite home. The speed, travel and acceleration/deceleration speed match the specifications in user defined parameter No. 26 (PIO jog speed), user parameter No. 48 (PIO inching distance) and rated jog acceleration/deceleration, respectively.
8	JOG-	Jog- command	0: Normal 1: Jog- command • The actuator jogs in the direction of home as long as this bit is 1 if the 14th JISL bit is 0. The speed and acceleration/deceleration speed match the specifications in user parameter No. 26 “PIO jog speed” and rated acceleration/deceleration speed. If this bit is set to 0 or the 9th bit of the jog-command is changed to 1, the actuator will decelerate to a stop. • If a positive edge (this bit: 0 → 1) is detected for the jog+ command while the 14th JISL bit is 0, the actuator inches in the direction of home. The speed, travel and acceleration/deceleration speed match the specifications in user defined parameter No. 26 (PIO jog speed), user parameter No. 48 (PIO inching distance) and rated jog acceleration/deceleration, respectively.
7	ST7	Start position 7	(If either of these bits is enabled) The actuator moves to the position of the specified position number. These bits are only valid when PIO patterns 4 or 5 (solenoid valve mode) is selected. The move is started if either of the ST0 to ST7 bits is set to 1 (this bit: 0 → 1). If a position other than the enabled start poison is selected, the alarm “085 Position No. error at moving” is generated. You can select the signal input method as “Level” or “Edge” in user parameter No. 27, “Movement command type.” If multiple positions are entered at the same time, the smallest number takes the priority.
6	ST6	Start position 6	
5	ST5	Start position 5	
4	ST4	Start position 4	
3	ST3	Start position 3	
2	ST2	Start position 2	
1	ST1	Start position 1	
0	ST0	Start position 0	

(3) Data of position number specification registers (Address = $0D03_H$) (POSR)
Position movement command register details (Address = 9800_H) (POSR)

Bit	Symbol	Name	Function
15	-	Cannot be used	
14	-	Cannot be used	
13	-	Cannot be used	
12	-	Cannot be used	
11	-	Cannot be used	
10	-	Cannot be used	
9	PC512	Position command bit 512	These bits indicate position numbers to be moved using binary codes. Note that the maximum position number varies depending on the model and PIO pattern. [When address = $0D03_H$ is used] After specifying a position number, set the CSTR (start signal) of device control register 1 to 1, and the actuator will move to the specified position. [Refer to 5.5.1 or 6.6.1.] [When address = 9800_H is used] This register is such that once a position number is specified, the actuator will move to the specified position. You need not set the CSTR (start signal).
8	PC256	Position command bit 256	
7	PC128	Position command bit 128	
6	PC64	Position command bit 64	
5	PC32	Position command bit 32	
4	PC16	Position command bit 16	
3	PC8	Position command bit 8	
2	PC4	Position command bit 4	
1	PC2	Position command bit 2	
0	PC1	Position command bit 1	

(4) Data of device status register 1 (Address = 9005_H) (DSS1)

Bit	Symbol	Name	Function
15	EMGS	EMG status	0: Emergency stop not actuated 1: Emergency stop actuated This bit indicates whether or not the controller is currently in the emergency stop mode due to an emergency stop input, cutoff of the drive source, etc.
14	SFTY	Safety speed enabled status	0: Safety status disabled 1: Safety status enabled Enable/disable the safety speed of the controller using the “safety speed command bit” of device control register 1.
13	PWR	Controller ready status	0: Controller busy 1: Controller ready This bit indicates whether or not the controller can be controlled externally. Normally this bit does not become 0 (busy).
12	SV	Servo ON status	0: Servo OFF 1: Servo ON The servo ON status is indicated. After a servo ON command is issued, this bit will remain 0 until the servo ON delay time set by a parameter elapses. If the servo cannot be turned ON for some reason even after a servo ON command is received, this bit will remain 0. The RC controller does not accept any movement command while this bit is 0.
11	PSFL	Missed work part in push-motion operation	0: Normal 1: Missed work part in push-motion operation This bit turns 1 when the actuator has moved to the end of the push band without contacting the work part (= the actuator has missed the work part) according to a push-motion operation command. Operation commands other than push-motion do not change this bit.
10	ALMH	Major failure status	0: Normal 1: Major failure alarm present This bit will turn 1 if any alarm at the cold start level or operation cancellation level is generated. Alarms at the operation cancellation level can be reset by using an alarm reset command, but resetting alarms at the cold start level requires turning the power supply off and then on again.
9	ALML	Minor failure status	0: Normal 1: Minor failure alarm present This bit will turn 1 when a message level alarm is generated.
8	ABER	Absolute error status	0: Normal 1: Absolute error present This bit will turn 1 if an absolute error occurs in case the absolute specification is set.
7	BKRL	Brake forced-release status	0: Brake actuated 1: Brake released This bit indicates the status of brake operation. Normally the bit remains 1 while the servo is ON. Even when the servo is OFF, changing the “brake forced-release command bit” in device control register 1 to 1 will change this bit to 1.
6	-	Cannot be used	
5	STP	Pause status	0: Normal 1: Pause command active This bit remains 1 while a pause command is input. If the PIO/Modbus Switch Setting (5.4.16 or 6.5.16) is PIO enabled, paused PIO signals are monitored (set the switch to AUTO in case of RC controllers with a mode toggle switch). If Modbus is enabled, the Pause Commands (5.4.6 or 6.5.6) are monitored.
4	HEND	Home return completion status	0: Home return not yet complete 1: Home return complete This bit will become 1 when home return is completed. In case the absolute specification is set, the bit is set to 1 from the startup if absolute reset has been completed. If a movement command is issued while this bit is 0, an alarm will generate.
3	PEND	Position complete status	0: Positioning not yet complete 1: Position complete This bit turns 1 when the actuator has moved close enough the target position and entered the positioning band. It also turns 1 when the servo turns on after the actuator has started, because the controller recognizes that the actuator has completed a positioning to the current position. This bit will also become 1 during the push-motion operation as well as at the completion.
2	CEND	Load cell calibration complete	0: Calibration not yet complete 1: Calibration complete This bit turns 1 when the load cell calibration command (CLBR) has been successfully executed.
1	CLBS	Load cell calibration status	0: Calibration not yet complete 1: Calibration complete Regardless of whether or not a load cell calibration command has been issued, this bit is 1 as long as a calibration has completed in the past.
0	-	Cannot be used	

(5) Data of device status register 2 (Address = 9006_H) (DSS2)

Bit	Symbol	Name	Function
15	ENBS	Enable	0: Disabling state (an operation stop, the servo OFF) 1: When the teaching tool which carries the enabling SW in the model which carries an enabling state (normal operation) enabling function is connected, the state of the enabling SW is shown. (Note) From the model which does not carry the time of AUTO mode, or an enabling function, it is fixation 1.
14	-	Cannot be used	
13	LOAD	Load output judgment status	0: Normal 1: Load output judgment If a load current threshold or check range (individual zone boundaries: only supported by PCON-CF) is set when a movement command is issued, this bit indicates whether or not the motor current has reached the threshold inside the check range. This bit maintains the current value until the next position command is received.
12	TRQS	Torque level status	0: Normal 1: Torque level achieved This bit turns 1 when the current has reached a level corresponding to the specified push torque during a push-motion operation. Since this bit indicates a level, its status will change when the current level changes.
11	MODS	Teaching mode status	0: Normal operation mode 1: Teaching mode This bit becomes 1 when the teaching mode is selected by the “teach mode command bit” of device control register 2.
10	TEAC	Position-data load command status	0: Normal 1: Position data load complete Setting the “position-data load command bit” in device control register 2 to 1 will change this bit to 0. This bit will turn 1 once position data has been written to the EEPROM successfully.
9	JOG+	Jog+ status	0: Normal 1: “Jog+” command active This bit becomes 1 while the “jog+ command bit” of device control register 2 is selected.
8	JOG-	Jog- status	0: Normal 1: “Jog-“ command active This bit becomes 1 while the “jog- command bit” of device control register 2 is selected.
7	PE7	Position complete 7	These bits output a position complete number as a binary value in PIO pattern 4 or 5 (solenoid valve mode). Each of these bits turns 1 when the actuator has completed a position movement and become close enough to the target position by entering the positioning band according to a position movement command (ST0 to ST7 in device control register 2). Although the bit turns 0 once the servo is turned off, when the servo is turned on again the bit will turn 1 if the actuator is still within the positioning band of the specified command position data. Moreover, they will become 1 when push-motion is completed or missed in push-motion operation.
6	PE6	Position complete 6	
5	PE5	Position complete 5	
4	PE4	Position complete 4	
3	PE3	Position complete 3	
2	PE2	Position complete 2	
1	PE1	Position complete 1	
0	PE0	Position complete 0	

(6) Data of expansion device status register (Address = 9007_H) (DSSE)

Bit	Symbol	Name	Function
15	EMGP	Emergency stop status	0: Emergency stop input OFF 1: Emergency stop input ON This bit indicates the status of the emergency stop input port.
14	MPUV	Motor voltage low status	0: Normal 1: Motor drive source cut off This bit becomes 1 if there is no input from the motor drive power supply.
13	RMDS	Operation mode status	0: AUTO mode 1: MANU mode This bit becomes 1 when the RC controller is in the MANU mode. Note that the controller is always in the MANU mode in cases of models not equipped with an operation mode switch (ERC2, PCON-SE, ACON-SE, PCON-CY and ACON-CY).
12	-	Cannot be used	
11	GHMS	Home return status	0: Normal 1: Home return This bit remains 1 for as long as home return is in progress. This bit will be 0 in other cases.
10	PUSH	Push-motion operation in progress	0: Normal 1: Push-motion operation in progress This bit remains 1 while the actuator is performing a push-motion operation (excluding an approach operation. It will turn 0 under the following conditions: 1. The actuator has missed the push motion operation. 2. The actuator has paused. 3. The next movement command has been issued. 4. The servo has turned OFF.
9	PSNS	Excitation detection status	0: Excitation detection not yet complete 1: Excitation detection complete PCON/ERC2 Series controllers perform excitation detection at the first servo ON command received after the controller has started. This bit becomes 1 when excitation detection is completed. This bit remains 0 if the excitation detection has failed. Even after a successful detection, the bit will return to 0 when a software reset is performed. This bit becomes 1 if pole sensing is performed with the first servo ON command after startup and the operation is completed in case of ACON series controllers. On SCON Series controllers, this bit is always 0.
8	PMSS	PIO/Modbus switching status	0: PIO commands enabled 1: PIO command disabled The result of switching according to the PIO/Modbus switching setting explained in 5.4.16 or 6.5.16, or the current status, is indicated.
7	-	Cannot be used	
6		Cannot be used	
5	MOVE	Moving signal	0: Stopped 1: Moving This bit indicates whether or not the actuator is moving (conditions during home return and push-motion operation included). This bit remains 0 while the actuator is paused.
4	-	Cannot be used	
3	-	Cannot be used	
2	-	Cannot be used	
1		Cannot be used	
0	-	Cannot be used	

(7) Data of system status registers (Address = 9008H) (STAT)

Bit	Symbol	Name	Function
31	BATL	Absolute battery sag (SCON limitation)	0: Normal 1: Battery Sag If the voltage of an absolute battery becomes below a warning level, it will be set to 1. Axial operation is possible, if the major failure status bit of the device status register 1 becomes zero even if this bit is 1.
30 to 18	-	Cannot be used	
17	ASOF	Auto servo OFF	0: Normal 1: Auto servo OFF If “Auto servo OFF delay time” is set with a parameter of the RC controller, this bit becomes 1 when the servo is turned OFF automatically after the specified time has elapsed following the position complete.
16	AEEP	Nonvolatile memory being accessed	0: Normal 1: Nonvolatile memory being accessed This bit turns 1 as soon as the nonvolatile memory in the RC controller is accessed to read or write the controller’s parameter position table, etc. The bit becomes 0 when the access is completed or a timeout error occurs.
15 to 5	-	Cannot be used	
4	RMDS	Operation mode status	0: AUTO mode 1: MANU mode This bit becomes 1 when the RC controller is in the MANU mode. Note that the controller is always in the MANU mode in cases of models not equipped with an operation mode switch (ERC2, PCON-SE/CY and ACON-SE/CY).
3	HEND	Home return completion status	0: Home return not yet complete 1: Home return completion This bit will become 1 when home return is completed. In case the absolute specification is set, the bit is set to 1 from the startup if absolute reset has been completed. If a movement command is issued while this bit is 0, an alarm will generate.
2	SV	Servo status	0: Servo OFF 1: Servo ON The servo ON status is indicated. After a servo ON command is issued, this bit will remain 0 until the servo ON delay time set by a parameter elapses. If the servo cannot be turned ON for some reason even after a servo ON command is received, this bit will remain 0. The RC controller does not accept any movement command while this bit is 0.
1	SON	Servo command status	0: Servo OFF 1: Servo ON This bit indicates the servo ON/OFF command status. This bit will turn 1 when the following conditions are met: · The EMG status bit in device status register 1 is 0. [Refer to 5.3.6 or 6.4.6.] · The major failure status bit in device status register 1 is 0. [Refer to 5.3.6 or 6.4.6.] · The enable status bit in device status register 2 is 1. [Refer to 5.3.7 or 6.4.7.] · The auto servo OFF status in the system status register is 0. [Refer to 5.3.9 or 6.4.9.]
0	MPOW	Drive source ON	0: Drive source cut off 1: Normal This bit will turn 0 when the motor drive-source cutoff terminal is released.

(8) Data of special port monitor registers (Address = 9012$_H$) (SIPM)

Bit	Symbol	Name	Function
15	-	Cannot be used	
14	NP	Command pulse NP signal status	This bit indicates the status of the command pulse NP signal.
13	-	Cannot be used	
12	PP	Command pulse PP signal status	This bit indicates the status of the command pulse PP signal.
11	-	Cannot be used	
10	-	Cannot be used	
9	-	Cannot be used	
8	MDSW	Mode switch status	0: AUTO mode 1: MANU mode This bit becomes 1 when the RC controller is in the MANU mode. Note that the controller is always in the MANU mode in cases of models not equipped with an operation mode switch (ERC2, PCON-SE/CY, ACON-SE/CY).
7	-	Cannot be used	
6	-	Cannot be used	
5	-	Cannot be used	
4	BLCT	Belt breakage sensor (SCON only)	0: Belt broken 1: Normal
3	HMCK	Home-check sensor monitor	0: Sensor OFF 1: Sensor ON On a model equipped with a home-check sensor function, this bit indicates the status of sensor input. It is always 0 on any other model.
2	OT	Overtravel sensor monitor	0: Sensor OFF 1: Sensor ON This bit indicates the status of the overtravel sensor signal in the encoder connector. It is always 0 on a model not equipped with an overtravel sensor.
1	CREP	Creep sensor monitor	0: Sensor OFF 1: Sensor ON This bit indicates the status of the creep sensor signal in the encoder connector. It is always 0 on a model not equipped with a creep sensor.
0	LS	Limit sensor monitor	0: Sensor OFF 1: Sensor ON This bit indicates the status of the limit sensor signal in the encoder connector. It is always 0 on a model not equipped with a limit sensor.

(9) Data of zone status register (Address = 9013_H) (ZONS)

Bit	Symbol	Name	Function
15	-	Cannot be used	
14	LS2	Limit sensor output monitor 2 (PCON-C/CG, ACON-C/CG, SCON PIO pattern 5)	0: Out of range 1: In range The negative boundary of the positioning band is obtained by subtracting the positioning band size from target position No. 2 while the positive boundary of the positioning band is obtained by adding the positioning band size to target position No. 2. This bit will become 1 when the current position is within the band and 0 when it is outside the band. This bit becomes effective upon home return completion. It remains effective even while the servo is OFF.
13	LS1	Limit sensor output monitor 1 (PCON-C/CG, ACON-C/CG, SCON PIO pattern 5)	0: Out of range 1: In range The negative boundary of the positioning band is obtained by subtracting the positioning band size from target position No. 1 while the positive boundary of the positioning band is obtained by adding the positioning band size to target position No. 1. This bit remains 1 as long as the current position is within these boundaries. This bit will become 1 when the current position is within the band and 0 when it is outside the band. This bit becomes effective upon home return completion. It remains effective even while the servo is OFF.
12	LS0	Limit sensor output monitor 0 (PCON-C/CG, ACON-C/CG, SCON PIO pattern 5)	0: Out of range 1: In range The negative boundary of the positioning band is obtained by subtracting the positioning band size from target position No. 0 while the positive boundary of the positioning band is obtained by adding the positioning band size to target position No. 0. This bit remains 1 as long as the current position is within these boundaries. This bit will become 1 when the current position is within the band and 0 when it is outside the band. This bit becomes effective upon home return completion. It remains effective even while the servo is OFF.
11	-	Cannot be used	
10	-	Cannot be used	
9	-	Cannot be used	
8	ZP	Position zone output monitor	0: Out of range 1: In range This bit remains 1 while the current position is within the zone range specified for each position and becomes 0 when it is outside the range. This bit becomes effective upon home return completion. It remains effective even while the servo is OFF.
7	-	Cannot be used	
6	-	Cannot be used	
5	-	Cannot be used	
4	-	Cannot be used	
3	-	Cannot be used	
2	-	Cannot be used	
1	Z2	Zone output monitor 2	0: Out of range 1: In range This bit remains 1 while the current position is within the range where the zone boundary 2 parameter is set and becomes 0 when it is outside the range. This bit becomes effective upon home return completion. It remains effective even while the servo is OFF.
0	Z1	Zone output monitor 1	0: Out of range 1: In range This bit remains 1 while the current position is within the range where the zone boundary 1 parameter is set and becomes 0 when it is outside the range. This bit becomes effective upon home return completion. It remains effective even while the servo is OFF.

(10) Data of position number status register (Address = 9014_H) (POSS)

Bit	Symbol	Name	Function
15	-	Cannot be used	
14	-	Cannot be used	
13	-	Cannot be used	
12	-	Cannot be used	
11	-	Cannot be used	
10	-	Cannot be used	
9	PM512	Position complete number status bit 512	These bits indicate position numbers for which positioning has been completed (valid in cases other than PIO patterns 4 and 5 (solenoid valve mode)). The position complete is read as binary code. It becomes possible to read position complete numbers when the current position gets close to the target position (within the positioning band in either the positive or negative directions). 0 is read in other cases. Although all the bits will change to 0 once the servo turns OFF, the position complete becomes valid again if the current position is still inside the positioning band when the servo is turned ON subsequently. In push-motion, the position complete numbers can be read at both the completion and miss of push-motion.
8	PM256	Position complete number status bit 256	
7	PM128	Position complete number status bit 128	
6	PM64	Position complete number status bit 64	
5	PM32	Position complete number status bit 32	
4	PM16	Position complete number status bit 16	
3	PM8	Position complete number status bit 8	
2	PM4	Position complete number status bit 4	
1	PM2	Position complete number status bit 2	
0	PM1	Position complete number status bit 1	

4.3.3 Structure of Mudbus Status Registers

The layout of the Mudbus status registers is shown below.

Address	Register
0000_H	(Reserved for system) (Note)
0100_H ~ $010F_H$	Device status register 1 [DSS1]
0110_H ~ $011F_H$	Device status register 2 [DSS2]
0120_H ~ $012F_H$	Expansion device status register [DSSE]
0130_H ~ $013F_H$	Position number status register [POSS]
0140_H ~ $014F_H$	Zone status register [ZONS]
0150_H ~ $015F_H$	Input port monitor register [DIPM]
0160_H ~ $016F_H$	Output port monitor register [DOPM]
0170_H ~ $017F_H$	Special input port register [SIPM]
	(Reserved for system) (Note)
0400_H ~ $040F_H$	Device control register 1 [DRG1]
0410_H ~ $041F_H$	Device control register 2 [DRG2]
0420_H ~ $042F_H$	Expansion device control register [DRGE]
0430_H ~ $043F_H$	Position number specification register [POSR]
$FFFF_H$	(Reserved for system) (Note)

(Note) Areas reserved for the system cannot be used for communication.

4.3.4 Data of Mudbus Status Registers

Address [hex]	Area name	Description	Symbol	Reference page RTU		Reference page ASCII	
0000 to 0CFF	Reserved for system						
0100	Device status register 1 (DSS1)	EMG status	EMGS	(66)	32	(177)	32
0101		Safety speed enabled status	SFTY				
0102		Controller ready status	PWR				
0103		Servo ON status	SV				
0104		Missed work part in push-motion operation	PSFL				
0105		Major failure status	ALMH				
0106		Minor failure status	ALML				
0107		Absolute error status	ABER				
0108		Brake forced-release status	BKRL				
0109		Cannot be used					
010A		Pause status	STP		32		32
010B		Home return status	HEND				
010C		Position complete status	PEND				
010D		Load cell calibration complete	CEND				
010E		Load cell calibration status	CLBS				
010F		Cannot be used					
0110	Device status register 2 (DSS2)	Cannot be used		(68)		(179)	
0111		Cannot be used					
0112		Load output judgment status	LOAD		33		33
0113		Torque level status	TRQS				
0114		Teaching mode status	MODS				
0115		Position-data load command status	TEAC				
0116		Jog+ status	JOG+				
0117		Jog- status	JOG-				
0118		Position complete 7	PE7				
0119		Position complete 6	PE6				
011A		Position complete 5	PE5				
011B		Position complete 4	PE4				
011C		Position complete 3	PE3				
011D		Position complete 2	PE2				
011E		Position complete 1	PE1				
011F		Position complete 0	PE0				
0120	Expansion device status register (DSSE)	Emergency stop status	EMGP	(70)	34	(181)	34
0121		Motor voltage low status	MPUV				
0122		Operation mode status	RMDS				
0123		Cannot be used					
0124		Home return status	GHMS		34		34
0125		Push-motion operation in progress	PUSH				
0126		Excitation detection status	PSNS				
0127		PIO/Modbus switching status	PMSS				
0128		Cannot be used					
0129		Cannot be used					
012A		Moving signal	MOVE		34		34
012B to 012F		Cannot be used					

Address [hex]	Area name	Description	Symbol	Reference page			
				RTU		ASCII	
0130 to 0135	Position number status register (POSS)	Cannot be used		(86)		(197)	
0136		Position complete number status bit 512	PM512		38		38
0137		Position complete number status bit 256	PM256				
0138		Position complete number status bit 128	PM128				
0139		Position complete number status bit 64	PM64				
013A		Position complete number status bit 32	PM32				
013B		Position complete number status bit 16	PM16				
013C		Position complete number status bit 8	PM8				
013D		Position complete number status bit 4	PM4				
013E		Position complete number status bit 2	PM2				
013F		Position complete number status bit 1	PM1				
0140	Zone status register (ZONS)	Cannot be used		(84)		(195)	
0141		Limit sensor output monitor 2	LS2		37		37
0142		Limit sensor output monitor 1	LS1				
0143		Limit sensor output monitor 0	LS0				
0144 to 0146		Cannot be used					
0147		Position zone output monitor	ZP		37		37
0148 to 014D		Cannot be used					
014E		Zone output monitor 2	Z2		37		37
014F		Zone output monitor 1	Z1				
0150 to 015F	Input port monitor register (DIPM)	PIO connector pin numbers 20A (IN15) to 5A (IN0)		58		169	
0160 to 016F	Output port monitor register (DOPM)	PIO connector pin numbers 16B (OUT15) to 1B (OUT0)		62		173	
0170	Special input port monitor register (SIPM)	Cannot be used		(82)		(193)	
0171		Command pulse NP signal status	NP		36		36
0172		Cannot be used					
0173		Command pulse PP signal status	PP		36		36
0174 to 0175		Cannot be used					
0176		Cannot be used					
0177		Mode switch status	MDSW		36		36
0178		Cannot be used					
0179 to 017A		Cannot be used					
017B		Belt breakage sensor monitor	BLCT				
017C		Home-check sensor monitor	HMCK		36		36
017D		Overtravel sensor	OT				
017E		Creep sensor	CREP				
017F		Limit sensor	LS				
0180 to 03FF	Reserved for system						

Address [hex]	Area name	Description	Symbol	Reference page			
				RTU		ASCII	
0400	Device control register 1 (DRG1)	EMG operation specification	EMG	(123)	29	(234)	29
0401		Safety speed command	SFTY				
0402		Cannot be used					
0403		Servo ON command	SON		29		29
0404 to 0406		Cannot be used					
0407		Alarm reset command	ALRS		29		29
0408		Brake forced-release command	BKRL				
0409		Cannot be used					
040A		Pause command	STP		29		29
040B		Home return command	HOME				
040C		Positioning start command	CSTR				
040D to 040F		Cannot be used					
0410	Device control register 2 (DRG2)	Cannot be used		(123)		(234)	
0411		Jog/inch switching	JISL		30		30
0412 to 0413		Cannot be used					
0414		Teaching mode command	MOD		30		30
0415		Position data load command	TEAC				
0416		Jog+ command	JOG+				
0417		Jog command	JOG-				
0418		Start position 7	ST7				
0419		Start position 6	ST6				
041A		Start position 5	ST5				
041B		Start position 4	ST4				
041C		Start position 3	ST3				
041D		Start position 2	ST2				
041E		Start position 1	ST1				
041F		Start position 0	ST0				
0420 to 0425	Expansion device control register (DRGE)	Cannot be used					
0426		Load cell calibration command	CLBR		117		228
0427		PIO/Modbus switching specification	PMSL		119		230
0428 to 042B		Cannot be used					
042C		Deceleration stop	STOP		121		232
042D to 042F		Cannot be used					
0430 to 0435	Position number specification register (POSR)	Cannot be used		(123)		(234)	
0436		Position command bit 512	PC512		31		31
0437		Position command bit 256	PC256				
0438		Position command bit 128	PC128				
0439		Position command bit 64	PC64				
043A		Position command bit 32	PC32				
043B		Position command bit 16	PC16				
043C		Position command bit 8	PC8				
043D		Position command bit 4	PC4				
043E		Position command bit 2	PC2				
043F		Position command bit 1	PC1				
0440 to FFFF	Reserved for system						

5 Modbus RTU

5.1 Message Frames (Query and Response)

Start	Address	Function code	Data	CRC Check	End
Silent interval	1 byte	1 byte	n byte	2 byte	Silent interval

(1) Start
This field contains a silent interval (non communication time) of 3.5 characters or longer.
(1 character = 10 bits)
Example: In case of 9600 bps, (10 x 3.5) bits x 1/9600 bps = 3.65 ms
(Note) If the response timeout error occurs, change parameter No. 45, "Silent interval multiplier" or No. 17, "Min. delay for activating local transmitter" using the IAI teaching tool as required.

(2) Address
This field specifies the addresses of connected RC controllers (01_H to 10_H).
Address = axis number + 1

⚠ Caution: The address is not equal to the corresponding axis number: be careful when making settings.

(3) Function
The table below summarizes the function codes and functions that can be used with RC controllers.

Code [Hex]	Name	Function
01_H	Read Coil Status	Read coils/DOs.
02_H	Read Input Status	Read input statuses/DIs.
03_H	Read Holding Registers	Read holding registers.
04_H	Read Input Registers	Read input registers.
05_H	Force Single Coil	Write one coil/DO.
06_H	Preset Single Register	Write holding register.
07_H	Read Exception Status	Read exception statuses.
$0F_H$	Force Multiple Coils	Write multiple coils/DOs at once.
10_H	Preset Multiple Registers	Write multiple holding registers at once.
11_H	Report Slave ID	Query a slave's ID.
17_H	Read / Write Registers	Read/write registers.

(Note) This manual explains about ▭ mark function codes.
(Reference) The ROBONET gateway supports three types of function codes (03_H, 06_H and 10_H).
[Please refer to the separate ROBONET Instruction Manual.]

(4) Data
Use this field to add data specified by a function code. It is also allowed to omit data if data addition is not specified by a function code.

(5) CRC check
In the RTU mode, an error check field confirming to the CRC method is automatically (Note) included in order to check contents of all messages. Moreover, checking is carried out regardless of the parity check method of individual characters in messages.
The CRC check consists of 16-bit binary values. The CRC value is calculated by the sender that appends the CRC field to a message. The recipient recalculates the CRC value again while receiving the message, and compares the calculation result against the actual value received in the CRC field. If the two values do not match, an error will generate.
(Note) When using a PC or a PLC not supporting Modbus are used as the host, it is necessary to create a function for calculating CRC.
Programs written in C language are included in 10, "CRC Check Calculation."
Generation polynomial equation: **x 16 + x15 + x2 + 1** (CRC-16 method)

(Reference) CRC calculation is automatically carried out with the FINS command supporting Modbus RTU communication of the PLC CJ1 series made by Omron.

(6) End
This field contains a silent interval (non communication time) of 3.5 characters or longer.
(Note) If the response timeout error occurs, change parameter No. 45, "Silent interval multiplier" or No. 17, "Min. delay for activating local transmitter" using the IAI's teaching tools as required.

(7) Broadcast
It is possible to send a query containing same data to all connected axes by specifying the address 00_H. In this case, no response is returned from an RC controller.
Note, however, that the function codes etc. that can be used with this function are limited; care should be taken when using the function. Please check the function codes that can be used in 5.2, "List of RTU Mode Queries."

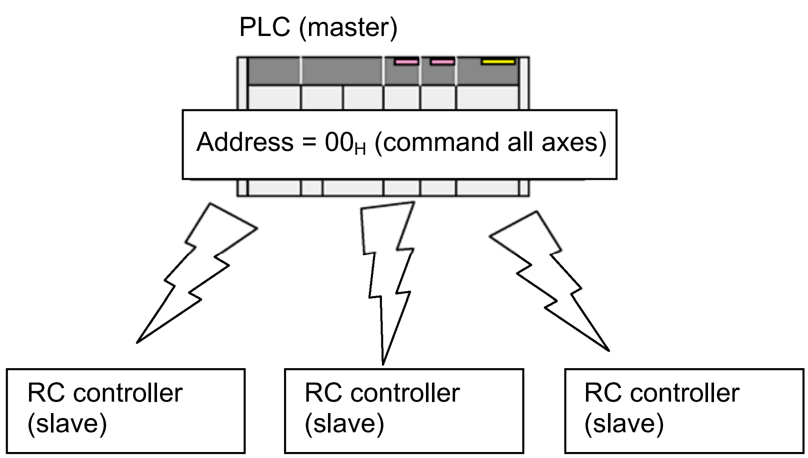


Fig. 5.1

⚠ Caution

The sizes of send/receive buffers are set to 256 bytes for an RC controller, respectively. Make sure to keep the messages small enough such that messages sent from the host side do not exceed the receive buffer and data requests do not exceed send buffer.

5.2 List of RTU Mode Queries

The table below lists queries available in the RTU mode.

FC	Function	Symbol	Function Summary	Combination with PIO	Broad-cast	Page
03	Multiple FC03 register read	None	This function can be used to successively read multiple registers that use function 03.	○		51
03	Current position read	PNOW	This function reads the current actuator position in units of 0.01 mm.	○		54
03	Present alarm code read	ALMC	This function reads alarm codes that are presently detected.	○		56
03	I/O port input status read	DIPM	This function reads the ON/OFF statuses of PIO input ports.	○		58
03	I/O port output status read	DOPM	This function reads the ON/OFF statuses of PIO output ports.	○		62
03	Controller status signal read 1 (device status 1) (Operation preparation status)	DSS1	This function reads the following 12 statuses: [1] Emergency stop [2] Safety speed enabled/disabled [3] Controller ready [4] Servo ON/OFF [5] Missed work part in push-motion operation [6] Major failure [7] Minor failure [8] Absolute error [9] Brake [10] Pause [11] Home return completion [12] Position complete [13] Load cell calibration complete [14] Load cell calibration status	○		66
03	Controller status signal read 2 (device status 2) (Operation preparation 1 status)	DSS2	This function reads the following 15 statuses: [1] Enable [2] Load output judgment (check-range load current threshold) [3] Torque level (load current threshold) [4] Teaching mode (normal/teaching) [5] Position data load (normal/complete) [6] Jog+ (normal/command active) [7] Jog- (normal/command active) [8] Position complete 7 [9] Position complete 6 [10] Position complete 5 [11] Position complete 4 [12] Position complete 3 [13] Position complete 2 [14] Position complete 1 [15] Position complete 0	○		68

FC: Function code
PIO: Parallel I/O (input/output of an I/O connector)
The circle marks in the Combination use with PIO and Broadcast columns indicate queries that can be combined with PIO and in broadcast communication, respectively.

FC	Function	Symbol	Function Summary	Combination with PIO	Broad-cast	Page
03	Controller status signal read 3 (extended device status) (Operation preparation 2 status)	DSSE	This function reads the following 9 statuses: [1] Emergency stop (emergency stop input port) [2] Motor voltage low [3] Operation mode (AUTO/MANU) [4] Home return [5] Push-motion operation in progress [6] Excitation detection [7] PIO/Modbus switching [8] Position-data write completion status [9] Moving	○		70
03	Controller status signal read 4 (System status) (Controller status)	STAT	This function reads the following 7 statuses: [1] Automatic servo OFF [2] Nonvolatile memory being accessed [3] Operation mode (AUTO/MANU) [4] Home return completion [5] Servo ON/OFF [6] Servo command [7] Drive source ON (normal/cut off)	○		72
03	Current speed read	VNOW	This function reads the current actuator speed in units of 0.01 mm/sec.	○		74
03	Current read	CNOW	This function reads the motor-torque current command value of the actuator in mA.	○		76
03	Deviation read	DEVI	This function reads the deviation over a 1-ms period in pulses.	○		78
03	Total power on time read	STIM	This function reads the total time in msec since the controller power was turned on.	○		80
03	Special input port input signal status read (Sensor input status)	SIPM	This function reads the following 8 statuses: [1] Command pulse NP [2] Command pulse PP [3] Mode switch [4] Belt breakage sensor [5] Home check sensor [6] Overtravel sensor [7] Creep sensor [8] Limit sensor	○		82

FC	Function	Symbol	Function Summary	Combination with PIO	Broad-cast	Page
03	Zone output signal read	ZONS	This function reads the following 6 statuses: [1] LS2 (PIO pattern solenoid valve mode [3-point type] [2] LS1 (PIO pattern solenoid valve mode [3-point type] [3] LS0 (PIO pattern solenoid valve mode [3-point type] [4] Position zone [5] Zone 2 [6] Zone 1	○		84
03	Positioning completed position number read	POSS	This function reads the following 9 statuses: [1] Position complete number bit 256 [2] Position complete number bit 128 [3] Position complete number bit 64 [4] Position complete number bit 32 [5] Position complete number bit 16 [6] Position complete number bit 8 [7] Position complete number bit 4 [8] Position complete number bit 2 [9] Position complete number bit 1	○		86
03	Force feedback data write	FBFC	The current measurement on the load cell is read in units of 0.01 N.	○		88
05	Safety speed enable/disable switching	SFTY	This function issues a command to enable/disable the safety speed.		○	91
05	Servo ON/OFF	SON	This function issues a command to turn the servo ON/OFF.		○	93
05	Alarm reset	ALRS	This function issues a command to reset alarms/cancel the remaining travel.		○	95
05	Forced brake release	BKRL	This function issues a command to forcibly release the brake.		○	97
05	Pause	STP	This function issues a pause command.		○	99
05	Home return	HOME	This function issues a home return operation command.		○	101
05	Positioning operation start	CSTR	This signal starts a position number specified movement.		○	103
05	Jog/inch switching	JISL	This function switches between the jogging mode and the inching mode		○	105
05	Teaching mode command	MOD	This function switches between the normal mode and the teaching mode		○	107
05	Position data load command	TEAC	This function issues a current position load command in the teaching mode.		○	109
05	Jog+ command	JOG+	This function issues a jogging/inching command in the direction opposite home.		○	111
05	Jog- command	JOG-	This function issues a jogging/inching command in the direction of home.		○	113

FC	Function	Symbol	Function Summary	Combination with PIO	Broad-cast	Page
05	Start position 0 to 7 <<ST0 to ST7>> move command	ST0 to ST7	This function specifies position numbers effective only in the solenoid valve mode. The actuator can be operated with this command alone (note).		○	115
05	Load cell calibration command	CLBR	Calibrate the load cell.		○	117
05	PIO/Modbus switching setting	PMSL	This function issues a command to enable/disable PIO external command signals.		○	119
05	Deceleration stop	STOP	This function can decelerate the actuator to a stop.		○	121
06	Direct control information write		Change (write) the content of the controller's register.		○	123
10	Numerical movement command	None	This function can be used to send the target position, positioning band, speed, acceleration/deceleration, push, and control setting in a single message to operate the actuator. Normal movement, relative movement and push-motion operation are supported.		○	127
10	Position data table write	None	This function can be used to change all data of the specified position number for the specified axis.		○	145
Indeter-minable	Exception response	None	This response will be returned when the message contains invalid data.			264

5.3. Data and Status Reading (Queries Using Code 03)

5.3.1 Reading Consecutive Multiple Registers

(1) Function

These registers read the contents of registers in a slave.

This function is not supported in broadcast communication.

(2) Start address list

With RC Series controllers, the sizes of send/receive buffers are set to 256 bytes, respectively. Accordingly, a maximum of 126 registers' worth of data consisting of 252 bytes (one register uses two bytes), except 4 bytes (slave address + function code + error check) of the above 256 bytes, can be queried in the RTU mode. In other words, all of the data listed below (total 21 registers) can be queried in a single communication.

Address [H]	Symbol	Name	Sign	Register size	Byte
9000, 9001	PNOW	Current position monitor	○	2	4
9002	ALMC	Present alarm code query		1	2
9003	DIPM	Input port query		1	2
9004	DOPM	Output port monitor query		1	2
9005	DSS1	Device status query 1		1	2
9006	DSS2	Device status query 2		1	2
9007	DSSE	Expansion device status query		1	2
9008, 9009	STAT	System status query		2	4
900A, 900B	VNOW	Current speed monitor	○	2	4
900C, 900D	CNOW	Current ampere monitor	○	2	4
900E, 900F	DEVI	Deviation monitor	○	2	4
9010, 9011	STIM	System timer query		2	4
9012	SIPM	Special input port query		1	2
9013	ZONS	Zone status query		1	2
9014	POSS	Position complete number status query		1	2
901E	FBFC	Force feedback data monitor	○	2	4

(3) Query format

In a query message, specify the address of the register from which to start reading data, and number of bytes in registers to be read.
1 register (1 address) = 2 bytes = 16-bit data

Field	RTU mode 8-bit data	Number of data items (number of bytes)	Remarks
Start	None		Silent interval
Slave address [H]	Arbitrary	1	Axis No. + 1 (01_H to 10_H)
Function code [H]	03	1	Register reading code
Start address [H]	Arbitrary	2	Refer to 5.3.1 (2), "Start address list"
Number of registers [H]	Arbitrary	2	Refer to the start address list.
Error check [H]	CRC (16 bits)	2	
End	None		Silent interval
Total number of bytes		8	

(4) Response format

Field	RTU mode 8-bit data	Number of data items (number of bytes)	Remarks
Start			Silent interval
Slave address [H]	Arbitrary	1	Axis No. + 1 (01_H to 10_H)
Function code [H]	03	1	Register reading code
Number of data bytes [H]		1	Total number of bytes of registers specified in the query
Data 1 [H]		Number of bytes of register specified in the query	
Data 2 [H]		Number of bytes of register specified in the query	
Data 3 [H]		Number of bytes of register specified in the query	
Data 4 [H]		Number of bytes of register specified in the query	
:		:	
:		:	
Error check [H]	CRC (16 bits)	2	Silent interval
End			
Total number of bytes		256 max.	

(5) Query sample

A sample query that queries addresses 9000_H to 9009_H of a controller of axis No. 0 is shown below.

● Query (silent intervals are inserted before and after the query)

01 03 90 00 00 0A E8 CD

Field	RTU mode 8-bit data
Start	Silent interval
Slave address [H]	01
Function code [H]	03
Start address [H]	9000
Number of registers [H]	000A (10 registers)
Error check [H]	E8CD (in accordance with CRC calculation)
End	Silent interval

The response to the query is as follows.

● Response (silent intervals are inserted before and after the response)

01 03 14 00 00 00 00 00 00 00 00 6E 00 60 18 80 00 23 C7 00 00 00 19 18 A6

Field	RTU mode 8-bit data
Start	Silent interval
Slave address [H]	01
Function code [H]	03
Number of data bytes [H]	14 (20 bytes = 10 registers)
Data 1 [H]	00 00 00 00 (current position monitor)
Data 2 [H]	00 00 (present alarm code query)
Data 3 [H]	00 00 (input port query)
Data 4 [H]	6E 00 (output port query)
Data 5 [H]	60 18 (device status 1 query)
Data 6 [H]	80 00 (device status 2 query)
Data 7 [H]	23 C7 (expansion device status query)
Data 8 [H]	00 00 00 19 (system status query)
Error check [H]	18A6 (in accordance with CRC calculation)
End	Silent interval

(Note) If the response example is simply an example and will vary depending on various conditions.

5.3.2 Current Position Reading <<PNOW>> (in 0.01 mm units)

(1) Function

This bit reads the current position in units of 0.01 mm. The sign is effective.

(2) Query format

Field	Number of data items (number of bytes)	RTU mode 8-bit data	Remarks
Start	None		Silent interval
Slave address [H]	1	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	1	03	Register reading
Start address [H]	2	9000	Current position monitor
Number of registers [H]	2	0002	Reading addresses 9000_H to 9001_H
Error check [H]	2	CRC (16 bits)	
End	None		Silent interval
Total number of bytes	8		

(3) Response format

A response message contains 16 bits of data per register.

Field	Number of data items (number of bytes)	RTU mode 8-bit data	Remarks
Start	None		Silent interval
Slave address [H]	1	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	1	03	Register reading
Number of data bytes [H]	1	04	Reading 2 registers = 4 bytes
Data 1 [H]	2	In accordance with the current value	Current data [Hex] (most significant digit)
Data 2 [H]	2	In accordance with the current value	Current data [Hex] (least significant digit)
Error check [H]	2	CRC (16 bits)	
End	None		Silent interval
Total number of bytes	9		

(4) Query sample

A sample query that reads the current position (addresses 9000_H to 9001_H) of a controller with axis No. 0 is shown below.

- Query (silent intervals are inserted before and after the query)
  01 03 90 00 00 02 E9 0B

Field	RTU mode 8-bit data
Start	Silent interval
Slave address [H]	01
Function code [H]	03
Start address [H]	9000
Number of registers [H]	0002
Error check [H]	E90B (in accordance with CRC calculation)
End	Silent interval

The response to the query is as follows.

- Response (silent intervals are inserted before and after the response)
  01 03 04 00 00 0B FE 7C 83

Field	RTU mode 8-bit data
Start	Silent interval
Slave address [H]	01
Function code [H]	03
Number of data bytes [H]	04 (4 bytes = 2 registers)
Data 1 [H]	00 00
Data 2 [H]	0B FE
Error check [H]	7C83 (in accordance with CRC calculation)
End	Silent interval

The current position is "$00000BFE_H$" → Converted to a decimal value → 3070 (x 0.01 mm)
The current position is 30.7 mm.
Example 2: If the current position is read "$FFFFFFF5_H$" (negative position) →
$FFFFFFFF_H$ – $FFFFFFF5_H$ + 1 (make sure to add 1) →
Convert to a decimal value → 11 (x 0.01 mm)
The current position is -0.11 mm

(Note) The data of the response example is simply an example and will vary depending on various conditions.

5.3.3 Present Alarm Code Reading <<ALMC>>

(1) Function

This query reads the code indicating the normal status or alarm status of the controller.
In the normal status, 00_H is stored.
[For details on alarm codes, refer to the operation manual for each controller.]

(2) Query format

Field	Number of data items (number of bytes)	RTU mode 8-bits data	Remarks
Start	None		Silent interval
Slave address [H]	1	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	1	03	Register reading
Start address [H]	2	9002	Present alarm code
Number of registers [H]	2	0001	Reading address 9002_H
Error check [H]	2	CRC (16 bits)	
End	None		Silent interval
Total number of bytes	8		

(3) Response format

A response message contains 16 bits of data per register.

Field	Number of data items (number of bytes)	RTU mode 8-bits data	Remarks
Start	None		Silent interval
Slave address [H]	1	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	1	03	Register reading
Number of data bytes [H]	1	02	Reading 1 register = 2 bytes
Data 1 [H]	2	Alarm code	Alarm code [Hex]
Error check [H]	2	CRC (16 bits)	
End	None		Silent interval
Total number of bytes	7		

(4) Query sample

A sample query that reads the alarm code position (address 9002_H) of a controller with axis No. 0 is shown below.

- Query (silent intervals are inserted before and after the query)
  01 03 90 02 00 01 08 CA

Field	RTU mode 8-bit data
Start	Silent interval
Slave address [H]	01
Function code [H]	03
Start address [H]	9002
Number of registers [H]	0001
Error check [H]	08CA (in accordance with CRC calculation)
End	Silent interval

The response to the query is as follows.

- Response (silent intervals are inserted before and after the response)
  01 03 02 00 E8 B8 0A

Field	RTU mode 8-bit data
Start	Silent interval
Slave address [H]	01
Function code [H]	03
Number of data bytes [H]	02 (2 bytes = 1 register)
Data 1 [H]	00 E8
Error check [H]	B80A (in accordance with CRC calculation)
End	Silent interval

The most important alarm presently detected is "0E8"$_H$, which is a phase A/B open alarm.
[For details on alarm codes, refer to the operation manual that comes with each controller.]

(Note) The data of the response example is simply an example and will vary depending on various conditions.

5.3.4 I/O Port Input Signal Status Reading <<DIPM>>

(1) Function

This query reads the port input value of the RC controller regardless of the PIO pattern.
The status of the port to which a signal is currently input as recognized by the RC controller is read.

(2) Query format

Field	Number of data items (number of bytes)	RTU mode 8-bit data	Remarks
Start	None		Silent interval
Slave address [H]	1	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	1	03	Register reading
Start address [H]	2	9003	Input port monitor register
Number of registers [H]	2	0001	Address 9003_H
Error check [H]	2	CRC (16 bits)	
End	None		Silent interval
Total number of bytes	8		

(3) Response format

A response message contains 16 bits of data per address.

Field	Number of data items (number of bytes)	RTU mode 8-bit data	Remarks
Start	None		Silent interval
Slave address [H]	1	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	1	03	Register reading
Number of data bytes [H]	1	02	Reading 1 register = 2 bytes
Data 1 [H]	2	Port input value	Port input value [Hex]
Error check [H]	2	CRC (16 bits)	
End	None		Silent interval
Total number of bytes	7		

(4) Query sample

A sample query that reads the current position (address 9003_H) of a controller with axis No. 0 is shown below.

- Query (silent intervals are inserted before and after the query)
  01 03 90 03 00 01 59 0A

Field	RTU mode 8-bit data
Start	Silent interval
Slave address [H]	01
Function code [H]	03
Start address [H]	9003
Number of registers [H]	0001
Error check [H]	590A (in accordance with CRC calculation)
End	Silent interval

The response to the query is as follows.

- Response (silent intervals are inserted before and after the response)
  01 03 02 90 00 D4 44

Field	RTU mode 8-bit data
Start	Silent interval
Slave address [H]	01
Function code [H]	03
Number of data bytes [H]	02 (2 bytes = 1 register)
Data 1 [H]	90 00
Error check [H]	D444 (in accordance with CRC calculation)
End	Silent interval

The input data area address is "9000"$_H$, which is converted to binary value "1001000000000000."

↑ ↑

INT15 -------------------- INT 1

(Note) The data of the response example is simply an example and will vary depending on various conditions.

(5) Port assignment [For details, refer to the operation manual that comes with each RC controller]
Write the port assignment of PIO patterns to each RC controller.
0 indicates that response data is always 0.

Port	ERC2 (PIO type)			
	PIO pattern 0	PIO pattern 1	PIO pattern 2	PIO pattern 3
IN0	PC1	ST0	PC1	PC1
IN1	PC2	ST1	PC2	PC2
IN2	PC4	ST2	PC4	PC4
IN3	HOME	0	PC8	PC8
IN4	CSTR	RES	CSTR	CSTR
IN5	*STP	*STP	*STP	*STP
IN6	0	0	0	0
IN7	0	0	0	0
IN8	0	0	0	0
IN9	0	0	0	0
IN10	0	0	0	0
IN11	0	0	0	0
IN12	0	0	0	0
IN13	0	0	0	0
IN14	0	0	0	0
IN15	0	0	0	0

Port	PCON-C/CG						PCON-PL/PO	
	PIO pattern 0	PIO pattern 1	PIO pattern 2	PIO pattern 3	PIO pattern 4	PIO pattern 5	PIO pattern 0	PIO pattern 1
IN0	PC1	PC1	PC1	PC1	ST0	ST0	SON	SON
IN1	PC2	PC2	PC2	PC2	ST1	ST1	TL	TL
IN2	PC4	PC4	PC4	PC4	ST2	ST2	HOME	HOME
IN3	PC8	PC8	PC8	PC8	ST3	0	RES	RES/ DCLR
IN4	PC16	PC16	PC16	PC16	ST4	0	0	0
IN5	PC32	PC32	PC32	PC32	ST5	0	0	0
IN6	0	MODE	PC64	PC64	ST6	0	0	0
IN7	0	JISL	PC128	PC128	0	0	0	0
IN8	0	JOG+	0	PC256	0	0	0	0
IN9	BKRL	JOG−	BKRL	BKRL	BKRL	BKRL	0	0
IN10	RMOD	RMOD	RMOD	RMOD	RMOD	RMOD	0	0
IN11	HOME	HOME	HOME	HOME	HOME	0	0	0
IN12	*STP	*STP	*STP	*STP	*STP	0	0	0
IN13	CSTR	CSTR/ PWRT	CSTR	CSTR	0	0	0	0
IN14	RES	RES	RES	RES	RES	RES	0	0
IN15	SON	SON	SON	SON	SON	SON	0	0

Port	ACON-C/CG PIO pattern 0	PIO pattern 1	PIO pattern 2	PIO pattern 3	PIO pattern 4	PIO pattern 5	ACON-PL/PO PIO pattern 0	PIO pattern 1
IN0	PC1	PC1	PC1	PC1	ST0	ST0	SON	SON
IN1	PC2	PC2	PC2	PC2	ST1	ST1	TL	TL
IN2	PC4	PC4	PC4	PC4	ST2	ST2	HOME	HOME
IN3	PC8	PC8	PC8	PC8	ST3	0	RES	RES/ DCLR
IN4	PC16	PC16	PC16	PC16	ST4	0	0	0
IN5	PC32	PC32	PC32	PC32	ST5	0	0	0
IN6	0	MODE	PC64	PC64	ST6	0	0	0
IN7	0	JISL	PC128	PC128	0	0	0	0
IN8	0	JOG+	0	PC256	0	0	0	0
IN9	BKRL	JOG−	BKRL	BKRL	BKRL	BKRL	0	0
IN10	RMOD	RMOD	RMOD	RMOD	RMOD	RMOD	0	0
IN11	HOME	HOME	HOME	HOME	HOME	0	0	0
IN12	*STP	*STP	*STP	*STP	*STP	0	0	0
IN13	CSTR	CSTR/ PWRT	CSTR	CSTR	0	0	0	0
IN14	RES	RES	RES	RES	RES	RES	0	0
IN15	SON	SON	SON	SON	SON	SON	0	0

Port	SCON-C/CA PIO pattern 0	PIO pattern 1	PIO pattern 2	PIO pattern 3	PIO pattern 4	PIO pattern 5	PIO pattern 6	PIO pattern 7	SCON-CA (Pulse train mode) PIO pattern 0
IN0	PC1	PC1	PC1	PC1	ST0	ST0	PC1	ST0	SON
IN1	PC2	PC2	PC2	PC2	ST1	ST1	PC2	ST1	RES
IN2	PC4	PC4	PC4	PC4	ST2	ST2	PC4	ST2	HOME
IN3	PC8	PC8	PC8	PC8	ST3	0	PC8	ST3	TL
IN4	PC16	PC16	PC16	PC16	ST4	0	PC16	ST4	CSTR
IN5	PC32	PC32	PC32	PC32	ST5	0	0	0	DCLR
IN6	0	MODE	PC64	PC64	ST6	0	0	0	BKRL
IN7	0	JISL	PC128	PC128	0	0	0	0	RMOD
IN8	0	JOG+	0	PC256	0	0	CLBR	CLBR	0
IN9	BKRL	JOG−	BKRL	BKRL	BKRL	BKRL	BKRL	BKRL	0
IN10	RMOD	RMOD	RMOD	RMOD	RMOD	RMOD	RMOD	RMOD	0
IN11	HOME	HOME	HOME	HOME	HOME	0	HOME	HOME	0
IN12	*STP	*STP	*STP	*STP	*STP	0	*STP	*STP	0
IN13	CSTR	CSTR/ PWRT	CSTR	CSTR	0	0	CSTR	0	0
IN14	RES	RES	RES	RES	RES	RES	RES	RES	0
IN15	SON	SON	SON	SON	SON	SON	SON	SON	0

5.3.5 I/O Port Output Signal Status Reading<<DOPM>>

(1) Function

This query reads the port output value of the RC controller regardless of the PIO pattern.

(2) Query format

Field	Number of data items (number of bytes)	RTU mode 8-bit data	Remarks
Start	None		Silent interval
Slave address [H]	1	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	1	03	Register reading
Start address [H]	2	9004	Output port monitor register
Number of registers [H]	2	0001	Reading addresses 9004_H
Error check [H]	2	CRC (16 bits)	
End	None		Silent interval
Total number of bytes	8		

(3) Response format

A response message contains 16 bits of data per register.

Field	Number of data items (number of bytes)	RTU mode 8-bit data	Remarks
Start	None		Silent interval
Slave address [H]	1	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	1	03	Register reading
Number of data bytes [H]	1	02	Reading 1 register = 2 bytes
Data 1 [H]	2	D0 output value	Port output value [Hex]
Error check [H]	2	CRC (16 bits)	
End	None		Silent interval
Total number of bytes	7		

(4) Query sample

A sample query that queries address 9004_H of a controller of axis No. 0 is shown below.

- Query (silent intervals are inserted before and after the query)
  01 03 90 04 00 01 E8 CB

Field	RTU mode 8-bit data
Start	Silent interval
Slave address [H]	01
Function code [H]	03
Start address [H]	9004
Number of registers [H]	0001
Error check [H]	E8CB (in accordance with CRC calculation)
End	Silent interval

The response to the query is as follows.

- Response (silent intervals are inserted before and after the response)
  01 03 02 68 00 97 84

Field	RTU mode 8-bit data
Start	Silent interval
Slave address [H]	01
Function code [H]	03
Number of data bytes [H]	02 (2 bytes = 1 register)
Data 1 [H]	68 00
Error check [H]	9784 (in accordance with CRC calculation)
End	Silent interval

The input data area address is "6800"$_H$, which is converted to binary value "0110100000000000."
↑ ↑
INT15 -------------------- INT 1

(Note) The data of the response example is simply an example and will vary depending on various conditions.

(5) Port assignment [For details, refer to the operation manual that comes with each RC controller.]
Write the port assignment of PIO patterns to each RC controller.
0 indicates that response data is always 0.

Port	ERC2 (PIO type)			
	PIO pattern 0	PIO pattern 1	PIO pattern 2	PIO pattern 3
OUT0	PEND	PE0	PEND	PEND
OUT1	HEND	PE1	HEND	HEND
OUT2	ZONE	PE2	ZONE	ZONE
OUT3	*ALM	*ALM	*ALM	*ALM
OUT4	0	0	0	0
OUT5	0	0	0	0
OUT6	0	0	0	0
OUT7	0	0	0	0
OUT8	0	0	0	0
OUT9	0	0	0	0
OUT10	0	0	0	0
OUT11	0	0	0	0
OUT12	0	0	0	0
OUT13	0	0	0	0
OUT14	0	0	0	0
OUT15	0	0	0	0

Port	PCON-C/CG						PCON-PL/PO	
	PIO pattern 0	PIO pattern 1	PIO pattern 2	PIO pattern 3	PIO pattern 4	PIO pattern 5	PIO pattern 0	PIO pattern 1
OUT0	PM1	PM1	PM1	PM1	PE0	LS0	SV	SV
OUT1	PM2	PM2	PM2	PM2	PE1	LS1	INP	INP/ TLR
OUT2	PM4	PM4	PM4	PM4	PE2	LS2	HEND	HEND
OUT3	PM8	PM8	PM8	PM8	PE3	0	*ALM	*ALM
OUT4	PM16	PM16	PM16	PM16	PE4	0	0	0
OUT5	PM32	PM32	PM32	PM32	PE5	0	0	0
OUT6	MOVE	MOVE	PM64	PM64	PE6	0	0	0
OUT7	ZONE1	MODES	PM128	PM128	ZONE1	ZONE1	0	0
OUT8	PZONE	PZONE	PZONE	PM256	PZONE	PZONE	0	0
OUT9	RMDS	RMDS	RMDS	RMDS	RMDS	RMDS	0	0
OUT10	HEND	HEND	HEND	HEND	HEND	HEND	0	0
OUT11	PEND	PEND/ WEND	PEND	PEND	PEND	0	0	0
OUT12	SV	SV	SV	SV	SV	SV	0	0
OUT13	*EMGS	*EMGS	*EMGS	*EMGS	*EMGS	*EMGS	0	0
OUT14	*ALM	*ALM	*ALM	*ALM	*ALM	*ALM	0	0
OUT15	0	0	0	0	0	0	0	0

Port	ACON-C/CG						ACON-PL/PO	
	PIO pattern 0	PIO pattern 1	PIO pattern 2	PIO pattern 3	PIO pattern 4	PIO pattern 5	PIO pattern 0	PIO pattern 1
OUT0	PM1	PM1	PM1	PM1	PE0	LS0	SV	SV
OUT1	PM2	PM2	PM2	PM2	PE1	LS1	INP	INP/ TLR
OUT2	PM4	PM4	PM4	PM4	PE2	LS2	HEND	HEND
OUT3	PM8	PM8	PM8	PM8	PE3	0	*ALM	*ALM
OUT4	PM16	PM16	PM16	PM16	PE4	0	0	0
OUT5	PM32	PM32	PM32	PM32	PE5	0	0	0
OUT6	MOVE	MOVE	PM64	PM64	PE6	0	0	0
OUT7	ZONE1	MODES	PM128	PM128	ZONE1	ZONE1	0	0
OUT8	PZONE	PZONE	PZONE	PM256	PZONE	PZONE	0	0
OUT9	RMDS	RMDS	RMDS	RMDS	RMDS	RMDS	0	0
OUT10	HEND	HEND	HEND	HEND	HEND	HEND	0	0
OUT11	PEND	PEND/ WEND	PEND	PEND	PEND	0	0	0
OUT12	SV	SV	SV	SV	SV	SV	0	0
OUT13	*EMGS	*EMGS	*EMGS	*EMGS	*EMGS	*EMGS	0	0
OUT14	*ALM	*ALM	*ALM	*ALM	*ALM	*ALM	0	0
OUT15	TRQS	0	TRQS	TRQS	TRQS	0	0	0

Port	SCON-C/CA						SCON-CA		(Pulse train mode)
	PIO pattern 0	PIO pattern 1	PIO pattern 2	PIO pattern 3	PIO pattern 4	PIO pattern 5	PIO pattern 6	PIO pattern 7	PIO pattern 0
OUT0	PM1	PM1	PM1	PM1	PE0	LS0	PM1	PE0	PWR
OUT1	PM2	PM2	PM2	PM2	PE1	LS1	PM2	PE1	SV
OUT2	PM4	PM4	PM4	PM4	PE2	LS2	PM4	PE2	INP
OUT3	PM8	PM8	PM8	PM8	PE3	0	PM8	PE3	HEND
OUT4	PM16	PM16	PM16	PM16	PE4	0	PM16	PE4	TLR
OUT5	PM32	PM32	PM32	PM32	PE5	0	TRQS	TRQS	*ALM
OUT6	MOVE	MOVE	PM64	PM64	PE6	0	LOAD	LOAD	*EMGS
OUT7	ZONE1	MODES	PM128	PM128	ZONE1	ZONE1	CEND	CEND	RMDS
OUT8	PZONE	PZONE	PZONE	PM256	PZONE	PZONE	PZONE	PZONE	ALM1
OUT9	RMDS	RMDS	RMDS	RMDS	RMDS	RMDS	RMDS	RMDS	ALM2
OUT10	HEND	HEND	HEND	HEND	HEND	HEND	HEND	HEND	ALM4
OUT11	PEND	PEND/ WEND	PEND	PEND	PEND	0	PEND	PEND	ALM8
OUT12	SV	SV	SV	SV	SV	SV	SV	SV	0
OUT13	*EMGS	*EMGS	*EMGS	*EMGS	*EMGS	*EMGS	*EMGS	*EMGS	0
OUT14	*ALM	*ALM	*ALM	*ALM	*ALM	*ALM	*ALM	*ALM	ZONE1
OUT15	*BALM	*BALM	*BALM	*BALM	*BALM	*BALM	*BALM	*BALM	ZONE2

5.3.6 Device Status Reading 1 <<DSS1>>

(1) Function

This query reads the internal status of the controller.
[Refer to 4.3.2 (4), "Data of device status register 1."]

(2) Query format

Field	Number of data items (number of bytes)	RTU mode 8-bit data	Remarks
Start	None		Silent interval
Slave address [H]	1	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	1	03	Register reading
Start address [H]	2	9005	Device status register 1
Number of registers [H]	2	0001	Reading address 9005_H
Error check [H]	2	CRC (16 bits)	
End	None		Silent interval
Total number of bytes	8		

(3) Response format

A response message contains 16 bits of data per address.

Field	Number of data items (number of bytes)	RTU mode 8-bit data	Remarks
Start	None		Silent interval
Slave address [H]	1	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	1	03	Register reading
Number of data bytes [H]	1	02	Reading 1 register = 2 bytes
Data [H]	2	Status 1	Status 1 [Hex]
Error check [H]	2	CRC (16 bits)	
End	None		Silent interval
Total number of bytes	7		

(4) Query sample

A sample query that reads the current position (address 9005_H) of a controller with axis No. 0 is shown below.

- Query (silent intervals are inserted before and after the query)
  01 03 90 05 00 01 B9 0B

Field	RTU mode 8-bit data
Start	Silent interval
Slave address [H]	01
Function code [H]	03
Start address [H]	9005
Number of registers [H]	0001
Error check [H]	B90B (in accordance with CRC calculation)
End	Silent interval

The response to the query is as follows.

- Response (silent intervals are inserted before and after the response)
  01 03 02 70 98 9C 2E

Field	RTU mode 8-bit data
Start	Silent interval
Slave address [H]	01
Function code [H]	03
Number of data bytes [H]	02 (2 bytes = 1 register)
Data 1 [H]	70 98
Error check [H]	9C2E (in accordance with CRC calculation)
End	Silent interval

(Note) The data of the response example is simply an example and will vary depending on various conditions.

5.3.7 Device Status Reading 2 <<DSS2>>

(1) Function

This query reads the internal status of the controller.
[Refer to 4.3.2 (5), "Data of device status register 2."]

(2) Query format

Field	Number of data items (number of bytes)	RTU mode 8-bit data	Remarks
Start	None		Silent interval
Slave address [H]	1	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	1	03	Register reading
Start address [H]	2	9006	Device status register 2
Number of registers [H]	2	0001	Reading address 9006_H
Error check [H]	2	CRC (16 bits)	
End	None		Silent interval
Total number of bytes	8		

(3) Response format

A response message contains 16 bits of data per register.

Field	Number of data items (number of bytes)	RTU mode 8-bit data	Remarks
Start	None		Silent interval
Slave address [H]	1	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	1	03	Internal status of controller
Number of data bytes [H]	1	02	Reading 1 register = 2 bytes
Data [H]	2	Status 2	Status 2 [Hex]
Error check [H]	2	CRC (16 bits)	
End	None		Silent interval
Total number of bytes	7		

(4) Query sample

A sample query that reads the current position (address 9006_H) of a controller with axis No. 0 is shown below.

- Query (silent intervals are inserted before and after the query)
  01 03 90 06 00 01 49 0B

Field	RTU mode 8-bit data
Start	Silent interval
Slave address [H]	01
Function code [H]	03
Start address [H]	9006
Number of registers [H]	0001
Error check [H]	490B (in accordance with CRC calculation)
End	Silent interval

The response to the query is as follows.

- Response (silent intervals are inserted before and after the response)
  01 03 02 80 00 D9 84

Field	RTU mode 8-bit data
Start	Silent interval
Slave address [H]	01
Function code [H]	03
Number of data bytes [H]	02 (2 bytes = 1 register)
Data 1 [H]	80 00
Error check [H]	D984 (in accordance with CRC calculation)
End	Silent interval

(Note) The data of the response example is simply an example and will vary depending on various conditions.

5.3.8 Device Status Reading 3 <<DSSE>>

(1) Function

This query reads internal status (expansion device) of the controller.
[Refer to 4.3.2 (6), "Data of expansion device status register."]

(2) Query format

Field	Number of data items (number of bytes)	RTU mode 8-bit data	Remarks
Start	None		Silent interval
Slave address [H]	1	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	1	03	Register reading
Start address [H]	2	9007	Expansion device status register
Number of registers [H]	2	0001	Reading address 9007_H
Error check [H]	2	CRC (16 bits)	
End	None		Silent interval
Total number of bytes	8		

(3) Response format

A response message contains 16 bits of data per register.

Field	Number of data items (number of bytes)	RTU mode 8-bit data	Remarks
Start	None		Silent interval
Slave address [H]	1	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	1	03	Register reading
Number of data bytes [H]	1	02	Reading 1 register = 2 bytes
Data [H]	2	Expansion status	Expansion status [Hex]
Error check [H]	2	CRC (16 bits)	
End	None		Silent interval
Total number of bytes	7		

(4) Query sample

A sample query that reads the expansion device status (address 9007_H) of a controller of axis No. 0 is shown below.

- Query (silent intervals are inserted before and after the query)
  01 03 90 07 00 01 18 CB

Field	RTU mode 8-bit data
Start	Silent interval
Slave address [H]	01
Function code [H]	03
Start address [H]	9007
Number of registers [H]	0001
Error check [H]	18CB (in accordance with CRC calculation)
End	Silent interval

The response to the query is as follows.

- Response (silent intervals are inserted before and after the response)
  01 03 02 33 C7 ED 26

Field	RTU mode 8-bit data
Start	Silent interval
Slave address [H]	01
Function code [H]	03
Number of data bytes [H]	02 (2 bytes = 1 register)
Data 1 [H]	33 C7
Error check [H]	ED26 (in accordance with CRC calculation)
End	Silent interval

(Note) The data of the response example is simply an example and will vary depending on various conditions

5.3.9 Device Status Reading4 <<STAT>>

(1) Function

This query reads the internal operation status of the controller.
[Refer to 4.3.2 (7), "Data of system status register."]

(2) Query format

Field	Number of data items (number of bytes)	RTU mode 8-bit data	Remarks
Start	None		Silent interval
Slave address [H]	1	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	1	03	Register reading
Start address [H]	2	9008	System status register
Number of registers [H]	4	0002	Reading addresses 9008_H to 9009_H
Error check [H]	2	CRC (16 bits)	
End	None		Silent interval
Total number of bytes	8		

(3) Response format

A response message contains 16 bits of data per register.

Field	Number of data items (number of bytes)	RTU mode 8-bit data	Remarks
Start	None		Silent interval
Slave address [H]	1	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	1	03	Internal status of controller
Number of data bytes [H]	1	04	Reading 2 registers = 4 bytes
Data [H]	4	System status	System status [Hex]
Error check [H]	2	CRC (16 bits)	
End	None		Silent interval
Total number of bytes	9		

(4) Query sample

A sample query that reads the system status (from address 9008_H) of a controller of axis No. 0 is shown below.

- Query (silent intervals are inserted before and after the query)
  01 03 90 08 00 02 68 C9

Field	RTU mode 8-bit data
Start	Silent interval
Slave address [H]	01
Function code [H]	03
Start address [H]	9008
Number of registers [H]	0002
Error check [H]	68C9 (in accordance with CRC calculation)
End	Silent interval

The response to the query is as follows.

- Response (silent intervals are inserted before and after the response)
  01 03 04 00 0C 00 17 7A 3E

Field	RTU mode 8-bit data
Start	Silent interval
Slave address [H]	01
Function code [H]	03
Number of data bytes [H]	04 (4 bytes = 2 registers)
Data 1 [H]	00 0C 00 17
Error check [H]	7A3E (in accordance with CRC calculation)
End	Silent interval

(Note) The data of the response example is simply an example and will vary depending on various conditions.

5.3.10 Current Speed Reading <<VNOW>>

(1) Function

The monitored data of actual motor speed is read. The speed may be positive or negative depending on the moving direction of the actuator.
The unit is 0.01 mm/sec.

(2) Query format

Field	Number of data items (number of bytes)	RTU mode 8-bit data	Remarks
Start	None		Silent interval
Slave address [H]	1	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	1	03	Register reading
Start address [H]	2	900A	Current speed monitor
Number of registers [H]	4	0002	Reading addresses $900A_H$ to $900B_H$
Error check [H]	2	CRC (16 bits)	
End	None		Silent interval
Total number of bytes	8		

(3) Response format

A response message contains 16 bits of data per register.

Field	Number of data items (number of bytes)	RTU mode 8-bit data	Remarks
Start	None		Silent interval
Slave address [H]	1	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	1	03	Register reading
Number of data bytes [H]	1	04	Reading 2 registers = 4 bytes
Data [H]	4	Current speed	Current speed [Hex] Indicated in units of 0.01 mm/sec.
Error check [H]	2	CRC (16 bits)	
End	None		Silent interval
Total number of bytes	9		

(4) Query sample

A sample query that reads the current speed monitor (from address $900A_H$) of a controller of axis No. 0 is shown below.

- Query (silent intervals are inserted before and after the query)
  01 03 90 0A 00 02 C9 09

Field	RTU mode 8-bit data
Start	Silent interval
Slave address [H]	01
Function code [H]	03
Start address [H]	900A
Number of registers [H]	0002
Error check [H]	C909 (in accordance with CRC calculation)
End	Silent interval

The response to the query is as follows.

- Response (silent intervals are inserted before and after the response)
  01 03 04 00 00 03 E4 FA 88

Field	RTU mode 8-bit data
Start	Silent interval
Slave address [H]	01
Function code [H]	03
Number of data bytes [H]	04 (4 bytes = 2 registers)
Data 1 [H]	00 00 03 E4
Error check [H]	FA88 (in accordance with CRC calculation)
End	Silent interval

The current speed is "000003E4" → Converted to a decimal value → 996 (x 0.01 mm/sec)
The current speed monitor is 9.96 mm/sec.

Example 2: When the current speed reading is "FFFFFF35" (moving in the direction opposite to the example above) →
$FFFFFFFF_H$ – $FFFFFF35_H$ + 1 (make sure to add 1) →
Converted to a decimal value → 203 (x 0.01 mm/sec) →
The current speed is 2.03 mm/sec.

(Note) The data of the response example is simply an example and will vary depending on various conditions.

5.3.11 Current Ampere Reading <<CNOW>>

(1) Function

This query reads the monitor data of the motor current (torque current command value), indicated in units of mA.

(2) Query format

Field	Number of data items (number of bytes)	RTU mode 8-bit data	Remarks
Start	None		Silent interval
Slave address [H]	1	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	1	03	Register reading
Start address [H]	2	900C	Current ampere monitor
Number of registers [H]	4	0002	Reading addresses $900C_H$ to $900D_H$
Error check [H]	2	CRC (16 bits)	
End	None		Silent interval
Total number of bytes	8		

(3) Response format

A response message contains 16 bits of data per register.

Field	Number of data items (number of bytes)	RTU mode 8-bit data	Remarks
Start	None		Silent interval
Slave address [H]	1	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	1	03	Register reading
Number of data bytes [H]	1	04	Reading 2 register = 4 bytes
Data [H]	4	Motor current monitor	Motor current monitor [Hex] The unit is mA.
Error check [H]	2	CRC (16 bits)	
End	None		Silent interval
Total number of bytes	9		

(4) Query sample

A sample query that read the current ampere monitor (from address $900C_H$) of a controller of axis No. 0 is shown below.

- Query (silent intervals are inserted before and after the query)
  01 03 90 0C 00 02 29 08

Field	RTU mode 8-bit data
Start	Silent interval
Slave address [H]	01
Function code [H]	03
Start address [H]	900C
Number of registers [H]	0002
Error check [H]	2908 (in accordance with CRC calculation)
End	Silent interval

The response to the query is as follows.

- Response (silent intervals are inserted before and after the response)
  01 03 04 00 00 01 C8 FA 35

Field	RTU mode 8-bit data
Start	Silent interval
Slave address [H]	01
Function code [H]	03
Number of data bytes [H]	04 (4 bytes = 2 registers)
Data 1 [H]	00 00 01 C8
Error check [H]	FA35 (in accordance with CRC calculation)
End	Silent interval

The current ampere value is “000001C8” → Converted to a decimal value → 456
The current ampere monitor value is 456 mA.

(Note) The data of the response example is simply an example and will vary depending on various conditions.

5.3.12 Deviation Reading <<DEVI>>

(1) Function

This query reads the deviation over a 1-ms period between the position command value and the feedback value (actual position). The unit is pulse. The number of pulses per one motor revolution in mechanical angle varies depending on the encoder used.

(2) Query format

Field	Number of data items (number of bytes)	RTU mode 8-bit data	Remarks
Start	None		Silent interval
Slave address [H]	1	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	1	03	Register reading
Start address [H]	2	900E	Deviation monitor
Number of registers [H]	4	0002	Reading addresses $900E_H$ to $900F_H$
Error check [H]	2	CRC (16 bits)	
End	None		Silent interval
Total number of bytes	8		

(3) Response format

A response message contains 16 bits of data per register.

Field	Number of data items (number of bytes)	RTU mode 8-bit data	Remarks
Start	None		Silent interval
Slave address [H]	1	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	1	03	Register reading
Number of data bytes [H]	1	04	Reading 2 registers = 4 bytes
Data [H]	4	Deviation monitor	Deviation monitor [Hex] The unit is pulse.
Error check [H]	2	CRC (16 bits)	
End	None		Silent interval
Total number of bytes	9		

(4) Query sample

A sample query that reads the deviation monitor (from address $900E_H$) of a controller of axis No. 0 is shown below.

- Query (silent intervals are inserted before and after the query)
  01 03 90 0E 00 02 88 C8

Field	RTU mode 8-bit data
Start	Silent interval
Slave address [H]	01
Function code [H]	03
Start address [H]	900E
Number of registers [H]	0002
Error check [H]	88C8 (in accordance with CRC calculation)
End	Silent interval

The response to the query is as follows.

- Response (silent intervals are inserted before and after the response)
  01 03 04 00 00 00 0B BB F4

Field	RTU mode 8-bit data
Start	Silent interval
Slave address [H]	01
Function code [H]	03
Number of data bytes [H]	04 (4 bytes = 2 registers)
Data 1 [H]	00 00 00 0B
Error check [H]	BBF4 (in accordance with CRC calculation)
End	Silent interval

The deviation monitor is "0000000B" → Converted to a decimal value → 11
The deviation over a 1-ms period between the position command value and the feedback value (actual position) is 11 pulses.

(Note) The data of the response example is simply an example and will vary depending on various conditions.

5.3.13 Total Time after Power On Reading <<STIM>>

(1) Function

This query reads the total time since the controller power was turned on. The unit is msec.
The timer value is not cleared by software reset.

(2) Query format

Field	Number of data items (number of bytes)	RTU mode 8-bit data	Remarks
Start	None		Silent interval
Slave address [H]	1	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	1	03	Register reading
Start address [H]	2	9010	System timer
Number of registers [H]	4	0002	Reading addresses 9010_H to 9011_H
Error check [H]	2	CRC (16 bits)	
End	None		Silent interval
Total number of bytes	8		

(3) Response format

A response message contains 16 bits of data per register.

Field	Number of data items (number of bytes)	RTU mode 8-bit data	Remarks
Start	None		Silent interval
Slave address [H]	1	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	1	03	Register reading
Number of data bytes [H]	1	04	Reading 2 registers = 4 bytes
Data [H]	4	System timer	System timer [Hex] The unit is ms.
Error check [H]	2	CRC (16 bits)	
End	None		Silent interval
Total number of bytes	9		

(4) Query sample

A sample query that reads the system timer value (from address 9010_H) of a controller of axis No. 0 is shown below.

Query (silent intervals are inserted before and after the query)

01 03 90 10 00 02 E8 CE

Field	RTU mode 8-bit data
Start	Silent interval
Slave address [H]	01
Function code [H]	03
Start address [H]	9010
Number of registers [H]	0002
Error check [H]	E8CE (in accordance with CRC calculation)
End	Silent interval

The response to the query is as follows.

Response (silent intervals are inserted before and after the response)

01 03 04 00 02 7A 72 F8 B6

Field	RTU mode 8-bit data
Start	Silent interval
Slave address [H]	01
Function code [H]	03
Number of data bytes [H]	04 (4 bytes = 2 registers)
Data 1 [H]	00 02 7A 72
Error check [H]	F8B6 (in accordance with CRC calculation)
End	Silent interval

The system timer is “00027A72” → Converted to a decimal value → 162418 (ms)

The total time since the controller power was turned on is 162.418 sec.

*** The data of the response example is simply an example and will vary depending on various conditions.**

5.3.14 Special Input Port Input Signal Status Reading<<SIPM>>

(1) Function

This query reads the status of input ports other than the normal input port.
[Refer to 4.3.2 (8), "Data of special input port monitor registers" for the data input via the special input port.]

(2) Query format

Field	Number of data items (number of bytes)	RTU mode 8-bit data	Remarks
Start	None		Silent interval
Slave address [H]	1	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	1	03	Register reading
Start address [H]	2	9012	Special input port monitor
Number of registers [H]	2	0001	Reading addresses 9012_H
Error check [H]	2	CRC (16 bits)	
End	None		Silent interval
Total number of bytes	8		

(3) Response format

A response message contains 16 bits of data per register.

Field	Number of data items (number of bytes)	RTU mode 8-bit data	Remarks
Start	None		Silent interval
Slave address [H]	1	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	1	03	Register reading
Number of data bytes [H]	1	02	Reading 1 register = 2 bytes
Data [H]	2	Special port monitor	Refer to 4.3.2 (8), "List table."
Error check [H]	2	CRC (16 bits)	
End	None		Silent interval
Total number of bytes	7		

(4) Query sample

A sample query that reads the special input port (address 9012_H) of a controller of axis No. 0 is shown below.

Query (silent intervals are inserted before and after the query)

01 03 90 12 00 01 09 0F

Field	RTU mode 8-bit data
Start	Silent interval
Slave address [H]	01
Function code [H]	03
Start address [H]	9012
Number of registers [H]	0001
Error check [H]	090F (in accordance with CRC calculation)
End	Silent interval

The response to the query is as follows.

Response (silent intervals are inserted before and after the response)

01 03 02 43 00 89 74

Field	RTU mode 8-bit data
Start	Silent interval
Slave address [H]	01
Function code [H]	03
Number of data bytes [H]	02 (2 bytes = 1 register)
Data 1 [H]	43 00
Error check [H]	8974 (in accordance with CRC calculation)
End	Silent interval

*** The data of the response example is simply an example and will vary depending on various conditions.**

5.3.15 Zone Output Signal Status Reading<<ZONS>>

(1) Function

This query reads the status of zone output.
[Refer to 4.3.2 (9), "Data of zone status registers."]

(2) Query format

Field	Number of data items (number of bytes)	RTU mode 8-bit data	Remarks
Start	None		Silent interval
Slave address [H]	1	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	1	03	Register reading
Start address [H]	2	9013	Zone status query
Number of registers [H]	2	0001	Reading address 9013_H
Error check [H]	2	CRC (16 bits)	
End			Silent interval
Total number of bytes	8		

(3) Response format

A response message contains 16 bits of data per register.

Field	Number of data items (number of bytes)	RTU mode 8-bit data	Remarks
Start	None		Silent interval
Slave address [H]	1	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	1	03	Register reading
Number of data bytes [H]	1	02	Reading 1 register = 2 bytes
Data [H]	2	Zone status	Refer to 4.3.2 (9), "List table"
Error check [H]	2	CRC (16 bits)	
End	None		Silent interval
Total number of bytes	7		

(4) Query sample

A sample query that reads the zone output status (address 9013_H) of a controller of axis No. 0 is shown below.

Query (silent intervals are inserted before and after the query)

01 03 90 13 00 01 58 CF

Field	RTU mode 8-bit data
Start	Silent interval
Slave address [H]	01
Function code [H]	03
Start address [H]	9013
Number of registers [H]	0001
Error check [H]	58CF (in accordance with CRC calculation)
End	Silent interval

The response to the query is as follows.

Response (silent intervals are inserted before and after the response)

01 03 02 00 00 B8 44

Field	RTU mode 8-bit data
Start	Silent interval
Slave address [H]	01
Function code [H]	03
Number of data bytes [H]	02 (2 bytes = 1 register)
Data 1 [H]	00 00
Error check [H]	B844 (in accordance with CRC calculation)
End	Silent interval

*** The data of the response example is simply an example and will vary depending on various conditions.**

5.3.16 Position Complete Number Reading<<POSS>>

(1) Function

This query reads the position complete number.
[Refer to 4.3.2 (10), "Data of position number status register."]

(2) Query format

Field	Number of data items (number of bytes)	RTU mode 8-bit data	Remarks
Start	None		Silent interval
Slave address [H]	1	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	1	03	Register reading
Start address [H]	2	9014	Position number status
Number of registers [H]	2	0001	Reading address 9014_H
Error check [H]	2	CRC (16 bits)	
End	None		Silent interval
Total number of bytes	8		

(3) Response format

A response message contains 16 bits of data per register.

Field	Number of data items (number of bytes)	RTU mode 8-bit data	Remarks
Start	None		Silent interval
Slave address [H]	1	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	1	03	Register reading
Number of data bytes [H]	1	02	Reading 1 register = 2 bytes
Data [H]	2	Position number status	Refer to 4.3.2 (10), "List table."
Error check [H]	2	CRC (16 bits)	
End	None		Silent interval
Total number of bytes	7		

(4) Query sample

A sample query that reads the position complete (address 9014_H) of a controller of axis No. 0 is shown below.

Query (silent intervals are inserted before and after the query)

01 03 90 14 00 01 E9 0E

Field	RTU mode 8-bit data
Start	Silent interval
Slave address [H]	01
Function code [H]	03
Start address [H]	9014
Number of registers [H]	0001
Error check [H]	E90E (in accordance with CRC calculation)
End	Silent interval

The response to the query is as follows.

Response (silent intervals are inserted before and after the response)

01 03 02 00 00 B8 44

Field	RTU mode 8-bit data
Start	Silent interval
Slave address [H]	01
Function code [H]	03
Number of data bytes [H]	02 (2 bytes = 1 register)
Data 1 [H]	00 00
Error check [H]	B844 (in accordance with CRC calculation)
End	Silent interval

*** The data of the response example is simply an example and will vary depending on various conditions.**

5.3.17 Force Feedback Data Read <<FBFC>> --- SCON-CA Only

(1) Function

The monitored data of load cell measurement (push force) is read.
The unit is 0.01 N.

(2) Query format

Field	Number of data items (number of bytes)	RTU mode 8-bit data	Remarks
Start	None		Silent interval
Slave address [H]	1	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	1	03	Register reading
Start address [H]	2	901E	Force feedback monitor
Number of registers [H]	4	0002	Reading address $901F_H$~$901F_H$
Error check [H]	2	CRC (16 bits)	
End	None		Silent interval
Total number of bytes	8		

(3) Response format

A response message contains 16 bits of data per register.

Field	Number of data items (number of bytes)	RTU mode 8-bit data	Remarks
Start	None		Silent interval
Slave address [H]	1	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	1	03	Register reading
Number of data bytes [H]	1	04	Reading 1 register = 4 bytes
Data [H]	4	Load cell measurement	Current push force [N] Unit: 0.01 N
Error check [H]	2	CRC (16 bits)	
End	None		Silent interval
Total number of bytes	9		

(4) Query sample

An example of use is shown, where the current measurement on the load cell connected to controller axis 0 is read.

- Query (silent intervals are inserted before and after the query)
  01 03 90 0A 00 02 89 0D

Field	RTU mode 8-bit data
Start	Silent interval
Slave address [H]	01
Function code [H]	03
Start address [H]	901E
Number of registers [H]	0002
Error check [H]	890D (in accordance with CRC calculation)
End	Silent interval

The response to the query is as follows.

- Response (silent intervals are inserted before and after the response)
  01 03 04 00 00 03 E4 FA 88

Field	RTU mode 8-bit data
Start	Silent interval
Slave address [H]	01
Function code [H]	03
Number of data bytes [H]	04 (4 bytes = 2 register)
Data 1 [H]	00 00 03 E4
Error check [H]	FA88 (in accordance with CRC calculation)
End	Silent interval

Example 1) The current measurement on the load cell is “000003E4, ” which is converted to a decimal value, or 996 (x 0.01 N) → The current push force is 9.96 N.

Example 2) If the current measurement reading on the load cell is “FFFFFF35” (tensile state[(Note 2)]), the formula $FFFFFFFF_H$ – $FFFFFF35_H$ + 1 (1 must be added) applies. The result is converted to a decimal value, or 203 (x 0.01 N) → The current tensile force[(Note 2)] is 2.03 N.

(Note 1) This is only one example of response. The specific response varies depending on each situation.

(Note 2) If a force is applied in the tensile direction, the load cell will break.

5.4 Operation Commands and Data Rewrite (Query Using Code 05)

5.4.1 Writing to Coil

(1) Function

Change (write) the status of DO (Discrete Output) of a slave to either ON or OFF.
In case of broadcast transmission, the coils at the specified address of all slaves are rewritten.

(2) Start address list

Start address [H]	Symbol	Function
0401	SFTY	Safety speed command
0403	SON	Servo ON command
0407	ALRS	Alarm reset command
0408	BKRL	Brake forced-release command
040A	STP	Pause command
040B	HOME	Home return command
040C	CSTR	Positioning start command
0411	JISL	Jog/inch switching
0414	MOD	Teaching mode command
0415	TEAC	Position data load command
0416	JOG+	Jog+ command
0417	JOG-	Jog- command
0418	ST7	Start position 7 (solenoid valve mode)
0419	ST6	Start position 6 (solenoid valve mode)
041A	ST5	Start position 5 (solenoid valve mode)
041B	ST4	Start position 4 (solenoid valve mode)
041C	ST3	Start position 3 (solenoid valve mode)
041D	ST2	Start position 2 (solenoid valve mode)
041E	ST1	Start position 1 (solenoid valve mode)
041F	ST0	Start position 0 (solenoid valve mode)
0426	CLBR	Load cell calibration command
0427	PMSL	PIO/Modbus switching specification
042C	STOP	Deceleration stop

5.4.2 Safety Speed Enable/Disable Switching (SFTY)

(1) Function
This query enables/disables the speed specified by user parameter No. 35, “Safety speed.”
Enabling the safety speed in the MANU mode will limit the speeds of all movement commands.

(2) Query format

Field	Number of data items (number of bytes)	RTU mode 8-bit data	Remarks
Start	None		Silent interval
Slave address [H]	1	Arbitrary	Axis number + 1 (01_H to 10_H) 00_H when broadcast is specified
Function code [H]	1	05	Write to a single coil DO.
Start address [H]	2	0401	Safety speed command
Changed data [H]	2	Arbitrary	Safety speed enabled: $FF00_H$ Safety speed disabled: 0000_H
Error check [H]	2	CRC (16 bits)	
End	None		Silent interval
Total number of bytes	8		

(3) Response
If the change is successful, the response message will be the same as the query.
If invalid data is sent, an exception response (refer to section 7) will be returned, or no response will be returned.

(4) Query sample

A sample query that enables the safety speed of a controller of axis No. 0 is shown below.
Query (silent intervals are inserted before and after the query)
01 05 04 01 FF 00 DC CA

Field	RTU mode 8-bit data
Start	Silent interval
Slave address [H]	01
Function code [H]	05
Start address [H]	0401
Changed data [H]	FF00
Error check [H]	DCCA (in accordance with CRC calculation)
End	Silent interval

If the change is successful, the response message will be the same as the query.

5.4.3 Servo ON/OFF <<SON>>

(1) Function

Control ON/OFF of the servo.
When "Servo ON" is specified by the new data, the servo will turn ON after elapse of the manufacturer parameter "Servo ON delay time." However, the following conditions must be satisfied:

- The EMG status bit in device status register 1 is 0.
- The major failure status bit in device status register 1 is 0.
- The enable status bit in device status register 2 is 1.
- The auto servo OFF status in the system status register is 0.

(2) Query Format

Field	Number of data items (number of bytes)	RTU mode 8-bit data	Remarks
Start	None		Silent interval
Slave address [H]	1	Arbitrary	Axis number + 1 (01_H to 10_H) 00_H when broadcast is specified
Function code [H]	1	05	Write to a single coil DO.
Start address [H]	2	0403	Servo ON/OFF command
Changed data [H]	2	Arbitrary	Servo ON: $FF00_H$ Servo OFF: 0000_H
Error check [H]	2	CRC (16 bits)	
End	None		Silent interval
Total number of bytes	17		

* If a teaching pendant or PC software is connected before the control establishes communication with the host, the servo is turned OFF, and then the teaching pendant/PC software is removed, the servo cannot be turned ON/OFF via commands received from with the host.
In this case, restore the RC controller power, or make sure the SIO port connection is removed while the servo is ON.

(3) Response

If the change is successful, the response message will be the same as the query.
If invalid data is sent, an exception response (refer to section 7) will be returned, or no response will be returned.

(4) Query sample

A sample query that turns on the servo of a controller of axis No. 0 is shown below.
Query (silent intervals are inserted before and after the query)
01 05 04 03 FF 00 7D 0A

Field	RTU mode 8-bit data
Start	Silent interval
Slave address [H]	01
Function code [H]	05
Start address [H]	0403
Changed data [H]	FF00
Error check [H]	7D0A (in accordance with CRC calculation)
End	Silent interval

If the change is successful, the response message will be the same as the query.

5.4.4 Alarm Reset <<ALRS>>

(1) Function

When the alarm reset edge is turned on (the data is first set to FF00$_H$ and then changed to 0000$_H$), **alarms will be reset**.
If any alarm cause has not been removed, the same alarm will be generated again. If the alarm reset edge is turned on while the actuator is paused, **the remaining travel will be cancelled**.
When alarms are reset, make sure to write changed data of 0000$_H$ to restore the normal status.

(2) Query format

Field	Number of data items (number of bytes)	RTU mode 8-bit data	Remarks
Start	None		Silent interval
Slave address [H]	1	Arbitrary	Axis number + 1 (01$_H$ to 10$_H$) 00$_H$ when broadcast is specified
Function code [H]	1	05	Write to a single coil DO.
Start address [H]	2	0407	Alarm reset command
Changed data [H]	2	Arbitrary	Execute alarm reset: FF00$_H$ Normal: 0000$_H$
Error check [H]	2	CRC (16 bits)	
End	None		Silent interval
Total number of bytes	8		

(3) Response

If the change is successful, the response message will be the same as the query.
If invalid data is sent, an exception response (refer to section 7) will be returned, or no response will be returned.

(4) Query sample

A sample query that resets the alarms of a controller of axis No. 0 is shown below.
Query (silent intervals are inserted before and after the query)
First time 01 05 04 07 FF 00 3C CB --- Execute alarm reset
Second time 01 05 04 07 00 00 7D 3B --- Restore normal status

Field	RTU mode 8-bit data
Start	Silent interval
Slave address [H]	01
Function code [H]	05
Start address [H]	0407
Changed data [H]	First time: FF00 Second time: 0000 (Write 0000_H after resetting alarms to restore the normal status.)
Error check [H]	First time: 3CCB (in accordance with CRC calculation) Second time: 7D3B (in accordance with CRC calculation)
End	Silent interval

If the change is successful, the response message will be the same as the query.

5.4.5 Brake Forced Release <<BKRL>>

(1) Function

Brake control is linked to servo ON/OFF. The brake can be forcefully released even when the servo is ON.

(2) Query format

Field	Number of data items (number of bytes)	RTU mode 8-bit data	Remarks
Start	None		Silent interval
Slave address [H]	1	Arbitrary	Axis number + 1 (01_H to 10_H) 00_H when broadcast is specified
Function code [H]	1	05	Write to a single coil DO.
Start address [H]	2	0408	Break forced release command
Changed data [H]	2	Arbitrary	Brake forced release: $FF00_H$ Normal: 0000_H
Error check [H]	2	CRC (16 bits)	
End	None		Silent interval
Total number of bytes	8		

* If a teaching pendant or PC software is connected before the control establishes communication with the host, the servo is turned OFF, and then the teaching pendant/PC software is removed, the servo cannot be turned ON/OFF via commands received from with the host.
In this case, restore the RC controller power, or make sure the SIO port connection is removed while the servo is ON.

(3) Response

If the change is successful, the response message will be the same as the query.
If invalid data is sent, an exception response (refer to section 7) will be returned, or no response will be returned.

(4) Query sample

A sample query that forcefully releases the break of a controller of axis No. 0 is shown below.
Query (silent intervals are inserted before and after the query)
01 05 04 08 FF 00 0C C8

Field	RTU mode 8-bit data
Start	Silent interval
Slave address [H]	01
Function code [H]	05
Start address [H]	0408
Changed data [H]	FF00
Error check [H]	0CC8 (in accordance with CRC calculation)
End	Silent interval

If the change is successful, the response message will be the same as the query.

5.4.6 Pause <<STP>>

(1) Function

If the pause command is transmitted during movement, the actuator decelerates and stops. If the status is set back to normal again, the actuator resumes moving for the remaining distance.
As long as the pause command is being transmitted, all motor movement is inhibited.
If the alarm reset command bit is set while the actuator is paused, the remaining travel will be cancelled.
If this bit is set during home return, the movement command will be held if the actuator has not yet reversed after contacting the mechanical end. If the actuator has already reversed after contacting the mechanical end, home return will be repeated from the beginning.

(2) Query format

Field	Number of data items (number of bytes)	RTU mode 8-bit data	Remarks
Start	None		Silent interval
Slave address [H]	1	Arbitrary	Axis number + 1 (01_H to 10_H) 00_H when broadcast is specified
Function code [H]	1	05	Write to a single coil DO.
Start address [H]	2	040A	Pause command
Changed data [H]	2	Arbitrary	Pause command: $FF00_H$ Normal: 0000_H
Error check [H]	2	CRC (16 bits)	
End	None		Silent interval
Total number of bytes	8		

(3) Response

If the change is successful, the response message will be the same as the query.
If invalid data is sent, an exception response (refer to section 7) will be returned, or no response will be returned.

(4) Query sample

A sample query that pauses a controller of axis No. 0 is shown below.
Query (silent intervals are inserted before and after the query)
01 05 04 0A FF 00 AD 08

Field	RTU mode 8-bit data
Start	Silent interval
Slave address [H]	01
Function code [H]	05
Start address [H]	040A
Changed data [H]	FF00
Error check [H]	AD08 (in accordance with CRC calculation)
End	Silent interval

If the change is successful, the response message will be the same as the query.

5.4.7 Home Return <<HOME>>

(1) Function

Home return operation will start if a rising edge in the home return command signal is detected (the data is first set to 0000_H and then changed to $FF00_H$). Upon home return completion, the HEND bit will become 1. This command can be input as many times as desired even after home return completion.

(2) Query format

Field	Number of data items (number of bytes)	RTU mode 8-bit data	Remarks
Start	None		Silent interval
Slave address [H]	1	Arbitrary	Axis number + 1 (01_H to 10_H) 00_H when broadcast is specified
Function code [H]	1	05	Write to a single coil DO.
Start address [H]	2	040B	Home return command
Changed data [H]	2	Arbitrary	Execute home return: $FF00_H$ Normal: 0000_H
Error check [H]	2	CRC (16 bits)	
End	None		Silent interval
Total number of bytes	8		

* The servo must be ON before a home return command is issued.
If a teaching pendant or PC software is connected before the control establishes communication with the host, the servo is turned OFF, and then the teaching pendant/PC software is removed, the servo cannot be turned ON/OFF via commands received from with the host.
In this case, restore the RC controller power, or make sure the SIO port connection is removed while the servo is ON.

(3) Response

If the change is successful, the response message will be the same as the query.
If invalid data is sent, an exception response (refer to section 7) will be returned, or no response will be returned.

(4) Query sample

A query example that executes home return operation of a controller of axis No. 0 is shown here.

Query (silent intervals are inserted before and after the query)

First time 01 05 04 0B 00 00 BD 38 --- Set normal status

Second time 01 05 04 0B FF 00 FC C8 --- Execute home return

Field	RTU mode 8-bit data
Start	Silent interval
Slave address [H]	01
Function code [H]	05
Start address [H]	040B
Changed data [H]	First time: 0000 Second time: FF00 (Send data twice to set the rising edge.)
Error check [H]	First time: 3CCB (in accordance with CRC calculation) Second time: 7D3B (in accordance with CRC calculation)
End	Silent interval

If the change is successful, the response message will be the same as the query.

5.4.8 Positioning Start Command <<CSTR>>

(1) Function

If the rising edge of the positioning start command is detected (the data is first set to 0000_H and then changed to $FF00_H$), the actuator will move to the position specified by the position number stored in the position number specification register (POSR:$0D03_H$). If nothing is done after the position start command ($FF00_H$ is read and no new data is written), a position complete will not be output even when the actuator enters the positioning band (write 0000_H and restore the normal status).
If this command is executed when home return has never been performed after the power was turned on (when the HEND bit is 0), the actuator will perform home return and then start moving to the target position.

* The target position, speed and all other operation parameters must be set in the position table (nonvolatile memory) of the controller in advance.

(2) Query format

Field	Number of data items (number of bytes)	RTU mode 8-bit data	Remarks
Start	None		Silent interval
Slave address [H]	1	Arbitrary	Axis number + 1 (01_H to 10_H) 00_H when broadcast is specified
Function code [H]	1	05	Write to a single coil DO.
Start address [H]	2	040C	Positioning start command
Changed data [H]	2	Arbitrary	Positioning start command: $FF00_H$ Normal: 0000_H
Error check [H]	2	CRC (16 bits)	
End	None		Silent interval
Total number of bytes	8		

(3) Response

If the change is successful, the response message will be the same as the query.
If invalid data is sent, an exception response (refer to section 7) will be returned, or no response will be returned.

(4) Query sample

A sample query that moves the actuator of a controller of axis No. 0 to the position specified by the position number stored in the position number specification register (POSR: $0D03_H$) is shown below.

Query (silent intervals are inserted before and after the query)

First time 01 05 04 0C FF 00 4D 09 --- Move to the specified position

Second time 01 05 04 0C 00 00 0C F9 --- Restore to the normal status

Field	RTU mode 8-bit data
Start	Silent interval
Slave address [H]	01
Function code [H]	05
Start address [H]	040C
Changed data [H]	First time: FF00 Second time: 0000 (Restore to the normal status.)
Error check [H]	First time: 4D09 (in accordance with CRC calculation) Second time: 0CF9 (in accordance with CRC calculation)
End	Silent interval

If the change is successful, the response message will be the same as the query.

5.4.9 Jog/Inch Switching <<JISL>>

(1) Function

This bit switches between jogging and inching.
If this bit switches while the actuator is jogging, the actuator will decelerate to a stop.
If this bit switches while the actuator is inching, the inching movement will continue.

(2) Query format

Field	Number of data items (number of bytes)	RTU mode 8-bit data	Remarks
Start	None		Silent interval
Slave address [H]	1	Arbitrary	Axis number + 1 (01_H to 10_H) 00_H when broadcast is specified
Function code [H]	1	05	Write to a single coil DO.
Start address [H]	2	0411	Jog/inch switching
Changed data [H]	2	Arbitrary	Inching operation status: $FF00_H$ Jogging operation status: 0000_H
Error check [H]	2	CRC (16 bits)	
End	None		Silent interval
Total number of bytes	8		

(3) Response

If the change is successful, the response message will be the same as the query.
If invalid data is sent, an exception response (refer to section 7) will be returned, or no response will be returned.

(4) Query sample

A sample query that switches the operation of a controller of axis No. 0 to inching is shown below.
Query (silent intervals are inserted before and after the query)
01 05 04 11 FF 00 AD 08

Field	RTU mode 8-bit data
Start	Silent interval
Slave address [H]	01
Function code [H]	05
Start address [H]	0411
Changed data [H]	FF00
Error check [H]	AD08 (in accordance with CRC calculation)
End	Silent interval

If the change is successful, the response message will be the same as the query.

5.4.10 Teaching Mode Command <<MOD>>

(1) Function

This bit switches between the normal operation mode and teaching mode.

(2) Query format

Field	Number of data items (number of bytes)	RTU mode 8-bit data	Remarks
Start	None		Silent interval
Slave address [H]	1	Arbitrary	Axis number + 1 (01_H to 10_H) 00_H when broadcast is specified
Function code [H]	1	05	Write to a single coil DO.
Start address [H]	2	0414	Switch between the normal mode and the teaching mode.
Changed data [H]	2	Arbitrary	Teaching mode: $FF00_H$ Normal operation mode: 0000_H
Error check [H]	2	CRC (16 bits)	
End	None		Silent interval
Total number of bytes	8		

(3) Response

If the change is successful, the response message will be the same as the query.
If invalid data is sent, an exception response (refer to section 7) will be returned, or no response will be returned.

(4) Query sample

A sample query that switches the operation mode of a controller of axis No. 0 to teaching mode is shown below.

Query (silent intervals are inserted before and after the query)

01 05 04 14 FF 00 CD 0E

Field	RTU mode 8-bit data
Start	Silent interval
Slave address [H]	01
Function code [H]	05
Start address [H]	0414
Changed data [H]	FF00
Error check [H]	CD0E (in accordance with CRC calculation)
End	Silent interval

If the change is successful, the response message will be the same as the query.

5.4.11 Position Data Load Command <<TEAC>>

(1) Function

The current position is acquired by writing this command (write FF00H) when the teaching mode command (5.4.10) is FF00H (teaching command).
The current position data will be written in the position number specified by the position number specification register when the aforementioned condition was detected.
If other position data fields are empty, the default parameter values will be written at the same time in the empty fields other than the target position (positioning band INP, speed VCMD, acceleration/deceleration speed ACMD, and control flag CTLF).
After sending this command (write $FF00_H$), keep the status as is for 20 ms or longer.

(2) Query format

Field	Number of data items (number of bytes)	RTU mode 8-bit data	Remarks
Start	None		Silent interval
Slave address [H]	1	Arbitrary	Axis number + 1 (01_H to 10_H) 00_H when broadcast is specified
Function code [H]	1	05	Write to a single coil DO.
Start address [H]	2	0415	Position data load command
Changed data [H]	2	Arbitrary	Position data load command: $FF00_H$ Normal: 0000_H
Error check [H]	2	CRC (16 bits)	
End	None		Silent interval
Total number of bytes	8		

(3) Response

If the change is successful, the response message will be the same as the query.
If invalid data is sent, an exception response (refer to section 7) will be returned, or no response will be returned.

(4) Query sample

A sample query that acquires the current position when a controller of axis No. 0 is in the teaching mode is shown below.

Query (silent intervals are inserted before and after the query)

01 05 04 15 FF 00 9C CE

Field	RTU mode 8-bit data
Start	Silent interval
Slave address [H]	01
Function code [H]	05
Start address [H]	0415
Changed data [H]	FF00
Error check [H]	9CCE (in accordance with CRC calculation)
End	Silent interval

If the change is successful, the response message will be the same as the query.

5.4.12 Jog+ Command <<JOG+>>

(1) Function

- The actuator performs either jog or inching operation.
  If the jog+ command (changed data $FF00_H$) is sent when the jog/inch switching command (5.4.9) is set to 0000_H (set to jog), the actuator will jog in the direction opposite home. The speed and acceleration/deceleration speed conform to the PIO jog speed set by user parameter No. 26 and rated acceleration/deceleration speed, respectively.
  If the jog+ command (changed data 0000_H) is sent or the jog- command (5.4.13, changed data $FF00_H$) is sent while the actuator is moving jog, the actuator will decelerate to a stop.
- If the jog+ command rising edge is set (the data is first set to 0000H and changed to FF00H) while the jog/inch switching command (5.4.9) is $FF00_H$ (set to inching), the actuator will inch in the direction opposite home. The speed, travel and acceleration/deceleration speed conform to user parameter No. 26 (PIO jogging speed), user parameter No. 48 (PIO inching distance), and rated acceleration/deceleration speed, respectively.

(2) Query format

Field	Number of data items (number of bytes)	RTU mode 8-bit data	Remarks
Start	None		Silent interval
Slave address [H]	1	Arbitrary	Axis number + 1 (01_H to 10_H) 00_H when broadcast is specified
Function code [H]	1	05	Write to a single coil DO.
Start address [H]	2	0416	Jog+ command
Changed data [H]	2	Arbitrary	Jog+ command: $FF00_H$ Normal: 0000_H
Error check [H]	2	CRC (16 bits)	
End	None		Silent interval
Total number of bytes	8		

(3) Response

If the change is successful, the response message will be the same as the query.
If invalid data is sent, an exception response (refer to section 7) will be returned, or no response will be returned.

(4) Query sample

[1] A sample query that makes a controller of axis No. 0 jog is shown below.
Query (silent intervals are inserted before and after the query)
01 05 04 16 FF 00 6C CE

Field	RTU mode 8-bit data
Start	Silent interval
Slave address [H]	01
Function code [H]	05
Start address [H]	0416
Changed data [H]	FF00
Error check [H]	6CCE (in accordance with CRC calculation)
End	Silent interval

If the change is successful, the response message will be the same as the query.

[2] A sample query that makes a controller of axis No. 0 inch is shown below.
Query (silent intervals are inserted before and after the query)
First time 01 05 04 16 FF 00 6C CE --- Perform inching movement
Second time 01 05 04 16 00 00 2D 3E --- Restore the normal status

Field	RTU mode 8-bit data
Start	Silent interval
Slave address [H]	01
Function code [H]	05
Start address [H]	0416
Changed data [H]	First time: FF00 Second time: 0000 (Restore the normal status.)
Error check [H]	First time: 6CCE (in accordance with CRC calculation) Second time: 2D3E (in accordance with CRC calculation)
End	Silent interval

If the change is successful, the response message will be the same as the query.

5.4.13 Jog- Command <<JOG->>

(1) Function

- The actuator performs either jog or inching operation.
  If the jog- command (changed data $FF00_H$) is sent when the jog/inch switching command (5.4.9) is set to 0000_H (set to jog), the actuator will jog in the direction of home. The speed and acceleration/deceleration speed conform to the PIO jog speed set by user parameter No. 26 and rated acceleration/deceleration speed, respectively.
  If the jog- command (changed data 0000_H) is sent or the jog+ command (5.4.12, changed data $FF00_H$) is sent while the actuator is moving, the actuator will decelerate to a stop.
- If the jog- command rising edge is set while the jog/inch switching command (5.4.9) is $FF00_H$ (set to inching), the actuator will inch in the direction of home. The speed, travel and acceleration/deceleration speed conform to user parameter No. 26 (PIO jogging sped), user parameter No. 48 (PIO inching distance), and rated acceleration/deceleration speed, respectively.

(2) Query format

Field	Number of data items (number of bytes)	RTU mode 8-bit data	Remarks
Start	None		Silent interval
Slave address [H]	1	Arbitrary	Axis number + 1 (01_H to 10_H) 00_H when broadcast is specified
Function code [H]	1	05	Write to a single coil DO.
Start address [H]	2	0417	Jog- command
Changed data [H]	2	Arbitrary	Jog- command: $FF00_H$ Normal: 0000_H
Error check [H]	2	CRC (16 bits)	
End	None		Silent interval
Total number of bytes	8		

(3) Response

If the change is successful, the response message will be the same as the query.
If invalid data is sent, an exception response (refer to section 7) will be returned, or no response will be returned.

(4) Query sample

[1] A sample query that makes a controller of axis No. 0 jog is shown below.
Query (silent intervals are inserted before and after the query)
01 05 04 17 FF 00 3D 0E

Field	RTU mode 8-bit data
Start	Silent interval
Slave address [H]	01
Function code [H]	05
Start address [H]	0417
Changed data [H]	FF00
Error check [H]	3D0E (in accordance with CRC calculation)
End	Silent interval

If the change is successful, the response message will be the same as the query.

[2] A sample query that makes a controller of axis No. 0 inch is shown below.
Query (silent intervals are inserted before and after the query)
First time 01 05 04 17 FF 00 3D 0E --- Perform inching movement
Second time 01 05 04 17 00 00 7C FE --- Restore the normal status

Field	RTU mode 8-bit data
Start	Silent interval
Slave address [H]	01
Function code [H]	05
Start address [H]	0417
Changed data [H]	First time: FF00 Second time: 0000 (Restore the normal status)
Error check [H]	First time: 3D0E (in accordance with CRC calculation) Second time: 7CFE (in accordance with CRC calculation)
End	Silent interval

If the change is successful, the response message will be the same as the query.

5.4.14 Start Positions 0 to 7 <<ST0 to ST7>> Movement Command (Limited to PIO Patterns 4 and 5)

(1) Function

The actuator moves to the specified position number position.
The movement command for start position 0 to 7 is effective only when PIO pattern 4 or 5 (solenoid valve mode) is selected.
The movement command is sent by enabling either one of ST0 to ST7 in 5.4.14 (5), "Start address" (write new value $FF00_H$ when 0000_H is set).
If a position other than the valid start positions is selected, "085: Moving position number error" will be generated.
Either level operation or edge operation can be selected using user parameter No. 27, "Movement command type."

(2) Query format

Field	Number of data items (number of bytes)	RTU mode 8-bit data	Remarks
Start	None		Silent interval
Slave address [H]	1	Arbitrary	Axis number + 1 (01_H to 10_H) 00_H when broadcast is specified
Function code [H]	1	05	Write to a single coil DO
Start address [H]	2	Arbitrary	Refer to 5.4.14 (5), "Start address."
Changed data [H]	2	Arbitrary	*1 Operation command ON: $FF00_H$ Operation command OFF: 0000_H
Error check [H]	2	CRC (16 bits)	
End	None		Silent interval
Total number of bytes	8		

*1 If user parameter No. 27, "Movement command type" is set to "level operation, " the actuator decelerates to a stop by overwriting $FF00_H$ with 0000_H.

(3) Response

If the change is successful, the response message will be the same as the query.
If invalid data is sent, an exception response (refer to section 7) will be returned, or no response will be returned.

(4) Query sample

A sample query that moves a controller of axis No. 0 to start position 2 is shown below.
An example of start position setting.

0	0.00	150.00	0.30	0.30
1	25.00	150.00	0.30	0.30
2	50.00	150.00	0.30	0.30
3	0.00	150.00	0.20	0.20

Fig. 5.2

Query (silent intervals are inserted before and after the query)
First time 01 05 04 1D 00 00 5C FC --- Write 0000_H to set the edge
Second time 01 05 04 1D FF 00 1D 0C --- Movement command

Field	RTU mode 8-bit data
Start	Silent interval
Slave address [H]	01
Function code [H]	05
Start address [H]	041D
Changed data [H]	First time: 0000 Second time: FF00
Error check [H]	First time: 5CFC (in accordance with CRC calculation) Second time: 1D0C (in accordance with CRC calculation)
End	Silent interval

If the change is successful, the response message will be the same as the query.

(5) Start address

Address	Symbol	Name	Function
0418	ST7	Start position 7	Move to position 7
0419	ST6	Start position 6	Move to position 6
041A	ST5	Start position 5	Move to position 5
041B	ST4	Start position 4	Move to position 4
041C	ST3	Start position 3	Move to position 3
041D	ST2	Start position 2	Move to position 2
041E	ST1	Start position 1	Move to position 1
041F	ST0	Start position 0	Move to position 0

5.4.15 Load Cell Calibration Command <<CLBR>> --- A dedicated load cell must be connected.

(1) Function --- SCON-CA only

The dedicated load cell is calibrated.
The factory setting of your load cell is that the ON status corresponds to a no-load state. If you want to define the reference state as a condition where a work part (load) is installed, calibrate the load cell.
Also calibrate the load cell in other situations as necessary (readjustment, inspection, etc.).

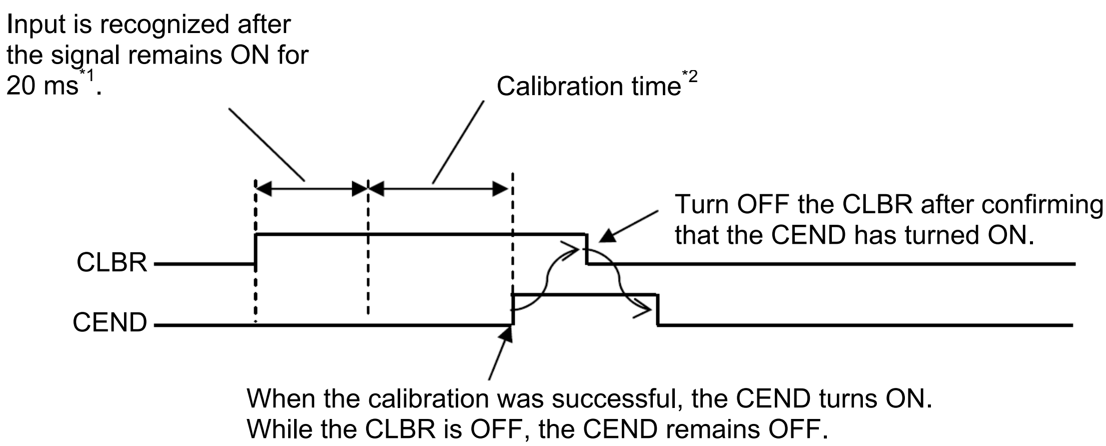
(2) Query format

Field	Number of data items (number of bytes)	RTU mode 8-bit data	Remarks
Start	None		Silent interval
Slave address [H]	1	Arbitrary	Axis number + 1 (01_H to 10_H) 00_H when broadcast is specified
Function code [H]	1	05	Write to a single coil DO
Start address [H]	2	0426	Load cell calibration command
Changed data [H]	2	Arbitrary	Calibration command: $FF00_H$ Normal operation: 0000_H
Error check [H]	2	CRC (16 bits)	
End	None		Silent interval
Total number of bytes	8		

(3) Calibration procedure

[1] Stop the actuator operation. (The load cell cannot be calibrated while the actuator is performing any axis operation or push-motion operation or being paused, in which case 0E1 (load cell calibration error) alarm generates.)
[2] Turn this signal ON and keep it ON for at least 20 ms.
[3] When the calibration is complete, the calibration complete signal (CEND of device status register 1 explained in 4.3.2 (4)) turns ON. After confirming that the CEND has turned ON, turn OFF the CLBR.
If the calibration was unsuccessful, a 0E1 (load cell calibration error) alarm generates.

Caution: Normal operation commands are not accepted while the CLBR is ON.



*1 If the CLBR is turned OFF during this period, calibration will not be performed because the signal is not yet recognized as having been input.
*2 If the CLBR is turned OFF during this period, an alarm will generate.

(4) Response

A response message to be sent following a successful change should be the same as the query.
If any invalid data has been sent, an exception response (refer to 7) will be returned or no response will be returned at all.

(5) Example of use

Calibrate the dedicated load cell connected to controller axis 0.
Query (Silent intervals are inserted before and after the query.)
01 05 04 26 FF 00 6C C1

Field	RTU mode 8-bit data
Start	Silent interval
Slave address [H]	01
Function code [H]	05
Start address [H]	0427
Changed data [H]	FF00
Error check [H]	6CC1 (in accordance with CRC calculation)
End	Silent interval

If the change is successful, the response message will be the same as the query.

5.4.16 PIO/Modbus Switching Setting <<PMSL>>

(1) Function

PIO external command signals can be enabled or disabled.

(2) Query format

Field	Number of data items (number of bytes)	RTU mode 8-bit data	Remarks
Start	None		Silent interval
Slave address [H]	1	Arbitrary	Axis number + 1 (01_H to 10_H) 00_H when broadcast is specified
Function code [H]	1	05	Write to a single coil DO.
Start address [H]	2	0427	PIO/Modbus switching setting
Changed data [H]	2	Arbitrary	*1 Enable Modus commands: $FF00_H$ Disable Modbus commands: 0000_H
Error check [H]	2	CRC (16 bits)	
End	None		Silent interval
Total number of bytes	8		

*1 • Enable Modbus commands (ON) (disable PIO command): $FF00_H$
Operation via PIO signals is not possible.
• Disable Modbus commands (OFF) (enable PIO command): 0000_H
Operation via external PIO signals is possible.

Supplement: If the Modbus command is enabled, the PIO status at change is maintained. If the Modbus command is switched to disabled, the operation status changes according to the current PIO status. Note that even if the status of signals that operate via edge detection has been changed, edge detection is ignored.

(3) Precaution

- On a model equipped with an operation mode switch, "Enable PIO commands" will be specified when the switch is set to the AUTO mode, or "Disable PIO commands" will be specified when the switch is set to the MANU mode.
- On a non-PIO model, the default setting is "Disable PIO commands."
- If IAI's tool (teaching pendant or PC software) is connected, "Teaching modes 1, 2" and "Monitor modes 1, 2" are available as tool modes. The correspondence between these modes and PIO enable/disable specifications are as follows:
  "Monitor modes 1, 2" → "Enable PIO commands"
  "Teaching modes 1, 2" → "Disable PIO commands"

(4) Response

If the change is successful, the response message will be the same as the query.
If invalid data is sent, an exception response (refer to section 7) will be returned, or no response will be returned.

(5) Query sample

A sample query that enables the Modbus command of the operation of a controller of axis No. 0 is shown below.
Query (silent intervals are inserted before and after the query)
01 05 04 27 FF 00 3D 01

Field	RTU mode 8-bit data
Start	Silent interval
Slave address [H]	01
Function code [H]	05
Start address [H]	0427
Changed data [H]	FF00
Error check [H]	3D01 (in accordance with CRC calculation)
End	Silent interval

If the change is successful, the response message will be the same as the query.

5.4.17 Deceleration Stop <<STOP>>

(1) Function

The actuator will start decelerating to a stop when the deceleration stop command edge (write $FF00_H$) is turned on.

(2) Query format

Field	Number of data items (number of bytes)	RTU mode 8-bit data	Remarks
Start	None		Silent interval
Slave address [H]	1	Arbitrary	Axis number + 1 (01_H to 10_H) 00_H when broadcast is specified
Function code [H]	1	05	Write to a single coil DO.
Start address [H]	2	042C	Deceleration stop setting
Changed data [H]	2	Arbitrary	Deceleration stop command (ON): $FF00_H$ * The controller automatically resets the value to 0000_H.
Error check [H]	2	CRC (16 bits)	
End	None		Silent interval
Total number of bytes	8		

(3) Response

If the change is successful, the response message will be the same as the query.
If invalid data is sent, an exception response (refer to section 7) will be returned, or no response will be returned.

(4) Query sample

A sample query that decelerates to a stop of a controller of axis No. 0 is shown below.
Query (silent intervals are inserted before and after the query)
01 05 04 2C FF 00 4C C3

Field	RTU mode 8-bit data
Start	Silent interval
Slave address [H]	01
Function code [H]	05
Start address [H]	0427
Changed data [H]	FF00
Error check [H]	4CC3 (in accordance with CRC calculation)
End	Silent interval

If the change is successful, the response message will be the same as the query.

5.5 Direct Writing of Control Information (Queries Using Code 06)

5.5.1 Writing to Registers

(1) Function

These queries change (write) data in registers of a slave.
In case of broadcast, data of registers of the same address of all slaves is changed.
[Refer to the details of device controller register 1 in 4.3.2 (1).]
[Refer to the details of device controller register 2 in 4.3.2 (2).]
[Refer to the details of the position number specification register and position movement specification register in 4.3.2 (3).]

(2) Start address list

Address	Symbol	Name	Byte
0D00	DRG1	Device control register 1	2
0D01	DRG2	Device control register 2	2
0D03	POSR	Position number specification register	2
9800	POSR	Position movement specification register	2

The registers above are control command registers. The bits of these registers are assigned to input ports by PIO patterns when "PIO/Modbus Switch Status (PMSS) (refer to 4.3.2 (6)) is set to "disable Modbus commands (enable PIO commands). These registers can be rewritten when the Modbus commands are enabled (PIO commands are disabled).

(3) Query format

Specify the address and data of the register whose data is to be changed in the query message.
Data to be changed shall be specified as 16-bit data in the changed data area of the query.

Field	Number of data items (number of bytes)	RTU mode 8-bit data	Remarks
Start	None		Silent interval
Slave address [H]	1	Arbitrary	Axis number + 1 (01_H to 10_H) 00_H when broadcast is specified
Function code [H]	1	06	Writing to registers
Start address [H]	2	Arbitrary	Refer to 5.5.1 (2), "Start address list."
Changed data [H]	2	Arbitrary	4.3.2 (1) to 4.3.2. (3) Refer to List of changed data.
Error check [H]	2	In accordance with the calculation result	
End	None		Silent interval
Total number of bytes	10		

(4) Response

If the change is successful, the response message will be the same as the query.
If invalid data is sent, an exception response (refer to section 7) will be returned, or no response will be returned.

(5) Query sample

Examples of different operations are shown in [1] to [3] below.

[1] A sample query that turns the servo of a controller of axis No. 0 on and then executes home return operation is performed.

Query (silent intervals are inserted before and after the query)

First time 01 06 0D 00 10 00 86 A6 --- Servo ON

Second time 01 06 0D 00 10 10 87 6A --- Home return

Field	RTU mode 8-bit data
Start	Silent interval
Slave address [H]	01
Function code [H]	06
Start address [H]	0D00
Changed data [H]	First time: 1000 Second time: 1010 (Keep the servo ON bit 1 in cases other than when the servo is OFF.)
Error check [H]	First time: 86A6 (in accordance with CRC calculation) Second time: 876A (in accordance with CRC calculation)
End	Silent interval

*1 Home return is not performed even if 1010_H is sent to change the data while the servo is OFF (refer to the timing chart at startup of each RC controller).

*2 To keep the previous status, send the previous status even if there is no change. As in the example above, keep the servo ON bit as 1 at home return as well.

If the change is successful, the response message will be the same as the query.

[2] Move to position No. 1 using the position movement specification register (address 9800_H).

Before this operation, perform the operation in example [1] above to complete a home return.

Query (Silent intervals are inserted before and after the query.)

01 06 98 00 00 01 67 6A

Field	RTU mode 8-bit data
Start	Silent interval
Slave address [H]	01
Function code [H]	06
Start address [H]	9800
Changed data [H]	0001
Error check [H]	676A (in accordance with CRC calculation)
End	Silent interval

* As soon as a position number is written to this register, the actuator starts moving. The CSTR (start signal) is not required.

A response message to be sent following a successful change should be the same as the query.

[3] Move to position No. 1 using the position number specification register (address $0D03_H$).
Before this operation, perform the operation in example [1] above to complete a home return.
Query (Silent intervals are inserted before and after the query.)
First time 01 06 0D 03 00 01 BA A6 --- Specify position No. 1
Second time 01 06 0D 00 10 00 86 A6--- Turn OFF the CSTR (start signal)
Third time 01 06 0D 00 10 08 87 60--- Turn ON the CSTR (start signal)

Field	RTU mode 8-bit data
Start	Silent interval
Slave address [H]	01
Function code [H]	06
Start address [H]	First time: 0D03 Second time: 0D00 Third time: 0D00
Changed data [H]	First time: 0001 Second time: 1000 Third time: 1008
Error check [H]	First time: BAA6 (in accordance with CRC calculation) Second time: 86A6 (in accordance with CRC calculation) Third time: 8760 (in accordance with CRC calculation)
End	Silent interval

* To keep the previous status, send the previous status even if there is no change. As in the example above, keep the servo ON bit as 1 at other than servo OFF.

If the change is successful, the response message will be the same as the query.

5.6 Direct Writing of Positioning Data (Queries Using Code 10)

5.6.1 Numerical Value Movement Command

(1) Function

Specify the target position in PTP positioning operation using absolute coordinates. It is possible to command the actuator to move via numerical values by writing directly to the group of registers at addresses from 9900_H to 9908_H (can be set in one message).
Values of all registers, other than the control flag specification register (address: 9908_H), will become effective once the values are sent. If there is no need to change the target position, positioning band, speed, acceleration/deceleration, push-current limiting value and control specification, therefore, each subsequent numerical movement command can be issued simply by writing a desired register that can effect an actual movement command based on changing of the applicable register alone (refer to "Start address list").

(2) Start address list

This group of registers is used to move the actuator by specifying the target position coordinates, positioning band, speed acceleration/deceleration, push-operation current limit control specification flags and so on as numerical values.
Data of start addresses in the list (6 registers in total) can be changed with one transmission.

Address [H]	Symbol	Name	Sign	Able to effect an actual movement command by changing the applicable register alone	Register size	Byte size	Unit
9900	PCMD	Target position specification register	○	○	2	4	0.01 mm
9902	INP	Positioning band specification register		x	2	4	0.01 mm
9904	VCMD	Speed specification register		○	2	4	0.01 mm/sec
9906	ACMD	Acceleration/deceleration specification register		○	1	2	0.01 G
9907	PPOW	Push-current limiting value specification register		○	1	2	%
9908	CTLF	Control flag specification register		x Initialization after each movement	1	2	-

(3) Query format

1 register = 2 bytes = 16-bit data

Field	Number of data items (number of bytes)	RTU mode 8-bit data	Remarks
Start		None	Silent interval
Slave address [H]	1	Arbitrary	Axis number + 1 (01_H to 10_H) 00_H if broadcast is specified
Function code [H]	1	10	Numerical value specification
Start address [H]	2	Arbitrary	Refer to 5.6.1 (2), "Start address list"
Number of registers [H]	2	Arbitrary	Refer to 5.6.1 (2), "Start address list"
Number of bytes [H]	1	In accordance with the number of registers above	Enter the value twice as large as the number of registers specified above
Changed data 1 [H]	2		Refer to 5.6.1 (2), "Start address list "
Changed data 2 [H]	2		Refer to 5.6.1 (2), "Start address list"
Changed data 3 [H]	2		Refer to 5.6.1 (2), "Start address list"
:	:		:
Error check [H]	2	CRC (16 bits)	
End		None	Silent interval
Total number of bytes	Up to 256		

(4) Response format

When normally changed, the response message responds with a copy of the query message excluding the number of bytes and changed data.

Field	Number of data items (number of bytes)	RTU mode 8-bit data	Remarks
Start		None	Silent interval
Slave address [H]	1	Arbitrary	Axis number + 1 (01_H to 10_H) 00_H if broadcast is specified
Function code [H]	1	10	Numerical value specification
Start address [H]	2	Arbitrary	Refer to 5.6.1 (2), "Start address list"
Number of registers [H]	2	Arbitrary	Refer to 5.6.1 (2), "Start address list"
Error check [H]	2	CRC (16 bits)	
End		None	Silent interval
Total number of bytes	Up to 256	Up to 256	

(5) Detailed explanation of registers

- Target position specification register (PCMD)
  This register specifies the target position in PTP positioning operation using absolute coordinates. The value of this register is set in units of 0.01 mm in a range of –999999 to 999999 (FFF0BDC1(Note) ~000F423F$_H$). When a position is specified using absolute coordinates and the specified value exceeds the soft limit set by a parameter, an alarm will generate the moment a movement start command is issued. The actuator will start moving when the lower word of this register (symbol: PCMD, address: 9900$_H$) is rewritten. In other words, **a numerical movement command can be issued simply by writing a target position in this register.**
  (Note) To set a negative value, use a two's complement.

- Positioning band register (INP)
  This register is used in two different ways depending on the type of operation.
  The first way is the normal positioning operation, where it specifies the allowable difference between the target position and current position to be used in the detection of position complete. The second way is the push-motion operation, where it specifies the push-motion band. The value of this register is set in units of 0.01 mm in a range of 1 to 999999 (1~000F423F$_H$). Whether the normal operation or push-motion operation is specified by the applicable bit in the control flag specification register as explained later.
  Changing this register alone will not start actuator movement.

- Speed specification register (VCMD)
  This register specifies the moving speed. The value of this register is set in units of 0.01 mm/sec in a range of 1 to 999999 (1~000F423F$_H$). If the specified value exceeds the maximum speed set by a parameter, an alarm will generate the moment a movement start command is issued.
  The actuator will start moving when this lower word of this register is rewritten. In other words, the speed can be changed while the actuator is moving, simply by rewriting this register.

■ Acceleration/deceleration specification register (ACMD)
This register specifies the acceleration or deceleration. The value of this register is set in units of 0.01 G in a range of 1 to 300 (1~012C_H). If the specified value exceeds the maximum acceleration or deceleration set by a parameter, an alarm will generate the moment a movement start command is issued.
The actuator will start moving when this register is rewritten. In other words, the acceleration/deceleration can be changed while the actuator is moving, simply by rewriting this register.

■ Push-current limiting value (PPOW)
Set the current limit during push-motion operation in PPOW. Set an appropriate value by referring to the table below.

Actuator model name	Pushable range [%]	Settable range (input value) [H]
Actuator other than RCS2-RA13R	20~70(Note)	33~B2
RCS2-RA13R	20~200	33~1FE

(Note) The setting ranges may vary depending on the actuator.
[For details, refer to the IAI catalog or operation manual of each actuator.]

The actuator will start moving when this register is rewritten. In other words, the current limiting value can be changed during push-motion operation simply by rewriting this register.
Sample push-motion current setting
● When setting the current to 20%
255(100%) x 0.2 (20%)= 51 → 33_H (converted to a hexadecimal value)

■ Control Flag Specification Register (CTLF)
Set the method of operation.
If push-motion operation or incremental operation (pitch feed) is selected, set this register every time a movement command is issued. (This is because the register will be overwritten with the default value every time the actuator moves.)

CTLF bit structure

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
MSB	-	-	NTC1	NTC0	-	-	-	-	MOD1	MOD0	GSL1	GSL0	INC	DIR	PUSH	-	LSB

Bit 1 (PUSH) = 0: Normal operation (default)
1: Push-motion operation
Bit 2 (DIR) = 0: The direction of push-motion operation after completion of approach is defined as the forward direction (default).
1: The direction of push-motion operation after completion of approach is defined as the reverse direction.
This bit is used to calculate the direction of final stop position from PCMD. If this bit is set incorrectly, therefore, the target position will deviate from the specified position by a distance corresponding to "2 x INP, " as shown in Fig. 5.3 below.
If bit 1 is set to 0, the setting of this bit is invalid.

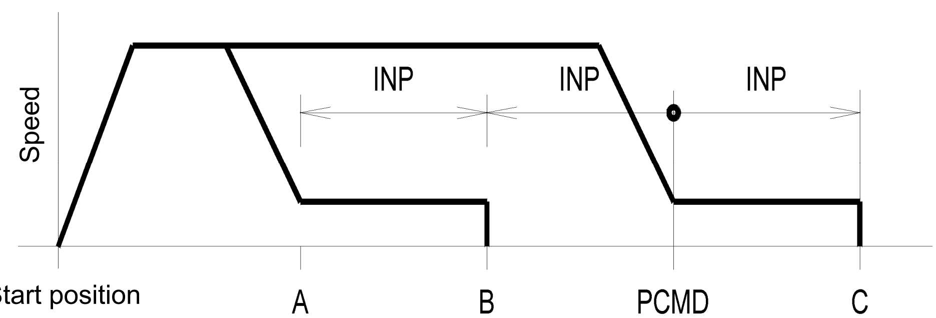


Fig. 5.3 Operating Direction in Push-motion Operation

Bit 3 (INC) = 0: Normal operation (default)
1: Incremental operation (pitch feed)

Setting this bit to 1 will enable the actuator to operate relative to the current position.
In this operation, the actuator behaves differently between normal operation and push-motion operation (CTLF bit 1). While the travel is calculated with respect to the target position (PCMD) in normal operation, it is calculated relative to the current position in push-motion operation (when bit 1 = 1).
Here, since relative coordinate calculation involves adding up pulses in mm, followed by conversion, unlike a calculation method involving addition after pulse conversion, **"repeated relative movements will not cause position deviation as a result of cumulative errors corresponding to fraction pulses that are not divisible with certain lead settings".**

Bit 4 (GSL0), 5 (GSL1) = Refer to the table below. (These bits can be set only on SCON-CA controllers.)

GSL1	GSL0	Function
0	0	Select parameter set 0 (default).
0	1	Select parameter set 1
1	0	Select parameter set 2
1	1	Select parameter set 3

You can register a maximum of four servo gain parameter sets consisting of six parameters and move the actuator to each position by selecting a different parameter set every time. [For details, refer to the operation manual for your controller.]

Bit 6 (MOD0), 7 (MOD1) = Refer to the table below. These bits cannot be set on PCON-* and ERC2 controllers.)

MOD1	MOD0	Function
0	0	Trapezoid pattern (default)
0	1	S-motion
1	0	Primary delay filter
1	1	Cannot be used.

These signals are used to select the acceleration/deceleration pattern characteristics. Set one of the patterns before issuing an actuator movement command. [For details, refer to the operation manual for your controller.]

Bit 12 (NTC0), 13 (NTC1) = Refer to the table below. (These bits can be set only on SCON-CA controllers.)

NTC1	NTC0	Function
0	0	Do not use vibration control (default).
0	1	Select parameter set 1
1	0	Select parameter set 2
1	1	Select parameter set 3

When vibration control is used, you can register a maximum of three parameter sets and move the actuator to each position by selecting a different parameter set every time. [For details, refer to the operation manual for your controller.]

(6) Example of use

Examples of different operations are shown in [1] to [7] below.

[1] Move by changing the target position. (All data other than the target position are the default values of their respective parameters.)

Conditions: The operation conditions conform to the default speed, default acceleration/deceleration and default positioning band set by the controller's user parameters. Only the target position is changed to move the actuator.

Supplement: Controller's user parameters

- Default speed (parameter No. 8) → Maximum speed of the applicable actuator as specified in the catalog
- Default acceleration/deceleration (parameter No. 9) → Rated acceleration of the applicable actuator as specified in the catalog
- Default positioning band (parameter No. 10) → Default value = 0.1 mm

Write the target position specification register (9900_H) (Example 1)

↓

Start of movement

(Example 1) Target position: 50 mm

Target position [mm]	Positioning band [mm]	Speed [mm/s]	Acceleration/ deceleration [G]	Push [%]	Control flag
50	Need not be set.				

■ Query :01 10 9900 0002 04 0000 1388 3655

■ Response :01 10 9900 0002 6F54

--- The query message is copied, except for the number of bytes and new data, and returned as a response.

■ Breakdown of Query Message

Field	RTU mode 8-bit data	Remarks
Start	None	Silent interval
Slave address	01_H	Axis number + 1
Function code	10_H	
Start address	9900_H	The starting address corresponds to the setting of target position specification register 9900_H.
Number of registers	0002_H	Addresses 9900_H to 9901_H are written.
Number of bytes	04_H	2 (registers) × 2 = 4 (bytes) → 4_H
New data 1, 2 (target position) Input unit (0.01 mm)	0000_H	All upper bits of the 32-bit data are 0.
	$07D0_H$	50 [mm] x 100 = 5000 → 1388_H
Error check	3655_H	CRC checksum calculation result → 3655_H
End	None	Silent interval
Total number of bytes	27	

[2] Move by changing the target position. (As well as data other than the target position).

Conditions: Want to move the actuator by changing the target position, speed and acceleration/deceleration every time.

Write the target position specification register (9900_H) through acceleration/deceleration specification register (9906_H)(Example 2)

↓

Start of movement

(Example 2) Target position: 50 mm

Target position [mm]	Positioning band [mm]	Speed [mm/s]	Acceleration/ deceleration [G]	Push [%]	Control flag
50	0.1	100	0.3	Need not be set.	

■ Query : 01 10 9900 0007 0E 0000 1388 0000 000A 0000 2710 001E 50CF
■ Response : 01 10 9900 0007 AF57
--- The query message is copied, except for the number of bytes and new data, and returned as a response.

■ Breakdown of Query Message

Field	RTU mode 8-bit data	Remarks
Start	None	Silent interval
Slave address	01_H	Axis number + 1
Function code	10_H	
Start address	9900_H	The starting address corresponds to the setting of target position specification register 9900_H.
Number of registers	0007_H	Addresses 9900_H to 9906_H are written.
Number of bytes	$0E_H$	7 (registers) × 2 = 14 (bytes) → E_H
New data 1, 2 (target position) Input unit (0.01 mm)	0000_H	All upper bits of the 32-bit data are 0.
	1388_H	50 [mm] × 100 = 5000 → 1388_H
New data 3, 4 (Positioning band) Input unit (0.01 mm)	0000_H	All upper bits of the 32-bit data are 0.
	$000A_H$	0.1 [mm] × 100 = 10 → $000A_H$
New data 5, 6 (Speed) Input unit (0.01 mm/sec)	0000_H	All upper bits of the 32-bit data are 0.
	2710_H	100 [mm/s] ×100 = 10000 → 2710_H
New data 7 (Acceleration/deceleration) Input unit (0.01 G)	$001E_H$	0.3 [G] × 100 = 30 → $001E_H$
Error check	$50CF_H$	CRC checksum calculation result → $50CF_H$
End	None	Silent interval
Total number of bytes	23	

[3] Change the speed while the actuator is moving.

Conditions: Change the target position, speed and acceleration/deceleration each time the actuator is moved, with the actuator speed changed at a given timing during movement.

Write the target position specification register (9900_H) through acceleration/deceleration specification register (9906_H)(Example 2)

↓

Start of movement

↓

Write the speed specification registers (9904_H and 9905_H)(Example 3)

↓

The actuator continues with the normal operation at the new speed

(Example 3) Change the speed from 100 mm/s to 50 mm/s while the actuator is moving.

Target position [mm]	Positioning band [mm]	Speed [mm/s]	Acceleration/ deceleration [G]	Push [%]	Control flag
50	0.1	100 → 50	0.3	Need not be set.	

(1) Start the movement at a speed of 100 mm/s. [Refer to Example [2], "Move by changing the speed" above.]
- Query : 01 10 9900 0007 0E 0000 1388 0000 000A 0000 2710 001E 50CF
- Response : 01 10 9900 0007 AF57

(2) Change the speed to 50 mm/s.
- Query : 01 10 9904 0002 04 0000 1388 395C
- Response : 01 10 9904 0002 2E95

--- The query message is copied, except for the number of bytes and new data, and returned as a response.

■ Breakdown of Query Message (Change the speed to 50 mm/s. [Refer to the above example for the query message used to start the movement at 100 mm/s.])

Field	RTU mode 8-bit data	Remarks
Start	None	Silent interval
Slave address	01_H	Axis number + 1
Function code	10_H	
Start address	9904_H	The starting address corresponds to the setting of target position specification register 9904_H.
Number of registers	0002_H	Addresses 9904_H to 9905_H are written.
Number of bytes	04_H	2 (registers) × 2 = 4 (bytes) → 4_H
New data 5, 6 (Speed) Input unit (0.01 mm/s)	0000_H	All upper bits of the 32-bit data are 0.
	1388_H	50 [mm/s] × 100 = 5000 → 1388_H
Error check	$395C_H$	CRC checksum calculation result → $395C_H$
End	None	Silent interval
Total number of bytes	13	

[4] Move in the incremental (pitch feed) mode.

Conditions: The operation conditions conform to the default speed, default acceleration/deceleration and default positioning band set by the controller's user parameters. Only the pitch width is changed to move the actuator.

Write the target position specification register (9900_H) through control flag specification register (9908_H: Incremental setting) (Example 4)

↓

Start of movement

Supplement: Addresses 9900_H and 9908_H alone cannot be changed in a single data transmission. Since all addresses are sequential, send two messages if 9900_H and 9908_H alone are changed. If you want to send only one message, write all addresses from 9900_H to 9908_H.

(Example 4) Move in the incremental mode by setting the pitch to 10 mm.

Pitch [mm]	Positioning band [mm]	Speed [mm/s]	Acceleration/ deceleration [G]	Push [%]	Control flag
10	0.1	100	0.3	0	Incremental (bit3 = 1)

- Query: 01 10 9900 0009 12 0000 03E8 0000 000A 0000 2710 001E 0000 0008 F3A0
- Response: 01 10 9900 0009 2E93

--- The query message is copied, except for the number of bytes and new data, and returned as a response.

■ Breakdown of Query Message

Field	RTU mode 8-bit data	Remarks
Start	None	Silent interval
Slave address	01_H	Axis No. 0 + 1
Function code	10_H	
Start address	9900_H	The start address is the target position specification register 9900_H.
Number of registers	0009_H	Specify 9900_H through 9908_H as the addresses to be written.
Number of bytes	12_H	9 (registers) x 2 = 18 (bytes) → 12_H
New data 1, 2 (target position) Input unit (0.01 mm)	0000_H	All upper bits of the 32-bit data are 0.
	$03E8_H$	10 [mm] × 100 = 1000 → $03E8_H$
New data 3, 4 (positioning band) Input unit (0.01 mm)	0000_H	All upper bits of the 32-bit data are 0.
	$000A_H$	0.1 [mm] × 100 = 10 → $000A_H$
New data 5, 6 (speed) Input unit (0.01 mm/sec)	0000_H	All upper bits of the 32-bit data are 0.
	2710_H	100 [mm/s] × 100 = 10000 → 2710_H
New data 7 (acceleration/deceleration) Input unit (0.01 G)	$001E_H$	0.3 [G] × 100 = 30 → $001E_H$
New data 8 (push) Input unit (%)	0000_H	0 [%] → 0_H
New data 9 (control flag)	0008_H	(Incremental setting) 1000b → 0008_H
Error check	$F3A0_H$	CRC check calculation result → $F3A0_H$
End	None	Silent interval
Total number of bytes	27	

[5] Change the speed during incremental movement (pitch feed).

Conditions: Change the target position, speed and acceleration/deceleration each time the actuator is moved, with the positioning band changed at a given timing during movement.

Write the target position specification register (9900_H) through control flag specification register (9908_H: Incremental setting) (Example 5)

↓

Start of incremental movement

↓

Write the speed specification register (9904_H) through control flag specification register (9908_H: Incremental setting) (Example 5).

↓

The actuator continues with the incremental movement at the new speed.

Supplement: After the control flag specification register (9908_H) is set, the register will return to the default value (0_H: Normal movement) once the actuator starts moving. Accordingly, you must set the control flag specification register (9908_H) and send it again if another incremental or push-motion operation is to be performed.

(Example 5) Change the speed from 100 mm/s to 50 mm/s while the actuator is moving.

Pitch [mm]	Positioning band [mm]	Speed [mm/s]	Acceleration/ deceleration [G]	Push [%]	Control flag
10	0.1	100 → 50	0.3	0	Incremental (bit3 = 1)

(1) Start moving at a speed of 100 mm/s. [Refer to Example [4], “Moving in the incremental (pitch feed) mode” above.]
- ■ Query : 01 10 9900 0009 12 0000 03E8 0000 000A 0000 2710 001E 0000 0008 F3A0
- ■ Response : 01 10 9900 0009 2E93

(2) Change the speed to 50 mm/s.
- ■ Query : 01 10 9904 0005 0A 0000 1388 001E 0000 0008 BD83
- ■ Response: 01 10 9904 0005 6F57
  --- The query message is copied, except for the number of bytes and new data, and returned as a response.

■ Breakdown of Query Message (Change the speed to 50 mm/s. [Refer to the above example for the query message used to start the movement at 100 mm/s.])

Field	RTU mode 8-bit data	Remarks
Start	None	Silent interval
Slave address	01_H	Axis No. 0 + 1
Function code	10_H	
Start address	9904_H	The start address is the target position specification register 9904_H.
Number of registers	0005_H	Specify 9904_H through 9908_H as the addresses to be written.
Number of bytes	$0A_H$	5 (registers) x 2 = 10 (bytes) → A_H
New data 5, 6 (speed) Input unit (0.01 mm/s)	0000_H	All upper bits of the 32-bit data are 0.
	1388_H	50 [mm/s] × 100 = 5000 → 1388_H
New data 7 (acceleration/deceleration) Input unit (0.01 G)	$001E_H$	0.3 [G] × 100 = 30 → $001E_H$
New data 8 (push) Input unit (%)	0000_H	0 [%]→ 0_H
New data 9 (control flag)	0008_H	(Incremental setting) 1000b→0008_H
Error check	$BD83_H$	CRC check calculation result → $BD83_H$
End	None	Silent interval
Total number of bytes	19	

[6] Perform a push-motion operation. (changing pushing force during push-operation)
Conditions: Perform push-motion operation by changing the push force at a desired timing while the actuator is pushing the work part.

Write the target position specification register (9900_H) through control flag specification register (9908_H: Push-motion setting) (Example 6)

↓

Start push-motion operation

↓

Write the push-current limit specification register (9907_H) through control flag specification register (9908_H: Push-motion setting) (Example 7)

↓

The actuator continues with the push-motion operation with the new push force

(Example 6) Perform a push-motion operation for 20 mm from the 50-mm position at a current-limiting value of 70%.

Target position [mm]	Positioning band [mm]	Speed [mm/s]	Acceleration/ deceleration [G]	Push [%]	Control flag
50	20	100	0.3	70	Push-motion operation (bit1 = 1, bit2 = 0, 1)

- Query: 01 10 9900 0009 12 0000 1388 0000 07D0 0000 2710 001E 00B2 0006 C377
- Response: 01 10 9900 0009 2E93
  --- The query message is copied, except for the number of bytes and new data, and returned as a response.

■ Breakdown of Query Message

Field	RTU mode 8-bit data	Remarks
Start	None	Silent interval
Slave address	01_H	Axis No. 0 + 1
Function code	10_H	
Start address	9900_H	The start address is the target position specification register 9900_H.
Number of registers	0009_H	Specify 9900_H through 9908_H as the addresses to be written.
Number of bytes	12_H	9 (registers) x 2 = 18 (bytes) → 12_H
New data 1, 2 (target position) Input unit (0.01 mm)	0000_H	All upper bits of the 32-bit data are 0.
	1388_H	50 [mm] × 100 = 5000 → 1388_H
New data 3, 4 (positioning band) Input unit (0.01 mm)	0000_H	All upper bits of the 32-bit data are 0.
	$07D0_H$	20 [mm] × 100 = 2000 → $07D0_H$
New data 5, 6 (speed) Input unit (0.01 mm/sec)	0000_H	All upper bits of the 32-bit data are 0.
	2710_H	100 [mm] × 100 = 10000 → 2710_H
New data 7 (acceleration/deceleration) Input unit (0.01 G)	$001E_H$	0.3 [G] × 100 = 30 → $001E_H$
New data 8 (push) Input unit (%)	$00B2_H$	70 [%] → $B2_H$
New data 9 (control flag)	0006_H	(Push setting) 0110b → 0006_H
Error check	$C377_H$	CRC check calculation result → $C377_H$
End	None	Silent interval
Total number of bytes	27	

(Example 7) Change the push current limit from 70% to 50% during a push-motion operation.

Target position [mm]	Positioning band [mm]	Speed [mm/s]	Acceleration/ deceleration [G]	Push [%]	Control flag
50	20	100	0.3	70 → 50	Push-motion operation (bit1 = 1, bit2 = 1)

■ Query : 01 10 9907 0002 04 007F 0006 C5C5
■ Response : 01 10 9907 0002 DE95
--- The query message is copied, except for the number of bytes and new data, and returned as a response.

■ Breakdown of Query Message

Field	RTU mode 8-bit data	Remarks
Start	None	Silent interval
Slave address	01_H	Axis No. 0 + 1
Function code	10_H	
Start address	9907_H	The start address is the target position specification register 9907_H.
Number of registers	0002_H	Specify 9907_H through 9908_H as the addresses to be written.
Number of bytes	04_H	2 (registers) x 2 = 4 (bytes) → 4_H
New data 8 (push) Input unit (%)	$007F_H$	50 [%]→$7F_H$
New data 9 (control flag)	0006_H	(Push setting) 0110b→0006_H
Error check	$C5C5_H$	CRC check calculation result → $C5C5_H$
End	None	Silent interval
Total number of bytes	12	

[7] Note (changing positioning band during movement)

The positioning band cannot be changed while the actuator is moving.

Conditions: Change the target position, speed and acceleration/deceleration each time the actuator is moved, with the positioning band changed at a given timing during movement.
(Cannot be changed. If data is written, the data is reflected in the next positioning.)

Write the target position specification register (9900_H) through acceleration/deceleration specification register (9906_H)

↓

Start normal operation

↓

Write the positioning band specification registers (9902_H and 9903_H)

↓

The actuator continues with the normal operation at the original positioning band setting

Supplement: Writing the positioning band specification registers alone cannot effect an actual movement command.
Therefore, the data changed by writing the positioning band specification registers (9902_H and 9903_H) will become effective when the next movement command is executed.

5.6.2 Writing Position Table Data

(1) Function

Position table data can be changed using this query.
Each time an address specified in “Start address list” (addresses +0000$_H$ through +000E$_H$) is accessed, the applicable data will be read from the EEPROM in units of position data. After having been written, the data will be stored in the EEPROM again.

* The EEPROM has a rewrite life of approx. 100, 000 times due to device limitations. If the position table data is written frequently, the EEPROM will reach its rewrite life quickly and a failure may occur. Accordingly, be careful not to let unexpected loops, etc., occur due to the logics on the host side.

(2) Start address list

In a query input, each address is calculated using the formula below:
1000$_H$ + (16 x Position number) $_H$ + Address (Offset) $_H$

Example Change the speed command register for position No. 200
1000$_H$ + (16 x 200 = 3200) $_H$ + 4$_H$
= 1000$_H$ + C80$_H$ + 4$_H$
= 1C84$_H$
“1C84” becomes the input value for the start address field of this query.

* The maximum position number varies depending on the controller model and the PIO pattern currently specified.

■ Position data change resisters

Address	Symbol	Name	Sign	Register size	Byte size	Input unit
+0000	PCMD	Target position	○	2	4	0.01 mm
+0002	INP	Positioning band		2	4	0.01 mm
+0004	VCMD	Speed command		2	4	0.01 mm/sec
+0006	ZNMP	Individual zone boundary +	○	2	4	0.01 mm
+0008	ZNLP	Individual zone boundary -	○	2	4	0.01 mm
+000A	ACMD	Acceleration command		1	2	0.01 G
+000B	DCMD	Deceleration command		1	2	0.01 G
+000C	PPOW	Push-current limiting value		1	2	%
+000D	LPOW	Load current threshold		1	2	%
+000E	CTLF	Control flag specification		1	2	

* Addresses starting with “+” indicate offsets.

(3) Query format

1 register = 2 bytes = 16-bit data

Field	Number of data items (Number of bytes)	RTU mode 8-bit data	Remarks
Start	None		Silent interval
Slave address [H]	1	Arbitrary	Axis number + 1 (01_H to 10_H) 00_H when broadcast is specified
Function code [H]	1	10	Numerical value command
Start address [H]	2	Arbitrary	Refer to 5.6.2 (2), "Start address list."
Number of registers [H]	2	Arbitrary	Refer to 5.6.2 (2), "Start address list."
Number of bytes [H]	1	In accordance with the above registers	A value corresponding to twice the number of registers specified above is input.
Changed data 1 [H]	2		Refer to "5.6.2 (2) Start address list."
Changed data 2 [H]	2		Refer to "5.6.2 (2) Start address list."
Changed data 3 [H]	2		Refer to "5.6.2 (2) Start address list."
:	:		:
Error check [H]		CRC (16 bits)	
End	None		Silent interval
Total number of bytes	Up to 256		

(4) Response format

If the change is successful, a response message that is effectively a copy of the query message, except for the byte count and new data, will be returned.

Field	Number of data items (Number of bytes)	RTU mode 8-bit data	Remarks
Start	None		Silent mode
Slave address [H]	1	Arbitrary	Axis number + 1 (01_H to 10_H) 00_H when broadcast is specified
Function code [H]	1	10	Numerical value command
Start address [H]	2	Arbitrary	Refer to 5.6.2 (2), "Start address list."
Number of registers [H]	2	Arbitrary	Refer to 5.6.2 (2), "Start address list."
Error check [H]	2	CRC (16 bits)	
End	None		Silent interval
Total number of bytes	Up to 256		

(5) Detailed explanation of registers

- Target Position (PCMD)
  This register specifies the target position using absolute coordinates or by an relative distance. The value of this register is set in units of 0.01 mm in a range of -999999 to 999999 (FFF0BDC1(Note)~000F423F$_H$). When a position is specified using absolute coordinates and the specified value exceeds the soft limit set by a parameter, an alarm will generate the moment a movement start command is issued. Specify whether to use absolute coordinates or relative distance using the applicable bit in the control flag specification register as explained later.
  (Note) To set a negative value, use a two's complement.

- Positioning band Specification Register (INP)
  This register is used in two different ways depending on the type of operation. The first way is the normal positioning operation, where it specifies the allowable difference between the target position and current position to be used in the detection of position complete. The second way is the push-motion operation, where it specifies the push-motion band. The value of this register is set in units of 0.01 mm in a range of 1 to 999999 (1~000F423F$_H$).
  Whether the normal operation or push-motion operation is specified by the applicable bit in the control flag specification register as explained later.

- Speed Specification Register (VCMD)
  This register specifies the moving speed in positioning. The value of this register is set in units of 0.01 mm/sec in a range of 1 to 999999 (1~000F423F$_H$). If the specified value exceeds the maximum speed set by a parameter, an alarm will generate the moment a movement start command is issued.

- Individual Zone Boundaries ± (ZNMP, ZNLP)
  These registers output zone signals that are effective only during positioning, separately from the zone boundaries set by parameters.
  Set in ZNMP the positive zone signal output boundary expressed using absolute coordinates, and set the negative zone signal output boundary in ZNLP. The corresponding bit in the zone register remains ON while the current position is within these positive and negative boundaries. The value of this register is set in units of 0.01 mm, and in a range of -999999 to 999999 (FFF0BDC1(Note ~000F423F$_H$) for both registers. However, ZNMP must be greater than ZNLP. Set the same value in both ZNMP and ZNLP to disable the individual zone output.
  (Note) To set a negative value, use a two's complement.

- Acceleration specification register (ACMD)
  This register specifies the acceleration during positioning. The value of this register is set in units of 0.01 G in a range of 1 to 300 (1~012C$_H$). If the specified value exceeds the maximum acceleration set by a parameter, an alarm will generate the moment a movement start command is issued.

- Deceleration specification register (DCMD)
  This register specifies the deceleration during positioning. The value of this register is set in units of 0.01 G in a range of 1 to 300 ($1 \sim 012C_H$). If the specified value exceeds the maximum deceleration set by a parameter, an alarm will generate the moment a movement start command is issued.

- Push-current limiting value (PPOW)
  Set the current limit during push-motion operation in PPOW. Set an appropriate value by referring to the table below.

Actuator model name	Pushable range [%]	Settable range (input value) [H]
Actuator other than RCS2-RA13R	20~70(Note)	33~B2
RCS2-RA13R	20~200	33~1FE

(Note) The setting ranges may vary depending on the actuator.
[For details, refer to the IAI catalog or operation manual of each actuator.]

Sample push-motion current setting
● When setting the current to 20%
255 (100%) x 0.2 (20%)= 51 → 33_H (converted to a hexadecimal value)

- Load Output Current Threshold (LPOW)
  To perform load output judgment, set the current threshold in LPOW. Set an appropriate value according to the actuator used, just like the push current limit (PPOW). If load output judgment is not performed, set 0.

- Control Flag Specification Register (CTLF)
  [Refer to the control flag specification register in 5.6.1 (5).]

(6) Sample query

A sample query that rewrites all data of position No. 12 of axis No. 0 is shown below.

Axis No. 0

Target position (mm)	Positioning band (mm)	Speed (mm/sec)	Individual zone boundary+ (mm)	Individual zone boundary- (mm)	Acceleration (G)	Deceleration (G)	Push (%)	Threshold	Movement control
100	0.1	200	60	40	0.01	0.3	0	0	Normal movement

■ Query (silent intervals are inserted before and after the query)
01 10 10 C0 00 0F 1E 00 00 27 10 00 00 00 0A 00 00 4E 20 00 00 17 70 00 00 0F A0 00 01
00 1E 00 00 00 00 00 00 70 1E

■ Received response 01 10 10 C0 00 0F 84 F1

--- The query message is copied, except for the number of bytes and new data, and returned as a response.

■ Breakdown of Query Message

Field	RTU mode 8-bit data	Remarks
Start	None	Silent interval
Slave address	01_H	Axis No. 0 + 1
Function code	10_H	
Start address	$10C0_H$	The start address is the target position specification register $10C0_H$ for position No. 12. *1
Number of registers	$000F_H$	Total 15 registers of register symbols PCMD to CTLF are specified to be written.
Number of bytes	$1E_H$	15 (registers) x 2 = 30 (bytes) → $1E_H$
New data 1, 2 (target position) Input unit (0.01 mm)	0000_H	All upper bits of the 32-bit data are 0.
	2710_H	100 (mm) x 100 = 10000 → 2710_H
New data 3, 4 (positioning band) Input unit (0.01 mm)	0000_H	All upper bits of the 32-bit data are 0.
	$000A_H$	0.1 (mm) x 100 = 10 → $000A_H$
New data 5, 6 (speed) Input unit (0.01 mm/sec)	0000_H	All upper bits of the 32-bit data are 0.
	$4E20_H$	200 (mm/sec) x 100 = 20000 → $4E20_H$
	$0FA0_H$	40 (mm) x 100 = 4000 → $0FA0_H$

Continue to the next page

Continued from the previous page

Field	RTU mode 8-bit data	Remarks
New data 7, 8 (individual zone boundary +) Input unit (0.01 mm)	0000_H	All upper bits of the 32-bit data are 0.
	1770_H	60 (mm) x 100 = 6000 → 1770_H
New data 9, 10 (individual zone boundary -) Input unit (0.01 mm)	0000_H	All upper bits of the 32-bit data are 0.
	$0FA0_H$	40 (mm) x 100 = 4000 → $0FA0_H$
New data 11 (acceleration) Input unit (0.01 G)	0001_H	0.01 (G) x 100 = 1 → 0001_H
New data 12 (deceleration) Input unit (0.01 G)	$001E_H$	0.3 (G) x 100 = 30 → $001E_H$
New data 13 (push) Input unit (%)	0000_H	0 (%) → 0_H
New data 14 (threshold) Input unit (%)	0000_H	0 (%) → 0_H
New data 15 (control flag)	0000_H	All bits are 0, because normal operation is specified. 0000_b → 0000_H
Error check	$701E_H$	CRC check calculation result → $701E_H$
End		Silent interval
Total number of bytes	39	

*1) Calculation of start address
In the example, all data of position No. 12 is changed. Accordingly, the target position address of position No. 12 is set in the start address field of this query.

1000_H + (16 x 12 =192) $_H$ + 0_H
= 1000_H + $C0_H$ + 0_H
= $10C0_H$

"10C0" becomes the input value for the start address field of this query.

Shown below are the screens of IAI's PC software for RC controllers, indicating how position data changes before and after a query message is sent.

■ Before a query is sent

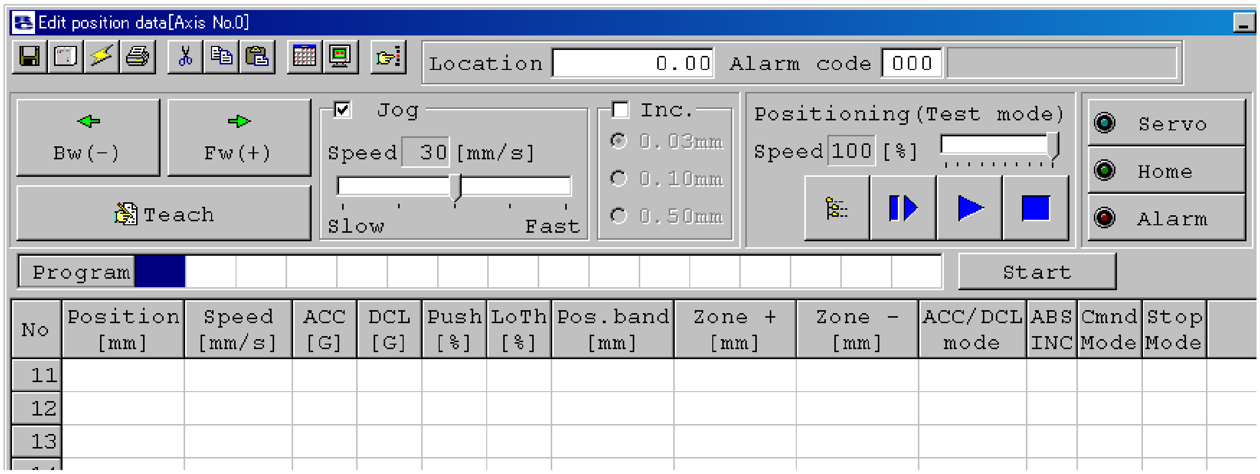


Fig. 5.4

■ After a query is sent

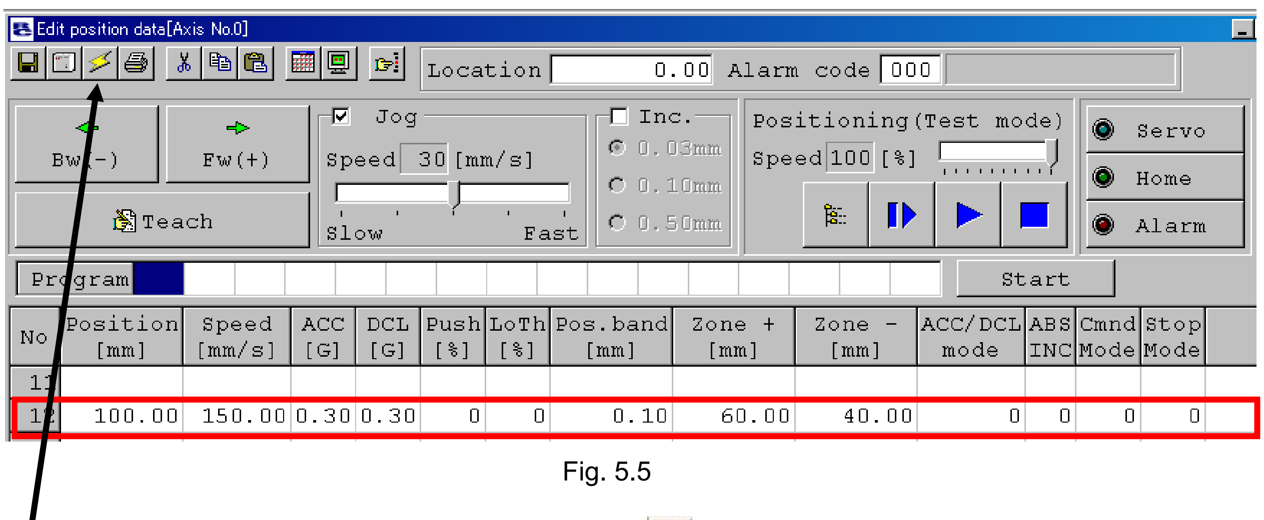


Fig. 5.5

* The overwritten data is not displayed until the button  is pressed or the Edit Position Data window is reopened.

6 Modbus ASCII

6.1 Message Frames (Query and Response)

Start	Address	Function code	Data	LRC Check	End
1 character	2 characters	2 characters	n characters	2 characters	2 characters
1 byte	2 bytes	2 bytes	nx2 bytes	2 bytes	2 bytes

* 1 character is expressed with 1 byte (2 characters) in ASCII code (refer to 6.2 ASCII "Code Table").

(1) Start
The Start field is equivalent to the header field and ":" (colon) is used in the ASCII mode.
It is expressed as $3A_H$ in ASCII code.

(2) Address
This field specifies the addresses of connected RC controllers (01_H to 10_H).
Set Address = axis number + 1
in ASCII code. Example) The axis number/is 30_H32_H.

Note: The address is not equal to the corresponding axis number: be careful when making settings.

(3) Function
The table below summarizes the function codes and functions that can be used with RC controllers.

Code		Name	Function
[Hex]	(ASCII)		
01_H	30_H31_H.	Read Coil Status	Read coils/DOs.
02_H	30_H32_H.	Read Input Status	Read input statuses/DIs.
03_H	30_H33_H.	Read Holding Registers	Read holding registers.
04_H	30_H34_H.	Read Input Registers	Read input registers.
05_H	30_H35_H.	Force Single Coil	Write one coil/DO.
06_H	30_H36_H.	Preset Single Register	Write holding register.
07_H	30_H37_H.	Read Exception Status	Read exception statuses.
$0F_H$	30_H46_H.	Force Multiple Coils	Write multiple coils/DOs at once.
10_H	31_H30_H.	Preset Multiple Registers	Write multiple holding registers at once.
11_H	31_H31_H.	Report Slave ID	Query a slave's ID.
17_H	31_H37_H.	Read / Write Registers	Read/write registers.

* This manual uses ☐ mark function codes.
* The ROBONET gateway supports three types of function codes (03_H, 06_H and 10_H).
(Refer to the separate ROBONET Instruction Manual.)
The ROBONET gateway does not support the ASCII mode.

(4) Data
Use this field to add data specified by a function code. It is also allowed to omit data if data addition is not specified by function codes.

(5) LRC Check
In the ASCII mode, an error check field conforming to the LRC method is automatically (*) included in order to check the message content excluding the first colon and CR/LF. Moreover, checking is carried out regardless of the parity check method of individual characters in messages.
The LRC field consists of two ASCII code characters. The LRC value is calculated by the sender that appends the LRC field to the message. The recipient recalculates the LRC value while receiving the message, and compares the calculation result against the actual value received in the LRC field. If the two values do not match, an error will generate.
* The host side must create a function that calculates the LRC value.

● <LRC check calculation example> area is the target range of error check

In case the message query is as follows: [':'] ["01] ["05"] ["040B"] ["0000"] [LRC] [CR] [LF]

[1] First, add all numerical values in units of bytes.
Total value added = $01_H + 05_H + 04_H + 0B_H + 00_H + 00_H = 15_H$

[2] Next, an 8-bit-based 2's complement of this value is computed, yielding the value $FFFFFFEB_H$. The LRC value is obtained by extracting the least significant byte. Thus the LRC value is "EB."

(6) End
This is equivalent to the trailer, and use "CR/LF" in the ASCII mode. In ASCII code, 00_H and $0A_H$ are displayed.

(7) Broadcast
It is possible to send a query containing same data to all connected axes by specifying the address 00_H. In this case, no response is returned from the RC controllers.
Note, however, that the function codes etc. that can be used with this function are limited; care should be taken when using the function. Please check the function codes that can be used in 6.3, "List of ASCII Mode Queries."

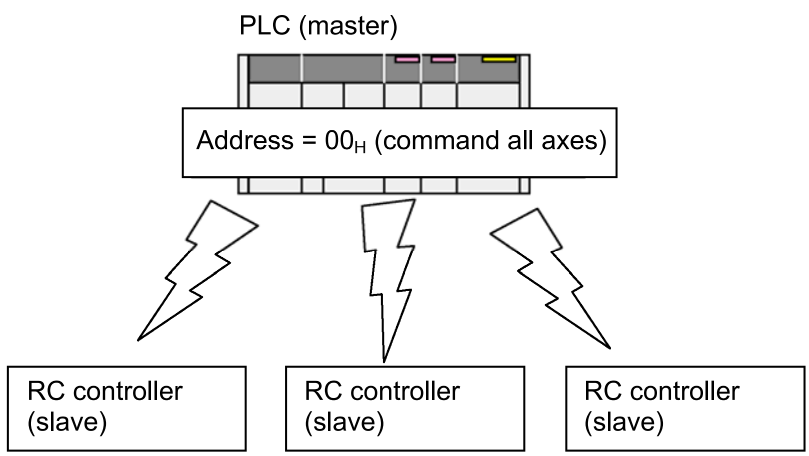


Fig.6.1

⚠ Caution

- The sizes of send/receive buffers are set to 256 bytes for the RC controllers, respectively. Make sure to keep the messages small enough such that messages sent from the host side do not exceed the receive buffer and data requests do not exceed send buffer.

- If the number of data items results in an odd number of bytes, caution must be taken for the reasons below.
The data is communicated on a byte-by-byte basis in Modbus communication.
In many cases, however, the data is treated in units of 2 bytes on the master side. If the number of data items becomes odd, 00_H (i.e., NULL) may be added automatically at the end of a packet in some cases.
RC controllers are configured such that the Modbus RTU is basically used as the interface on the master side. Since the controller normally stands by for reception in the RTU mode, and then makes judgment whether the code is ASCII or not after the reception, it cannot manage header/delimiter fields. For this reason, communication in the ASCII mode is disabled in such cases.
Example: In case of querying output ports of axis No. 0

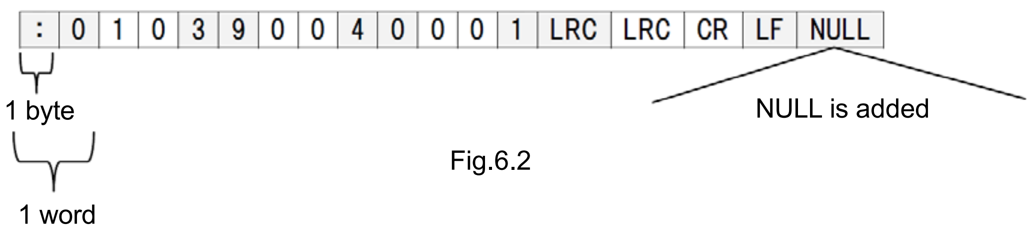


Fig.6.2

6.2 ASCII Code Table

ASCII Code (numbers and characters enclosed with ☐ are converted and sent.)

Most significant 3bit / Least significant 4bit	0	1	2	3	4	5	6	7
0	NUL	DLE	SP	0	@	P		p
1	SOH	DC1	!	1	A	Q	a	q
2	STX	DC2	“	2	B	R	b	r
3	ETX	DC3	#	3	C	S	c	s
4	EOT	DC4	$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	‘	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[	k	{
C	FF	IS4	,	<	L	¥	l	|
D	CR	IS4	—	=	M	]	m	}
E	SO	IS4	.	>	N	^	n	
F	SI	IS4	/	?	O	_	o	DEL

- NUL: Null character
- ETX: End of text
- ACK: Acknowledgment
- HT: Horizontal tab
- FF: Form feed
- SI: Shift in
- NAC: Negative acknowledgment
- CAN: Cancel
- ESC: Escape
- SOH: Start of header
- EOT: End of transmission
- BEL: Bell
- LF: Line feed
- CR: Carriage return
- DLE: Data link escape
- SYN: Synchronized characters
- EM: End of media
- SP: Space
- STX: Start of text
- ENQ: Enquiry
- BS: Backspace
- VT: Vertical tab
- SO: Shift out
- DC*: Device control *
- ETB: End of transmission block
- DEL: Delete

Example: “1” is 31_H in ASCII code and “00110001” in binary number presentation.

6.3 List of ASCII Mode Queries

FC	Function	Symbol	Function	Combination with PIO	Broad-cast	Page
03	Multiple FC03 register read	None	This function can be used to successively read multiple registers that use function 03.	○		162
03	Current position read	PNOW	This function reads the current actuator position in units of 0.01 mm.	○		165
03	Present alarm code read	ALMC	This function reads alarm codes that are presently detected.	○		167
03	I/O port input status read	DIPM	This function reads the ON/OFF statuses of PIO input ports.	○		169
03	I/O port output status read	DOPM	This function reads the ON/OFF statuses of PIO output ports.	○		173
03	Controller status signal read 1 (device status 1) (Operation preparation status)	DSS1	This function reads the following 12 statuses: [1] Emergency stop [2] Safety speed enabled/disabled [3] Controller ready [4] Servo ON/OFF [5] Missed work part in push-motion operation [6] Major failure [7] Minor failure [8] Absolute error [9] Brake [10] Pause [11] Home return completion [12] Position complete [13] Load cell calibration complete [14] Load cell calibration status	○		177
03	Controller status signal read 2 (device status 2) (Operation preparation 1 status)	DSS2	This function reads the following 15 statuses: [1] Enable [2] Load output judgment (check-range load current threshold) [3] Torque level (load current threshold) [4] Teaching mode (normal/teaching) [5] Position data load (normal/complete) [6] Jog+ (normal/command active) [7] Jog- (normal/command active) [8] Position complete 7 [9] Position complete 6 [10] Position complete 5 [11] Position complete 4 [12] Position complete 3 [13] Position complete 2 [14] Position complete 1 [15] Position complete 0	○		179

FC: Function code
PIO: Parallel I/O (input/output of an I/O connector)
The circle marks in the Simultaneous use with PIO and Broadcast columns indicate queries that can be used simultaneously with PIO and in broadcast communication, respectively.

FC	Function	Symbol	Function	Combination with PIO	Broad-cast	Page
03	Controller status signal read 3 (extended device status) (Operation preparation 2 status)	DSSE	This function reads the following 9 statuses: [1] Emergency stop (emergency stop input port) [2] Motor voltage low [3] Operation mode (AUTO/MANU) [4] Home return [5] Push-motion operation in progress [6] Excitation detection [7] PIO/Modbus switching [8] Position-data write completion status [9] Moving	○		181
03	Controller status signal read 4 (System status) (Controller status)	STAT	This function reads the following 7 statuses: [1] Automatic servo OFF [2] EEPROM accessed [3] Operation mode (AUTO/MANU) [4] Home return completion [5] Servo ON/OFF [6] Servo command [7] Drive source ON (normal/cut off)	○		183
03	Current speed read	VNOW	This function reads the current actuator speed in units of 0.01 mm/sec.	○		185
03	Current read	CNOW	This function reads the motor-torque current command value of the actuator in mA.	○		187
03	Deviation read	DEVI	This function reads the deviation over a 1-ms period in pulses.	○		189
03	Total power on time read	STIM	This function reads the total time in msec since the controller power was turned on.	○		191
03	Special input port input signal status read (Sensor input status)	SIPM	This function reads the following 9 statuses: [1] Command pulse NP [2] Command pulse PP [3] Mode switch [4] Belt breakage sensor [5] Home check sensor [6] Overtravel sensor [7] Creep sensor [8] Limit sensor	○		193

FC	Function	Symbol	Function	Combination with PIO	Broad-cast	Page
03	Zone status query	ZONS	This function reads the following 6 statuses: [1] LS2 (PIO pattern solenoid valve mode [3-point type] [2] LS1 (PIO pattern solenoid valve mode [3-point type] [3] LS0 (PIO pattern solenoid valve mode [3-point type] [4] Position zone [5] Zone 2 [6] Zone 1	○		195
03	Position complete number query	POSS	This function reads the following 9 statuses: [1] Position complete number bit 256 [2] Position complete number bit 128 [3] Position complete number bit 64 [4] Position complete number bit 32 [5] Position complete number bit 16 [6] Position complete number bit 8 [7] Position complete number bit 4 [8] Position complete number bit 2 [9] Position complete number bit 1	○		197
03	Force feedback data write	FBFC	The current measurement on the load cell is read in units of 0.01 N.	○		199
05	Safety speed mode switching	SFTY	This function issues a command to enable/disable the safety speed.		○	202
05	Servo ON/OFF	SON	This function issues a command to turn the servo ON/OFF.		○	204
05	Alarm reset	ALRS	This function issues a command to reset alarms/cancel the remaining travel.		○	206
05	Brake release	BKRL	This function issues a command to forcibly release the brake.		○	208
05	Pause	STP	This function issues a pause command.		○	210
05	Home return	HOME	This function issues a home return operation command.		○	212
05	Teaching mode command	CSTR	This function turns the start signal ON/OFF for movement by position number specification. (This function is not used in this manual.)		○	214
05	Jog/inch switching	JISL	This function switches between the jogging mode and the inching mode		○	216
05	Teaching mode command	MOD	This function switches between the normal mode and the teaching mode		○	218
05	Position data load	TEAC	This function issues a current position load command in the teaching mode.		○	220
05	Jog+	JOG+	This function issues a jogging/inching command in the direction opposite home.		○	222
05	Jog-	JOG-	This function issues a jogging/inching command in the direction of home.		○	224

FC	Function	Symbol	Function	Combination with PIO	Broad-cast	Page
05	Position number specification 0 to 7	ST0 to ST7	This function specifies position numbers effective only in the solenoid valve mode. The actuator can be operated with this command alone. *		○	226
05	Load cell calibration command	CLBR	Calibrate the load cell.		○	228
05	PIO/Modbus switching setting	PMSL	This function issues a command to enable/disable PIO external command signals.		○	230
05	Deceleration stop	STOP	This function can decelerate the actuator to a stop.		○	232
06					○	234
10	Numerical movement command	None	This function can be used to send the target position, positioning band, speed, acceleration/deceleration, push, and control setting in a single message to operate the actuator. Normal movement, relative movement and push-motion operation are supported.		○	238
10	Position data table write	None	This function can be used to change all data of the specified position number for the specified axis.		○	256
Indeter-minable	Exception response	None	This response will be returned when the message contains invalid data.			264

6.4. Data and Status Reading (queries Using Code 03)

6.4.1 Reading Consecutive Multiple Registers

*) Please refer to "6.2 ASCII Code Table."

(1) Function

These registers read the contents of registers in a slave.

This function is not supported in broadcast communication.

(2) Start address list

With RC Series controllers, the sizes of send/receive buffers are set to 256 bytes, respectively. Accordingly, a maximum of 123 registers' worth of data consisting of 247 bytes (one register uses two bytes), which is 9 bytes (header + slave address + function code + error check + trailer) of 256 bytes, can be queried in the ASCII mode. In other words, all of the data listed below (total 21 registers) can be queried in a single communication.

Address [H]	Symbol	Name	Sign	Register size	Byte
9000	PNOW	Current position monitor	○	2	4
9002	ALMC	Present alarm code query		1	2
9003	DIPM	Input port query		1	2
9004	DOPM	Output port monitor query		1	2
9005	DSS1	Device status query 1		1	2
9006	DSS2	Device status query 2		1	2
9007	DSSE	Expansion device status query		1	2
9008	STAT	System status query		2	4
900A	VNOW	Current speed monitor	○	2	4
900C	CNOW	Current ampere monitor	○	2	4
900E	DEVI	Deviation monitor	○	2	4
9010	STIM	System timer query		2	4
9012	SIPM	Special input port query		1	2
9013	ZONS	Zone status query		1	2
9014	POSS	Position complete number status query		1	2
901E	FBFC	Force feedback data monitor	○	2	4

(3) Query format

In a query message, specify the address of the register from which to start reading data, and number of bytes in registers to be read.
1 register = 2 bytes = 16-bit data

Field	Number of characters (Number of bytes)	ASCII mode fixed character string	Remarks
Header	1	':'	
Slave address [H]	2	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	2	'0', '3'	Register reading code
Start address [H]	4	Arbitrary	Refer to 6.4.1 (2), "Start address list."
Number of registers [H]	4	Arbitrary	Refer to "Start address list."
Error check [H]	2	LRC calculation result	
Trailer	2	CR/LF	
Total number of bytes	17	17	

(4) Response format

Field	Number of characters (Number of bytes)	ASCII mode fixed character string	Remarks
Header	1	':'	
Slave address [H]	2	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	2	'0', '3'	"Read Holding Registers" code
Number of data bytes [H]	2		Number of specified registers in a query format x 2
Data 1 [H]	4		
Data 2 [H]	4		
Data 3 [H]	4		
Data 4 [H]	4		
:	:		
:	:		
Error check [H]	2	LRC calculation result	
Trailer		CR/LF	
Total number of bytes			

(5) Sample query

A sample query that reads addresses 9000_H to 9009_H in a RC controller of axis No. 0 is shown below:

Query: 01039000000A62[CR][LF]

Field	ASCII mode fixed character string	Converted ASCII code data [H]
Header	‘:’	3A
Slave address [H]	‘0’, ‘1’	3031
Function code [H]	‘0’, ‘3’	3033
Start address [H]	‘9’, ‘0’, ‘0’, ‘0’	39303030
Number of registers [H]	‘0’, ‘0’, ‘0’, ‘A’	30303041
Error check [H]	‘6’, ‘2’	3632
Trailer	‘CR’, ‘LF’	0D0A

The response to the query is as follows.

Response: 0103140000000000000B80162002000800031C7000800111C[CR][LF]

Field	ASCII mode fixed character string	Converted ASCII code data [H]
Header	‘:’	3A
Slave address [H]	‘0’, ‘1’	3031
Function code [H]	‘0’, ‘3’	3033
Number of data bytes [H]	‘1’, ‘4’ (20 bytes = 10 registers)	3134
Data 1 [H]	‘0’, ‘0’, ‘0’, ‘0’, ‘0’, ‘0’, ‘0’, ‘0’	3030303030303030
Data 2 [H]	‘0’, ‘0’, ‘0’, ‘0’	30303030
Data 3 [H]	‘B’, ‘8’, ‘0’, ‘1’	42383031
Data 4 [H]	‘6’, ‘2’, ‘0’, ‘0’	36323030
Data 5 [H]	‘2’, ‘0’, ‘0’, ‘0’	32303030
Data 6 [H]	‘8’, ‘0’, ‘0’, ‘0’	38303030
Data 7 [H]	‘3’, ‘1’, ‘C’, ‘7’	33314337
Data 8 [H]	‘0’, ‘0’, ‘0’, ‘8’, ‘0’, ‘0’, ‘1’, ‘1’	3030303830303131
Error check [H]	‘1’, ‘C’ (in accordance with LRC calculation)	3143
Trailer	‘CR’, ‘LF’	0D0A

* **The data of the response example is simply an example and will vary depending on various conditions.**

6.4.2 Current Position Reading (in 0.01 mm units) Monitor <<PNOW>>

(1) Function

This query reads the current in units of 0.01 mm. The sign is effective.

(2) Query format

Field	Number of characters	ASCII mode character string (fixed)	Remarks
Header	1	':'	
Slave address [H]	2	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	2	'0', '3'	Reading registers
Start address [H]	4	'9', '0', '0', '0'	Current position monitor
Number of registers [H]	4	'0', '0', '0', '2'	Reading addresses 9000_H to 9001_H
Error check [H]	2	LRC calculation result	
Trailer	2	'CR', 'LF'	
Total number of bytes	17		

(3) Response format

A response message contains 16 bits of data per register.

Field	Number of characters	ASCII mode character string (fixed)	Remarks
Header	1	':'	
Slave address [H]	2	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	2	'0', '3'	Reading registers
Number of data bytes [H]	2	'0', '4'	Reading two registers = 4 bytes
Data 1 [H]	4	In accordance with the current value	Current value data [Hex]
Data 2 [H]	4	In accordance with the current value	Current value data [Hex]
Error check [H]	2	LRC calculation result	
Trailer	2	'CR', 'LF'	
Total number of bytes	19		

(4) Sample query (Axis No. 0)

A sample query that reads address 9000_H in a controller of axis No. 0 is shown below:
Query: 0103900000026A[CR][LF]

Field	ASCII mode fixed character string	Converted ASCII code data [H]
Header	‘:’	3A
Slave address	‘0’, ‘1’	3031
Function code	‘0’, ‘3’	3033
Start address	‘9’, ‘0’, ‘0’, ‘0’	39303030
Number of registers	‘0’, ‘0’, ‘0’, ‘2’	30303032
Error check	‘6’, ‘A’	3641
Trailer	‘CR’, ‘LF’	0D0A

The response to the query is as follows.
Response: 010304000013885D[CR][LF]

Field	ASCII mode fixed character string	Converted ASCII code data [H]
Header	‘:’	3A
Slave address [H]	‘0’, ‘1’	3031
Function code [H]	‘0’, ‘3’	3033
Number of data bytes [H]	‘0’, ‘4’ (4 bytes = 2 registers)	3034
Data 1 [H]	‘0’, ‘0’, ‘0’, ‘0’	30303030
Data 2 [H]	‘1’, ‘3’, ‘8’, ‘8’	31333838
Error check [H]	‘5’, ‘D’ (in accordance with LRC calculation)	3544
Trailer	‘CR’, ‘LF’	0D0A

The current position is “00001388” → Converted to a decimal value → 5000 (x 0.01 mm)
The current position is 50 mm.

*** The data of the response example is simply an example and will vary depending on various conditions.**

6.4.3 Present Alarm Code Query <<ALMC>>

(1) Function

Whether the controller is normal or any alarm presently detected is indicated by a code.
If no alarm is present, 00_H is stored.
[For details on alarm codes, refer to the operation manual that comes with each controller.]

(2) Query format

Field	Number of characters	ASCII mode character string (fixed)	Remarks
Header	1	‘:’	
Slave address [H]	2	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	2	‘0’, ‘3’	Reading registers
Start address [H]	4	‘9’, ‘0’, ‘0’, ‘2’	Present alarm codes
Number of registers [H]	4	‘0’, ‘0’, ‘0’, ‘1’	Reading address 9002_H
Error check [H]	2	LRC calculation result	
Trailer	2	‘CR’, ‘LF’	
Total number of bytes	17		

(3) Response format

A response message contains 16 bits of data per register.

Field	Number of characters	ASCII mode character string (fixed)	Remarks
Header	1	‘:’	
Slave address [H]	2	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	2	‘0’, ‘3’	Reading registers
Number of data bytes [H]	2	‘0’, ‘2’	Reading one register (2 bytes)
Data 1 [H]	4	Alarm code	Alarm code [Hex]
Error check [H]	2	LRC calculation result	
Trailer	2	‘CR’, ‘LF’	
Total number of bytes	15		

(4) Sample query (Axis No. 0)

A sample query that reads address 9002_H in an RC controller of axis No. 0 is shown below:
Query: 01039002000169[CR][LF]

Field	ASCII mode fixed character string	Converted ASCII code data [H]
Header	‘:’	3A
Slave address	‘0’, ‘1’	3031
Function code	‘0’, ‘3’	3033
Start address	‘9’, ‘0’, ‘0’, ‘2’	39303032
Number of registers	‘0’, ‘0’, ‘0’, ‘1’	30303031
Error check	‘6’, ‘9’	3639
Trailer	‘CR’, ‘LF’	0D0A

The response to the query is as follows.
Response: 01030200E812[CR][LF]

Field	ASCII mode fixed character string	Converted ASCII code data [H]
Header	‘:’	3A
Slave address [H]	‘0’, ‘1’	3031
Function code [H]	‘0’, ‘3’	3033
Number of data bytes [H]	‘0’, ‘2’ (2 bytes = 1 register)	3032
Data 1 [H]	‘0’, ‘0’, ‘E’, ‘8’	30304538
Error check [H]	‘1’, ‘2’ (in accordance with LRC calculation)	3132
Trailer	‘CR’, ‘LF’	0D0A

The most important alarm presently detected is “0E8”$_H$, which is a phase A/B open alarm.
[For details on alarm codes, refer to the operation manual that comes with each controller.]

* **The data of the response example is simply an example and will vary depending on various conditions.**

6.4.4 I/O Port Input Signal Status Reading <<DIPM>>

(1) Function

Port input values of the RC controller are read directly regardless of the PIO pattern. Note that the values are the states of ports recognized by the RC controller as inputs.

(2) Query format

Field	Number of characters	ASCII mode character string (fixed)	Remarks
Header	1	‘:’	
Slave address [H]	2	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	2	‘0’, ‘3’	Reading registers
Start address [H]	4	‘9’, ‘0’, ‘0’, ‘3’	Input port monitor register
Number of registers [H]	4	‘0’, ‘0’, ‘0’, ‘1’	Reading address 9003_H
Error check [H]	2	LRC calculation result	
Trailer	2	‘CR’, ‘LF’	
Total number of bytes	17		

(3) Response format

A response message contains 16 bits of data per address.

Field	Number of characters	ASCII mode character string (fixed)	Remarks
Header	1	‘:’	
Slave address [H]	2	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	2	‘0’, ‘3’	Reading registers
Number of data bytes [H]	2	‘0’, ‘2’	Reading one register (2 bytes).
Data 1 [H]	4	DI input value	DI input value [Hex]
Error check [H]	2	LRC calculation result	
Trailer	2	‘CR’, ‘LF’	
Total number of bytes	15		

(4) Sample query (Axis No. 0)

A sample query that reads input ports (address 9003_H) in a controller of axis No. 0 is shown below.

Query: 01 03 90 03 00 01 68 [CR] [LF]

Field	ASCII mode fixed character string	Converted ASCII code data [H]
Start	‘:’	3A
Slave address [H]	‘0’, ‘1’	3031
Function code [H]	‘0’, ‘3’	3033
Start address [H]	‘9’, ‘0’, ‘0’, ‘3’	39303033
Number of registers [H]	‘0’, ‘0’, ‘0’, ‘1’	30303031
Error check [H]	‘6’, ‘8’ (In accordance with LRC calculation)’	3638
End	‘CR’, ‘LF’	0D0A

The response to the query is as follows.

Response: 01 03 02 B8 01 14 [CR] [LF]

Field	ASCII mode fixed character string	Converted ASCII code data [H]
Start	‘:’	3A
Slave address [H]	‘0’, ‘1’	3031
Function code [H]	‘0’, ‘3’	3033
Number of data bytes [H]	‘0’, ‘2’ (2 bytes = 1 register)	3032
Data 1 [H]	‘B’, ‘8’, ‘0’, ‘1’	42383031
Error check [H]	‘1’, ‘4’ (in accordance with LRC calculation)	3134
End	‘CR’, ‘LF’	0D0A

The input port data area is “B801”$_H$ → Converted to the binary value “1011100000000001”

*** The data of the response example is simply an example and will vary depending on various conditions.**

(5) Port assignment [For details, refer to the operation manual that comes with each RC controller.]
Write the port assignment of PIO patterns to each RC controller.
0 indicates that response data is always 0.

Port	ERC2 (PIO type)			
	PIO pattern 0	PIO pattern 1	PIO pattern 2	PIO pattern 3
IN0	PC1	ST0	PC1	PC1
IN1	PC2	ST1	PC2	PC2
IN2	PC4	ST2	PC4	PC4
IN3	HOME	0	PC8	PC8
IN4	CSTR	RES	CSTR	CSTR
IN5	*STP	*STP	*STP	*STP
IN6	0	0	0	0
IN7	0	0	0	0
IN8	0	0	0	0
IN9	0	0	0	0
IN10	0	0	0	0
IN11	0	0	0	0
IN12	0	0	0	0
IN13	0	0	0	0
IN14	0	0	0	0
IN15	0	0	0	0

Port	PCON-C/CG						PCON-PL/PO	
	PIO pattern 0	PIO pattern 1	PIO pattern 2	PIO pattern 3	PIO pattern 4	PIO pattern 5	PIO pattern 6	PIO pattern 7
IN0	PC1	PC1	PC1	PC1	ST0	ST0	SON	SON
IN1	PC2	PC2	PC2	PC2	ST1	ST1	TL	TL
IN2	PC4	PC4	PC4	PC4	ST2	ST2	HOME	HOME
IN3	PC8	PC8	PC8	PC8	ST3	0	RES	RES/ DCLR
IN4	PC16	PC16	PC16	PC16	ST4	0	0	0
IN5	PC32	PC32	PC32	PC32	ST5	0	0	0
IN6	0	MODE	PC64	PC64	ST6	0	0	0
IN7	0	JISL	PC128	PC128	0	0	0	0
IN8	0	JOG+	0	PC256	0	0	0	0
IN9	BKRL	JOG−	BKRL	BKRL	BKRL	BKRL	0	0
IN10	RMOD	RMOD	RMOD	RMOD	RMOD	RMOD	0	0
IN11	HOME	HOME	HOME	HOME	HOME	0	0	0
IN12	*STP	*STP	*STP	*STP	*STP	0	0	0
IN13	CSTR	CSTR/ PWRT	CSTR	CSTR	0	0	0	0
IN14	RES	RES	RES	RES	RES	RES	0	0
IN15	SON	SON	SON	SON	SON	SON	0	0

Port	ACON-C/CG PIO pattern 0	PIO pattern 1	PIO pattern 2	PIO pattern 3	PIO pattern 4	PIO pattern 5	ACON-PL/PO PIO pattern 0	PIO pattern 1
IN0	PC1	PC1	PC1	PC1	ST0	ST0	SON	SON
IN1	PC2	PC2	PC2	PC2	ST1	ST1	TL	TL
IN2	PC4	PC4	PC4	PC4	ST2	ST2	HOME	HOME
IN3	PC8	PC8	PC8	PC8	ST3	0	RES	RES/ DCLR
IN4	PC16	PC16	PC16	PC16	ST4	0	0	0
IN5	PC32	PC32	PC32	PC32	ST5	0	0	0
IN6	0	MODE	PC64	PC64	ST6	0	0	0
IN7	0	JISL	PC128	PC128	0	0	0	0
IN8	0	JOG+	0	PC256	0	0	0	0
IN9	BKRL	JOG−	BKRL	BKRL	BKRL	BKRL	0	0
IN10	RMOD	RMOD	RMOD	RMOD	RMOD	RMOD	0	0
IN11	HOME	HOME	HOME	HOME	HOME	0	0	0
IN12	*STP	*STP	*STP	*STP	*STP	0	0	0
IN13	CSTR	CSTR/ PWRT	CSTR	CSTR	0	0	0	0
IN14	RES	RES	RES	RES	RES	RES	0	0
IN15	SON	SON	SON	SON	SON	SON	0	0

Port	SCON-C/CA PIO pattern 0	PIO pattern 1	PIO pattern 2	PIO pattern 3	PIO pattern 4	PIO pattern 5	SCON-CA PIO pattern 6	PIO pattern 7	(Pulse train mode) PIO pattern 0
IN0	PC1	PC1	PC1	PC1	ST0	ST0	PC1	ST0	SON
IN1	PC2	PC2	PC2	PC2	ST1	ST1	PC2	ST1	RES
IN2	PC4	PC4	PC4	PC4	ST2	ST2	PC4	ST2	HOME
IN3	PC8	PC8	PC8	PC8	ST3	0	PC8	ST3	TL
IN4	PC16	PC16	PC16	PC16	ST4	0	PC16	ST4	CSTR
IN5	PC32	PC32	PC32	PC32	ST5	0	0	0	DCLR
IN6	0	MODE	PC64	PC64	ST6	0	0	0	BKRL
IN7	0	JISL	PC128	PC128	0	0	0	0	RMOD
IN8	0	JOG+	0	PC256	0	0	CLBR	CLBR	0
IN9	BKRL	JOG−	BKRL	BKRL	BKRL	BKRL	BKRL	BKRL	0
IN10	RMOD	RMOD	RMOD	RMOD	RMOD	RMOD	RMOD	RMOD	0
IN11	HOME	HOME	HOME	HOME	HOME	0	HOME	HOME	0
IN12	*STP	*STP	*STP	*STP	*STP	0	*STP	*STP	0
IN13	CSTR	CSTR/ PWRT	CSTR	CSTR	0	0	CSTR	0	0
IN14	RES	RES	RES	RES	RES	RES	RES	RES	0
IN15	SON	SON	SON	SON	SON	SON	SON	SON	0

6.4.5 I/O Port Output Signal Status Reading <<DOPM>>

(1) Function

Port output values of the RC controller are stored directly regardless of the PIO pattern.

(2) Query format

Field	Number of characters	ASCII mode character string (fixed)	Remarks
Header	1	‘:’	
Slave address [H]	2	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	2	‘0’, ‘3’	Reading registers
Start address [H]	4	‘9’, ‘0’, ‘0’, ‘4’	Output port monitor register
Number of registers [H]	4	‘0’, ‘0’, ‘0’, ‘1’	Reading address 9004_H
Error check [H]	2	LRC calculation result	
Trailer	2	‘CR’, ‘LF’	
Total number of bytes	17		

(3) Response format

A response message contains 16 bits of data per register.

Field	Number of characters	ASCII mode character string (fixed)	Remarks
Header	1	‘:’	
Slave address [H]	2	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	2	‘0’, ‘3’	Reading registers
Number of data bytes [H]	2	‘0’, ‘2’	Reading one register (2 bytes)
Data 1 [H]	4	DO output value	DI output value [Hex]
Error check [H]	2	LRC calculation result	
Trailer	2	‘CR’, ‘LF’	
Total number of bytes	15		

(4) Sample query (Axis No. 0)

A sample query that reads input ports (address 9004_H) in a controller of axis No. 0 is shown below.

Query: 01039004000167[CR][LF]

Field	ASCII mode fixed character string	Converted ASCII code data [H]
Start	‘:’	3A
Slave address [H]	‘0’, ‘1’	3031
Function code [H]	‘0’, ‘3’	3033
Start address [H]	‘9’, ‘0’, ‘0’, ‘4’	39303034
Number of registers [H]	‘0’, ‘0’, ‘0’, ‘1’	30303031
Error check [H]	‘6’, ‘7’ (in accordance with LRC calculation)’	3637
End	‘CR’, ‘LF’	0D0A

The response to the query is as follows.

Response: 010302740086[CR][LF]

Field	ASCII mode fixed character string	Converted ASCII code data [H]
Start	‘:’	3A
Slave address [H]	‘0’, ‘1’	3031
Function code [H]	‘0’, ‘3’	3033
Number of data bytes [H]	‘0’, ‘2’ (2 bytes = 1 register)	3032
Data 1 [H]	‘7’, ‘4’, ‘0’, ‘0’	37343030
Error check [H]	‘8’, ‘6’ (in accordance with LRC calculation)	3836
End	‘CR’, ‘LF’	0D0A

The output port data area is “7400”$_H$ → Converted to the binary value “0111010000000000”

*** The data of the response example is simply an example and will vary depending on various conditions**

(5) Port assignment [For details, refer to the operation manual that comes with each RC controller.]
Write the port assignment of PIO patterns to each RC controller.
0 indicates that response data is always 0.

Port	ERC2 (PIO type) PIO pattern 0	PIO pattern 1	PIO pattern 2	PIO pattern 3
OUT0	PEND	PE0	PEND	PEND
OUT1	HEND	PE1	HEND	HEND
OUT2	ZONE	PE2	ZONE	ZONE
OUT3	*ALM	*ALM	*ALM	*ALM
OUT4	0	0	0	0
OUT5	0	0	0	0
OUT6	0	0	0	0
OUT7	0	0	0	0
OUT8	0	0	0	0
OUT9	0	0	0	0
OUT10	0	0	0	0
OUT11	0	0	0	0
OUT12	0	0	0	0
OUT13	0	0	0	0
OUT14	0	0	0	0
OUT15	0	0	0	0

Port	PCON-C/CG PIO pattern 0	PIO pattern 1	PIO pattern 2	PIO pattern 3	PIO pattern 4	PIO pattern 5	PCON-PL/PO PIO pattern 0	PIO pattern 1
OUT0	PM1	PM1	PM1	PM1	PE0	LS0	SV	SV
OUT1	PM2	PM2	PM2	PM2	PE1	LS1	INP	INP/ TLR
OUT2	PM4	PM4	PM4	PM4	PE2	LS2	HEND	HEND
OUT3	PM8	PM8	PM8	PM8	PE3	0	*ALM	*ALM
OUT4	PM16	PM16	PM16	PM16	PE4	0	0	0
OUT5	PM32	PM32	PM32	PM32	PE5	0	0	0
OUT6	MOVE	MOVE	PM64	PM64	PE6	0	0	0
OUT7	ZONE1	MODES	PM128	PM128	ZONE1	ZONE1	0	0
OUT8	PZONE	PZONE	PZONE	PM256	PZONE	PZONE	0	0
OUT9	RMDS	RMDS	RMDS	RMDS	RMDS	RMDS	0	0
OUT10	HEND	HEND	HEND	HEND	HEND	HEND	0	0
OUT11	PEND	PEND/ WEND	PEND	PEND	PEND	0	0	0
OUT12	SV	SV	SV	SV	SV	SV	0	0
OUT13	*EMGS	*EMGS	*EMGS	*EMGS	*EMGS	*EMGS	0	0
OUT14	*ALM	*ALM	*ALM	*ALM	*ALM	*ALM	0	0
OUT15	0	0	0	0	0	0	0	0

Port	ACON-C/CG						ACON-PL/PO	
	PIO pattern 0	PIO pattern 1	PIO pattern 2	PIO pattern 3	PIO pattern 4	PIO pattern 5	PIO pattern 0	PIO pattern 1
OUT0	PM1	PM1	PM1	PM1	PE0	LS0	SV	SV
OUT1	PM2	PM2	PM2	PM2	PE1	LS1	INP	INP/ TLR
OUT2	PM4	PM4	PM4	PM4	PE2	LS2	HEND	HEND
OUT3	PM8	PM8	PM8	PM8	PE3	0	*ALM	*ALM
OUT4	PM16	PM16	PM16	PM16	PE4	0	0	0
OUT5	PM32	PM32	PM32	PM32	PE5	0	0	0
OUT6	MOVE	MOVE	PM64	PM64	PE6	0	0	0
OUT7	ZONE1	MODES	PM128	PM128	ZONE1	ZONE1	0	0
OUT8	PZONE	PZONE	PZONE	PM256	PZONE	PZONE	0	0
OUT9	RMDS	RMDS	RMDS	RMDS	RMDS	RMDS	0	0
OUT10	HEND	HEND	HEND	HEND	HEND	HEND	0	0
OUT11	PEND	PEND/ WEND	PEND	PEND	PEND	0	0	0
OUT12	SV	SV	SV	SV	SV	SV	0	0
OUT13	*EMGS	*EMGS	*EMGS	*EMGS	*EMGS	*EMGS	0	0
OUT14	*ALM	*ALM	*ALM	*ALM	*ALM	*ALM	0	0
OUT15	TRQS	0	TRQS	TRQS	TRQS	0	0	0

Port	SCON-C/CA						SCON-CA		(Pulse train mode)
	PIO pattern 0	PIO pattern 1	PIO pattern 2	PIO pattern 3	PIO pattern 4	PIO pattern 5	PIO pattern 6	PIO pattern 7	PIO pattern 0
OUT0	PM1	PM1	PM1	PM1	PE0	LS0	PM1	PE0	PWR
OUT1	PM2	PM2	PM2	PM2	PE1	LS1	PM2	PE1	SV
OUT2	PM4	PM4	PM4	PM4	PE2	LS2	PM4	PE2	INP
OUT3	PM8	PM8	PM8	PM8	PE3	0	PM8	PE3	HEND
OUT4	PM16	PM16	PM16	PM16	PE4	0	PM16	PE4	TLR
OUT5	PM32	PM32	PM32	PM32	PE5	0	TRQS	TRQS	*ALM
OUT6	MOVE	MOVE	PM64	PM64	PE6	0	LOAD	LOAD	*EMGS
OUT7	ZONE1	MODES	PM128	PM128	ZONE1	ZONE1	CEND	CEND	RMDS
OUT8	PZONE	PZONE	PZONE	PM256	PZONE	PZONE	PZONE	PZONE	ALM1
OUT9	RMDS	RMDS	RMDS	RMDS	RMDS	RMDS	RMDS	RMDS	ALM2
OUT10	HEND	HEND	HEND	HEND	HEND	HEND	HEND	HEND	ALM4
OUT11	PEND	PEND/ WEND	PEND	PEND	PEND	0	PEND	PEND	ALM8
OUT12	SV	SV	SV	SV	SV	SV	SV	SV	0
OUT13	*EMGS	*EMGS	*EMGS	*EMGS	*EMGS	*EMGS	*EMGS	*EMGS	0
OUT14	*ALM	*ALM	*ALM	*ALM	*ALM	*ALM	*ALM	*ALM	ZONE1
OUT15	*BALM	*BALM	*BALM	*BALM	*BALM	*BALM	*BALM	*BALM	ZONE2

6.4.6 Controller Status Signal Reading <<DSS1>>

(1) Function

This query reads the internal status of the controller.
[Refer to 4.3.2 (4), "Data of device status register 1."]

(2) Query format

Field	Number of characters	ASCII mode character string (fixed)	Remarks
Header	1	':'	
Slave address [H]	2	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	2	'0', '3'	Reading registers
Start address [H]	4	'9', '0', '0', '5'	Device status register 1
Number of registers [H]	4	'0', '0', '0', '1'	Reading address 9005_H
Error check [H]	2	LRC calculation result	
Trailer	2	'CR', 'LF'	
Total number of bytes	17		

(3) Response format

A response message contains 16 bits of data per address.

Field	Number of characters	ASCII mode character string (fixed)	Remarks
Header	1	':'	
Slave address [H]	2	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	2	'0', '3'	Reading registers
Number of data bytes [H]	2	'0', '2'	Reading one register (2 bytes)
Data [H]	4	Status 1	Status 1 [Hex]
Error check [H]	2	LRC calculation result	
Trailer	2	'CR', 'LF'	
Total number of bytes	15		

(4) Sample query

A sample query that reads the device status (address 9005_H) in a controller of axis No. 0 is shown below.

Query: 01 03 90 05 00 01 66 [CR] [LF]

Field	ASCII mode fixed character string	Converted ASCII code data [H]
Start	‘:’	3A
Slave address [H]	‘0’, ‘1’	3031
Function code [H]	‘0’, ‘3’	3033
Start address [H]	‘9’, ‘0’, ‘0’, ‘5’	39303035
Number of registers [H]	‘0’, ‘0’, ‘0’, ‘1’	30303031
Error check [H]	‘6’, ‘6’ (in accordance with LRC calculation)	3636
End	‘CR’, ‘LF’	0D0A

The response to the query is as follows.

Response: 01 03 02 30 88 42 [CR] [LF]

Field	ASCII mode fixed character string	Converted ASCII code data [H]
Start	‘:’	3A
Slave address [H]	‘0’, ‘1’	3031
Function code [H]	‘0’, ‘3’	3033
Number of data bytes [H]	‘0’, ‘2’ (2 bytes = 1 register)	3032
Data 1 [H]	‘3’, ‘0’, ‘8’, ‘8’	33303838
Error check [H]	‘4’, ‘2’ (in accordance with LRC calculation)	3432
End	‘CR’, ‘LF’	0D0A

* **The data of the response example is simply an example and will vary depending on various conditions.**

6.4.7 Controller Status Reading 2 <<DSS2>>

(1) Function

This query reads the internal status2 of the controller.
[Refer to 4.3.2 (5), "Data of device status register 2."]

(2) Query format

Field	Number of characters	ASCII mode character string (fixed)	Remarks
Header	1	':'	
Slave address [H]	2	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	2	'0', '3'	Reading registers
Start address [H]	4	'9', '0', '0', '6'	Device status register 2
Number of registers [H]	4	'0', '0', '0', '1'	Reading address 9006_H
Error check [H]	2	LRC calculation result	
Trailer	2	'CR', 'LF'	
Total number of bytes	17		

(3) Response format

A response message contains 16 bits of data per register.

Field	Number of characters	ASCII mode character string (fixed)	Remarks
Header	1	':'	
Slave address [H]	2	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	2	'0', '3'	Internal controller status
Number of data bytes [H]	2	'0', '2'	Reading one register (2 bytes).
Data [H]	4	Status 2	Status 2 [Hex]
Error check [H]	2	LRC calculation result	
Trailer	2	'CR', 'LF'	
Total number of bytes	15		

(4) Sample query

A sample query that reads the device status (address 9006_H) in a controller of axis No. 0 is shown below.

Query: 01 03 90 06 00 01 65 [CR] [LF]

Field	ASCII mode fixed character string	Converted ASCII code data [H]
Start	‘:’	3A
Slave address [H]	‘0’, ‘1’	3031
Function code [H]	‘0’, ‘3’	3033
Start address [H]	‘9’, ‘0’, ‘0’, ‘6’	39303036
Number of registers [H]	‘0’, ‘0’, ‘0’, ‘1’	30303031
Error check [H]	‘6’, ‘5’ (In accordance with LRC calculation)	3635
End	‘CR’, ‘LF’	0D0A

The response to the query is as follows.

Response: 01 03 02 80 00 7A [CR] [LF]

Field	ASCII mode fixed character string	Converted ASCII code data [H]
Start	‘:’	3A
Slave address [H]	‘0’, ‘1’	3031
Function code [H]	‘0’, ‘3’	3033
Number of data bytes [H]	‘0’, ‘2’ (2 bytes = 1 register)	3032
Data 1 [H]	‘8’, ‘0’, ‘0’, ‘0’	38303030
Error check [H]	‘7’, ‘A’ (In accordance with LRC calculation)	3741
End	‘CR’, ‘LF’	0D0A

* **The data of the response example is simply an example and will vary depending on various conditions.**

6.4.8 Controller Status Reading 3 <<DSSE>>

(1) Function

Internal statuses (expansion device) of the controller are indicated.
[Refer to 4.3.2 (6), "Data of expansion device status registers."]

(2) Query format

Field	Number of characters	ASCII mode character string (fixed)	Remarks
Header	1	':'	
Slave address [H]	2	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	2	'0', '3'	Reading registers
Start address [H]	4	'9', '0', '0', '7'	Expansion device status register
Number of registers [H]	4	'0', '0', '0', '1'	Reading address 9007_H
Error check [H]	2	LRC calculation result	
Trailer	2	'CR', 'LF'	
Total number of bytes	17		

(3) Response format

A response message contains 16 bits of data per register.

Field	Number of characters	ASCII mode character string (fixed)	Remarks
Header	1	':'	
Slave address [H]	2	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	2	'0', '3'	Reading registers
Number of data bytes [H]	2	'0', '2'	Reading one register (2 bytes).
Data [H]	4	Expansion status	Expansion status [Hex]
Error check [H]	2	LRC calculation result	
Trailer	2	'CR', 'LF'	
Total number of bytes	15		

(4) Sample query

A sample query that reads the expansion device status (address 9007_H) in a controller of axis No. 0 is shown below.

Query: 01 03 90 07 00 01 64 [CR] [LF]

Field	ASCII mode fixed character string	Converted ASCII code data [H]
Start	‘:’	3A
Slave address [H]	‘0’, ‘1’	3031
Function code [H]	‘0’, ‘3’	3033
Start address [H]	‘9’, ‘0’, ‘0’, ‘7’	39303037
Number of registers [H]	‘0’, ‘0’, ‘0’, ‘1’	30303031
Error check [H]	‘6’, ‘4’ (In accordance with LRC calculation)	3634
End	‘CR’, ‘LF’	0D0A

The response to the query is as follows.

Response: 01 03 02 33 C7 00 [CR] [LF]

Field	ASCII mode fixed character string	Converted ASCII code data [H]
Start	‘:’	3A
Slave address [H]	‘0’, ‘1’	3031
Function code [H]	‘0’, ‘3’	3033
Number of data bytes [H]	‘0’, ‘2’ (2 bytes = 1 register)	3032
Data 1 [H]	‘3’, ‘3’, ‘C’, ‘7’	33334337
Error check [H]	‘0’, ‘0’ (In accordance with LRC calculation)	3030
End	‘CR’, ‘LF’	0D0A

* **The data of the response example is simply an example and will vary depending on various conditions.**

6.4.9 Controller Status Reading 4 <<STAT>>

(1) Function

This query reads the internal operation status of the controller.
[Refer to "4.3.2 (7) Data of system status register."]

(2) Query format

Field	Number of characters	ASCII mode character string (fixed)	Remarks
Header	1	':'	
Slave address [H]	2	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	2	'0', '3'	Reading registers
Start address [H]	4	'9', '0', '0', '8'	System status register
Number of registers [H]	4	'0', '0', '0', '2'	Reading addresses 9008_H to 9009_H
Error check [H]	2	LRC calculation result	
Trailer	2	'CR', 'LF'	
Total number of bytes	17		

(3) Response format

A response message contains 16 bits of data per register.

Field	Number of characters	ASCII mode character string (fixed)	Remarks
Header	1	':'	
Slave address [H]	2	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	2	'0', '3'	Internal controller status
Number of data bytes [H]	2	'0', '4'	Reading two registers (4 bytes).
Data [H]	8	System status	System status [Hex]
Error check [H]	2	LRC calculation result	
Trailer	2	'CR', 'LF'	
Total number of bytes	19		

(4) Sample query

A sample query that reads the system status (address 9008_H) in a controller of axis No. 0 is shown below.

Query: 01 03 90 08 00 02 62 [CR] [LF]

Field	ASCII mode fixed character string	Converted ASCII code data [H]
Start	‘:’	3A
Slave address [H]	‘0’, ‘1’	3031
Function code [H]	‘0’, ‘3’	3033
Start address [H]	‘9’, ‘0’, ‘0’, ‘8’	39303038
Number of registers [H]	‘0’, ‘0’, ‘0’, ‘2’	30303032
Error check [H]	‘6’, ‘2’ (In accordance with LRC calculation)	3632
End	‘CR’, ‘LF’	0D0A

The response to the query is as follows.

Response: 01 03 04 00 0C 00 11 DB [CR] [LF]

Field	ASCII mode fixed character string	Converted ASCII code data [H]
Start	‘:’	3A
Slave address [H]	‘0’, ‘1’	3031
Function code [H]	‘0’, ‘3’	3033
Number of data bytes [H]	‘0’, ‘4’ (4 bytes = 2 registers)	3034
Data 1 [H]	‘0’, ‘0’, ‘0’, ‘C’	30303043
Data 2 [H]	‘0’, ‘0’, ‘1’, ‘1’	30303131
Error check [H]	‘D’, ‘B’ (In accordance with LRC calculation)	4442
End	‘CR’, ‘LF’	0D0A

* **The data of the response example is simply an example and will vary depending on various conditions.**

6.4.10 Current Speed Query <<VNOW>>

(1) Function

The monitored data of actual motor speed is indicated. The value becomes positive or negative depending on the operating direction of the motor. The unit is 0.01 mm/sec.

(2) Query format

Field	Number of characters	ASCII mode character string (fixed)	Remarks
Header	1	‘:’	
Slave address [H]	2	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	2	‘0’, ‘3’	Reading registers
Start address [H]	4	‘9’, ‘0’, ‘0’, ‘A’	Current speed monitor
Number of registers [H]	4	‘0’, ‘0’, ‘0’, ‘2’	Reading addresses $900A_H$ to $900B_H$
Error check [H]	2	LRC calculation result	
Trailer	2	‘CR’, ‘LF’	
Total number of bytes	17		

(3) Response format

A response message contains 16 bits of data per register.

Field	Number of characters	ASCII mode character string (fixed)	Remarks
Header	1	‘:’	
Slave address [H]	2	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	2	‘0’, ‘3’	Reading registers
Number of data bytes [H]	2	‘0’, ‘4’	Reading two registers = 4 bytes
Data [H]	8	Current speed	Current speed [Hex] Indicated in units of 0.01 mm/sec.
Error check [H]	2	LRC calculation result	
Trailer	2	‘CR’, ‘LF’	
Total number of bytes	19		

(4) Sample query

A sample query that reads the speed (address $900A_H$) of a controller of axis No. 0 is shown below.
Query: 01 03 90 0A 00 02 60 [CR] [LF]

Field	ASCII mode fixed character string	Converted ASCII code data [H]
Start	‘:’	3A
Slave address [H]	‘0’, ‘1’	3031
Function code [H]	‘0’, ‘3’	3033
Start address [H]	‘9’, ‘0’, ‘0’, ‘A’	39303041
Number of registers [H]	‘0’, ‘0’, ‘0’, ‘2’	30303032
Error check [H]	‘6’, ‘0’ (In accordance with LRC calculation)	3630
End	‘CR’, ‘LF’	0D0A

The response to the query is as follows.
Response: 01 03 04 00 00 26 FC D6 [CR] [LF]

Field	ASCII mode fixed character string	Converted ASCII code data [H]
Start	‘:’	3A
Slave address [H]	‘0’, ‘1’	3031
Function code [H]	‘0’, ‘3’	3033
Number of data bytes [H]	‘0’, ‘4’ (4 bytes = 2 registers)	3034
Data 1 [H]	‘0’, ‘0’, ‘0’, ‘0’	30303030
Data 2 [H]	‘2’, ‘6’, ‘F’, ‘C’	32364643
Error check [H]	‘D’, ‘6’ (In accordance with LRC calculation)	4436
End	‘CR’, ‘LF’	0D0A

The current speed is “000026FC” → Converted to a decimal value → 9980 (x 0.01 mm/sec)
The current speed monitor is 99.8 mm/sec.

*** The data of the response example is simply an example and will vary depending on various conditions.**

6.4.11 Current Ampere Reading <<CNOW>>

(1) Function

The monitored data of motor current is indicated in mA.
The torque current command value is stored.

(2) Query format

Field	Number of characters	ASCII mode character string (fixed)	Remarks
Header	1	':'	
Slave address [H]	2	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	2	'0', '3'	Reading registers
Start address [H]	4	'9', '0', '0', 'C'	Current ampere monitor
Number of registers [H]	4	'0', '0', '0', '2'	Reading addresses $900C_H$ to $900D_H$
Error check [H]	2	LRC calculation result	
Trailer	2	'CR', 'LF'	
Total number of bytes	17		

(3) Response format

A response message contains 16 bits of data per register.

Field	Number of characters	ASCII mode character string (fixed)	Remarks
Header	1	':'	
Slave address [H]	2	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	2	'0', '3'	Reading registers
Number of data bytes [H]	2	'0', '4'	Reading two registers (4 bytes).
Data [H]	8	Motor current monitor	Motor current monitor [Hex] Indicated in mA.
Error check [H]	2	LRC calculation result	
Trailer	2	'CR', 'LF'	
Total number of bytes	19		

(4) Sample query

A sample query that reads the current ampere value (address $900C_H$) of a controller of axis No. 0 is shown below.

Query: 01 03 90 0C 00 02 5E [CR] [LF]

Field	ASCII mode fixed character string	Converted ASCII code data [H]
Start	‘:’	3A
Slave address [H]	‘0’, ‘1’	3031
Function code [H]	‘0’, ‘3’	3033
Start address [H]	‘9’, ‘0’, ‘0’, ‘C’	39303043
Number of registers [H]	‘0’, ‘0’, ‘0’, ‘2’	30303032
Error check [H]	‘5’, ‘E’ (In accordance with LRC calculation)	3545
End	‘CR’, ‘LF’	0D0A

The response to the query is as follows.

Response: 01 03 04 00 00 01 C8 2F [CR] [LF]

Field	ASCII mode fixed character string	Converted ASCII code data [H]
Start	‘:’	3A
Slave address [H]	‘0’, ‘1’	3031
Function code [H]	‘0’, ‘3’	3033
Number of data bytes [H]	‘0’, ‘4’ (4 bytes = 2 registers)	3034
Data 1 [H]	‘0’, ‘0’, ‘0’, ‘0’	30303030
Data 2 [H]	‘0’, ‘1’, ‘C’, ‘8’	30314338
Error check [H]	‘2’, ‘F’ (In accordance with LRC calculation)	3246
End	‘CR’, ‘LF’	0D0A

The current ampere value is “000001C8” → Converted to a decimal value → 456 (mA)
The current ampere monitor value is 456 mA.

*** The data of the response example is simply an example and will vary depending on various conditions.**

6.4.12 Deviation Reading <<DEVI>>

(1) Function

This query reads the deviation over a 1-ms period between the position command value and the feedback value (actual position). The unit is pulse. The number of pulses per one motor revolution in mechanical angle varies depending on the encoder used.

(2) Query format

Field	Number of characters	ASCII mode character string (fixed)	Remarks
Header	1	‘:’	
Slave address [H]	2	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	2	‘0’, ‘3’	Reading registers
Start address [H]	4	‘9’, ‘0’, ‘0’, ‘E’	Deviation monitor
Number of registers [H]	4	‘0’, ‘0’, ‘0’, ‘2’	Reading addresses $900E_H$ to $900F_H$
Error check [H]	2	LRC calculation result	
Trailer	2	‘CR’, ‘LF’	
Total number of bytes	17		

(3) Response format

A response message contains 16 bits of data per register.

Field	Number of characters	ASCII mode character string (fixed)	Remarks
Header	1	‘:’	
Slave address [H]	2	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	2	‘0’, ‘3’	Reading registers
Number of data bytes [H]	2	‘0’, ‘4’	Reading two registers (4 bytes)
Data [H]	8	Deviation monitor	Deviation monitor [Hex] Indicated in pulses.
Error check [H]	2	LRC calculation result	
Trailer	2	‘CR’, ‘LF’	
Total number of bytes	19		

(4) Sample query

A sample query that reads the deviation (address $900E_H$) of a controller of axis No. 0 is shown below.

Query: 01 03 90 0E 00 02 5C [CR] [LF]

Field	ASCII mode fixed character string	Converted ASCII code data [H]
Start	‘:’	3A
Slave address [H]	‘0’, ‘1’	3031
Function code [H]	‘0’, ‘3’	3033
Start address [H]	‘9’, ‘0’, ‘0’, ‘E’	39303045
Number of registers [H]	‘0’, ‘0’, ‘0’, ‘2’	30303032
Error check [H]	‘5’, ‘C’ (In accordance with LRC calculation)	3543
End	‘CR’, ‘LF’	0D0A

The response to the query is as follows.

Response: 01 03 04 00 00 00 83 75 [CR] [LF]

Field	ASCII mode fixed character string	Converted ASCII code data [H]
Start	‘:’	3A
Slave address [H]	‘0’, ‘1’	3031
Function code [H]	‘0’, ‘3’	3033
Number of data bytes [H]	‘0’, ‘4’ (4 bytes = 2 registers)	3034
Data 1 [H]	‘0’, ‘0’, ‘0’, ‘0’	30303030
Data 2 [H]	‘0’, ‘0’, ‘8’, ‘3’	30303833
Error check [H]	‘7’, ‘5’ (In accordance with LRC calculation)	3735
End	‘CR’, ‘LF’	0D0A

The deviation monitor is “00000083” → Converted to a decimal value → 131 pulse

The deviation over a 1-ms period between the position command value and the feedback value (actual position) is 131 pulses.

* **The data of the response example is simply an example and will vary depending on various conditions.**

6.4.13 Total Time after Power On Reading <<STIM>>

(1) Function

This query reads the total time since the controller power was turned on. The unit is ms.
This value is not cleared by a software reset.

(2) Query format

Field	Number of characters	ASCII mode character string (fixed)	Remarks
Header	1	‘:’	
Slave address [H]	2	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	2	‘0’, ‘3’	Reading registers
Start address [H]	4	‘9’, ‘0’, ‘1’, ‘0’	System timer
Number of registers [H]	4	‘0’, ‘0’, ‘0’, ‘2’	Reading addresses 9010_H to 9011_H
Error check [H]	2	LRC calculation result	
Trailer	2	‘CR’, ‘LF’	
Total number of bytes	17		

(3) Response format

A response message contains 16 bits of data per register.

Field	Number of characters	ASCII mode character string (fixed)	Remarks
Header	1	‘:’	
Slave address [H]	2	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	2	‘0’, ‘3’	Reading registers
Number of data bytes [H]	2	‘0’, ‘4’	Reading two registers (4 bytes).
Data [H]	8	System timer	System timer [Hex] Indicated in ms.
Error check [H]	2	LRC calculation result	
Trailer	2	‘CR’, ‘LF’	
Total number of bytes	19		

(4) Sample query

A sample query that reads the startup time (address 9010_H) of a controller of axis No. 0 is shown below.

Query: 01 03 90 10 00 02 5A [CR] [LF]

Field	ASCII mode fixed character string	Converted ASCII code data [H]
Start	‘:’	3A
Slave address [H]	‘0’, ‘1’	3031
Function code [H]	‘0’, ‘3’	3033
Start address [H]	‘9’, ‘0’, ‘1’, ‘0’	39303130
Number of registers [H]	‘0’, ‘0’, ‘0’, ‘2’	30303032
Error check [H]	‘5’, ‘A’ (In accordance with LRC calculation)	3541
End	‘CR’, ‘LF’	0D0A

The response to the query is as follows.

Response: 01 03 04 02 38 C0 94 6A [CR] [LF]

Field	ASCII mode fixed character string	Converted ASCII code data [H]
Start	‘:’	3A
Slave address [H]	‘0’, ‘1’	3031
Function code [H]	‘0’, ‘3’	3033
Number of data bytes [H]	‘0’, ‘4’ (4 bytes = 2 registers)	3034
Data 1 [H]	‘0’, ‘2’, ‘3’, ‘8’	30323338
Data 2 [H]	‘C’, ‘0’, ‘9’, ‘4’	43303934
Error check [H]	‘6’, ‘A’ (In accordance with LRC calculation)	3641
End	‘CR’, ‘LF’	0D0A

The system timer value is “0238C094” → Converted to a decimal value → 37273748 ms
The total time since the controller power is turned on is 10.3538 hours.

* **The data of the response example is simply an example and will vary depending on various conditions.**

6.4.14 Special Input Port Input Signal Status Query <<SIPM>>

(1) Function

This query reads the status of input ports other than the normal input port.
[Refer to 4.3.2 (8), “Data of special input port monitor registers” for the data input via the special input port.]

(2) Query format

Field	Number of characters	ASCII mode character string (fixed)	Remarks
Header	1	‘:’	
Slave address [H]	2	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	2	‘0’, ‘3’	Reading registers
Start address [H]	4	‘9’, ‘0’, ‘1’, ‘2’	Special input port monitor
Number of registers [H]	4	‘0’, ‘0’, ‘0’, ‘1’	Reading address 9012_H
Error check [H]	2	LRC calculation result	
Trailer	2	‘CR’, ‘LF’	
Total number of bytes	17		

(3) Response format

A response message contains 16 bits of data per register.

Field	Number of characters	ASCII mode character string (fixed)	Remarks
Header	1	‘:’	
Slave address [H]	2	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	2	‘0’, ‘3’	Reading registers
Number of data bytes [H]	2	‘0’, ‘2’	Reading one register `(2 bytes).
Data [H]	4	Special port monitor	Refer to 4.3.2 (8), “List table.”
Error check [H]	2	LRC calculation result	
Trailer	2	‘CR’, ‘LF’	
Total number of bytes	15		

(4) Sample query

A sample query that reads the special input port (address 9012_H) of a controller of axis No. 0 is shown below.

Query: 01 03 90 12 00 01 59 [CR] [LF]

Field	ASCII mode fixed character string	Converted ASCII code data [H]
Start	‘:’	3A
Slave address [H]	‘0’, ‘1’	3031
Function code [H]	‘0’, ‘3’	3033
Start address [H]	‘9’, ‘0’, ‘1’, ‘2’	39303132
Number of registers [H]	‘0’, ‘0’, ‘0’, ‘1’	30303031
Error check [H]	‘5’, ‘9’ (in accordance with LRC calculation)	3539
End	‘CR’, ‘LF’	0D0A

The response to the query is as follows.

Response: 01 03 02 03 00 F7

Field	ASCII mode fixed character string	Converted ASCII code data [H]
Start	‘:’	3A
Slave address [H]	‘0’, ‘1’	3031
Function code [H]	‘0’, ‘3’	3033
Number of data bytes [H]	‘0’, ‘2’ (2 bytes = 1 register)	3032
Data 1 [H]	‘0’, ‘3’, ‘0’, ‘0’	30333030
Error check [H]	‘F’, ‘7’ (in accordance with LRC calculation)	4637
End	‘CR’, ‘LF’	0D0A

* **The data of the response example is simply an example and will vary depending on various conditions.**

6.4.15 Zone Output Signal Status Reading <<ZONS>>

(1) Function

This query reads the status of zone output.
[Refer to 4.3.2 (9), "Data of zone status registers."]

(2) Query format

Field	Number of characters	ASCII mode character string (fixed)	Remarks
Header	1	':'	
Slave address [H]	2	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	2	'0', '3'	Reading registers
Start address [H]	4	'9', '0', '1', '3'	Zone status query
Number of registers [H]	4	'0', '0', '0', '1'	Reading address 9013_H
Error check [H]	2	LRC calculation result	
Trailer	2	'CR', 'LF'	
Total number of bytes	17		

(3) Response format

A response message contains 16 bits of data per register.

Field	Number of characters	ASCII mode character string (fixed)	Remarks
Header	1	':'	
Slave address [H]	2	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	2	'0', '3'	Reading registers
Number of data bytes [H]	2	'0', '2'	Reading one register (2 bytes).
Data [H]	4	Zone status	Refer to 4.3.2 (9), "List table."
Error check [H]	2	LRC calculation result	
Trailer	2	'CR', 'LF'	
Total number of bytes	15		

(4) Sample query

A sample query that reads the zone status (address 9013_H) of a controller of axis No. 0 is shown below.

Query: 01 03 90 13 00 01 58 [CR] [LF]

Field	ASCII mode fixed character string	Converted ASCII code data [H]
Start	‘:’	3A
Slave address [H]	‘0’, ‘1’	3031
Function code [H]	‘0’, ‘3’	3033
Start address [H]	‘9’, ‘0’, ‘1’, ‘3’	39303133
Number of registers [H]	‘0’, ‘0’, ‘0’, ‘1’	30303031
Error check [H]	‘5’, ‘8’ (In accordance with LRC calculation)	3538
End	‘CR’, ‘LF’	0D0A

The response to the query is as follows.

Response: 01 03 02 00 00 FA [CR] [LF]

Field	ASCII mode fixed character string	Converted ASCII code data [H]
Start	‘:’	3A
Slave address [H]	‘0’, ‘1’	3031
Function code [H]	‘0’, ‘3’	3033
Number of data bytes [H]	‘0’, ‘2’ (2 bytes = 1 register)	3032
Data 1 [H]	‘0’, ‘0’, ‘0’, ‘0’	30303030
Error check [H]	‘F’, ‘A’ (In accordance with LRC calculation)	4641
End	‘CR’, ‘LF’	0D0A

* **The data of the response example is simply an example and will vary depending on various conditions.**

6.4.16 Position Complete Number Query <<POSS>>

(1) Function

This query reads the position complete number.
[Refer to “4.3.2 (10) Data of position number status register.”]

(2) Query format

Field	Number of characters	ASCII mode character string (fixed)	Remarks
Header	1	‘:’	
Slave address [H]	2	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	2	‘0’, ‘3’	Reading registers
Start address [H]	4	‘9’, ‘0’, ‘1’, ‘4’	Position number status
Number of registers [H]	4	‘0’, ‘0’, ‘0’, ‘1’	Reading address 9014_H
Error check [H]	2	LRC calculation result	
Trailer	2	‘CR’, ‘LF’	
Total number of bytes	17		

(3) Response format

A response message contains 16 bits of data per register.

Field	Number of characters	ASCII mode character string (fixed)	Remarks
Header	1	‘:’	
Slave address [H]	2	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	2	‘0’, ‘3’	Reading registers
Number of data bytes [H]	2	‘0’, ‘2’	Reading one register (2 bytes).
Data [H]	4	Position number status	Refer to 4.3.2 (10), “List table.”
Error check [H]	2	LRC calculation result	
Trailer	2	‘CR’, ‘LF’	
Total number of bytes	15		

(4) Sample query

A sample query that reads the position complete (address 9014_H) of a controller of axis No. 0 is shown below.

Query: 01 03 90 14 00 01 57 [CR] [LF]

Field	ASCII mode fixed character string	Converted ASCII code data [H]
Start	‘:’	3A
Slave address [H]	‘0’, ‘1’	3031
Function code [H]	‘0’, ‘3’	3033
Start address [H]	‘9’, ‘0’, ‘1’, ‘4’	39303134
Number of registers [H]	‘0’, ‘0’, ‘0’, ‘1’	30303031
Error check [H]	‘5’, ‘7’ (in accordance with LRC calculation)	3537
End	‘CR’, ‘LF’	0D0A

The response to the query is as follows.

Response: 01 03 02 00 00 FA [CR] [LF]

Field	ASCII mode fixed character string	Converted ASCII code data [H]
Start	‘:’	3A
Slave address [H]	‘0’, ‘1’	3031
Function code [H]	‘0’, ‘3’	3033
Number of data bytes [H]	‘0’, ‘2’ (2 bytes = 1 register)	3032
Data 1 [H]	‘0’, ‘0’, ‘0’, ‘0’	30303030
Error check [H]	‘F’, ‘A’ (in accordance with LRC calculation)	4641
End	‘CR’, ‘LF’	0D0A

* **The data of the response example is simply an example and will vary depending on various conditions**

6.4.17 Force Feedback Data Read <<FBFC>> --- SCON-CA Only

(1) Function

The monitored data of load cell measurement (push force) is read.
The unit is 0.01 N.

(2) Query format

Field	Number of characters	ASC II mode character string (fixed)	Remarks
Header	1	‘:’	Silent interval
Slave address [H]	2	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	2	‘0’, ‘3’	Register reading
Start address [H]	4	‘9’, ‘0’, ‘1’, ‘E’	Force feedback monitor
Number of registers [H]	4	‘0, ‘0, ‘0’, 2’	Reading address $901E_H$ ~$901F_H$
Error check [H]	2	LRC calculation result	
Trailer	2	‘CR’, ‘LF’	Silent interval
Total number of bytes	17		

(3) Response format

A response message contains 16 bits of data per register.

Field	Number of characters	ASC II mode character string (fixed)	Remarks
Header	1	‘:’	Silent interval
Slave address [H]	2	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	2	‘0’, ‘3’	Register reading
Number of data bytes [H]	2	‘0, ‘4’	Reading 2 register = 4 bytes
Data [H]	8	Position number status	Current push force [N] Unit: 0.01 N
Error check [H]	2	LRC calculation result	
Trailer	2	‘CR’, ‘LF’	Silent interval
Total number of bytes	19		

(4) Query sample

An example of use is shown, where the current measurement on the load cell connected to controller axis 0 is read.

● Query (silent intervals are inserted before and after the query)

01 03 90 0A 00 02 89 0D

Field	Fixed character strings in ASCII mode	Converted ASCII code data [H]
Start	‘:’	3A
Slave address [H]	‘0’, ‘1’	3031
Function code [H]	‘0’, ‘3’	3033
Start address [H]	‘9’, ‘0’, ‘1’, ‘E’	39393145
Number of registers [H]	‘0’, ‘0’, ‘0’, ‘2’	30303032
Error check [H]	‘4’, ‘C’ (in accordance with LRC calculation)	3443
End	‘CR’, ‘LF’	0D0A

The response to the query is as follows.

● Response (silent intervals are inserted before and after the response)

01 03 04 00 00 03 E4 FA 88

Field	Fixed character strings in ASCII mode	Converted ASCII code data [H]
Start	‘:’	3A
Slave address [H]	‘0’, ‘1’	3031
Function code [H]	‘0’, ‘3’	3033
Number of data bytes [H]	‘0’, ‘4’ (2 bytes = 1 register)	3034
Data 1 [H]	‘0’, ‘0’, ‘0’, ‘0’	30303030
Data 2 [H]	‘0’, ‘3’, ‘E’, ‘4’	30334534
Error check [H]	‘1’, ‘1’ (in accordance with LRC calculation)	3131
End	‘CR’, ‘LF’	0D0A

Example 1) The current measurement on the load cell is “000003E4, ” which is converted to a decimal value, or 996 (x 0.01 N) → The current push force is 9.96 N.

Example 2) If the current measurement reading on the load cell is “FFFFFF35” (tensile state(Note 2)), the formula $FFFFFFFF_H$ – $FFFFFF35_H$ + 1 (1 must be added) applies. The result is converted to a decimal value, or 203 (x 0.01 N) → The current tensile force(Note 2) is 2.03 N.

(Note 1) This is only one example of response. The specific response varies depending on each situation.

(Note 2) If a force is applied in the tensile direction, the load cell will break.

6.5 Operation Commands and Data Rewrite (Query Using Code 05)

6.5.1 Writing to Coil

*) Please refer to "6.2 ASCII Code Table."

(1) Function

Change (write) the status of DO (Discrete Output) of a slave to either ON or OFF.
In case of broadcast transmission, the coils at the specified address of all slaves are rewritten.

(2) Start address list

Start address [H]	Symbol	Function
0401	SFTY	Safety speed command
0403	SON	Servo ON command
0407	ALRS	Alarm reset command
0408	BKRL	Brake forced-release command
040A	STP	Pause command
040B	HOME	Home return command
040C	CSTR	Positioning start command
0411	JISL	Jog/inch switching
0414	MOD	Teaching mode command
0415	TEAC	Position data load command
0416	JOG+	Jog+ command
0417	JOG-	Jog- command
0418	ST7	Start position 7 (solenoid valve mode)
0419	ST6	Start position 6 (solenoid valve mode)
041A	ST5	Start position 5 (solenoid valve mode)
041B	ST4	Start position 4 (solenoid valve mode)
041C	ST3	Start position 3 (solenoid valve mode)
041D	ST2	Start position 2 (solenoid valve mode)
041E	ST1	Start position 1 (solenoid valve mode)
041F	ST0	Start position 0 (solenoid valve mode)
0426	CLBR	Load cell calibration command
0427	PMSL	PIO/Modbus switching specification
042C	STOP	Deceleration stop

6.5.2 Safety Speed Enable/Disable Switching (SFTY)

(1) Function

This query enables/disables the speed specified by user parameter No. 35, “Safety speed.”
Enabling the safety speed in the MANU mode will limit the speeds of all movement commands.

(2) Query format

Field	Number of characters	ASCII mode character string (fixed)	Remarks
Header	1	‘:’	
Slave address [H]	2	Arbitrary	Axis number + 1 (01_H to 10_H) 00_H when broadcast is specified
Function code [H]	2	‘0’, ‘5’	Write to a single coil DO.
Start address [H]	4	‘0’, ‘4’ ‘0’, ‘1’	Safety speed command
Changed data [H]	4	Arbitrary	Safety speed enabled: ‘F’, ‘F’, ‘0’, ‘0’ Safety speed disabled: ‘0’, ‘0’, ‘0’, ‘0’
Error check [H]	2	LRC calculation result	
Trailer	2	‘CR’, ‘LF’	
Total number of bytes	17		

(3) Response

If the change is successful, the response message will be the same as the query.
If invalid data is sent, an exception response (refer to section 7) will be returned, or no response will be returned.

(4) Sample query

A sample query that enables the safety speed of a controller of axis No. 0 is shown below.

Query: 01 05 04 01 FF 00 DC CA

Field	ASCII mode 8-bit data	Converted ASCII code data [H]
Start	‘:’	3A
Slave address [H]	‘0’, ‘1’	3031
Function code [H]	‘0’, ‘5’	3035
Start address [H]	‘0’, ‘4’, ‘0’, ‘1’	30343031
Changed data [H]	‘F’, ‘F’, ‘0’, ‘0’	46463030
Error check [H]	‘D’, ‘C’ (In accordance with LRC calculation)	4443
End	‘CR’, ‘LF’	0D0A

If the change is successful, the response message will be the same as the query.

6.5.3 Servo ON/OFF <<SON>>

(1) Function

Control ON/OFF of the servo.
When "Servo ON" is specified by the new data, the servo will turn ON after elapse of the manufacturer parameter "Servo ON delay time." However, the following conditions must be satisfied:

- The EMG status bit in device status register 1 is 0.
- The major failure status bit in device status register 1 is 0.
- The enable status bit in device status register 2 is 1.
- The auto servo OFF status in the system status register is 0.

(2) Query Format

Field	Number of characters	ASCII mode character string (fixed)	Remarks
Header	1	':'	
Slave address [H]	2	Arbitrary	Axis number + 1 (01_H to 10_H) 00_H when broadcast is specified
Function code [H]	2	'0', '5'	Write to a single coil DO
Start address [H]	4	'0', '4' '0', '3'	Servo ON/OFF command
Changed data [H]	4	Arbitrary	Servo ON: 'F', 'F', '0', '0' Servo OFF: '0', '0', '0', '0'
Error check [H]	2	LRC calculation result	
Trailer	2	'CR', 'LF'	
Total number of bytes	17		

* If a teaching pendant or PC software is connected, the servo is turned OFF, and then the teaching pendant/PC software is removed, all before the control establishes communication with the host, the servo cannot be turned ON/OFF via commands received from with the host.
In this case, restore the controller power, or make sure the connected tool of the SIO port is removed while the servo is ON.

(3) Response

If the change is successful, the response message will be the same as the query.
If invalid data is sent, an exception response (refer to section 7) will be returned, or no response will be returned.

(4) Sample query

A sample query that turns the servo of a controller of axis No. 0 on is shown below.

Query: 01 05 04 03 FF 00 F4

Field	ASCII mode 8-bit data	Converted ASCII code data [H]
Start	‘:’	3A
Slave address [H]	‘0’, ‘1’	3031
Function code [H]	‘0’, ‘5’	3035
Start address [H]	‘0’, ‘4’, ‘0’, ‘3’	30343033
Changed data [H]	‘F’, ‘F’, ‘0’, ‘0’	46463030
Error check [H]	‘F’, ‘4’ (In accordance with LRC calculation)	4634
End	‘CR’, ‘LF’	0D0A

If the change is successful, the response message will be the same as the query.

6.5.4 Alarm Reset <<ALRS>>

(1) Function

When the alarm reset edge is turned on (the data is first set to $FF00_H$ and then changed to 0000_H), **alarms will be reset**.
If any alarm cause has not been removed, the same alarm will be generated again. If the alarm reset edge is turned on while the actuator is paused, **the remaining travel will be cancelled**.
When alarms are reset, make sure to write changed data of 0000_H to restore the normal status.

(2) Query format

Field	Number of characters	ASCII mode character string (fixed)	Remarks
Header	1	‘:’	
Slave address [H]	2	Arbitrary	Axis number + 1 (01_H to 10_H) 00_H when broadcast is specified
Function code [H]	2	‘0’, ‘5’	Write to a single coil DO.
Start address [H]	4	‘0’, ‘4’ ‘0’, ‘7’	Alarm reset command
Changed data [H]	4	Arbitrary	Execute alarm reset: ‘F’, ‘F’, ‘0’, ‘0’ Normal: ‘0’, ‘0’, ‘0’, ‘0’
Error check [H]	2	LRC calculation result	
Trailer	2	‘CR’, ‘LF’	
Total number of bytes	17		

(3) Response

If the change is successful, the response message will be the same as the query.
If invalid data is sent, an exception response (refer to section 7) will be returned, or no response will be returned.

(4) Query sample

A sample query that resets the alarms of a controller of axis No. 0 is shown below.

First time 01 05 04 07 FF 00 F0 --- Execute alarm reset
Second time 01 05 04 07 00 00 EF --- Restore normal status

Field	ASCII mode 8-bit data	Converted ASCII code data [H]
Start	‘:’	3A
Slave address [H]	‘0’, ‘1’	3031
Function code [H]	‘0’, ‘5’	3035
Start address [H]	‘0’, ‘4’, ‘0’, ‘7’	30343037
Changed data [H]	First time: ‘F’, ‘F’, ‘0’, ‘0’ Second time: ‘0’, ‘0’, ‘0’, ‘0’ (Write 0000_H after resetting alarms to restore the normal status.)	46463030 30303030
Error check [H]	First time: ‘F’, ‘0’ (in accordance with LRC calculation) Second time: ‘E’, ‘F’ (in accordance with LRC calculation)	4630 4546
End	‘CR’, ‘LF’	0D0A

If the change is successful, the response message will be the same as the query.

6.5.5 Brake Forced Release <<BKRL>>

(1) Function

Brake control is linked to servo ON/OFF. The brake can be forcefully released even when the servo is ON.

(2) Query format

Field	Number of characters	ASCII mode character string (fixed)	Remarks
Header	1	‘:’	
Slave address [H]	2	Arbitrary	Axis number + 1 (01_H to 10_H) 00_H when broadcast is specified
Function code [H]	2	‘0’, ‘5’	Write to a single coil DO
Start address [H]	4	‘0’, ‘4’ ‘0’, ‘8’	Break forced-release command
Changed data [H]	4	Arbitrary	Brake forced release: ‘F’, ‘F’, ‘0’, ‘0’ Normal: ‘0’, ‘0’, ‘0’, ‘0’
Error check [H]	2	LRC calculation result	
Trailer	2	‘CR’, ‘LF’	
Total number of bytes	17		

* If a teaching pendant or PC software is connected, the servo is turned OFF, and then the teaching pendant/PC software is removed, all before the control establishes communication with the host, the servo cannot be turned ON/OFF via commands received from with the host.
In this case, restore the controller power, or make sure the connected tool of the SIO port is removed while the servo is ON.

(3) Response

If the change is successful, the response message will be the same as the query.
If invalid data is sent, an exception response (refer to section 7) will be returned, or no response will be returned.

(4) Sample query

A sample query that forcefully releases the brake of a controller of axis No. 0 is shown below.

Query: 01 05 04 08 FF 00 F0

Field	ASCII mode 8-bit data	Converted ASCII code data [H]
Start	‘:’	3A
Slave address [H]	‘0’, ‘1’	3031
Function code [H]	‘0’, ‘5’	3035
Start address [H]	‘0’, ‘4’, ‘0’, ‘8’	30343038
Changed data [H]	‘F’, ‘F’, ‘0’, ‘0’	46463030
Error check [H]	‘F’, ‘0’ (In accordance with LRC calculation)	4630
End	‘CR’, ‘LF’	0D0A

If the change is successful, the response message will be the same as the query.

6.5.6 Pause <<STP>>

(1) Function

If the pause command is transmitted during movement, the actuator decelerates and stops. If the status is set back to normal again, the actuator resumes moving for the remaining distance.
As long as the pause command is being transmitted, all motor movement is inhibited.
If the alarm reset command bit is set while the actuator is paused, the remaining travel will be cancelled.
If this bit is set during home return, the movement command will be held if the actuator has not yet reversed after contacting the mechanical end. If the actuator has already reversed after contacting the mechanical end, home return will be repeated from the beginning.

(2) Query format

Field	Number of characters	ASCII mode character string (fixed)	Remarks
Header	1	‘:’	
Slave address [H]	2	Arbitrary	Axis number + 1 (01_H to 10_H)
Function code [H]	2	‘0’, ‘5’	Write to a single coil DO.
Start address [H]	4	‘0’, ‘4’ ‘0’, ‘A’	Pause command
Changed data [H]	4	Arbitrary	Pause command: ‘F’, ‘F’, ‘0’, ‘0’ Normal: ‘0’, ‘0’, ‘0’, ‘0’
Error check [H]	2	LRC calculation result	
Trailer	2	‘CR’, ‘LF’	
Total number of bytes	17		

(3) Response

If the change is successful, the response message will be the same as the query.
If invalid data is sent, an exception response (refer to section 7) will be returned, or no response will be returned.

(4) Sample query

A sample query that pauses a controller of axis No. 0 is shown below.

Query: 01 05 04 0A FF 00 ED

Field	ASCII mode 8-bit data	Converted ASCII code data [H]
Start	‘:’	3A
Slave address [H]	‘0’, ‘1’	3031
Function code [H]	‘0’, ‘5’	3035
Start address [H]	‘0’, ‘4’, ‘0’, ‘A’	30343041
Changed data [H]	‘F’, ‘F’, ‘0’, ‘0’	46463030
Error check [H]	‘E’, ‘D’ (in accordance with LRC calculation)	4544
End	‘CR’, ‘LF’	0D0A

If the change is successful, the response message will be the same as the query.

6.5.7 Home return <<HOME>>

(1) Function

Home return operation will start if a rising edge in the home return command signal is detected (the data is first set to 0000_H and then changed to $FF00_H$). Once the home return is completed, the HEND bit will become 1. This command can be input as many times as desired even after home return has been completed once.

(2) Query format

Field	Number of characters	ASCII mode character string (fixed)	Remarks
Header	1	‘:’	
Slave address [H]	2	Arbitrary	Axis number + 1 (01_H to 10_H) 00_H when broadcast is specified
Function code [H]	2	‘0’, ‘5’	Write to a single coil DO.
Start address [H]	4	‘0’, ‘4’ ‘0’, ‘B’	Home return command
Changed data [H]	4	Arbitrary	Execute home return: ‘F’, ‘F’, ‘0’, ‘0’ Normal: ‘0’, ‘0’, ‘0’, ‘0’
Error check [H]	2	LRC calculation result	
Trailer	2	‘CR’, ‘LF’	
Total number of bytes	17		

* If a teaching pendant or PC software is connected, the servo is turned OFF, and then the teaching pendant/PC software is removed, all before the control establishes communication with the host, the servo cannot be turned ON/OFF via commands received from with the host.
In this case, restore the controller power, or make sure the connected tool of the SIO port is removed while the servo is ON.

(3) Response

If the change is successful, the response message will be the same as the query.
If invalid data is sent, an exception response (refer to section 7) will be returned, or no response will be returned.

(4) Sample query

A sample query that executes home return operation of a controller of axis No. 0 is shown below.

Query:

First time: 01 05 04 0B 00 00 EB --- Set normal status

Second time: 01 05 04 0B FF 00 EC --- Execute home return

Field	ASCII mode 8-bit data	Converted ASCII code data [H]
Start	‘:’	3A
Slave address [H]	‘0’, ‘1’	3031
Function code [H]	‘0’, ‘5’	3035
Start address [H]	‘0’, ‘4’, ‘0’, ‘B’	30343042
Changed data [H]	First time: ‘0’, ‘0’, ‘0’, ‘0’ Second time: ‘F’, ‘F’, ‘0’, ‘0’ (Send data twice to set the edge.)	30303030 46463030
Error check [H]	First time: ‘E’, ‘B’ (In accordance with LRC calculation) Second time: ‘E’, ‘C’ (In accordance with LRC calculation)	4542 4543
End	‘CR’, ‘LF’	0D0A

If the change is successful, the response message will be the same as the query.

6.5.8 Positioning Start Command <<CSTR>>

(1) Function

If the rising edge of the positioning start command is detected (the data is first set to $FF00_H$ and then changed to 0000_H), the actuator will move to the position specified by the position number stored in the position number specification register (POSR:$0D03_H$). If nothing is done after the position start command ($FF00_H$ is read and no new data is written), a position complete will not be output even when the actuator enters the positioning band (write 0000_H and restore the normal status).

If this command is executed when home return has never been performed after the power is turned on (when the HEND bit is 0), the actuator will perform home return and then start moving to the target position.

* The target position, speed and all other operation parameters must be set in the position table (nonvolatile memory) of the controller in advance.

(2) Query format

Field	Number of characters	ASCII mode character string (fixed)	Remarks
Header	1	‘:’	
Slave address [H]	2	Arbitrary	Axis number + 1 (01_H to 10_H) 00_H when broadcast is specified
Function code [H]	2	‘0’, ‘5’	Write to a single coil DO.
Start address [H]	4	‘0’, ‘4’ ‘0’, ‘C’	Positioning start command
Changed data [H]	4	Arbitrary	Positioning start command: ‘F’, ‘F’, ‘0’, ‘0’ Normal: ‘0’, ‘0’, ‘0’, ‘0’
Error check [H]	2	LRC calculation result	
Trailer	2	‘CR’, ‘LF’	
Total number of bytes	17		

(3) Response

If the change is successful, the response message will be the same as the query.

If invalid data is sent, an exception response (refer to section 7) will be returned, or no response will be returned.

(4) Sample query

A sample query that moves the actuator of a controller of axis No. 0 to the position specified by the position number stored in the position number specification register (POSR:0D03$_H$) is shown below.
Query:
First time: 01 05 04 0C FF 00 EB --- Movement command
Second time: 01 05 04 0C 00 00 EA --- Normal status

Field	ASCII mode 8-bit data	Converted ASCII code data [H]
Start	‘:’	3A
Slave address [H]	‘0’, ‘1’	3031
Function code [H]	‘0’, ‘5’	3035
Start address [H]	‘0’, ‘4’, ‘0’, ‘C’	30343043
Changed data [H]	First time: ‘F’, ‘F’, ‘0’, ‘0’ Second time: ‘0’, ‘0’, ‘0’, ‘0’ (Restore the normal status.)	46463030 30303030
Error check [H]	First time: ‘E’, ‘B’ (In accordance with LRC calculation) Second time: ‘E’, ‘A’ (In accordance with LRC calculation)	4542 4541
End	‘CR’, ‘LF’	0D0A

If the change is successful, the response message will be the same as the query.

6.5.9 Jog/Inch Switching <<JISL>>

(1) Function

This bit switches between jogging and inching.
If this bit switches while the actuator is jogging, the actuator will decelerate to a stop.
If this bit switches while the actuator is inching, the inching movement will continue.

(2) Query format

Field	Number of characters	ASCII mode character string (fixed)	Remarks
Header	1	‘:’	
Slave address [H]	2	Arbitrary	Axis number + 1 (01_H to 10_H) 00_H when broadcast is specified
Function code [H]	2	‘0’, ‘5’	Write to a single coil DO.
Start address [H]	4	‘0’, ‘4’ ‘1’, ‘1’	Jog/Inch Switching
Changed data [H]	4	Arbitrary	Inching operation status: ‘F’, ‘F’, ‘0’, ‘0’ Jogging operation status: ‘0’, ‘0’, ‘0’, ‘0’
Error check [H]	2	LRC calculation result	
Trailer	2	‘CR’, ‘LF’	
Total number of bytes	17		

(3) Response

If the change is successful, the response message will be the same as the query.
If invalid data is sent, an exception response (refer to section 7) will be returned, or no response will be returned.

(4) Sample query

A sample query that switches the operation of a controller of axis No. 0 to inching is shown below.

Query: 01 05 04 11 FF 00 E6

Field	ASCII mode 8-bit data	Converted ASCII code data [H]
Start	‘:’	3A
Slave address [H]	‘0’, ‘1’	3031
Function code [H]	‘0’, ‘5’	3035
Start address [H]	‘0’, ‘4’, ‘1’, ‘1’	30343131
Changed data [H]	‘F’, ‘F’, ‘0’, ‘0’	46463030
Error check [H]	‘E’, ‘6’ (In accordance with LRC calculation)	4536
End	‘CR’, ‘LF’	0D0A

If the change is successful, the response message will be the same as the query.

6.5.10 Teaching Mode Command <<MOD>>

(1) Function

This bit switches between the normal operation mode and teaching mode.

(2) Query format

Field	Number of characters	ASCII mode character string (fixed)	Remarks
Header	1	':'	
Slave address [H]	2	Arbitrary	Axis number + 1 (01_H to 10_H) 00_H when broadcast is specified
Function code [H]	2	'0', '5'	Write to a single coil DO.
Start address [H]	4	'0', '4' '1', '4'	Switch between the normal mode and the teaching mode.
Changed data [H]	4	Arbitrary	Teaching mode: 'F', 'F', '0', '0' Normal operation mode: '0', '0', '0', '0'
Error check [H]	2	LRC calculation result	
Trailer	2	'CR', 'LF'	
Total number of bytes	17		

(3) Response

If the change is successful, the response message will be the same as the query.
If invalid data is sent, an exception response (refer to section 7) will be returned, or no response will be returned.

(4) Sample query

A sample query that switches the operation mode of a controller of axis No. 0 to teaching mode is shown below.

Query: 01 05 04 14 FF 00 E3

Field	ASCII mode 8-bit data	Converted ASCII code data [H]
Start	‘:’	3A
Slave address [H]	‘0’, ‘1’	3031
Function code [H]	‘0’, ‘5’	3035
Start address [H]	‘0’, ‘4’, ‘1’, ‘4’	30343134
Changed data [H]	‘F’, ‘F’, ‘0’, ‘0’	46463030
Error check [H]	‘E’, ‘3’ (In accordance with LRC calculation)	4533
End	‘CR’, ‘LF’	0D0A

If the change is successful, the response message will be the same as the query.

6.5.11 Position Data Load Command <<TEAC>>

(1) Function

The current position is acquired by writing this command (write $FF00_H$) when the teaching mode command (6.5.10) is $FF00_H$ (teaching command).
The current position data will be written in the position number specified by the position number specification register when the aforementioned condition was detected.
If other position data fields are empty, the default parameter values will be written at the same time in the empty fields other than the target position (positioning band INP, speed VCMD, acceleration/deceleration speed ACMD, and control flag CTLF).
After sending this command (write $FF00_H$), keep the status as is for 20 ms or longer.

(2) Query format

Field	Number of characters	ASCII mode character string (fixed)	Remarks
Header	1	‘:’	
Slave address [H]	2	Arbitrary	Axis number + 1 (01_H to 10_H) 00_H when broadcast is specified
Function code [H]	2	‘0’, ‘5’	Write to a single coil DO.
Start address [H]	4	‘0’, ‘4’ ‘1’, ‘5’	Position data load command
Changed data [H]	4	Arbitrary	Position data load command: ‘F’, ‘F’, ‘0’, ‘0’ Normal: ‘0’, ‘0’, ‘0’, ‘0’
Error check [H]	2	LRC calculation result	
Trailer	2	‘CR’, ‘LF’	
Total number of bytes	17		

(3) Response

If the change is successful, the response message will be the same as the query.
If invalid data is sent, an exception response (refer to section 7) will be returned, or no response will be returned.

(4) Sample query

A sample query that acquires the current position when a controller of axis No. 0 is in the teaching mode is shown below.

Query: 01 05 04 15 FF 00 E2

Field	ASCII mode 8-bit data	Converted ASCII code data [H]
Start	‘:’	3A
Slave address [H]	‘0’, ‘1’	3031
Function code [H]	‘0’, ‘5’	3035
Start address [H]	‘0’, ‘4’, ‘1’, ‘5’	30343135
Changed data [H]	‘F’, ‘F’, ‘0’, ‘0’	46463030
Error check [H]	‘E’, ‘2’ (In accordance with LRC calculation)	4532
End	‘CR’, ‘LF’	0D0A

If the change is successful, the response message will be the same as the query.

6.5.12 Jog+ Command <<JOG+>>

(1) Function

- The actuator performs either jog or inching operation.
  If the jog+ command (changed data FF00$_H$) is sent when the jog/inch switching command (6.5.9) is set to 0000$_H$ (set to jog), the actuator will jog in the direction opposite home. The speed and acceleration/deceleration speed conform to the PIO jog speed set by user parameter No. 26 and rated acceleration/deceleration speed, respectively.
  If the jog+ command (changed data 0000$_H$) is sent or the jog- command (6.5.13, changed data FF00$_H$) is sent while the actuator is moving, the actuator will decelerate to a stop.
- If the jog+ command rising edge is set while the jog/inch switching command (6.5.9) is FF00$_H$ (set to inching), the actuator will inch in the direction opposite home. The speed, travel and acceleration/deceleration speed conform to user parameter No. 26 (PIO jogging speed), user parameter No. 48 (PIO inching distance), and rated acceleration/deceleration speed, respectively.

(2) Query format

Field	Number of characters	ASCII mode character string (fixed)	Remarks
Header	1	‘:’	
Slave address [H]	2	Arbitrary	Axis number + 1 (01$_H$ to 10$_H$) 00$_H$ when broadcast is specified
Function code [H]	2	‘0’, ‘5’	Write to a single coil DO.
Start address [H]	4	‘0’, ‘4’ ‘1’, ‘6’	Jog+ command
Changed data [H]	4	Arbitrary	Jog+ command: ‘F’, ‘F’, ‘0’, ‘0’ Normal: ‘0’, ‘0’, ‘0’, ‘0’
Error check [H]	2	LRC calculation result	
Trailer	2	‘CR’, ‘LF’	
Total number of bytes	17		

(3) Response

If the change is successful, the response message will be the same as the query.
If invalid data is sent, an exception response (refer to section 7) will be returned, or no response will be returned.

(4) Sample query

[1] A sample query that makes a controller of axis No. 0 jog is shown below.
Query: 01 05 04 16 FF 00 E1

Field	ASCII mode 8-bit data	Converted ASCII code data [H]
Start	‘:’	3A
Slave address [H]	‘0’, ‘1’	3031
Function code [H]	‘0’, ‘5’	3035
Start address [H]	‘0’, ‘4’, ‘1’, ‘6’	30343136
Changed data [H]	‘F’, ‘F’, ‘0’, ‘0’	46463030
Error check [H]	‘E’, ‘1’ (In accordance with LRC calculation)	4531
End	‘CR’, ‘LF’	0D0A

If the change was successful, the response message will be the same as the query.

[2] A sample query that makes a controller of axis No. 0 inch is shown below.
Query:
First time: 01 05 04 16 FF 00 E1 --- Inching movement
Second time: 01 05 04 16 00 00 E0 --- Restore normal status

Field	ASCII mode 8-bit data	Converted ASCII code data [H]
Start	‘:’	3A
Slave address [H]	‘0’, ‘1’	3031
Function code [H]	‘0’, ‘5’	3035
Start address [H]	‘0’, ‘4’, ‘1’, ‘6’	30343046
Changed data [H]	First time: ‘F’, ‘F’, ‘0’, ‘0’ Second time: ‘0’, ‘0’, ‘0’, ‘0’ (Restore the normal status.)	46463030 30303030
Error check [H]	First time: ‘E’, ‘1’ (In accordance with LRC calculation) Second time: ‘E’, ‘0’ (In accordance with LRC calculation)	4531 4530
End	‘CR’, ‘LF’	0D0A

If the change is successful, the response message will be the same as the query.

6.5.13 Jog- Command <<JOG->>

(1) Function

- The actuator performs either jog or inching operation.
  If the jog- command (changed data $FF00_H$) is sent when the jog/inch switching command (6.5.9) is set to 0000_H (set to jog), the actuator will jog in the direction of home. The speed and acceleration/deceleration speed conform to the PIO jog speed set by user parameter No. 26 and rated acceleration/deceleration speed, respectively.
  If the jog- command (changed data 0000_H) is sent or the jog+ command (6.5.12, changed data $FF00_H$) is sent while the actuator is moving, the actuator will decelerate to a stop.
- If the jog- command rising edge is set while the jog/inch switching command (6.5.9) is $FF00_H$ (set to inching), the actuator will inch in the direction of home. The speed, travel and acceleration/deceleration speed conform to user parameter No. 26 (PIO jogging speed), user parameter No. 48 (PIO inching distance), and rated acceleration/deceleration speed, respectively.

(2) Query format

Field	Number of characters	ASCII mode character string (fixed)	Remarks
Header	1	‘:’	
Slave address [H]	2	Arbitrary	Axis number + 1 (01_H to 10_H) 00_H when broadcast is specified
Function code [H]	2	‘0’, ‘5’	Write to a single coil DO.
Start address [H]	4	‘0’, ‘4’ ‘1’, ‘7’	Jog- command
Changed data [H]	4	Arbitrary	Jog- command: ‘F’, ‘F’, ‘0’, ‘0’ Normal: ‘0’, ‘0’, ‘0’, ‘0’
Error check [H]	2	LRC calculation result	
Trailer	2	‘CR’, ‘LF’	
Total number of bytes	17		

(3) Response

If the change is successful, the response message will be the same as the query.
If invalid data is sent, an exception response (refer to section 7) will be returned, or no response will be returned.

(4) Sample query

[1] A sample query that makes a controller of axis No. 0 jog is shown below.
Query: 01 05 04 17 EF 00 E0

Field	ASCII mode 8-bit data	Converted ASCII code data [H]
Start	‘:’	3A
Slave address [H]	‘0’, ‘1’	3031
Function code [H]	‘0’, ‘5’	3035
Start address [H]	‘0’, ‘4’, ‘1’, ‘7’	30343137
Changed data [H]	‘F’, ‘F’, ‘0’, ‘0’	46463030
Error check [H]	‘E’, ‘0’ (in accordance with LRC calculation)	4530
End	‘CR’, ‘LF’	0D0A

If the change was successful, the response message will be the same as the query.

[2] A sample query that makes a controller of axis No. 0 inch is shown below.
Query:
First time: 01 05 04 17 FF 00 E0 --- Inching movement
Second time: 01 05 04 17 00 00 DF --- Restore normal status

Field	ASCII mode 8-bit data	Converted ASCII code data [H]
Start	‘:’	3A
Slave address [H]	‘0’, ‘1’	3031
Function code [H]	‘0’, ‘5’	3035
Start address [H]	‘0’, ‘4’, ‘1’, ‘7’	30343047
Changed data [H]	First time: ‘F’, ‘F’, ‘0’, ‘0’ Second time: ‘0’, ‘0’, ‘0’, ‘0’ (Restore the normal status.)	46463030 30303030
Error check [H]	First time: ‘E’, ‘0’ (in accordance with LRC calculation) Second time: ‘D’, ‘F’ (in accordance with LRC calculation)	4530 4446
End	‘CR’, ‘LF’	0D0A

If the change is successful, the response message will be the same as the query.

6.5.14 Start Positions 0 to 7 <<ST0 to ST7>> (Limited to PIO Patterns 4 and 5)

(1) Function

The actuator moves to the specified position number position.
The movement command for start positions 0 to 7 is effective only when PIO pattern 4 or 5 (solenoid valve mode) is selected.
The movement command is sent by enabling either one of ST0 to ST7 in "6.5.14 (5) Start address" (write new value $FF00_H$ when 0000_H is set).
If a position other than the valid start positions is selected, "085: Moving position number error" will be generated.
Either level operation or edge operation can be selected using user parameter No. 27, "Movement command type."

(2) Query format

Field	Number of characters	ASCII mode character string (fixed)	Remarks
Header	1	':'	
Slave address [H]	2	Arbitrary	Axis number + 1 (01_H to 10_H) 00_H when broadcast is specified
Function code [H]	2	'0', '5'	Write to a single coil DO
Start address [H]	4	Arbitrary	Refer to 6.5.14 (5), "Start address."
Changed data [H]	4	Arbitrary	*1 Operation command: 'F', 'F', '0', '0' Operation command: '0', '0', '0', '0'
Error check [H]	2	LRC calculation result	
Trailer	2	'CR', 'LF'	
Total number of bytes	17		

*1 If user parameter No. 27 "Movement command type" is set to "level operation, " the actuator decelerates to a stop by overwriting $FF00_H$ with 0000_H.

(3) Response

If the change is successful, the response message will be the same as the query.
If invalid data is sent, an exception response (refer to section 7) will be returned, or no response will be returned.

(4) Query sample

A sample query that moves a controller of axis No. 0 to start position 2 is shown below.
Sample start position setting

0	0.00	150.00	0.30	0.30
1	25.00	150.00	0.30	0.30
2	50.00	150.00	0.30	0.30
3	0.00	150.00	0.20	0.20

Fig.6.2

Query
First time 01 05 04 1D 00 00 D9 --- Write 0000_H to set the edge
Second time 01 05 04 1D FF 00 DA --- Movement command

Field	ASCII mode 8-bit data	Converted ASCII code data [H]
Start	‘:’	3A
Slave address [H]	‘0’, ‘1’	3031
Function code [H]	‘0’, ‘5’	3035
Start address [H]	‘0’, ‘4’, ‘1’, ‘D’	30343044
Changed data [H]	First time: ‘0’, ‘0’, ‘0’, ‘0’ Second time: ‘F’, ‘F’, ‘0’, ‘0’	30303030 46463030
Error check [H]	First time: ‘D’, ‘9’ (In accordance with LRC calculation) Second time: ‘D’, ‘A’ (In accordance with LRC calculation)	4439 4441
End	‘CR’, ‘LF’	0D0A

If the change is successful, the response message will be the same as the query.

(5) Start address

Address	Symbol	Name	Function
0418	ST7	Start position 7	Move to position 7
0419	ST6	Start position 6	Move to position 6
041A	ST5	Start position 5	Move to position 5
041B	ST4	Start position 4	Move to position 4
041C	ST3	Start position 3	Move to position 3
041D	ST2	Start position 2	Move to position 2
041E	ST1	Start position 1	Move to position 1
041F	ST0	Start position 0	Move to position 0

6.5.15 Load Cell Calibration Command <<CLBR>> --- A dedicated load cell must be connected.

(1) Function --- SCON-CA only

The dedicated load cell is calibrated.

The factory setting of your load cell is that the ON status corresponds to a no-load state. If you want to define the reference state as a condition where a work part (load) is installed, calibrate the load cell.

Also calibrate the load cell in other situations as necessary (readjustment, inspection, etc.).

(2) Query format

Field	Number of characters	ASCII mode character string (fixed)	Remarks
Header	1	‘:’	
Slave address [H]	2	Arbitrary	Axis number + 1 (01_H to 10_H) 00_H when broadcast is specified
Function code [H]	2	‘0’, ‘5’	Write to a single coil DO
Start address [H]	2	‘0’, ‘4’ ‘2’, ‘6’	Load cell calibration command
Changed data [H]	2	Arbitrary	Calibration command: $FF00_H$ Normal operation: 0000_H
Error check [H]	2	LRC calculation result	
Trailer	2	‘CR’, ‘LF’	
Total number of bytes	17		

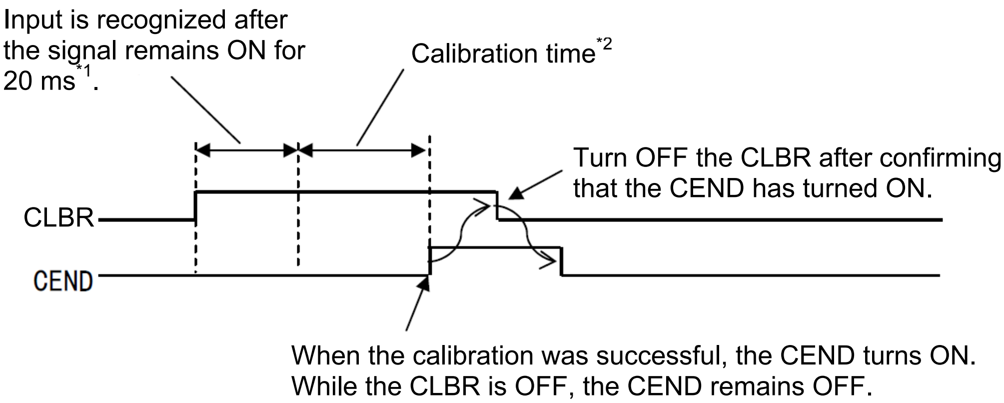
(3) Calibration procedure

[1] Stop the actuator operation. (The load cell cannot be calibrated while the actuator is performing any axis operation or push-motion operation or being paused, in which case 0E1 (load cell calibration error) alarm generates.)

[2] Turn this signal ON and keep it ON for at least 20 ms.

[3] When the calibration is complete, the calibration complete signal (CEND of device status register 1 explained in 4.3.2 (4)) turns ON. After confirming that the CEND has turned ON, turn OFF the CLBR.
If the calibration was unsuccessful, a 0E1 (load cell calibration error) alarm generates.

⚠ Caution: Normal operation commands are not accepted while the CLBR is ON.



*1 If the CLBR is turned OFF during this period, calibration will not be performed because the signal is not yet recognized as having been input.
*2 If the CLBR is turned OFF during this period, an alarm will generate.

(4) Response

A response message to be sent following a successful change should be the same as the query.
If any invalid data has been sent, an exception response (refer to 7) will be returned or no response will be returned at all.

(5) Example of use

Calibrate the dedicated load cell connected to controller axis 0.
Query 01 05 04 26 FF 00 D1

Field	ASCII mode 8-bit data	Converted ASCII code data [H]
Start	‘:’	3A
Slave address [H]	‘0’, ‘1’	3031
Function code [H]	‘0’, ‘5’	3035
Start address [H]	‘0’, ‘4’, ‘2’, ‘6’	30343236
Changed data [H]	‘F, , ‘F’; ‘0’, ‘0’	46463030
Error check [H]	‘D’, ‘1’ (In accordance with LRC calculation)	4431
End	‘CR’, ‘LF’	0D0A

If the change is successful, the response message will be the same as the query.

6.5.16 PIO/Modbus Switching Setting <<PMSL>>

(1) Function

PIO external command signals can be enabled or disabled.

(2) Query format

Field	Number of characters	ASCII mode character string (fixed)	Remarks
Header	1	‘:’	
Slave address [H]	2	Arbitrary	Axis number + 1 (01_H to 10_H) 00_H when broadcast is specified
Function code [H]	2	‘0’, ‘5’	Write to a single coil DO.
Start address [H]	4	‘0’, ‘4’, ‘2’, ‘7’	PIO/Modbus switching setting
Changed data [H]	4	Arbitrary	*1 Enable Modbus commands: ‘F’, ‘F’, ‘0’, ‘0’ Disable Modbus commands: ‘0’, ‘0’, ‘0’, ‘0’
Error check [H]	2	LRC calculation result	
Trailer	2	‘CR’, ‘LF’	
Total number of bytes	17		

*1 • Enable Modbus commands (ON) (disable PIO command): $FF00_H$
Operation via PIO signals is not possible.
• Disable Modbus commands (OFF) (enable PIO command): 0000_H
Operation via external PIO signals is possible.

Supplement: If the Modbus command is enabled, the PIO status at change is maintained. If the Modbus command is switched to disabled, the operation status changes according to the current PIO status. Note that even if the status of signals that operate via edge detection has been changed, edge detection is ignored.

(3) Precaution

- On a model equipped with an operation mode switch, “Enable PIO commands” will be specified when the switch is set to the AUTO mode, or “Disable PIO commands” will be specified when the switch is set to the MANU mode.
- On a non-PIO model, the default setting is “Disable PIO commands.”
- If IAI’s tool (teaching pendant or PC software) is connected, “Teaching modes 1, 2” and “Monitor modes 1, 2” are available as tool modes. The correspondence between these modes and PIO enable/disable specifications are as follows:
  “Monitor modes 1, 2” → “Enable PIO commands”
  “Teaching modes 1, 2” → “Disable PIO commands”

(4) Response

If the change is successful, the response message will be the same as the query.
If invalid data is sent, an exception response (refer to section 7) will be returned, or no response will be returned.

(5) Query sample

A sample query that enables the Modbus command of the operation of a controller of axis No. 0 is shown below.
Query: 01 05 04 27 FF 00 D0

Field	ASCII mode 8-bit data	Converted ASCII code data [H]
Start	‘:’	3A
Slave address [H]	‘0’, ‘1’	3031
Function code [H]	‘0’, ‘5’	3035
Start address [H]	‘0’, ‘4’, ‘2’, ‘7’	30343237
Changed data [H]	‘F’, ‘F’, ‘0’, ‘0’	46463030
Error check [H]	‘D’, ‘0’ (in accordance with LRC calculation)	4430
End	‘CR’, ‘LF’	0D0A

If the change is successful, the response message will be the same as the query.

6.5.17 Deceleration Stop <<STOP>>

(1) Function

The actuator will start decelerating to a stop upon detection of the deceleration stop command (write $FF00_H$) rising edge.

(2) Query format

Field	Number of characters	ASCII mode character string (fixed)	Remarks
Header	1	‘:’	
Slave address [H]	2	Arbitrary	Axis number + 1 (01_H to 10_H) 00_H when broadcast is specified
Function code [H]	2	‘0’, ‘5’	Write to a single coil DO
Start address [H]	4	‘0’, ‘4’, ‘2’, ‘C’	Deceleration stop setting
Changed data [H]	4	Arbitrary	Deceleration stop command (ON): ‘F’, ‘F’, ‘0’, ‘0’ * The controller automatically resets the value to 0000_H.
Error check [H]	2	LRC calculation result	
Trailer	2	‘CR’, ‘LF’	
Total number of bytes	17		

(3) Response

If the change is successful, the response message will be the same as the query.
If invalid data is sent, an exception response (refer to section 7) will be returned, or no response will be returned.

(4) Sample query

A sample query that sends the deceleration stop command to a controller of axis No. 0 is shown below.

Query: 01 05 04 2C FF 00 CB

Field	ASCII mode 8-bit data	Converted ASCII code data [H]
Start	‘:’	3A
Slave address [H]	‘0’, ‘1’	3031
Function code [H]	‘0’, ‘5’	3035
Start address [H]	‘0’, ‘4’, ‘2’, ‘C’	30343243
Changed data [H]	‘F’, ‘F’, ‘0’, ‘0’	46463030
Error check [H]	‘C’, ‘B’ (In accordance with LRC calculation)	4342
End	‘CR’, ‘LF’	0D0A

If the change is successful, the response message will be the same as the query.

6.6 Control Information Direct Writing (Queries Using Code 06)

6.6.1 Writing to Registers

*) Please refer to "6.2 ASCII Code Table."

(1) Function

These queries change (write) data in registers of a slave.
In case of broadcast, data of registers of the same address of all slaves is changed.
[Refer to the details of device controller register 1 in 4.3.2 (1).]
[Refer to the details of device controller register 2 in 4.3.2 (2).]
[Refer to the details of the position number specification register and position movement specification register in 4.3.2 (3).]

(2) Start address list

Address	Symbol	Name	Byte
0D00	DRG1	Device control register 1	2
0D01	DRG2	Device control register 2	2
0D03	POSR	Position number specification register	2
9800	POSR	Position movement specification register	2

The registers above are control command registers. The bits of these registers are assigned to input ports by PIO patterns when "PIO/Modbus Switch Status (PMSS) [refer to 6.4.8] is set to "disable Modbus commands (enable PIO commands). These registers can be rewritten when the Modbus commands are enabled (PIO commands are disabled).

(3) Query format

Specify the address and data of the register whose data is to be changed in the query message.
Data to be changed shall be specified as 16-bit data in the changed data area of the query.

Field	Number of characters (Number of bytes)	ASCII mode fixed character string	Remarks
Header	1	‘:’	
Slave address [H]	2	Arbitrary	Axis number + 1 (01_H to 10_H) 00_H when broadcast is specified
Function code [H]	2	‘0’, ‘6’	Writing to registers
Start address [H]	4	Arbitrary	Refer to 6.6.1 (2), “Start address list.”
Changed data [H]	4		4.3.2 (1) to 4.3.2. (3), Refer to “List of changed data.”
Error check [H]	2	LRC calculation result	
Trailer	2	‘CR’, ‘LF’	
Total number of bytes	17		

(4) Response

If the change is successful, the response message will be the same as the query.
If invalid data is sent, an exception response (refer to section 7) will be returned, or no response will be returned.

(5) Query sample

Examples of different operations are shown in [1] to [3] below.

[1] A sample query that turns the servo of a controller of axis No. 0 on and then executes home return operation is performed.

Query

First time 01 06 0D 00 10 00 DC --- Servo ON

Second time 01 06 0D 00 10 10 CC --- Home return

Field	ASCII mode 8-bit data	Converted ASCII code data [H]
Start	‘:’	3A
Slave address [H]	‘0’, ‘1’	3031
Function code [H]	‘0’, ‘6’	3036
Start address [H]	‘0’, ‘D’, ‘0’, ‘0’	30443030
Changed data [H]	First time: ‘1’, ‘0’, ‘0’, ‘0’ Second time: ‘1’, ‘0’, ‘1, ‘0’	31303030 31303130
Error check [H]	First time: ‘D’, ‘C’ (in accordance with CRC calculation) Second time: ‘C’, ‘C’ (in accordance with CRC calculation)	4443 4343
End	‘CR’, ‘LF’	0D0A

*1 Home return is not performed even if 1010_H is sent to change the data while the servo is OFF (refer to the timing chart at startup of each RC controller).

*2 To keep the previous status, send the previous status even if there is no change. As in the example above, keep the servo ON bit as 1 at home return as well.

If the change is successful, the response message will be the same as the query.

[2] Move to position No. 1 using the position movement specification register (address 9800_H). Before this operation, perform the operation in example [1] above to complete a home return. Query (Silent intervals are inserted before and after the query.)
01 06 98 00 00 01 60

Field	ASCII mode 8-bit data	Converted ASCII code data [H]
Start	‘:’	3A
Slave address [H]	‘0’; ‘1’	3031
Function code [H]	‘0, ’ ‘6’	3036
Start address [H]	‘9’, ‘8’, ‘0’, ‘0’	39383030
Changed data [H]	‘0’, ‘0’, ‘0’, ‘1’	30303031
Error check [H]	‘6’, ‘0’ (in accordance with CRC calculation)	3630
End	‘CR’, ‘LF’	0D0A

* As soon as a position number is written to this register, the actuator starts moving. The CSTR (start signal) is not required.

A response message to be sent following a successful change should be the same as the query.

[3] Move to position No. 1 using the position number specification register (address $0D03_H$).
Before this operation, perform the operation in example [1] above to complete a home return.
Query (Silent intervals are inserted before and after the query.)

First time 01 06 0D 03 00 01 E8 --- Specify position No. 1
Second time 01 06 0D 00 10 00 DC--- Turn OFF the CSTR (start signal)
Third time 01 06 0D 00 10 08 D4--- Turn ON the CSTR (start signal)

Field	ASCII mode 8-bit data	Converted ASCII code data [H]
Start	':'	3A
Slave address [H]	'0', '1'	3031
Function code [H]	'0', '6'	3036
Start address [H]	First time: '0', 'D', '0', '3' Second time: '0', 'D', '0', '0' Third time: '0', 'D', '0', '0'	30443033 30443030 30443030
Changed data [H]	First time: '0', '0', '0', '1' Second time: '1', '0', '0', '0' Third time: '1', '0', '0', '8'	30303031 31303030 31303038
Error check [H]	First time: 'E', '8'(in accordance with CRC calculation) Second time: 'D', 'C'(in accordance with CRC calculation) Third time: 'D', '4'(in accordance with CRC calculation)	4538 4443 4434
End	'CR', 'LF'	0D0A

* To keep the previous status, send the previous status even if there is no change. As in the example above, keep the servo ON bit as 1 at home return as well.

If the change is successful, the response message will be the same as the query.

6.7 Positioning Data Direct Writing (Queries Using Code 10)

6.7.1 Numerical Value Movement Command

*) Please refer to "6.2 ASCII Code Table."

(1) Function

Specify the target position in PTP positioning operation using absolute coordinates. It is possible to command the actuator to move via numerical values by writing directly to the group of registers at addresses from 9900_H to 9908_H (can be set in one message).

Values of all registers, other than the control flag specification register (address: 9908_H), will become effective once the values are sent after the power is supplied. If there is no need to change the target position, positioning band, speed, acceleration/deceleration, push-current limiting value and control specification, therefore, each subsequent numerical movement command can be issued simply by writing a desired register that can effect an actual movement command based on changing of the applicable register alone (refer to "Start address list").

(2) Start address list

This group of registers is used to move the actuator by specifying the target position coordinates, positioning band, speed, push-current limiting value, control flag specification and so on as numerical values.

Data of start addresses in the list (6 registers in total) can be changed with one transmission.

Address [H]	Symbol	Name	Sign	Able to effect an actual movement command by changing the applicable register alone	Register size	Byte size	Unit
9900	PCMD	Target position specification register	○	○	2	4	0.01 mm
9902	INP	Positioning band specification register		x	2	4	0.01 mm
9904	VCMD	Speed specification register		○	2	4	0.01 mm/sec
9906	ACMD	Acceleration/deceleration specification register		○	1	2	0.01 G
9907	PPOW	Push-current limiting value specification register		○	1	2	%
9908	CTLF	Control flag specification register		x Initialization after each movement	1	2	-

(3) Query format

1 register = 2 bytes = 16-bit data

Field	Number of characters (number of bytes)	ASCII mode fixed character string	Remarks
Header	1	‘:’	
Slave address [H]	2	Arbitrary	Axis number + 1 (01_H to 10_H) 00_H if broadcast is specified
Function code [H]	2	‘1’, ‘0’	Numerical value specification
Start address [H]	4	Arbitrary	Refer to 6.7.1. (2), “Start address list”
Number of registers [H]	4	Arbitrary	Refer to 6.7.1 (2), “Start address list”
Number of bytes [H]	2	In accordance with the number of registers above	Enter the value twice as large as the number of registers specified above
Changed data 1 [H]	4		Refer to 6.7.1 (2), “Start address list”
Changed data 2 [H]	4		Refer to 6.7.1 (2), “Start address list”
Changed data 3 [H]	4		Refer to 6.7.1 (2), “Start address list”
:	:		
Error check [H]	2	LRC calculation result	
Trailer	2	‘CR’, ‘LF’	
Total number of bytes	Up to 256		

(4) Response format

When normally changed, the response message responds with a copy of the query message excluding the number of bytes and changed data.

Field	Number of characters (number of bytes)	ASCII mode fixed character string	Remarks
Header	1	‘:’	
Slave address [H]	2	Arbitrary	Axis number + 1 (01_H to 10_H) 00_H if broadcast is specified
Function code [H]	2	‘1’, ‘0’	Numerical value specification
Start address [H]	4	Arbitrary	Refer to 6.7.1 (2), “Start address list”
Number of registers [H]	4	Arbitrary	Refer to 6.7.1 (2), “Start address list”
Error check [H]	2	LRC calculation result	
Trailer	2	‘CR’, ‘LF’	
Total number of bytes	Up to 256		

(5) Detailed explanation of registers

- Target position specification register (PCMD)

  This register specifies the target position in PTP positioning operation using absolute coordinates. The value of this register is set in units of 0.01 mm in a range of –999999 to 999999 (FFF0BDC1(Note) ~ $00F423F_H$). When a position is specified using absolute coordinates and the specified value exceeds the soft limit set by a parameter, an alarm will generate the moment a movement start command is issued. The actuator will start moving when the lower word of this register (symbol: PCMD, address: 9900H) is rewritten. In other words, **a numerical movement command can be issued simply by writing a target position in this register.**

  (Note) To set a negative value, use a two's complement.

- Positioning band register (INP)

  This register is used in two different ways depending on the type of operation.
  The first way is the normal positioning operation, where it specifies the allowable difference between the target position and current position to be used in the detection of position complete.
  The second way is the push-motion operation, where it specifies the push-motion band. The value of this register is set in units of 0.01 mm in a range of 1 to 999999 (1~$000F423F_H$). Whether the normal operation or push-motion operation is specified by the applicable bit in the control flag specification register as explained later.
  Changing this register alone will not start actuator movement.

- Speed specification register (VCMD)

  This register specifies the moving speed. The value of this register is set in units of 0.01 mm/sec in a range of 1 to 999999 (1~$000F423F_H$). If the specified value exceeds the maximum speed set by a parameter, an alarm will generate the moment a movement start command is issued.
  The actuator will start moving when this lower word of this register is rewritten. In other words, the speed can be changed while the actuator is moving, simply by rewriting this register.

■ Acceleration/deceleration specification register (ACMD)
This register specifies the acceleration or deceleration. The value of this register is set in units of 0.01 G in a range of 1 to 300 (1~012C$_H$). If the specified value exceeds the maximum acceleration or deceleration set by a parameter, an alarm will generate the moment a movement start command is issued.
The actuator will start moving when this register is rewritten. In other words, the acceleration/deceleration can be changed while the actuator is moving, simply by rewriting this register.

■ Push-current limiting value (PPOW)
Set the current limit during push-motion operation in PPOW. Set an appropriate value by referring to the table below.

Actuator model name	Pushable range [%]	Settable range (input value) [H]
Actuator other than RCS2-RA13R	20~70(Note)	33~B2
RCS2-RA13R	20~200	33~1FE

(Note) The setting ranges may vary depending on the actuator.
[For details, refer to the IAI catalog or operation manual of each actuator.]

The actuator will start moving when this register is rewritten. In other words, the current limiting value can be changed during push-motion operation simply by rewriting this register.
Sample push-motion current setting
● When setting the current to 20%
255 (100%) x 0.2 (20%)= 51 → 33$_H$ (converted to a hexadecimal value)

■ Control Flag Specification Register (CTLF)

Set the method of operation.

If push-motion operation or incremental operation (pitch feed) is selected, set this register every time a movement command is issued. (This is because the register will be overwritten with the default value every time the actuator moves.)

CTLF bit structure

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
MSB	–	–	NTC1	NTC0	–	–	–	–	MOD1	MOD0	GSL1	GSL0	INC	DIR	PUSH	–	LSB

Bit 1 (PUSH) = 0: Normal operation (default)
1: Push-motion operation

Bit 2 (DIR) = 0: The direction of push-motion operation after completion of approach is defined as the forward direction (default).
1: The direction of push-motion operation after completion of approach is defined as the reverse direction.
This bit is used to calculate the direction of final stop position from PCMD. If this bit is set incorrectly, therefore, the target position will deviate from the specified position by a distance corresponding to "2 x INP, " as shown in Fig. 5.3 below.
If bit 1 is set to 0, the setting of this bit is invalid.

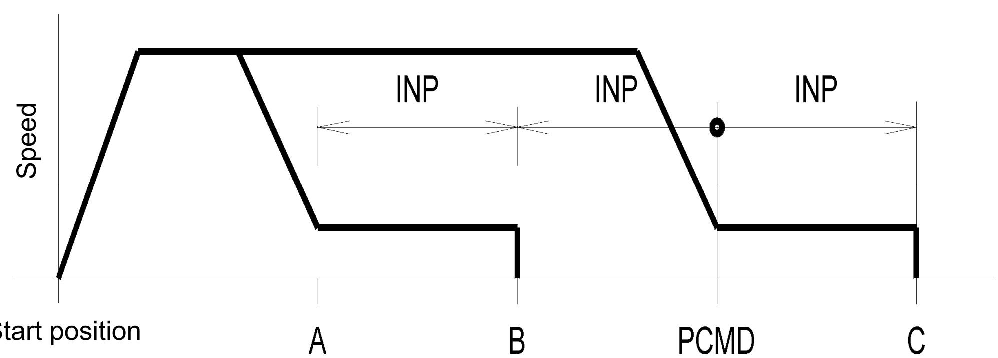


Fig. 6.3 Operating Direction in Push-motion Operation

Bit 3 (INC) = 0: Normal operation (default)
1: Incremental operation (pitch feed)

Setting this bit to 1 will enable the actuator to operate relative to the current position.
In this operation, the actuator behaves differently between normal operation and push-motion operation (CTLF bit 1). While the travel is calculated with respect to the target position (PCMD) in normal operation, it is calculated relative to the current position in push-motion operation (when bit 1 = 1).
Here, since relative coordinate calculation involves adding up pulses in mm, followed by conversion, unlike a calculation method involving addition after pulse conversion, **"repeated relative movements will not cause position deviation as a result of cumulative errors corresponding to fraction pulses that are not divisible with certain lead settings".**

Bit 4 (GSL0), 5 (GSL1) = Refer to the table below. (These bits can be set only on SCON-CA controllers.)

GSL1	GSL0	Function
0	0	Select parameter set 0 (default).
0	1	Select parameter set 1
1	0	Select parameter set 2
1	1	Select parameter set 3

You can register a maximum of four servo gain parameter sets consisting of six parameters and move the actuator to each position by selecting a different parameter set every time. [For details, refer to the operation manual for your controller.]

Bit 6 (MOD0), 7 (MOD1) = Refer to the table below. These bits cannot be set on PCON-* and ERC2 controllers.)

MOD1	MOD0	Function
0	0	Trapezoid pattern (default)
0	1	S-motion
1	0	Primary delay filter
1	1	Cannot be used.

These signals are used to select the acceleration/deceleration pattern characteristics. Set one of the patterns before issuing an actuator movement command. [For details, refer to the operation manual for your controller.]

Bit 12 (NTC0), 13 (NTC1) = Refer to the table below. (These bits can be set only on SCON-CA controllers.)

NTC1	NTC0	Function
0	0	Do not use vibration control (default).
0	1	Select parameter set 1
1	0	Select parameter set 2
1	1	Select parameter set 3

When vibration control is used, you can register a maximum of three parameter sets and move the actuator to each position by selecting a different parameter set every time. [For details, refer to the operation manual for your controller.]

(6) Example of use

Examples of different operations are shown in [1] to [7] below.

[1] Move by changing the target position. (All data other than the target position are the default values of their respective parameters.)

Conditions: The operation conditions conform to the default speed, default acceleration/deceleration and default positioning band set by the controller's user parameters. Only the target position is changed to move the actuator.

Supplement: Controller's user parameters

- Default speed (parameter No. 8) → Maximum speed of the applicable actuator as specified in the catalog
- Default acceleration/deceleration (parameter No. 9) → Rated acceleration of the applicable actuator as specified in the catalog
- Default positioning band (parameter No. 10) → Default value = 0.1 mm

Write the target position specification register (9900_H) (Example 1)

↓

Start of movement

(Example 1) Target position: 50 mm

Target position [mm]	Positioning band [mm]	Speed [mm/s]	Acceleration/ deceleration [G]	Push [%]	Control flag
50	Need not be set.				

■Query : 01 10 9900 0002 04 0000 1388 79[CR][LF]

■Response : 01 10 9900 0002 54[CR][LF]

--- The query message is copied, except for the number of bytes and new data, and returned as a response.

■ Breakdown of Query Message

Field	ASCII mode fixed character string	Converted ASCII code data [H]	Remarks
Header	':'	3A	
Slave address	'0', '1'	3031	Axis number + 1
Function code	'1', '0'	3130	
Start address	'9', '9', '0', '0'	39393030	The start address is the target position specification register 9900_H.
Number of registers	'0', '0', '0', '2'	30303032	Specify 9900_H through 9901_H as the addresses to be written.
Number of bytes	'0', '4'	3034	2 (registers) × 2 = 4 (bytes) → 4_H
	'0', '0', '0', '0'	30303030	
Changed data 2 [H]	'0', '7', 'D', '0'	30374430	50 [mm] x 100 = 5000 → 1388_H
Error check	'7', '9'	3739	LRC checksum calculation result → 79_H
Trailer	'CR', 'LF'	0D0A	
Total number of bytes	27		

[2] Move by changing the target position. (as well as data other than the target position).

Conditions: Want to move the actuator by changing the target position, speed and acceleration/deceleration every time.

Write the target position specification register (9900_H) through acceleration/deceleration specification register (9906_H)(Example 2)

↓

Start of movement

(Example 2) Target position: 50 mm

Target position [mm]	Positioning band [mm]	Speed [mm/s]	Acceleration/ deceleration [G]	Push [%]	Control flag
50	0.1	100	0.3	Need not be set.	

■ Query : 01 10 9900 0007 0E 0000 1388 0000 000A 0000 2710 001E 47[CR][LF]

■ Response : 01 10 9900 0007 4F[CR][LF]

--- The query message is copied, except for the number of bytes and new data, and returned as a response.

■ Breakdown of Query Message

Field	ASCII mode fixed character string	Converted ASCII code data [H]	Remarks
Header	':'	3A	
Slave address	'0', '1'	3031	Axis number + 1
Function code	'1', '0'	3130	
Start address	'9', '9', '0', '0'	39393030	The start address is the target position specification register 9900_H.
Number of registers	'0', '0', '0', '7'	30303039	Specify 9900_H through 9906_H as the addresses to be written.
Number of bytes	'0', 'E'	3132	7 (registers) × 2 = 14 (bytes) → E_H
New data 1, 2 (target position) Input unit (0.01 mm)	'0', '0', '0', '0'	30303030	All upper bits of the 32-bit data are 0.
	'1', '3', '8', '8'	31333838	50 [mm] × 100 = 5000 → 1388_H
New data 3, 4 (Positioning band) Input unit (0.01 mm)	'0', '0', '0', '0'	30303030	All upper bits of the 32-bit data are 0.
	'0', '0', '0', 'A'	30303041	0.1 [mm] × 100 = 10 → $000A_H$
New data 5, 6 (Speed) Input unit (0.01 mm/sec)	'0', '0', '0', '0'	30303030	All upper bits of the 32-bit data are 0.
	'2', '7', '1', '0'	32373130	100 [mm/s] ×100 = 10000 → 2710_H
New data 7 (Acceleration/deceleration) Input unit (0.01 G)	'0', '0', '1', 'E'	30303145	0.3 [G] × 100 =30 → $001E_H$
Error check	'4', '7'	3437	LCRC checksum calculation result → 47_H
Trailer	'CR', 'LF'	0D0A	
Total number of bytes	47		

[3] Change the speed while the actuator is moving.

Conditions: Change the target position, speed and acceleration/deceleration each time the actuator is moved, with the actuator speed changed at a given timing during movement.

Write the target position specification register (9900_H) through acceleration/deceleration specification register (9906_H)(Example 2)

↓

Start of movement

↓

Write the speed specification registers (9904_H and 9905_H)(Example 3)

↓

The actuator continues with the normal operation at the new speed

(Example 3) Change the speed from 100 mm/s to 50 mm/s while the actuator is moving.

Target position [mm]	Positioning band [mm]	Speed [mm/s]	Acceleration/ deceleration [G]	Push [%]	Control flag
50	0.1	100 → 50	0.3	Need not be set.	

(1) Start the movement at a speed of 100 mm/s. [Refer to Example [2], "Move by changing the speed" above.]
- Query : 01 10 9900 0007 0E 0000 1388 0000 000A 0000 2710 001E 47[CR][LF]
- Response : 01 10 9900 0007 4F[CR][LF]

(2) Change the speed to 50 mm/s.
- Query : 01 10 9904 0002 04 0000 1388 B1[CR][LF]
- Response : 01 10 9904 0002 50[CR][LF]
  --- The query message is copied, except for the number of bytes and new data, and returned as a response.

■ Breakdown of Query Message (Change the speed to 50 mm/s. [Refer to the above example for the query message used to start the movement at 100 mm/s.])

Field	ASCII mode fixed character string	Converted ASCII code data [H]	Remarks
Header	':'	3A	
Slave address	'0', '1'	3031	Axis number + 1
Function code	'1', '0'	3130	
Start address	'9', '9', '0', '4'	39393034	The start address is the target position specification register 9904_H.
Number of registers	'0', '0', '0', '2'	30303032	Specify 9904_H through 9905_H as the addresses to be written.
Number of bytes	'0', '4'	3034	2 (registers) × 2 = 4 (bytes) → 4_H
Changed data 5, 6 [H] Input unit (0.01 mm/s)	'0', '0', '0', '0'	30303030	All upper bits of the 32-bit data are 0.
	'1', '3', '8', '8'	31333838	50 [mm] × 100 = 5000 → 1388_H
Error check	'B', '1'	4231	LRC check calculation result → $B1_H$
Trailer	'CR', 'LF'	0D0A	
Total number of bytes	27		

[4] Move in the incremental (pitch feed) mode.

Conditions: The operation conditions conform to the default speed, default acceleration/deceleration and default positioning band set by the controller's user parameters. Only the pitch width is changed to move the actuator.

Write the target position specification register (9900_H) through control flag specification register (9908_H: Incremental setting) (Example 4)

↓

Start of movement

Supplement: Addresses 9900_H and 9908_H alone cannot be changed in a single data transmission. Since all addresses are sequential, send two messages if 9900_H and 9908_H alone are changed.
If you want to send only one message, write all addresses from 9900_H to 9908_H.

(Example 4) Move in the incremental mode by setting the pitch to 10 mm.

Pitch [mm]	Positioning band [mm]	Speed [mm/s]	Acceleration/ deceleration [G]	Push [%]	Control flag
10	0.1	100	0.3	0	Incremental (bit3 = 1)

- Query: 01 10 9900 0009 12 0000 03E8 0000 000A 0000 2710 001E 0000 0008 E9[CR][LF]
- Response: 01 10 9900 0009 4D[CR][LF]
  - -- The query message is copied, except for the number of bytes and new data, and returned as a response.

■ Breakdown of Query Message

Field	ASCII mode fixed character string	Converted ASCII code data [H]	Remarks
Header	‘:’	3A	
Slave address	‘0’, ‘1’	3031	Axis No. 0 + 1
Function code	‘1’, ‘0’	3130	
Start address	‘9’, ‘9’, ‘0’, ‘0’	39393030	The start address is the target position specification register 9900_H.
Number of registers	‘0’, ‘0’, ‘0’, ‘9’	30303039	Specify 9900_H through 9908_H as the addresses to be written.
Number of bytes	‘1’, ‘2’	3132	9 (registers) x 2 = 18 (bytes) → 12_H
Changed data 1, 2 (target position) Input unit (0.01 mm)	‘0’, ‘0’, ‘0’, ‘0’	30303030	All upper bits of the 32-bit data are 0.
	‘0’, ‘3’, ‘E’, ‘8’	30334538	10 [mm] × 100 = 1000 → $03E8_H$
Changed data 3, 4 (positioning band) Input unit (0.01 mm)	‘0’, ‘0’, ‘0’, ‘0’	30303030	All upper bits of the 32-bit data are 0.
	‘0’, ‘0’, ‘0’, ‘A’	30303041	0.1 [mm] × 100 = 10 → $000A_H$
Changed data 5, 6 (speed) Input unit (0.01 mm/sec)	‘0’, ‘0’, ‘0’, ‘0’	30303030	All upper bits of the 32-bit data are 0.
	‘2’, ‘7’, ‘1’, ‘0’	32373130	100 [mm/s] × 100 = 10000 → 2710_H
Changed data 7 (acceleration/deceleration) Input unit (0.01 G)	‘0’, ‘0’, ‘1’, ‘E’	30303145	0.3 [G] × 100 = 30 → $001E_H$
Changed data 8 (push) Input unit (%)	‘0’, ‘0’, ‘0’, ‘0’	30303030	0 [%] → 0_H
Changed data 9 (control flag)	‘0’, ‘0’, ‘0’, ‘8’	30303038	(Incremental setting) 1000b → 0008_H
Error check	‘E’, ‘9’	4539	LRC check calculation result → $E9_H$
Trailer	‘CR’, ‘LF’	0D0A	
Total number of bytes	55		

[5] Change the speed during incremental movement (pitch feed).

Conditions: Change the target position, speed and acceleration/deceleration each time the actuator is moved, with the positioning band changed at a given timing during movement.

Write the target position specification register (9900_H) through control flag specification register (9908_H: Incremental setting) (Example 4)

↓

Start of incremental movement

↓

Write the speed specification register (9904_H) through control flag specification register (9908_H: Incremental setting) (Example 5).

↓

The actuator continues with the incremental movement at the new speed.

Supplement: After the control flag specification register (9908_H) is set, the register will return to the default value (0_H: Normal movement) once the actuator starts moving. Accordingly, you must set the control flag specification register (9908_H) and send it again if another incremental or push-motion operation is to be performed.

(Example 5) Change the speed from 100 mm/s to 50 mm/s while the actuator is moving.

Pitch [mm]	Positioning band [mm]	Speed [mm/s]	Acceleration/ deceleration [G]	Push [%]	Control flag
10	0.1	100 → 50	0.3	0	Incremental (bit3 = 1)

(1) Start moving at a speed of 100 mm/s. [Refer to Example [4], “Moving in the incremental (pitch feed) mode” above.]

- Query : 01 10 9900 0009 12 0000 03E8 0000 000A 0000 2710 001E 0000 0008 E9[CR][LF]
- Response : 01 10 9900 0009 4D[CR][LF]

(2) Change the speed to 50 mm/s.

- Query : 01 10 9904 0005 0A 0000 1388 001E 0000 0008 82[CR][LF]
- Response: 01 10 9904 0005 4D[CR][LF]

--- The query message is copied, except for the number of bytes and new data, and returned as a response.

■ Breakdown of Query Message (Change the speed to 50 mm/s. [Refer to the above example for the query message used to start the movement at 100 mm/s.])

Field	ASCII mode fixed character string	Converted ASCII code data [H]	Remarks
Header	‘:’	3A	
Slave address	‘0’, ‘1’	3031	Axis No. 0 + 1
Function code	‘1’, ‘0’	3130	
Start address	‘9’, ‘9’, ‘0’, ‘4’	39393034	The start address is the target position specification register 9904_H.
Number of registers	‘0’, ‘0’, ‘0’, ‘5’	30303032	Specify 9904_H through 9908_H as the addresses to be written.
Number of bytes	‘0’, ‘A’	3034	5 (registers) x 2 = 10(bytes) → A_H
Changed data 1, 2 (target position) Input unit (0.01 mm)	‘0’, ‘0’, ‘0’, ‘0’	30303030	All upper bits of the 32-bit data are 0.
	‘1’, ‘3’, ‘8’, ‘8’	31333838	50 [mm/s] × 100 = 5000 → 1388_H
Changed data 7 (acceleration/deceleration) Input unit (0.01 G)	‘0’, ‘0’, ‘1’, ‘E’	30303145	0.3 [G] × 100 = 30 → $001E_H$
Changed data 8 (push) Input unit (%)	‘0’, ‘0’, ‘0’, ‘0’	30303030	0 [%] → 0_H
Changed data 9 (control flag)	‘0’, ‘0’, ‘0’, ‘8’	30303038	(Incremental setting) 1000b → 0008_H
Error check	‘8’, ‘2’	3832	LRC check calculation result → 82_H
Trailer	‘CR’, ‘LF’	0D0A	
Total number of bytes	39		

[6] Perform a push-motion operation. (changing pushing force during push-operation)

Conditions: Perform push-motion operation by changing the push force at a desired timing while the actuator is pushing the work part.

Write the target position specification register (9900_H) through control flag specification register (9908_H: Push-motion setting) (Example 6)

↓

Start push-motion operation

↓

Write the push-current limit specification register (9907_H) through control flag specification register (9908_H: Push-motion setting) (Example 7)

↓

The actuator continues with the push-motion operation with the new push force

(Example 6) Perform a push-motion operation for 20 mm from the 50-mm position at a current-limiting value of 70%.

Target position [mm]	Positioning band [mm]	Speed [mm/s]	Acceleration/ deceleration [G]	Push [%]	Control flag
50	20	100	0.3	70	Push-motion operation (bit1 = 1, bit2 = 0, 1)

- Query: 01 10 9900 0009 12 0000 1388 0000 07D0 0000 2710 001E 00B2 0006 BC[CR][LF]
- Response: 01 10 9900 0009 4D[CR][LF]

  --- The query message is copied, except for the number of bytes and new data, and returned as a response.

■ Breakdown of Query Message

Field	ASCII mode fixed character string	Converted ASCII code data [H]	Remarks
Header	‘:’	3A	
Slave address	‘0’, ‘1’	3031	Axis No. 0 + 1
Function code	‘1’, ‘0’	3130	
Start address	‘9’, ‘9’, ‘0’, ‘0’	39393030	The start address is the target position specification register 9900_H.
Number of registers	‘0’, ‘0’, ‘0’, ‘9’	30303039	Specify 9900_H through 9908_H as the addresses to be written.
Number of bytes	‘1’, ‘2’	3132	9 (registers) x 2 = 18 (bytes) → 12_H
New data 1, 2 (target position) Input unit (0.01 mm)	‘0’, ‘0’, ‘0’, ‘0’	30303030	All upper bits of the 32-bit data are 0.
	‘1’, ‘3’, ‘8’, ‘8’	31333838	50 [mm] × 100 = 5000 → 1388
New data 3, 4 (positioning band) Input unit (0.01 mm)	‘0’, ‘0’, ‘0’, ‘0’	30303030	All upper bits of the 32-bit data are 0.
	‘0’, ‘7’, ‘D’, ‘0’	30374430	20 [mm] × 100 = 2000 → $07D0_H$
New data 5, 6 (speed) Input unit (0.01 mm/sec)	‘0’, ‘0’, ‘0’, ‘0’	30303030	All upper bits of the 32-bit data are 0.
	‘2’, ‘7’, ‘1’, ‘0’	32373130	100 [mm/s] × 100 = 10000 → 2710_H
New data 7 (acceleration/deceleration) Input unit (0.01 G)	‘0’, ‘0’, ‘1’, ‘E’	30303145	0.3 [G] × 100 = 30 → $001E_H$
New data 8 (push) Input unit (%)	‘0’, ‘0’, ‘B’, ‘2’	30304232	70 [%] → $B2_H$
New data 9 (control flag)	‘0’, ‘0’, ‘0’, ‘6’	30303036	(Push setting) 1000b → 0006_H
Error check	‘B’, ‘C’	4243	LRC check calculation result → BC_H
Trailer	‘CR’, ‘LF’	0D0A	
Total number of bytes	55		

(Example 7) Change the push current limit from 70% to 50% during a push-motion operation.

Target position [mm]	Positioning band [mm]	Speed [mm/s]	Acceleration/ deceleration [G]	Push [%]	Control flag
50	20	100	0.3	70 → 50	Push-motion operation (bit1 = 1, bit2 = 1)

■Query: 01 10 9907 0002 04 007F 0006 C4[CR][LF]
■Response: 01 10 9907 0002 4D[CR][LF]

--- The query message is copied, except for the number of bytes and new data, and returned as a response.

■ Breakdown of Query Message

Field	ASCII mode fixed character string	Converted ASCII code data [H]	Remarks
Header	‘:’	3A	
Slave address	‘0’, ‘1’	3031	Axis No. 0 + 1
Function code	‘1’, ‘0’	3130	
Start address	‘9’, ‘9’, ‘0’, ‘7’	39393037	The start address is the target position specification register 9907_H.
Number of registers	‘0’, ‘0’, ‘0’, ‘2’	30303032	Specify 9907_H through 9908_H as the addresses to be written.
Number of bytes	‘0’, ‘4’	3034	2 (registers) x 2 = 4 (bytes) → 4_H
Changed data 8 (push) Input unit (%)	‘0’, ‘0’, ‘7’, ‘F’	30303746	50 [%] → $7F_H$
Changed data 9 (control flag)	‘0’, ‘0’, ‘0’, ‘6’	30303036	(Push setting) 1000b → 0006_H
Error check	‘C’, ‘4’	4334	LRC check calculation result → $C4_H$
Trailer	‘CR’, ‘LF’	0D0A	
Total number of bytes	27		

[7] Note (changing positioning band during movement)

The positioning band cannot be changed while the actuator is moving.

Conditions: Change the target position, speed and acceleration/deceleration each time the actuator is moved, with the positioning band changed at a given timing during movement.
(Cannot be changed. If data is written, the data is reflected in the next positioning.)

Write the target position specification register (9900_H) through acceleration/deceleration specification register (9906_H)

↓

Start normal operation

↓

Write the positioning band specification registers (9902_H and 9903_H)

↓

The actuator continues with the normal operation at the original positioning band setting

Supplement: Writing the positioning band specification registers alone cannot effect an actual movement command.
Therefore, the data changed by writing the positioning band specification registers (9902_H and 9903_H) will become effective when the next movement command is executed.

6.7.2 Writing Position Table Data

(1) Function

Position table data can be changed using this query.
Each time an address specified in "Start address list" (addresses $+0000_H$ through $+000E_H$) is accessed, the applicable data will be read from the EEPROM in units of position data. After having been written, the data will be stored in the EEPROM again.

* The number of times allowed to write to EEPROM is limited to approximately 100, 000 times due to device restriction. If the position table data is updated frequently, the number of times allowed to write to EEPROM may be exceeded, which may cause failures. Take caution not to cause unexpected loops and similar in logic on the master side.

(2) Start address list

In a query input, each address is calculated using the formula below:
1000_H + (16 x Position number) $_H$ + Address (Offset) $_H$

(Example) Change the speed command register for position No. 200
1000_H + (16 x 200 = 3200) $_H$ + 4_H
= 1000_H + $C80_H$ + 4_H
= $1C84_H$
"1C84" becomes the input value for the start address field of this query.

* The maximum position number varies depending on the controller model and the PIO pattern currently specified.

■ Position data change resisters

Address	Symbol	Name	Sign	Register size	Byte size	Input unit
+0000	PCMD	Target position	○	2	4	0.01 mm
+0002	INP	Positioning band		2	4	0.01 mm
+0004	VCMD	Speed command		2	4	0.01 mm/sec
+0006	ZNMP	Individual zone boundary +	○	2	4	0.01 mm
+0008	ZNLP	Individual zone boundary -	○	2	4	0.01 mm
+000A	ACMD	Acceleration command		1	2	0.01 G
+000B	DCMD	Deceleration command		1	2	0.01 G
+000C	PPOW	Push-current limiting value		1	2	%
+000D	LPOW	Load current threshold		1	2	%
+000E	CTLF	Control flag specification		1	2	

* Addresses starting with "+" indicate offsets.

(3) Query format

1 register = 2 bytes = 16-bit data

Field	ASCII mode fixed character string	Number of characters (Number of bytes)	Remarks
Header	‘:’	1	
Slave address [H]	Arbitrary	2	Axis number + 1 (01_H to 10_H) 00_H when broadcast is specified
Function code [H]	‘1’, ‘0’	2	
Start address [H]	Arbitrary	4	Refer to 6.7.2 (2), “Start address list.”
Number of registers [H]	Arbitrary	4	Refer to 6.7.2 (2), “Start address list.”
Number of bytes [H]	In accordance with the above registers	2	A value corresponding to twice the number of registers specified above is input.
Changed data 1 [H]		4	Refer to 6.7.2 (2), “Start address list.”
Changed data 2 [H]		4	Refer to 6.7.2 (2), “Start address list.”
Changed data 3 [H]		4	Refer to 6.7.2 (2), “Start address list.”
:		:	
Error check [H]	LRC calculation result	2	
Trailer	‘CR’, ‘LF’	2	
Total number of bytes		Up to 256	

(4) Response format

If the change is successful, a response message that is effectively a copy of the query message, except for the byte count and new data, will be returned.

Field	ASCII mode fixed character string	Number of characters (Number of bytes)	Remarks
Header	‘:’	1	
Slave address [H]	Arbitrary	2	Axis number + 1 (01_H to 10_H) 00_H when broadcast is specified
Function code [H]	‘1’, ‘0’	2	
Start address [H]	Arbitrary	4	Refer to 6.7.2 (2), “Start address list.”
Number of registers [H]	Arbitrary	4	Refer to 6.7.2 (2), “Start address list.”
Error check [H]	LRC calculation result	2	
Trailer	‘CR’, ‘LF’	2	
Total number of bytes		Up to 256	

(5) Detailed explanation of registers

- Target Position (PCMD)
This register specifies the target position using absolute coordinates or by an relative distance. The value of this register is set in units of 0.01 mm in a range of -999999 to 999999 (FFF0BDC1(Note)~000F423F$_H$). When a position is specified using absolute coordinates and the specified value exceeds the soft limit set by a parameter, an alarm will generate the moment a movement start command is issued. Specify whether to use absolute coordinates or relative distance using the applicable bit in the control flag specification register as explained later.
(Note) To set a negative value, use a two's complement.

- Positioning band Specification Register (INP)
This register is used in two different ways depending on the type of operation. The first way is the normal positioning operation, where it specifies the allowable difference between the target position and current position to be used in the detection of position complete. The second way is the push-motion operation, where it specifies the push-motion band. The value of this register is set in units of 0.01 mm in a range of 1 to 999999 (1~000F423F$_H$).
Whether the normal operation or push-motion operation is specified by the applicable bit in the control flag specification register as explained later.

- Speed Specification Register (VCMD)
This register specifies the moving speed in positioning. The value of this register is set in units of 0.01 mm/sec in a range of 1 to 999999 (1~000F423F$_H$). If the specified value exceeds the maximum speed set by a parameter, an alarm will generate the moment a movement start command is issued.

- Individual Zone Boundaries ± (ZNMP, ZNLP)
These registers output zone signals that are effective only during positioning, separately from the zone boundaries set by parameters.
Set in ZNMP the positive zone signal output boundary expressed using absolute coordinates, and set the negative zone signal output boundary in ZNLP. The corresponding bit in the zone register remains ON while the current position is within these positive and negative boundaries. The value of this register is set in units of 0.01 mm, and in a range of -999999 to 999999 (FFF0BDC1(Note ~000F423F$_H$) for both registers. However, ZNMP must be greater than ZNLP. Set the same value in both ZNMP and ZNLP to disable the individual zone output.
(Note) To set a negative value, use a two's complement.

- Acceleration specification register (ACMD)
This register specifies the acceleration during positioning. The value of this register is set in units of 0.01 G in a range of 1 to 300 (1~012C$_H$). If the specified value exceeds the maximum acceleration set by a parameter, an alarm will generate the moment a movement start command is issued.

■ Deceleration specification register (ACMD)
This register specifies the deceleration during positioning. The value of this register is set in units of 0.01 G in a range of 1 to 300 ($1\sim012C_H$). If the specified value exceeds the maximum deceleration set by a parameter, an alarm will generate the moment a movement start command is issued.

■ Push-current limiting value (PPOW)
Set the current limit during push-motion operation in PPOW. Set an appropriate value by referring to the table below.

Actuator model name	Pushable range [%]	Settable range (input value) [H]
Actuator other than RCS2-RA13R	20~70(Note)	33~B2
RCS2-RA13R	20~200	33~1FE

(Note) The setting ranges may vary depending on the actuator.
[For details, refer to the IAI catalog or operation manual of each actuator.]

Sample push-motion current setting
● When setting the current to 20%
256 (100%) x 0.2 (20%)= 51.2 → 33_H (converted to a hexadecimal value)

■ Load Output Current Threshold (LPOW)
To perform load output judgment, set the current threshold in LPOW. Set an appropriate value according to the actuator used, just like the push current limit (PPOW). If load output judgment is not performed, set 0.

■ Control Flag Specification Register (CTLF)
[Refer to the control flag specification register in 6.7.1 (5).]

(7) Sample query

A sample query that rewrites all data of position No. 12 of axis No. 0 is shown below.

Axis No. 0

Target position (mm)	Positioning band (mm)	Speed (mm/sec)	Individual zone boundary+ (mm)	Individual zone boundary- (mm)	Acceleration (G)	Deceleration (G)	Push (%)	Threshold	Movement control
100	0.1	200	60	40	0.01	0.3	0	0	Normal movement

■ Query 01 10 10C0 000F 1E 0000 2710 0000 000A 0000 4E20 0000 1770
0000 0FA0 0001 001E 0000 0000 0000 EE[CR][LF]

■ Received response 01 10 10C0 000F 10[CR][LF]

■ Breakdown of Query Message

Field	ASCII mode fixed character string	Converted ASCII code data [H]	Remarks
Header	‘:’	3A	
Slave address	‘0’, ‘1’	3031	Axis No. 0 + 1
Function code	‘1’, ‘0’	3130	
Start address	‘1’, ‘0’, ‘C’, ‘0’	31304330	The start address is the target position specification register $10C0_H$ for position No. 12. *1
Number of registers	‘0’, ‘0’, ‘0’, ‘F’	30303046	Total 15 registers of register symbols PCMD to CTLF are specified to be written.
Number of bytes	‘1’, ‘E’	3145	15 (registers) x 2 = 30 (bytes) → $1E_H$
New data 1, 2 (target position) Input unit (0.01 mm)	‘0’, ‘0’, ‘0’, ‘0’	30303030	All upper bits of the 32-bit data are 0.
	‘2’, ‘7’, ‘1’, ‘0’	32373130	100 (mm) x 100 = 10000 → 2710_H
New data 3, 4 (positioning band) Input unit (0.01 mm)	‘0’, ‘0’, ‘0’, ‘0’	30303030	All upper bits of the 32-bit data are 0.
	‘0’, ‘0’, ‘0’, ‘A’	30303041	0.1 (mm) x 100 = 10 → $000A_H$
New data 5, 6 (speed) Input unit (0.01 mm/sec)	‘0’, ‘0’, ‘0’, ‘0’	30303030	All upper bits of the 32-bit data are 0.
	‘4’, ‘E’, ‘2’, ‘0’	34453230	200 (mm/sec) x 100 = 20000 → $4E20_H$
New data 7, 8 (individual zone boundary +) Input unit (0.01 mm)	‘0’, ‘0’, ‘0’, ‘0’	30303030	All upper bits of the 32-bit data are 0.
	‘1’, ‘7’, ‘7’, ‘0’	31373730	60 (mm) x 100 = 6000 → 1770_H
	‘0’, ‘F’, ‘A’, ‘0’	30464130	40 (mm) x 100 = 4000 → $0FA0_H$

Continue to the next page

Field	ASCII mode fixed character string	Converted ASCII code data [H]	Remarks
Changed data 9, 10 (individual zone boundary -) Input unit (0.01 mm)	‘0’, ‘0’, ‘0’, ‘0’	30303030	All upper bits of the 32-bit data are 0.
	‘0’, ‘F’, ‘A’, ‘0’	30464130	40 (mm) x 100 = 4000 → $0FA0_H$
Changed data 11 (acceleration) Input unit (0.01 G)	‘0’, ‘0’, ‘0’, ‘1’	30303031	0.01 (G) x 100 = 1 → 0001_H
Changed data 12 (deceleration) Input unit (0.01 G)	‘0’, ‘0’, ‘1’, ‘E’	30303145	0.3 (G) x 100 = 30 → $001E_H$
Changed data 13 (push) Input unit [%]	‘0’, ‘0’, ‘0’, ‘0’	30303030	0 (%) → 0_H
Changed data 14 (threshold) Input unit [%]	‘0’, ‘0’, ‘0’, ‘0’	30303030	0 (%) → 0_H
Changed data 15 (control flag)	‘0’, ‘0’, ‘0’, ‘0’	30303030	All bits are 0 in the normal operation mode. 0000_b → 0000_H
Error check	‘E’, ‘E’	4545	LRC check calculation result → EE_H
Trailer	‘CR’, ‘LF’	0D0A	
Total number of bytes	79		

*1) Calculation of start address
In the example, all data of position No. 12 is changed. Accordingly, the target position address of position No. 12 is set in the start address field of this query.

1000_H + (16 x 12 =192) $_H$ + 0_H
= 1000_H + $C0_H$ + 0_H
= $10C0_H$

“10C0” becomes the input value for the start address field of this query.

Shown below are the screens of IAI's PC software for RC controller, indicating how position data changes before and after a query message is sent.

■ Before a query is sent

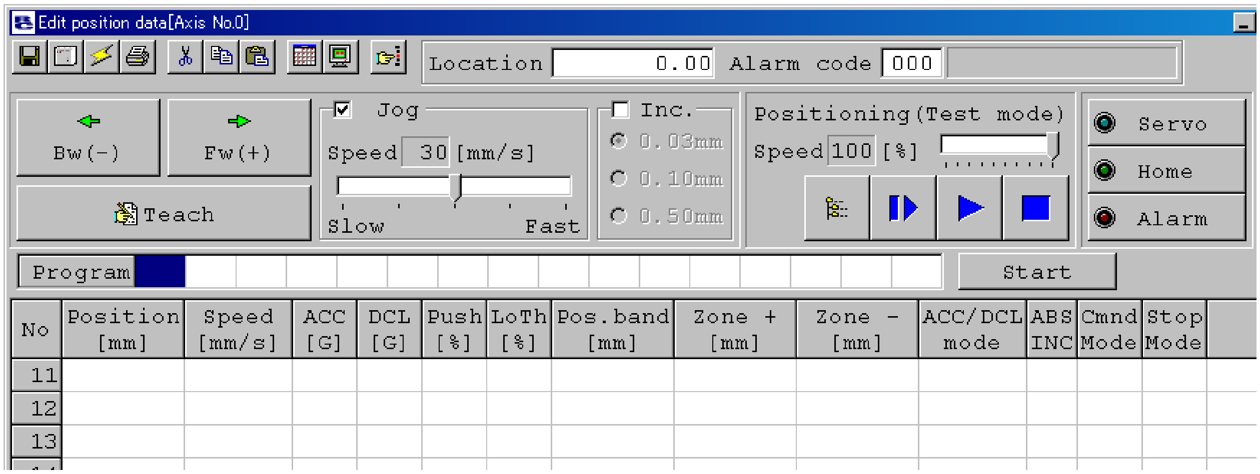


Fig. 6.4

■ After a query is sent

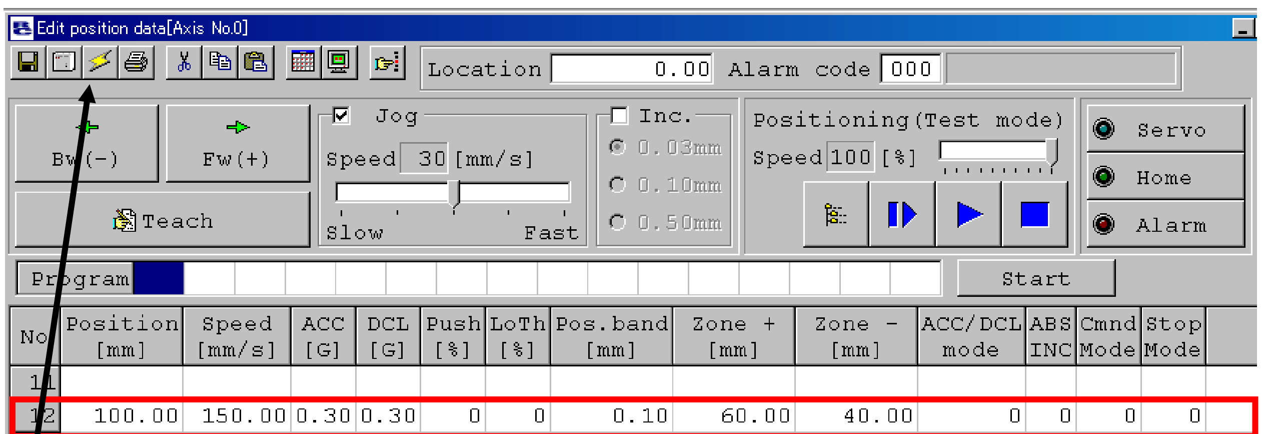


Fig. 6.5

* The overwritten data is not displayed until the button  is pressed or the Edit Position Data window is reopened.

7 Troubleshooting

7.1 Responses at Errors (Exception Responses)

In each query (command), except for a broadcast query message, the master issues a query by expecting a "successful" response(response), and the applicable slave must return a response to the query. If the query is processed successfully, the slave returns a "successful" response. If an error occurs, however, the slave returns an exception response.

The slave responds to a query in one of the following four ways:

(1) The slave receives the query successfully, processes it successfully, and then returns a "successful" response.
(2) The slave returns no response because the query could not be received due to a communication error, etc. The master generates a timeout error.
(3) The slave also returns no response if the query is received but is found invalid because a LRC/CRC error is detected. In this case, the master also generates a timeout error.
(4) If the query is received properly without generating errors but it cannot be processed for some reason (such as when the applicable register does not exist), the slave returns an exception response that contains an exception code indicating the content of exception.

Example of exception response generation
(Sample query message using Read Input Status)

Field	Sample value [Hex]	ASCII mode character string	RTU mode 8 bits [Hex]
Header		‘:’	None
Slave address	03_H	‘0, ’ ‘3’	03_H
Function code	02_H	‘0, ’ ‘2’	02_H
Start address [H]	04_H	‘0, ’ ‘4’	04_H
Start address (L)	$A1_H$	‘A, ’ ‘1’	$A1_H$
Number of DIs [H]	00_H	‘0, ’ ‘0’	00_H
Number of DIs (L)	14_H	‘1, ’ ‘4’	14_H
Error check		LRC (2 characters)	CRC (16 bits)
Trailer		CR/LF	None
	Total bytes	17	8

If input status 04A1H does not exist, the following exception response will be returned.

Sample exception response from a slave

Field	Sample value [Hex]	ASCII mode character string	RTU mode 8 bits [Hex]
Header		‘:’	None
Slave address	03_H	‘0, ’ ‘3’	03_H
Function code	82_H	‘8, ’ ‘2’	82_H
Exception code	02_H	‘0, ’ ‘2’	02_H
Error check		LRC (2 characters)	CRC (16 bits)
Trailer		CR/LF	None
	Total bytes	11	5

The exception response consists of the slave address field, function code field, and data field. In the slave address field, the applicable slave address is set as in the slave address field of a “successful” response. In the function code field, the function code in the query is set, and then the MSB (most significant bit of the function code) of this field is set to 1. This allows the master to recognize that the message is not a “successful” response, but an exception response. An exception code indicating the content of exception is set in the data field.

Example) Query function code “02_H” (00000010b)
→ Exception response function code “82_H” (10000010b)

■ Exception codes

The table below lists the exception codes that may generate in RC Series controllers, as well as the contents of respective codes.

Code [Hex]	Exception code	Function	Remarks
01_H	Illegal Function	Indicates that the function is invalid.	The query cannot be executed because a major error has occurred on the slave side due to function errors.
02_H	Illegal Data Address	Indicates that the data address is invalid.	Use of the data address value is not permitted.
03_H	Illegal Data Value	Indicates that the data is invalid.	Use of the data value is not permitted.
04_H	Slave Device Failure	Indicates that the query cannot be executed because an irremediable error occurred in the slave.	The query cannot be executed because a major error has occurred on the slave side.

7.2 Notes

- When referencing registers using Modbus functions, registers belonging to multiple categories cannot be read simultaneously using a single message. To reference registers belonging to multiple categories, read them using multiple messages by classifying the corresponding addresses by category.
- The explanations in this specification apply commonly to RC controller Series models supporting "Protocol M." For the specifications and other items specific to each model, refer to the RC controller's operation manual that comes with the applicable controller.

7.3 When Communication Fails

Select an applicable item and perform the processing enclosed with □.
The specific processing details are explained after the flowchart; check the details indicated by the * symbol.
○ = Yes, X = No

Symptom: Cannot communicate normally!
Was communication possible until now?

- ○ → Cannot communicate with all controllers?
  - ○ → Cannot communicate with any controllers. →
    - (1) Host failure → [Repair host]
    - (2) Disconnection of communication cables or poor contact due to vibration and similar → [Fix disconnection]
  - × → Cannot communicate with specific controllers. →
    - (1) Controller failure → [Replace the controller]
    - (2) Disconnection of communication cables between controllers and junctions → [Fix disconnection]
    - (3) Dislocation of terminating resistor → [Fix dislocation]
    - (4) Wrong controller axis number setting (when controllers are replaced, parameters are initialized, the version is upgraded, etc.) → [Check] *4
    - (5) Wrong controller communication speed setting (when controllers are replaced, parameters are initialized, the version is upgraded, etc.) → [Check] *6
    - (6) Wrong transmitter activation time setting (when controllers are replaced, parameters are initialized, the version is upgraded, etc.) → [Check parameter No. 17]
  - Can any of the parts of the communication cables be moved?
    - ○ There are movable parts → Disconnection of communication cables → [Fix disconnection]
  - Is the power supply voltage to each device normal? (each RC controller, SIO converter, etc.)
    - ○ Power supply voltage error → [Set normal power supply voltage]
  - Did the failure occur after the surrounding environment was changed? (Peripheral devices, etc. have been changed.)
    - ○ Occurred after change → Malfunction due to noise →
      - [(1) Decrease the communication speed] → [Check] *2
      - [(2) Change communication cable wiring] → [Check] *7
      - [(3) Check connection of FG of each RC controller] *7
      - [(4) Mount a clamp filter on the power lines]
      - [(5) Check grounding of the control board] *7
- ×↓

Continue to the next page

Continued from the previous page

Did the problem occur after moving facilities?

○ Occurred after moving facilities → (1) Disconnection of communication cables → Fix disconnection
(2) Loosening of connectors for communication cables → Check connection
(3) Wrong control communication speed setting → Refer to "3.2. (1) Precautions → Check *6

Did the problem occur after adding controllers?

○ Occurred after adding controllers → (1) Wrong axis number settings for additional controllers → Check *4
(2) Wrong communication speed setting for additional controllers → Check *6
(3) Wrong transmitter activation time setting for additional controllers → Check parameter No. 17
(4) Wrong terminating resistor mounting position → Mount the terminating resistor on the junction furthest from the master (host). (Refer to Section 3.)

Was a new device started up (including newly introduced devices with existing settings)?

○ ↓

Cannot communicate with any controllers?

○ ↓

Check connection using the PC software for RC. *1

○ → Cannot communicate with any controllers → (1) Wrong wiring of communication cables between the host and junction → Check wiring *3
(2) Device power supply error → Check power supply voltage → Check that the 0 V line is used commonly for all devices → Use the 0 V line commonly for all devices.
(3) Overlapping setting of controller axis numbers → Check *4
(4) Inconsistent controller communication speed setting → Check *6

× → Cannot communicate with specific controllers.→ (1) Wrong wiring of communication cables between controllers and junction → Check wiring *3
(2) Device power supply error → Check power supply voltage
(3) Overlapping setting of controller axis numbers → Check *4
(4) Inconsistent controller communication speed setting → Check *6

× → Can communicate. → [1] Inadequate host programs
(1) Check programs again *5
(2) Check communication speed setting again *2

Continue to the next page

Continued from the previous page

Y

- Cannot communicate with specific controllers.
  - ○ [1] Wrong wiring of communication cables between controllers and junction → Check wiring *3
  - [2] Device power supply error → Check power supply voltage → Check that the 0 V line is used commonly for all devices → Use the 0 V line commonly for all devices.
- Cannot communicate from time to time?
  - ○ [1] Malfunction due to noise →
    - (1) Decrease the communication speed Check *2
    - (2) Change communication cable wiring Check *7
    - (3) Check connection of FG of each RC controller *7
    - (4) Mount a clamp filter on the power lines
    - (5) Check grounding of the control board *7
  - [2] Inadequate host programs → Check programs again (occurrence of communication buffer overflow, etc.?)

*1 Connect a PC to the host following the procedure explained in sections 3.1, 3.2 and 3.3.
[1] Start the PC software.
[2] Select [Application Setting] from the [Setting] menu.
Check that the port is set to the port number of the PC used and that the last axis number is set to a value larger than the number of connected axes in the Communication Setting window.
(If any settings are wrong, correct the settings and then restart the PC for RC.)

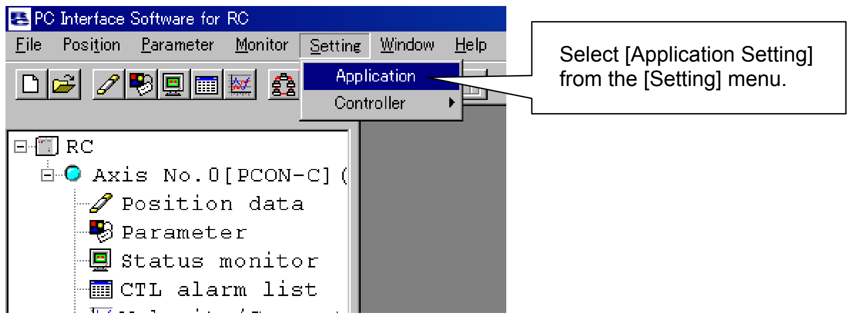


Fig.9.1

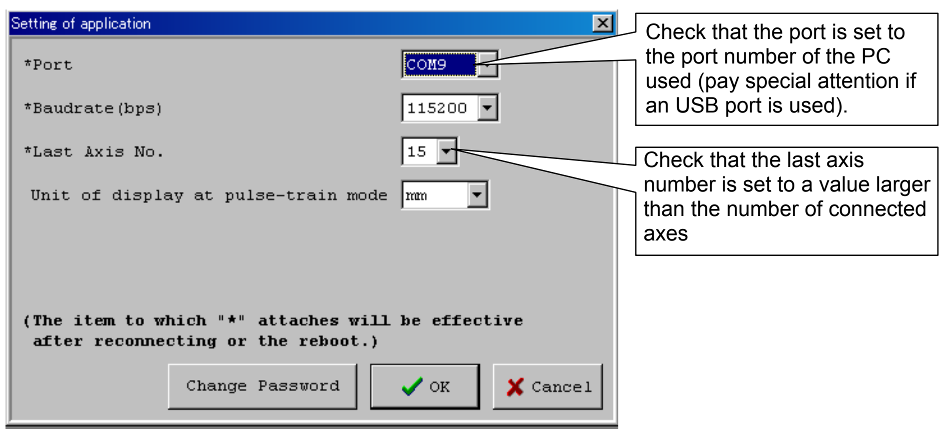


Fig. 9.2

[3] Select [Edit/Teach] from the [Position] menu.
The Position Data Edit Axis Selection window appears, displaying the connected axes.
Axes for which connected axis numbers are displayed can communicate normally.

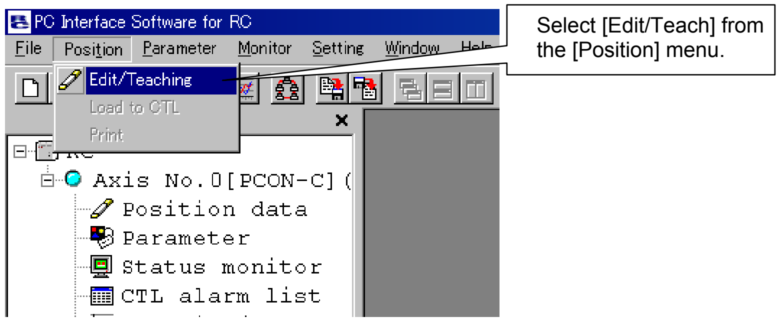


Fig.9.3

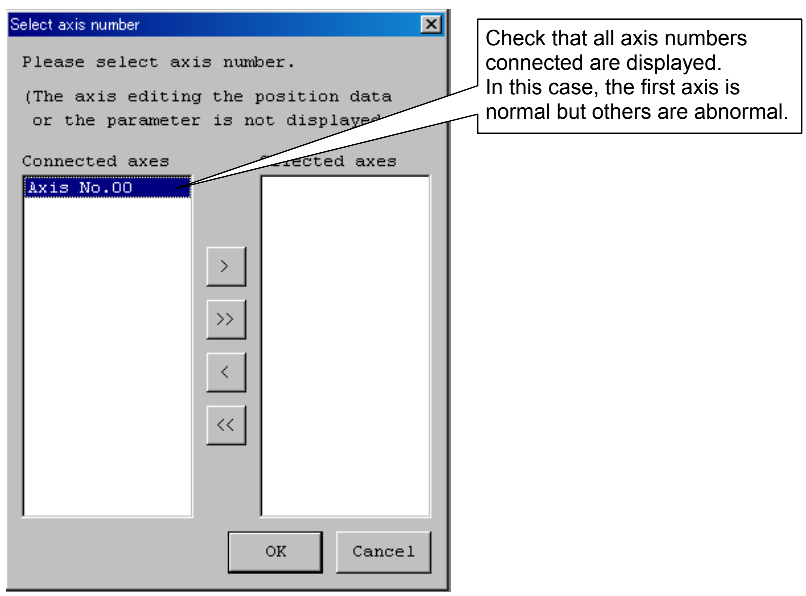


Fig. 9.4

*2 Refer to section 3.6 to decrease the communication speed.
*3 Refer to sections 3.1, 3.2 and 3.3 to check wiring again.
*4 Refer to section 3.5 to check the axis number settings again (check that there are no overlapping numbers).
*5 Check again that the procedure in section 3.4 is followed correctly.
[1] If queries other than those that use a function code 03 are used, check that the PIO/Modbus switching in sections 5.4.16 (RTU) and 6.5.16 (ASCII) is set to the Modbus side.
[2] Unless the RC controller is restarted using the PC software for RC, the communication speed setting selected when connecting the PC software for RC is maintained. In this case, restart the RC controller.

*6 Refer to section 3.6 to check the communication speed setting again.
Set the same communication speed for all RC controllers as well as the host.
Check (2) in *5.

*7 Wire communication cables such that they do not run in parallel with power cables and cables that send pulse signals.
Check that the communication cable is properly shielded (recommendation: 1-point ground).
Check that the setting environment and noise countermeasures live up to the specifications given in the instruction manual of each RC controller.

If the problems are not solved after checking above step, please contact us.
In this case, please let us know about the phenomena occurring and the result of checking the items in the flowchart as well.

8 Reference Materials

8.1 CRC Check Calculation

Sample C functions used for CRC calculation are shown below.
They are equivalent to the CRC calculation functions stated in the published Modbus Protocol Specification (PI-MBUS-300 Rev. J).

```
unsigned short CalcCRC16swap(
    unsigned char*   puchMsg,                  /* message to calculate */
    unsigned short   usDataLen)                /* quantity of bytes in message */
{
    unsigned char    uchCRCHi = 0xFF;          /* high byte of CRC initialized */
    unsigned char    uchCRCLo = 0xFF;          /* low byte of CRC initialized */
    unsigned int     uIndex;                   /* will index into CRC lookup table */

    while(usDataLen--)                         /* pass through message buffer */
    {                                          /* calculate the CRC */
        uIndex = uchCRCHi ^ *puchMsg++;
        uchCRCHi = uchCRCLo ^ auchCRCHi[uIndex];
        uchCRCLo = auchCRCLo[uIndex];
    }
    return (uchCRCHi << 8 | uchCRCLo);
}

const unsigned char auchCRCHi[] =
{/*  Table of CRC values for high-order byte */
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
    0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
    0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
    0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
    0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
```

```
    0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
    0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
    0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
    0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
};

const unsigned char auchCRCLo[] =
{/* Table of CRC values for low-order byte */
    0x00, 0xC0, 0xC1, 0x01, 0xC3, 0x03, 0x02, 0xC2, 0xC6, 0x06, 0x07, 0xC7, 0x05, 0xC5, 0xC4, 0x04,
    0xCC, 0x0C, 0x0D, 0xCD, 0x0F, 0xCF, 0xCE, 0x0E, 0x0A, 0xCA, 0xCB, 0x0B, 0xC9, 0x09, 0x08, 0xC8,
    0xD8, 0x18, 0x19, 0xD9, 0x1B, 0xDB, 0xDA, 0x1A, 0x1E, 0xDE, 0xDF, 0x1F, 0xDD, 0x1D, 0x1C, 0xDC,
    0x14, 0xD4, 0xD5, 0x15, 0xD7, 0x17, 0x16, 0xD6, 0xD2, 0x12, 0x13, 0xD3, 0x11, 0xD1, 0xD0, 0x10,
    0xF0, 0x30, 0x31, 0xF1, 0x33, 0xF3, 0xF2, 0x32, 0x36, 0xF6, 0xF7, 0x37, 0xF5, 0x35, 0x34, 0xF4,
    0x3C, 0xFC, 0xFD, 0x3D, 0xFF, 0x3F, 0x3E, 0xFE, 0xFA, 0x3A, 0x3B, 0xFB, 0x39, 0xF9, 0xF8, 0x38,
    0x28, 0xE8, 0xE9, 0x29, 0xEB, 0x2B, 0x2A, 0xEA, 0xEE, 0x2E, 0x2F, 0xEF, 0x2D, 0xED, 0xEC, 0x2C,
    0xE4, 0x24, 0x25, 0xE5, 0x27, 0xE7, 0xE6, 0x26, 0x22, 0xE2, 0xE3, 0x23, 0xE1, 0x21, 0x20, 0xE0,
    0xA0, 0x60, 0x61, 0xA1, 0x63, 0xA3, 0xA2, 0x62, 0x66, 0xA6, 0xA7, 0x67, 0xA5, 0x65, 0x64, 0xA4,
    0x6C, 0xAC, 0xAD, 0x6D, 0xAF, 0x6F, 0x6E, 0xAE, 0xAA, 0x6A, 0x6B, 0xAB, 0x69, 0xA9, 0xA8, 0x68,
    0x78, 0xB8, 0xB9, 0x79, 0xBB, 0x7B, 0x7A, 0xBA, 0xBE, 0x7E, 0x7F, 0xBF, 0x7D, 0xBD, 0xBC, 0x7C,
    0xB4, 0x74, 0x75, 0xB5, 0x77, 0xB7, 0xB6, 0x76, 0x72, 0xB2, 0xB3, 0x73, 0xB1, 0x71, 0x70, 0xB0,
    0x50, 0x90, 0x91, 0x51, 0x93, 0x53, 0x52, 0x92, 0x96, 0x56, 0x57, 0x97, 0x55, 0x95, 0x94, 0x54,
    0x9C, 0x5C, 0x5D, 0x9D, 0x5F, 0x9F, 0x9E, 0x5E, 0x5A, 0x9A, 0x9B, 0x5B, 0x99, 0x59, 0x58, 0x98,
    0x88, 0x48, 0x49, 0x89, 0x4B, 0x8B, 0x8A, 0x4A, 0x4E, 0x8E, 0x8F, 0x4F, 0x8D, 0x4D, 0x4C, 0x8C,
    0x44, 0x84, 0x85, 0x45, 0x87, 0x47, 0x46, 0x86, 0x82, 0x42, 0x43, 0x83, 0x41, 0x81, 0x80, 0x40,
};
```

8.2 Configuration of Systems that Use both SIO and PIO

It is possible to monitor the current position and other values via the SIO (communication) by running the RC controller with PIO. All queries that use function code 03 for either RTU and ASCII can be monitored. Set PIO/Modbus switching (section 5.4.16 or 6.5.16) to the PIO side and, in case of RC controllers equipped with a mode switch, set the switch to AUTO. The following RC controller models can use both PIO and SIO.

- PCON-C/CG/CF, PCON-CY, PCON-PL/PO,
- ACON-C/CG, ACON-CY, ACON-PL/PO,
- SCON-C/CA,
- ERC2

Example 1 of system configuration that uses both SIO and PIO

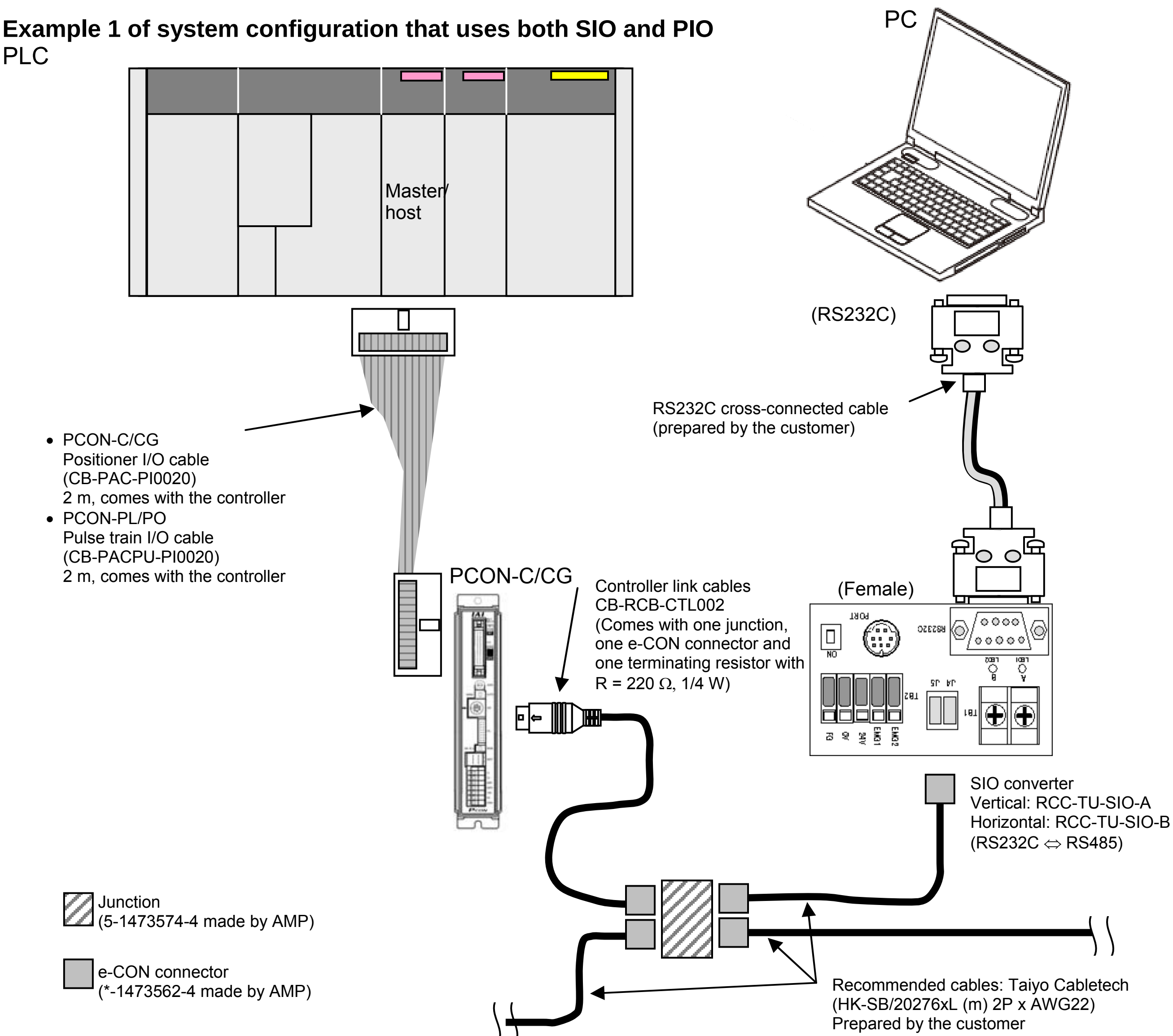


Fig. 11.1

Example 2 of system configuration that uses both SIO and PIO

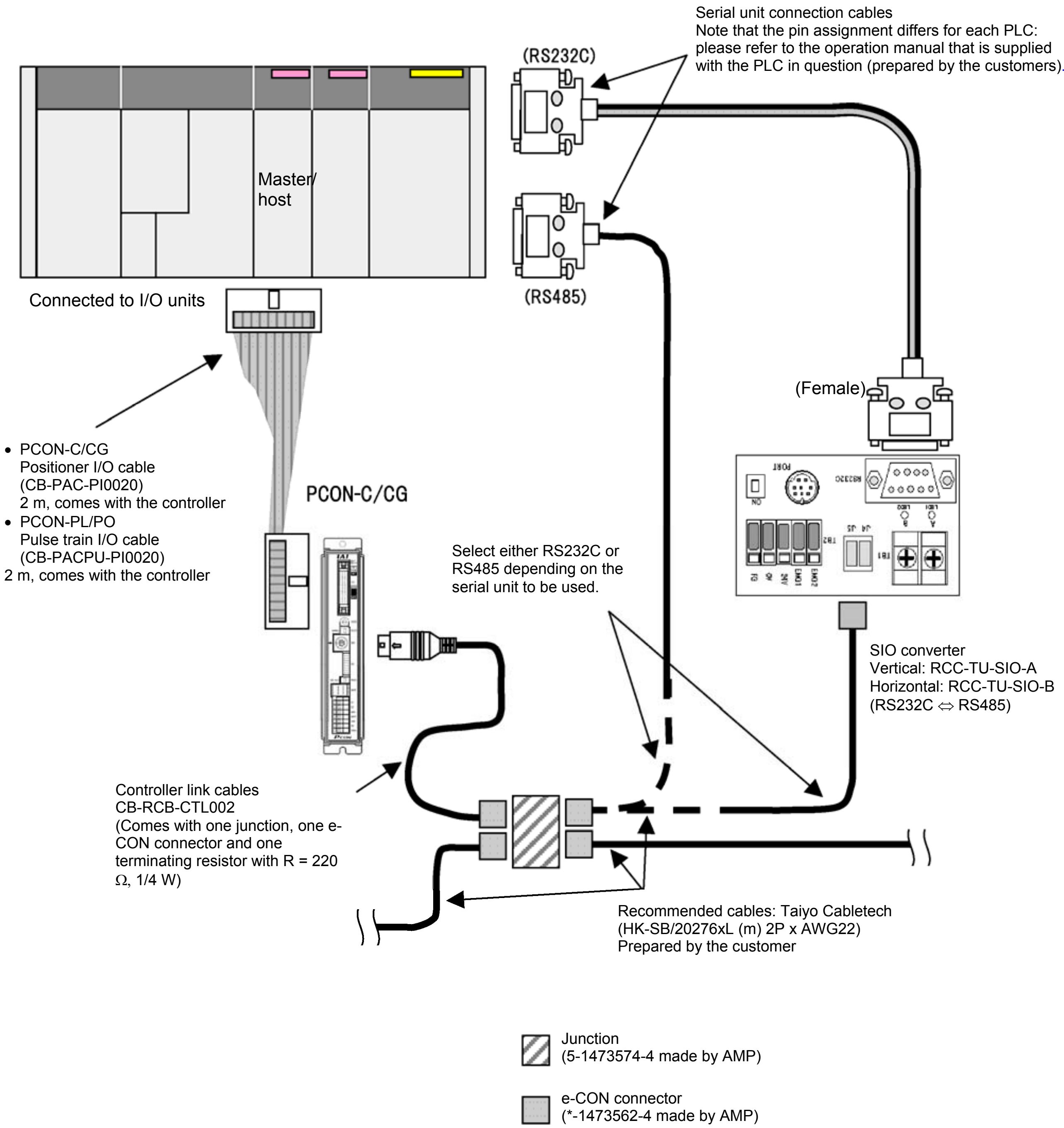


Fig. 11.2

Change History

Revision Date	Description of Revision
May 2010	Released Rev. 5. • Added “Safety Guide.” • Added SCON-CA to the supported models. (Added the load cell calibration command, complete and measurement read commands and registers.) • Readjusted the specification of query 06. • Readjusted the specification of query 10.

Catalog No.: ME0162-5A

IAI America, Inc.

Head Office: 2690 W. 237th Street, Torrance, CA 90505
TEL (310) 891-6015 FAX (310) 891-0815
Chicago Office: 1261 Hamilton Parkway, Itasca, IL 60143
TEL (630) 467-9900 FAX (630) 467-9912
Atlanta Office: 1220 Kennestone Circle, Suite 108, Marietta, GA 30066
TEL (678) 354-9470 FAX (678) 354-9471

website: www.intelligentactuator.com

IAI Industrieroboter GmbH

Ober der Röth 4, D-65824 Schwalbach am Taunus, Germany
TEL 06196-88950 FAX 06196-889524

IAI (Shanghai) Co., Ltd.

SHANGHAI JIAHUA BUSINESS CENTER A8-303, 808, Hongqiao Rd. Shanghai 200030, China
TEL 021+6448-4753 FX 021-6448-3992
website: www.iai-robot.com