FS100 OPTIONS
INSTRUCTIONS
FOR DATA TRANSMISSION FUNCTION

• Upon receipt of the product and prior to initial operation, read these instructions thoroughly, and retain for future reference.

• This instruction is applicable to both FS100 and FS100L controllers.

MOTOMAN INSTRUCTIONS
(FOR SMALL -SIZED MANIPULATORS)  (FOR LARGE AND MEDIUM-SIZED MANIPULATORS)
MOTOMAN-□□□ INSTRUCTIONS  MOTOMAN-□□□ INSTRUCTIONS
FS100 INSTRUCTIONS  FS100L INSTRUCTIONS
FS100 OPERATOR’S MANUAL  FS100 OPERATOR’S MANUAL
FS100 MAINTENANCE MANUAL  FS100L MAINTENANCE MANUAL

The FS100 OPERATOR’S MANUAL above is applicable to both FS100 and FS100L controllers.

Part Number: 159641-1CD
Revision: 1
Data Transmission Function

MANDATORY

- This manual explains the data transmission function of the FS100 system. Read this manual carefully and be sure to understand its contents before handling the FS100.
- General items related to safety are listed in Chapter 1: Safety of the FS100 Instructions. To ensure correct and safe operation, carefully read the FS100 Instructions before reading this manual.

CAUTION

- Some drawings in this manual are shown with the protective covers or shields removed for clarity. Be sure all covers and shields are replaced before operating this product.
- The drawings and photos in this manual are representative examples and differences may exist between them and the delivered product.
- YASKAWA may modify this model without notice when necessary due to product improvements, modifications, or changes in specifications. If such modification is made, the manual number will also be revised.
- If your copy of the manual is damaged or lost, contact a YASKAWA representative to order a new copy. The representatives are listed on the back cover. Be sure to tell the representative the manual number listed on the front cover.
- YASKAWA is not responsible for incidents arising from unauthorized modification of its products. Unauthorized modification voids your product’s warranty.

NOTE

This instruction manual is applicable to both the FS100 (a controller for small-sized manipulators) and FS100L (a controller for medium and large-sized manipulators).

The description of “FS100” refers to both the “FS100” and “FS100L” in this manual unless otherwise specified.
Notes for Safe Operation

Read this manual carefully before installation, operation, maintenance, or inspection of the FS100.

In this manual, the Notes for Safe Operation are classified as “DANGER”, “WARNING”, “CAUTION”, “MANDATORY”, or “PROHIBITED”.

⚠️ DANGER Indicates an imminent hazardous situation which, if not avoided, could result in death or serious injury to personnel.

⚠️ WARNING Indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury to personnel.

⚠️ CAUTION Indicates a potentially hazardous situation which, if not avoided, could result in minor or moderate injury to personnel and damage to equipment. It may also be used to alert against unsafe practices.

⚠️ MANDATORY Always be sure to follow explicitly the items listed under this heading.

🚫 PROHIBITED Must never be performed.

Even items described as “CAUTION” may result in a serious accident in some situations.

At any rate, be sure to follow these important items

To ensure safe and efficient operation at all times, be sure to follow all instructions, even if not designated as “DANGER”, “WARNING” and “CAUTION”.

WARNING

• Before operating the manipulator, check that servo power is turned off when the emergency stop button on the programing pendant is pressed. When the servo power is turned off, the SERVO ON LED on the programing pendant is turned off.

Injury or damage to machinery may result if the emergency stop circuit cannot stop the manipulator during an emergency. The manipulator should not be used if the emergency stop buttons do not function.

Figure 1: Emergency Stop Button

• In the case of not using the programming pendant, be sure to supply the emergency stop button on the equipment. Then before operating the manipulator, check to be sure that the servo power is turned OFF by pressing the emergency stop button. Connect the external emergency stop button to the 5-6 pin and 16-17 pin of the robot system signal connector (CN2).

• Upon shipment of the FS100, this signal is connected by a jumper cable in the dummy connector. To use the signal, make sure to prepare a new connector, and then input it.

If the signal is input with the jumper cable connected, it does not function, which may result in personal injury or equipment damage.

• Once the emergency stop button is released, clear the cell of all items which could interfere with the operation of the manipulator. Then turn the servo power ON.

Injury may result from unintentional or unexpected manipulator motion.

Figure 2: Release of EM

• Observe the following precautions when performing teaching operations within the P-point maximum envelope of the manipulator:
  – Be sure to use a lockout device to the safeguarding when going inside. Also, display the sign that the operation is being performed inside the safeguarding and make sure no one closes the safeguarding.
  – View the manipulator from the front whenever possible.
  – Always follow the predetermined operating procedure.
  – Ensure that you have a safe place to retreat in case of emergency.

Improper or unintended manipulator operation may result in injury.

The emergency stop button is located on the programing pendant.
Definition of Terms Used Often in This Manual

The MOTOMAN is the YASKAWA industrial robot product.

The MOTOMAN usually consists of the manipulator, the FS100 controller, manipulator cables, the FS100 programming pendant (optional), and the FS100 programming pendant dummy connector (optional).

In this manual, the equipment is designated as follows:

<table>
<thead>
<tr>
<th>Equipment</th>
<th>Manual Designation</th>
</tr>
</thead>
<tbody>
<tr>
<td>FS100 controller</td>
<td>FS100</td>
</tr>
<tr>
<td>FS100 programming pendant</td>
<td>Programming pendant</td>
</tr>
<tr>
<td>Cable between the manipulator and the controller</td>
<td>Manipulator Cable</td>
</tr>
<tr>
<td>FS100 programming pendant dummy connector</td>
<td>Programming pendant dummy connector</td>
</tr>
</tbody>
</table>
Descriptions of the programming pendant keys, buttons, displays and keyboard of the PC are shown as follows:

<table>
<thead>
<tr>
<th>Equipment</th>
<th>Manual Designation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Programming Pendant</td>
<td></td>
</tr>
<tr>
<td>Character Keys</td>
<td>The keys which have characters printed on them are denoted with []. ex. [ENTER]</td>
</tr>
<tr>
<td>Symbol Keys</td>
<td>The keys which have a symbol printed on them are not denoted with [] but depicted with a small picture. ex. PAGE key The Cursor is an exception, and a picture is not shown.</td>
</tr>
<tr>
<td>Axis Keys</td>
<td>“Axis Keys” and “Numeric Keys” are generic names for the keys for axis operation and number input.</td>
</tr>
<tr>
<td>Numeric Keys</td>
<td></td>
</tr>
<tr>
<td>Keys pressed simultaneously</td>
<td>When two keys are to be pressed simultaneously, the keys are shown with a “+” sign between them, ex.SHIFT key +COORD key</td>
</tr>
<tr>
<td>Mode Key</td>
<td>Three kinds of modes that can be selected by the mode key are denoted as follows: REMOTE, PLAY, or TEACH</td>
</tr>
<tr>
<td>Button</td>
<td>Three buttons on the upper side of the programming pendant are denoted as follows: HOLD button START button EMERGENCY STOP button</td>
</tr>
<tr>
<td>Displays</td>
<td>The menu displayed in the programming pendant is denoted with { }. ex. {JOB}</td>
</tr>
<tr>
<td>PC Keyboard</td>
<td>The name of the key is denoted ex. Ctrl key on the keyboard</td>
</tr>
</tbody>
</table>

**Description of the Operation Procedure**

In the explanation of the operation procedure, the expression “Select • • •” means that the Cursor is moved to the object item and the SELECT key is pressed, or that the item is directly selected by touching the screen.

**Registered Trademark**

In this manual, names of companies, corporations, or products are trademarks, registered trademarks, or brand names for each company or corporation. The indications of (R) and TM are omitted.
Table of Contents

1 Outline ................................................................. 1-1
   1.1 DCI Function .................................................. 1-2
   1.2 Stand-alone Function ....................................... 1-3
   1.3 Host Control Function ...................................... 1-4

2 For Using Data Transmission Function ...................... 2-1
   2.1 Remote Mode .................................................. 2-1
      2.1.1 Remote Mode ........................................... 2-1
      2.1.2 Command Remote Valid/Invalid .................... 2-3
      2.1.3 Display in Command Remote Mode .................. 2-4
   2.2 Serial I/F Port Assignment .................................. 2-5
   2.3 Parallel Operation of FS100 ............................... 2-6
      2.3.1 No Multiple-operation of DCI, Stand-alone, and Host Control Functions ....... 2-6
      2.3.2 File Access and Editing for a Single Target .......... 2-6
   2.4 Transmission Specifications ............................... 2-7
      2.4.1 Basic Specifications .................................... 2-7
      2.4.2 Transmission Control Characters .................... 2-7
      2.4.3 Transmission Format .................................... 2-8
      2.4.4 Error Control System ................................... 2-9
      2.4.5 Character Configuration ............................... 2-9
      2.4.6 Data Link Establishment ............................... 2-10
      2.4.7 Configuration of Heading and Text .................. 2-10
      2.4.8 Transmission Parameters ............................. 2-11
         2.4.8.1 Transmission Control Monitoring Timer .......... 2-11
         2.4.8.2 Transmission Control Resending Sequence ....... 2-12
      2.4.9 Connection of D-SUB Connector Pins ................ 2-13
      2.4.10 Connection .............................................. 2-13

3 DCI Function ....................................................... 3-1
   3.1 Outline ....................................................... 3-1
   3.2 Commands for Job Transmission .......................... 3-2
      3.2.1 LOADJ .................................................. 3-2
         3.2.1.1 Function ........................................... 3-2
         3.2.1.2 Configuration ..................................... 3-2
# Table of Contents

## 3.2 Data Transmission Function

- **3.2.2 SAVEJ**
  - 3.2.2.1 Function .................................................. 3-3
  - 3.2.2.2 Configuration .............................................. 3-3

- **3.2.3 DELETEJ**
  - 3.2.3.1 Function .................................................. 3-4
  - 3.2.3.2 Configuration .............................................. 3-4

- **3.2.4 SWAIT**
  - 3.2.4.1 Function .................................................. 3-4
  - 3.2.4.2 Configuration .............................................. 3-4

## 3.3 Commands for Variable Transmission

- **3.3.1 LOADV**
  - 3.3.1.1 Function .................................................. 3-5
  - 3.3.1.2 Configuration .............................................. 3-5

- **3.3.2 SAVEV**
  - 3.3.2.1 Function .................................................. 3-5
  - 3.3.2.2 Configuration .............................................. 3-5

## 3.4 Registering DCI Instruction

## 3.5 Concurrent Tasks from Multiple Jobs

## 3.6 DCI Parallel Execution

- **3.6.1 Parallel Execution Using NWAIT** .................................................. 3-11
- **3.6.2 Parallel Execution Using PSTART (Optional)** .................................................. 3-12

## 3.7 Transmission Procedure

- **3.7.1 Job Transmission** .................................................. 3-13
  - 3.7.1.1 Saving Procedure .............................................. 3-13
  - 3.7.1.2 Loading Procedure ............................................ 3-14

- **3.7.2 Variable Transmission** .................................................. 3-15
  - 3.7.2.1 Saving Procedure .............................................. 3-15
  - 3.7.2.2 Loading Procedure ............................................ 3-15

## 3.8 Axis Data Transmission Format

## 3.9 Alarm Codes

## 4 Stand-alone Function

- **4.1 Outline** .................................................. 4-1

- **4.2 Operation Flow** .................................................. 4-2

- **4.3 Operation** .................................................. 4-3
  - 4.3.1 Selecting External Memory Unit .............................................. 4-3
  - 4.3.2 Save .................................................. 4-4
    - 4.3.2.1 Saving Job .............................................. 4-4
    - 4.3.2.2 Saving File .............................................. 4-6
4.3.3 Load................................................................................................................... 4-8
  4.3.3.1 Loading Job............................................................................................. 4-8
  4.3.3.2 Loading File............................................................................................. 4-9

4.3.4 Job Selection Mode ............................................................................................ 4-11
  4.3.4.1 Single Selection Mode........................................................................... 4-11
  4.3.4.2 Related Selection Mode ........................................................................ 4-11
  4.3.4.3 Switching Selection Mode ..................................................................... 4-12

4.3.5 Selecting Job and Data File................................................................................ 4-13
  4.3.5.1 EACH Selection..................................................................................... 4-13
  4.3.5.2 BATCH Selection................................................................................... 4-13

4.4 Transmission Procedure................................................................................................. 4-13

5 Host Control Function of FS100.................................................................................... 5-1

  5.1 File Data Transmission Function ................................................................................... 5-1
    5.1.1 Transmission Procedure....................................................................................... 5-2
      5.1.1.1 Load ........................................................................................................ 5-2
      5.1.1.2 Save ........................................................................................................ 5-3
    5.1.2 Data Management ................................................................................................ 5-4

  5.2 Robot Control Function................................................................................................... 5-5
    5.2.1 Command Transmission....................................................................................... 5-5
    5.2.2 List of Interlock for Commands of Host Control Function ..................................... 5-7
    5.2.3 Command that Handle Axis Data .................................................................. 5-9
    5.2.4 Response to MOV-type Command.................................................................. 5-9
    5.2.5 Status Read Function ....................................................................................... 5-10
      5.2.5.1 Read/Monitor Command ....................................................................... 5-10
      5.2.5.2 Read/Data Access System Commands ................................................5-18
    5.2.6 System Control Function .................................................................................... 5-24
      5.2.6.1 Operation System Commands .............................................................. 5-24
      5.2.6.2 Start-up System Commands ..................................................................... 5-31
      5.2.6.3 Editing System Commands ..................................................................... 5-39
      5.2.6.4 Job Selection System Commands......................................................... 5-46
    5.2.7 I/O Read/Write Function ..................................................................................... 5-47
      5.2.7.1 Transmission Procedure........................................................................... 5-47
      5.2.7.2 Read-out of I/O Signal Status................................................................5-48
      5.2.7.3 Write-in of I/O Signal Status .................................................................. 5-49

  5.3 Commands for Multi-control Group and Independent Control Functions............... 5-50
    5.3.1 Commands for Multi-control Group............................................................ 5-50
    5.3.2 Commands for Independent Control Function.................................................. 5-51

  5.4 Alarm Codes ............................................................................................................. 5-52

  5.5 Interpreter Message List ............................................................................................ 5-53
# Data Transmission Function

## Table of Contents

6  Data List ........................................................................................................................................... 6-1
   6.1  Header Number List ...................................................................................................................... 6-1
   6.2  Parameter List ............................................................................................................................. 6-2

7  Remote Function Setting .................................................................................................................. 7-1
The data transmission function is for communication with a host computer such as a personal computer in BSC complying protocol.

The data transmission function adopts a serial transmission line and standard protocol, making easy connection to a host computer.

The data transmission function is not only for transmission of job but also for controlling robot system by a host computer using a set of commands.

The robot commands in the ASCII code command format are easy to use and helpful for a quick development of necessary software to be run on the host computer.

The data transmission function is divided into the following three functions.

- DCI (Data Communication by Instruction)
- Stand-alone function
- Host control function
1.1 DCI Function

The DCI function executes instructions described in a job to perform data transmission with a host computer.

This function loads and saves jobs and variables.

**Table 1-1: DCI Function**

<table>
<thead>
<tr>
<th>Job Transmission</th>
<th>Load</th>
<th>Save</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Job can be transmitted in either mode.</td>
<td>• Single job</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Related job</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Variable Transmission</th>
<th>Load</th>
<th>Save</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• Byte type global variables</td>
<td>• Double precision type global variables</td>
</tr>
<tr>
<td></td>
<td>• Integer type global variables</td>
<td>• Real number type global variables</td>
</tr>
<tr>
<td></td>
<td>• Position type global variables</td>
<td>• Position type global variables (Robot axes, base axes, station axes)</td>
</tr>
</tbody>
</table>

FS100
Host computer (personal computer, etc.)
1.2 Stand-alone Function

The stand-alone function is for data transmission with host computer by operation on the programming pendant.

This function loads and saves jobs and condition data.

Table 1-2: Stand-alone Function

<table>
<thead>
<tr>
<th>Job Transmission</th>
<th>Load</th>
<th>Save</th>
<th>Verify</th>
</tr>
</thead>
<tbody>
<tr>
<td>Job Transmission</td>
<td>Job can be transmitted in either mode.</td>
<td>• Single job</td>
<td>• Related job</td>
</tr>
<tr>
<td>Condition Data/ General Data Transmission</td>
<td>Load</td>
<td>• Tool data</td>
<td>• User coordinate data</td>
</tr>
<tr>
<td>System Information Transmission</td>
<td>Save</td>
<td>• System information</td>
<td>• Alarm history</td>
</tr>
</tbody>
</table>

FS100 Operation Host computer (personal computer, etc.)
1.3 **Host Control Function**

The host control function is for loading and saving jobs, reading robot status, and controlling the system by sending a command from a host computer.
# Outline

1.3 Host Control Function

### Table 1-3: Host Control Function

<table>
<thead>
<tr>
<th>File Data Transmission Function</th>
<th>Job Transmission</th>
<th>Load</th>
<th>Jobs can be transmitted in either mode:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Save</td>
<td>• Single job</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>• Related job</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Condition Data/General Data Transmission</th>
<th>Load</th>
<th>Tool data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Save</td>
<td>User coordinate data</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Variable data</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>System Information Transmission</th>
<th>Save</th>
<th>System information</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Alarm history</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Robot Control Function</th>
<th>Status Reading</th>
<th>• Read of error and alarm codes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>• Read of current position in a joint coordinate system</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Read of current position in a specified Cartesian coordinate system</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Read of mode, cycle, motion, alarm error and servo status</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Read of current job name, line No. and step No.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Read of all job names or related job names</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Monitoring completion of manipulator operation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Read of specified user coordinate data</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Read of control group and task selected status</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Read of variable data</td>
</tr>
</tbody>
</table>

| System Control | • Start, hold |
|               | • Reset, cancel |
|               | • Job deletion |
|               | • Master job setup |
|               | • Job, line No. and step No. setup |
|               | • Mode and cycle selection |
|               | • Servo power supply ON/OFF |
|               | • Programming pendant interlock setup/release |
|               | • Message display |
|               | • Joint motion and linear motion to a specified Cartesian coordinate system |
|               | • Linear motion by increments in a specified coordinate system |
|               | • Joint motion and linear motion to a specified joint coordinate system |
|               | • Conversion/reverse conversion of related job of a specified job (Relative job function is necessary) |
|               | • Write of specified user coordinate data |
|               | • Change of control group |
|               | • Change of task to be controlled |
|               | • Write of variable data |
2  For Using Data Transmission Function

2.1  Remote Mode

The data transmission function can be used with FS100 in remote mode.

2.1.1  Remote Mode

To use the data transmission function, set FS100 to remote mode.

In remote mode, the operation is ordered from a host computer; whereas in local mode, teach mode, and play mode, the programming pendant is used for operating the system.

To switch to the remote mode or the local mode, either
1. Set the mode key on the programming pendant to [REMOTE].

- The remote mode has two sub-modes; “I/O remote enable” and “Command remote enable”.
- Which sub-mode takes effect in remote mode is set in the pseudo input display.

Supplementary note: For details, refer to chapter 7 “Remote Function Setting”.

![Diagram of Mode Switching]

-  Local mode
-  Teach mode
-  Play mode
-  Remote mode
  - I/O remote enable
  - Command remote enable
2.1 Remote Mode

In remote mode, operations on the programming pendant are valid except the operation-related entries. This holds true in “I/O remote enable” and “Command remote enable” submodes. The concept is based on the conventional I/O control introduced to command control.

Note that the edit-related operations cannot be entered from more than one operating device.

In “Command remote enable” submode, to enable command remote controls only, issue the HLOCK command.

When the HLOCK command is ON, operations on the programming pendant are valid only hold and emergency stop.

Also the following I/O operations are disabled: selection between remote mode and local mode, external start, external servo ON, cycle selection, I/O prohibit, P.P/PANEL prohibit, and master job call. Other I/O operations are valid.

### Operation-site Mode | Operation-site | Condition to Enable the Operation
--- | --- | ---
Local Mode | Programming pendant | The remote lamp is OFF, or “INHIBIT PP/PANEL” in the pseudo input display is set to invalid.

### Remote Mode

| Mode | I/O remote enable | External I/O control board | The remote lamp is ON, and “INHIBIT IO” in the pseudo input display is set invalid.
| --- | --- | --- |
| Command remote enable | External computer | The remote lamp is ON, and “CMD REMOTE SEL” in the pseudo input display is set valid.

---

**NOTE**

- In remote mode, usually operations of the programming pendant is disabled, but they can be also enabled.
- To enable all operations, refer to chapter 7 “Remote Function Setting”.
- To selectively enable some of the operations, set the parameter S2C230. For details, refer to section 6.2 “Parameter List” on page 6-2.

In remote mode, operations on the programming pendant are valid except the operation-related entries.

This holds true in “I/O remote enable” and “Command remote enable” submodes. The concept is based on the conventional I/O control introduced to command control.

Note that the edit-related operations cannot be entered from more than one operating device.
2.1.2 Command Remote Valid/Invalid

Availability of each function of data transmission differs depending on the command remote setting (Enabled / Disabled).

When the command remote is set invalid, the read/monitor system commands (hereinafter called read-only function) in the host control function in addition to the DCI function and stand-alone function can be used.

For the details of read/monitor system commands, refer to section 5.2.2 “List of Interlock for Commands of Host Control Function” on page 5-7.

<table>
<thead>
<tr>
<th>Command Remote Setting</th>
<th>Function Availability</th>
</tr>
</thead>
</table>
| Invalid                | DCI function available  
                        | Stand-alone function available  
                        | Host control function (only read-only function) available |
| Valid                  | Host control function (all commands) available                 |

To validate the read-only function in the above host control function, set the parameter RS005 to “1”.

When the command remote is validated by pressing [REMOTE] with the read-only function valid, the command remote status is entered so that all commands can be used.

When the command remote is invalidated by pressing [REMOTE] again, the read-only function becomes validated again.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Contents and Set Value</th>
<th>Initial Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>RS005</td>
<td>BSC port function specification when the command remote is invalidated</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0 : DCI or stand-alone function</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 : Read-only function in host control</td>
<td></td>
</tr>
</tbody>
</table>
2.1.3 Display in Command Remote Mode

Even in command remote enabled submode, it is not necessary to call the command remote display because operations from FS100 is available.

To call the command remote display, select "REMOTE" from "I/O" under the top menu.

This display is used in common with the I/O remote mode display.

The message in the remote display changes according to the remote function selection. (Refer to chapter 7 “Remote Function Setting”.)

<table>
<thead>
<tr>
<th>Remote Select Status</th>
<th>Message</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>I/O Remote</td>
<td>Command Remote</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>&quot;Remote mode not specified&quot;</td>
<td>Same when the remote lamp is OFF.</td>
</tr>
<tr>
<td>i</td>
<td>i</td>
<td>&quot;I/O mode&quot;</td>
</tr>
<tr>
<td>i</td>
<td>i</td>
<td>&quot;Command mode&quot;</td>
</tr>
<tr>
<td>i</td>
<td>i</td>
<td>&quot;I/O and Command mode&quot;</td>
</tr>
<tr>
<td>Read-only Function</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Valid</td>
<td>&quot;Remote mode not specified&quot;</td>
<td>&quot;CURR&quot; and &quot;PREV&quot; are displayed.</td>
</tr>
</tbody>
</table>

○ : Valid, × : Invalid
2.2 Serial I/F Port Assignment

The FS100 has one serial interface port (RS-232Cl/F).

The FC1 protocol and the BSC complying protocol (for data transmission function: option) can be assigned to the port to communicate with external devices.

A change in assignment can be made only in local mode.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Contents and Set Value</th>
<th>Initial Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>RS000</td>
<td>Standard port protocol specification</td>
<td>2</td>
</tr>
</tbody>
</table>

0 : NON  
1 : System reserved  
2 : BSC LIKE (Data transmission function)  
3 : FC1

Fig. 2-1: CPU Unit Configuration
2.3 Parallel Operation of FS100

The FS100 is capable of parallel processing.

The parallel operation has the following restrictions. When an operation against these restrictions is made, a warning message is displayed.

<table>
<thead>
<tr>
<th>Operation</th>
<th>Warning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stand-alone Programming pendant</td>
<td>Error message for 5 seconds</td>
</tr>
<tr>
<td>DCI</td>
<td>Alarm</td>
</tr>
<tr>
<td>Host control</td>
<td>Interpreter message (or error message)</td>
</tr>
</tbody>
</table>

2.3.1 No Multiple-operation of DCI, Stand-alone, and Host Control Functions

All DCI, stand-alone, and host control use BSC LIKE protocol and the same port, therefore these functions cannot be performed by parallel processing.

- Warning message : Serial port not defined
- Warning message : Serial port being used
- Warning message : Protocol being used

2.3.2 File Access and Editing for a Single Target

Access to a single target file is available. Parallel processing of reads from two or more sources is impossible.

During access to a file for other function, the HLOCK command of the host control function cannot be issued.

Key operations are ignored while the HLOCK command is ON.

- Warning message : Data accessed with other functions
2.4 Transmission Specifications

This section explains the transmission specifications for the data transmission.

2.4.1 Basic Specifications

<table>
<thead>
<tr>
<th>Interface</th>
<th>Complies to RS-232C (RS/CS method)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transmission Speed</td>
<td>9600 bps</td>
</tr>
<tr>
<td>Transmission Mode</td>
<td>Half-duplex transmission system (point-to-point)</td>
</tr>
<tr>
<td>Synchronization system</td>
<td>Asynchronous (stop bit 1)(^{1})</td>
</tr>
<tr>
<td>Protocol</td>
<td>BSC LIKE</td>
</tr>
<tr>
<td>Transmission Code</td>
<td>ASCII, shift JIS 8-bit data length (^{1}), Even parity (^{1}), Nontransparent</td>
</tr>
<tr>
<td>Error Check</td>
<td>BCC</td>
</tr>
<tr>
<td>Response Method</td>
<td>ACK alternating response</td>
</tr>
</tbody>
</table>

1 Can be changed by transmission parameter setting

2.4.2 Transmission Control Characters

The transmission control characters are shown in the table below.

Table 2-1: Transmission Control Characters and Codes

<table>
<thead>
<tr>
<th>Control Character</th>
<th>Code (hexadecimal)</th>
<th>Meanings of Control Character</th>
</tr>
</thead>
<tbody>
<tr>
<td>DLE</td>
<td>10</td>
<td>Data Link Escape</td>
</tr>
<tr>
<td>SOH</td>
<td>01</td>
<td>Start of Heading</td>
</tr>
<tr>
<td>STX</td>
<td>02</td>
<td>Start of Text</td>
</tr>
<tr>
<td>ETX</td>
<td>03</td>
<td>End of Text</td>
</tr>
<tr>
<td>EOT</td>
<td>04</td>
<td>End of Transmission</td>
</tr>
<tr>
<td>ENQ</td>
<td>05</td>
<td>Enquiry</td>
</tr>
<tr>
<td>NAK</td>
<td>15</td>
<td>Negative Acknowledgment</td>
</tr>
<tr>
<td>ETB</td>
<td>17</td>
<td>End of Text Block</td>
</tr>
<tr>
<td>ACK0</td>
<td>10, 30</td>
<td>Even Affirmative Acknowledgment</td>
</tr>
<tr>
<td>ACK1</td>
<td>10, 31</td>
<td>Odd Affirmative Acknowledgment</td>
</tr>
</tbody>
</table>
### 2.4.3 Transmission Format

The transmission format is as follows.

```
SOH | HEADING | STX | TEXT | ETB | BCC
----|---------|-----|------|-----|-----
```

```
SOH | HEADING | STX | TEXT | ETX | BCC
----|---------|-----|------|-----|-----
```

```
STX | TEXT | ETB | BCC
-----|------|-----|-----
```

```
STX | TEXT | ETX | BCC
-----|------|-----|-----
```

ENQ

EOT

NAK

ACK0

ACK1
2.4.4 Error Control System

The error control is performed by a check sum of all the characters from SOH or STX to ETB or ETX.

The check sum is calculated as shown below.

<Example>

![Diagram of error control system]

- Start of calculation: Calculation is started when SOH or STX used as the block start sequence appears. These block start sequence are not included in the sum. As for a STX led by a SOH, STX is included in the sum.

- End of calculation: Calculation is ended when ETB or ETX used as the block end sequence appears, with the ETB or ETX included in the sum.

2.4.5 Character Configuration

The character configuration is as follows.
2.4.6 Data Link Establishment

A data link is established by responding ACK0 to ENQ.

2.4.7 Configuration of Heading and Text

The configuration of heading and text is as follows.

![Diagram showing data link configuration]

- Heading: 6 characters fixed, max. 256 characters
- Subcode No.
- (comma)
- Header No.
2.4.8 Transmission Parameters

2.4.8.1 Transmission Control Monitoring Timer

Two timers are provided for transmission control monitoring.
Both are transmission parameters so that their settings can be changed
for each system.

- Timer A: Sequence monitoring timer. Serves as protection against
  invalid or no response.
  Recommended value is 3 sec.
- Timer B: Text reception monitoring timer. Serves as protection
  against no response of text end character.
  Recommended value is 20 sec.
2.4.8.2 Transmission Control Resending Sequence

Two constants below are related to the transmission control resending sequence.

Both are transmission parameters like the transmission control monitoring timers, whose settings can be changed for each system.

- **Retry 1**: Number of resendings of a sequence character at an invalid or no response at all. Recommended value is 10 times.
- **Retry 2**: Number of resendings of a text at a block check error (reception of NAK). Recommended value is 3 times.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Contents and Set Value</th>
<th>Initial Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>RS030</td>
<td>Number of data bits</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>7 : 7 (bit)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>8 : 8 (bit)</td>
<td></td>
</tr>
<tr>
<td>RS031</td>
<td>Number of stop bits</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0 : 1 (bit)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 : 1.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2 : 2</td>
<td></td>
</tr>
<tr>
<td>RS032</td>
<td>Parity specification</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>0 : No specification</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 : Odd parity</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2 : Even parity</td>
<td></td>
</tr>
<tr>
<td>RS033</td>
<td>Transmission speed specification</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>1 : 150 (baud rate)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2 : 300</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3 : 600</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4 : 1200</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5 : 2400</td>
<td></td>
</tr>
<tr>
<td></td>
<td>6 : 4800</td>
<td></td>
</tr>
<tr>
<td></td>
<td>7 : 9600</td>
<td></td>
</tr>
<tr>
<td></td>
<td>8 : 19200</td>
<td></td>
</tr>
<tr>
<td>RS034</td>
<td>Timer A Sequence monitoring timer</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>Serves as protection against invalid or no response</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Unit : 0.1 sec. (Setting range : 0 to 100)</td>
<td></td>
</tr>
<tr>
<td>RS035</td>
<td>Timer B Text reception monitoring timer</td>
<td>200</td>
</tr>
<tr>
<td></td>
<td>Serves as protection against no response of text end character</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Unit : 0.1 sec. (Setting range : 0 to 255)</td>
<td></td>
</tr>
<tr>
<td>RS036</td>
<td>Retry 1 character Number of resendings of a sequence character at an invalid or no response</td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>(Setting range : 0 to 30)</td>
<td></td>
</tr>
<tr>
<td>RS037</td>
<td>Retry 2 Number of resendings of a text at a block check error (reception of NAK). (Setting range : 0 to 10)</td>
<td>3</td>
</tr>
<tr>
<td>RS038</td>
<td>Block check method 0 : Check sum</td>
<td>0</td>
</tr>
</tbody>
</table>
2.4.9 Connection of D-SUB Connector Pins

The connection of D-SUB connector pins is shown below.

*Fig. 2-2: YCP01 board (D-SUB9P)*

<table>
<thead>
<tr>
<th>Pin</th>
<th>FS100</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CD</td>
<td>1</td>
<td>Carrier detect</td>
</tr>
<tr>
<td>RD</td>
<td>2</td>
<td>Data receive</td>
</tr>
<tr>
<td>SD</td>
<td>3</td>
<td>Data send</td>
</tr>
<tr>
<td>ER</td>
<td>4</td>
<td>Data terminal ready</td>
</tr>
<tr>
<td>SG</td>
<td>5</td>
<td>Grounding for signal</td>
</tr>
<tr>
<td>RS</td>
<td>7</td>
<td>Request to send</td>
</tr>
<tr>
<td>CS</td>
<td>8</td>
<td>Sending enabled</td>
</tr>
<tr>
<td>FG</td>
<td>9</td>
<td>Protective grounding</td>
</tr>
</tbody>
</table>

2.4.10 Connection

Since the system is “null-modem”, connect the pins as shown below.

*Fig. 2-3: Board*

- Connect “RS” of the FS100 to “CS” of a host computer.
  This prevents data overrun when reception processing speed of the FS100 cannot catch up with data sending from the host computer.
  In other words, “RS” signal from the FS100 controls start-hold of data transmission from the host computer.
  The sending interface controller must be capable of coping with CS input displacement in units of a single byte.
- The FS100 sends data when the “CS” signal is ON.
3 DCI Function

3.1 Outline

The data communication by instruction (DCI) function loads, saves jobs and variables according to an instruction that executes data transmission with a host computer.

The DCI function is classified as follows.

- Job load, save and delete functions
- Variable load and save functions
3.2 Commands for Job Transmission

3.2.1 LOADJ

3.2.1.1 Function

Loads specified jobs as single or related jobs, from the external memory unit to the memory of the FS100.

3.2.1.2 Configuration

- If the FS100 memory already contains a job having the same name as the job to be loaded, the existing job is deleted and the new job is loaded. However, if the job to be loaded is as follows, an alarm occurs.
  - Execution starting job
  - Job under execution/halting
  - Job registered in job call stack

- Specify input group numbers (BCD/BIN, parity specification), and variable numbers in the same way as for the CALL command. If the pattern input value is 0, the operation is not executed. A variable number 0 is valid.

- Unit of loading: Select either a single job (JBI) or related jobs (JBR)

- When the NWAIT is specified, the next instruction is executed without waiting completion of job loading.

- While a job is being loaded by the LOADJ command for which NWAIT is specified, if an access is attempted to a job called by the CALL command or JUMP command, an alarm occurs. If a LOADJ or SAVEJ command has already been executed, a job is loaded after completion of the execution.
3.2.2 SAVEJ

3.2.2.1 Function

Saves a specified job as single or related jobs, from the memory of the FS100 to the external memory unit.

3.2.2.2 Configuration

- Specify input group numbers (BCD/BIN, parity specification), and variable numbers in the same way as for the CALL command. If the pattern input value is 0, the operation is not executed. A variable number 0 is valid.
- Unit of saving: Select either a single job (JBI) or related jobs (JBR).
- When the NWAIT is specified, the next command is executed without waiting completion of job saving. When a LOADJ or SAVEJ command has already been executed, a job is saved after completion of the execution.

SAVEJ

JOB:<Job name>
IG#:<Input group No.>
B#<Variable No.>

Unit of saving
JBI, JBR

NWAIT

IF statement
3.2.3 **DELETEJ**

3.2.3.1 Function

Deletes all jobs except its own job or specified jobs as single or related jobs, from the memory of the FS100.

3.2.3.2 Configuration

- Unit of deleting: Select either a single job (JBI) or related jobs (JBR).
- The following jobs cannot be deleted.
  - Execution starting job
  - Job under execution/halting
  - Job registered in job call stack

3.2.4 **SWAIT**

3.2.4.1 Function

Waits for completion of loading or saving jobs or variables.

Use this command to recognize a completion of LOADJ, SAVEJ, LOADV, and SAVEV commands when a NWAIT is specified for these instructions.

3.2.4.2 Configuration
3.3 Commands for Variable Transmission

3.3.1 LOADV

3.3.1.1 Function

Loads the specified global variables from an external memory unit to the FS100 memory.

3.3.1.2 Configuration

3.3.2 SAVEV

3.3.2.1 Function

Saves the specified global variables from the FS100 memory to an external memory unit.

3.3.2.2 Configuration
3.4 Registering DCI Instruction

1. Move the cursor to the address area.

2. Move the cursor to the line where an instruction is to be registered in the job content display.
   - In the job content display in teach mode, move the cursor to the line just above the place where an instruction is to be registered.

3. Press [INFORM LIST].

4. Select an instruction to be registered.
   - The instruction list dialog is displayed.
3.4 Registering DCI Instruction

- The cursor moves to the instruction list dialog while the cursor in the address area changed to an underline. The instruction where the cursor is positioned is displayed with the previously registered additional items in the input buffer line.

5. Change the additional items and variable data.
   - <To register items as displayed in the input buffer>
     1) Perform operation described in step 6 below.

   - <To edit any additional items>
     1) With the cursor on the instruction to be registered, press [SELECT].

- The cursor moves to the input buffer line.

• Changing a numerical value data of additional items
  I) Move the cursor to the additional item whose numerical value is to be changed. Pressing simultaneously [SHIFT] and the cursor key increments or decrements the value.

  II) To enter a value by pressing the number key, press [SELECT] to display the input line.

    Enter a value, then press [ENTER]. The value displayed in the input line is changed.
3 DCI Function
3.4 Registering DCI Instruction

- Adding, changing, or deleting the additional items
  - To add, change or delete the additional items, select an instruction in the input buffer line to display the detail edit display.

  ![Detail Edit Display]

- Adding the additional item
  I) Select “NOT USED” of an additional item selection status, then display the selection dialog.
  II) Select an additional item to be added.

- Changing the data type
  (1) To change the data type of additional item, move the cursor to the of the additional item and press [SELECT] to select a data type.

  ![Data Type Change]

  (2) After having added, changed or deleted the additional items, press [ENTER].
  - The detail edit display is ended and the job content display appears.
6. Press [INSERT] and [ENTER].

- The instruction displayed in the input buffer line is registered.
- To register an instruction just before an END instruction, it is not necessary to press [INSERT].
3.5 Concurrent Tasks from Multiple Jobs

As an option, commands related to DCI function can be executed from more than one job simultaneously. The operations are explained below.

- The DCI related commands can be executed in any job regardless of distinction among the ordinary job, concurrent job (option), or job activated in series (option).

- Multiplexing of DCI transmission function is not supported. Therefore, it is impossible to manipulate files on two or more external memory units (such as personal computer) connected to the FS100.

- If two or more commands related to DCI function are issued concurrently, the execution starts after completion of processing of the currently executing command. Therefore, if a module issues a command request while another module is executing DCI function, the request has to wait until the ongoing processing completes.
3 DCI Function
3.6 DCI Parallel Execution

3.6 DCI Parallel Execution

By using the function described below, the DCI instruction can be executed in parallel with general instructions such as a move instruction and operating instruction.

When this function is used, the robot can be moved or the calculation is executed during data transmission; this function is effective for reduction of tact time, etc.

3.6.1 Parallel Execution Using NWAIT

NOP
MOVJ VJ=50.00
MOVJ VJ=50.00
LOADJ JOB:ABC JBI NWAIT · · · ①
MOVJ VJ=50.00 · · · · · · · · · · · · · · · · · · · ②
MOVJ VJ=50.00 · · · · · · · · · · · · · · · · · · · · · ③
SWAIT · · · · · · · · · · · · · · · · · · · · · · · · · ④
CALL JOB:ABC · · · · · · · · · · · · · · · · · · · ⑤
  · · ·
END

In the above job, when the command ① is executed, loading of the host computer and the job are executed.

Normally, when NWAIT is not specified, the commands of ② and after are not executed until the job loading is completed. However, when NWAIT is specified, the commands ② and ③ are executed sequentially during the job loading; at execution of SWAIT command ④, the execution of command ⑤ is waited for the job “ABC” loading is completed.

At the time of completion of job “ABC” loading, the command ⑤ is executed to execute the job “ABC”.

At this time, if SWAIT command is not specified before the command ⑤, the command ⑤ is executed during the loading of job “ABC”, and an alarm occurs.

Therefore, be sure to verify that loading is completed before executing a job to be loaded, by using SWAIT command.

To load/save variables, be sure to input a SWAIT command before using variables to be loaded/saved as shown below.

(Correct)  (Wrong)
NOP
  · · ·
LOADV B000 NWAIT  LOADV B000 NWAIT
  · · ·
SWAIT  SET B001 B000
SET B001 B000
3.6.2 Parallel Execution Using PSTART (Optional)

By using an independent control command (optional), DCI commands can be executed in parallel with general commands.

For example, to execute the job “R1” for robot 1 is to be executed in parallel with the job “S1” for station 1 during job loading, the following procedure is taken:

Job “R1” : Job for robot 1
Job “S1” : Job for station 1

[JOB:R1]                                           [JOB:S1]
NOP                      NOP
MOVJ VJ=50.00            MOVJ VJ=50.00
MOVJ VJ=50.00            MOVJ VJ=50.00
PSTART JOB:S1 SUB1 ELSE  END
LOADJ JOB:ABC            END
PWAIT . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
CALL JOB:ABC             END

When PSTART command ELSE is executed, the job “S1” starts execution in parallel with the job “R1”.

The job “ABC” is loaded by the command END during execution of the job “S1”; when loading is completed, the FS100 waits for the job “S1” to be completed by the command END.

When the execution of job “S1” is completed, the job “ABC” is executed by the command END.
3.7 Transmission Procedure

3.7.1 Job Transmission

3.7.1.1 Saving Procedure

The transmission from the FS100 to a host computer proceeds as follows.

**FS100 → Host computer**

1. The ENQ code is sent out to establish a data link.
2. After the data link is established, data are sent out to the host computer.
3. After the transmission completes, the FS100 waits for a response from the host computer to verify the completion of transmission. Therefore, the host computer should return a response.
4. The transmission is terminated upon receipt of the response from the host computer.

The data type is distinguished by the header number and the subcode number. Refer to the header number list.

*1 File name: CR (File name does not include extension.)
*2 ACK0 or ACK1
*3 Normal completion: 0000CR (ASCII code)
   Abnormal completion: Integer except 0000 CR (ASCII code)
3.7.1.2 Loading Procedure

The transmission from a host computer to the FS100 proceeds as follows.

**Host computer → FS100**

1. The ENQ code is sent out to establish a data link.
2. After the data link is established, a request to send is sent out to the host computer.
3. When the request to send is accepted, the FS100 enters receiving status, waiting for the ENQ code from the host computer. Therefore, the host computer should send data after the data link is established.
4. The transmission is terminated at completion of data reception from the host computer.

A request to send consists of a header number and a subcode number. Refer to the header number list.

At transmission, memory capacity is checked and if received data cannot be stored, an alarm occurs.

If the transmission itself is normal, reception is continued and an alarm is displayed after the transmission is terminated.

If an error occurs during reception, the job data will not be stored.

---

*1 File name : CR (File name does not include extension.)
*2 ACK0 or ACK1
3.7.2 Variable Transmission

The variable transmission is performed in the same way as for the data as shown below.

3.7.2.1 Saving Procedure

For headers, refer to the header number list.

3.7.2.2 Loading Procedure
Data Transmission Function

3.7 Transmission Procedure

*1

Byte type global variable : [ ] [ ] (0 to 255)

Integer type global variable : ± [ ] [ ] (-32768 to +32767)

Double precision type global variable : ± [ ] [ ] [ ] [ ] [ ] [ ] (-2147483648 to 2137383647)

Real number type global variable : 7 significant digits (-1.70141E+38 to +1.70141E+38)

Position type (robot axis) global variable :
- Pulse type or XYZ type depending on the setting status
- The order varies depending on the number of robot's axes.

Pulse type
1. 6-axis robot
S, L, U, R, B, T (Unit : pulse)
(-999999999 to 999999999)

2. 7-axis robot
S, L, U, R, B, T, E (Unit : pulse)
(-999999999 to 999999999)

XYZ type
1. 6-axis robot
X, Y, Z, Rx, Ry, Rz, TYPE
   d0 = 0 : Flip   d0 = 1 : No flip
   d1 = 0 : Up    d1 = 1 : Back
   Unit : degree (°), significant 4 decimal points
   -9999.9999 to 9999.9999
   Unit : mm, significant 3 decimal points
   -999999.999 to 999999.999

2. 7-axis robot
X, Y, Z, Rx, Ry, Rz, Re, TYPE
   d0 = 0 : Flip   d0 = 1 : No flip
   d1 = 0 : Up    d1 = 1 : Back
   Unit : degree (°), significant 4 decimal points
   -9999.9999 to 9999.9999
   Unit : mm, significant 3 decimal points
   -999999.999 to 999999.999

Position type (base axis) global variable :

Pulse type or XYZ type depending on the internal setting status

Pulse type
1, 2, 3 (Unit : pulse)
(-999999999 to 999999999)

XYZ type
X, Y, Z (Unit : mm, significant 3 decimal points)
(-999999.999 to 999999.999)
The response is as follows when an error occurs in response.

```
SOH 90,004 STX DATA CR ETX
```

If a stop operation (hold and emergency stop) is done during data transmission (while jobs or variables are loaded or saved), the robot stops but the data transmission continues. In this case, the start lamp goes OFF.
The restart will not be accepted until completion of the data transmission.
Data Transmission Function

3. DCI Function
3.8 Axis Data Transmission Format

3.8 Axis Data Transmission Format

The FS100 data transmission function has the following restrictions on transmission of the FS100 internal data.

The robot axes are fixed to a 6-axis set.

A base axis and a station axis are recognized as an external axis.

Up to three base axes are available. With station axis data added after base axis data, up to six axes can be handled.

For example, SAVEV BP005 is read as SAVEV BP005 + EX005.

If the system lacks one of the variables, only the existing one is used.

If the system has both variables but not registered, an error occurs.

The definition of the robot, base, and station axes is used as it is, free of the predetermined axis data R1, B1, and S1.

<Example>

Transmission data of SAVEV in different system configurations are shown below.

- In a system having two base axes (X and Z) and no station axis
  If BP005 is pulse type and 1st axis is 100 and 2nd axis is 200, then
  SAVEV BP005 → 03, 007 100, 200, 0, 0, 0, 0
  If BP005 is XYZ type and X-axis is 123.456 and Z-axis is 234.567, then
  SAVEV BP005 → 03, 008 123.456, 234.567, 0,0, 0, 0

- In a system having no base axis and three station axes
  If EX005 is pulse type and 1st axis is 500, 2nd axis is 600, and 3rd axis is 700
  SAVEV EX005 → 03, 007 500, 600, 700, 0, 0, 0

- In a system having two base axes (X and Z) and three station axes
  If BP005 is pulse type, 1st axis is 100 and 2nd axis is 200, and
  EX005 is pulse type, 1st axis is 500, 2nd axis is 600, and 3rd axis is 700, then
  SAVEV BP005 → 03, 007 100, 200, 500, 600, 700, 0
  (Same as for SAVEV EX005)
  If BP005 is XYZ type, X axis is 123.456, and Z axis is 234.567, and
  EX005 is pulse type, 1st axis is 500, 2nd axis is 600, and 3rd axis is 700, then
  SAVEV BP005 → 03, 008 123.456, 234.567, 500, 600, 700, 0
  (same as for SAVEV EX005)
3.9 **Alarm Codes**

<table>
<thead>
<tr>
<th>Code</th>
<th>Message</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>4104</td>
<td>WRONG EXECUTION OF LOAD INST</td>
<td>Refer to the table below</td>
</tr>
<tr>
<td>4105</td>
<td>WRONG EXECUTION OF SAVE INST</td>
<td></td>
</tr>
<tr>
<td>4106</td>
<td>WRONG EXECUTION OF DELETE INST</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>Insufficient memory capacity</td>
</tr>
<tr>
<td>002</td>
<td>Job editing prohibited</td>
</tr>
<tr>
<td>003</td>
<td>Attempted to load or delete a job being executed.</td>
</tr>
<tr>
<td>004</td>
<td>No specified job</td>
</tr>
<tr>
<td>012</td>
<td>Position data destroyed</td>
</tr>
<tr>
<td>013</td>
<td>Position variable not registered</td>
</tr>
<tr>
<td>017</td>
<td>Instruction destroyed</td>
</tr>
<tr>
<td>019</td>
<td>Invalid character in job name</td>
</tr>
<tr>
<td>020</td>
<td>Invalid character in label</td>
</tr>
<tr>
<td>023</td>
<td>Invalid character in this system</td>
</tr>
<tr>
<td>024</td>
<td>Syntax error</td>
</tr>
<tr>
<td>090</td>
<td>Control command sending/receiving error (Ethernet)</td>
</tr>
<tr>
<td>104</td>
<td>Error response from host computer</td>
</tr>
<tr>
<td>111</td>
<td>Syntax error</td>
</tr>
<tr>
<td>112</td>
<td>Error in position data</td>
</tr>
<tr>
<td>113</td>
<td>No NOP or END instruction</td>
</tr>
<tr>
<td>117</td>
<td>Format error</td>
</tr>
<tr>
<td>118</td>
<td>Invalid number of data</td>
</tr>
<tr>
<td>120</td>
<td>Data range exceeded</td>
</tr>
<tr>
<td>122</td>
<td>Destroyed file exists</td>
</tr>
<tr>
<td>125</td>
<td>No serial port setting</td>
</tr>
<tr>
<td>126</td>
<td>This serial port already used</td>
</tr>
<tr>
<td>127</td>
<td>This protocol already used</td>
</tr>
<tr>
<td>128</td>
<td>File accessing in other function</td>
</tr>
<tr>
<td>211</td>
<td>System block error (Receiving EOT while waiting ACK)</td>
</tr>
<tr>
<td>212</td>
<td>System block error (Receiving EOT at starting receiving)</td>
</tr>
<tr>
<td>213</td>
<td>System block error (Receiving EOT before receiving the last block)</td>
</tr>
<tr>
<td>214</td>
<td>System block error (Receiving codes other than EOT before receiving the last block)</td>
</tr>
<tr>
<td>221</td>
<td>Sending error (Retry for NAK exceeded)</td>
</tr>
<tr>
<td>222</td>
<td>Sending error (Timeup for timer A retry)</td>
</tr>
<tr>
<td>223</td>
<td>Sending error (ACK0/ACK1 order error after retry)</td>
</tr>
<tr>
<td>231</td>
<td>Receiving error (Timeup for timer A while waiting ACK after ENQ, timeup for timer A while waiting ENQ response)</td>
</tr>
<tr>
<td>232</td>
<td>Receiving error (Timeup for timer B while receiving a text)</td>
</tr>
<tr>
<td>233</td>
<td>Receiving error (Heading length is shorter than 6 characters)</td>
</tr>
<tr>
<td>234</td>
<td>Receiving error (Heading length is longer than 6 characters)</td>
</tr>
</tbody>
</table>
### Data Transmission Function

#### 3 DCI Function

##### 3.9 Alarm Codes

<table>
<thead>
<tr>
<th>Data</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>235</td>
<td>Receiving error (Header number error)</td>
</tr>
<tr>
<td>236</td>
<td>Receiving error (Text length exceeds 256 bytes)</td>
</tr>
<tr>
<td>237</td>
<td>Receiving error (Receiving other than ENQ while waiting ENQ, receiving other than control code while waiting control code, receiving other than STX, SOH, ENQ, EOT while waiting text)</td>
</tr>
<tr>
<td>240</td>
<td>Software error</td>
</tr>
<tr>
<td>241</td>
<td>Hardware error (Overrun)</td>
</tr>
<tr>
<td>242</td>
<td>Hardware error (Parity error)</td>
</tr>
<tr>
<td>243</td>
<td>Hardware error (Framing error)</td>
</tr>
<tr>
<td>244</td>
<td>Hardware error (Sending timeup (timer A))</td>
</tr>
<tr>
<td>245</td>
<td>Hardware error (Sending timeup (timer B))</td>
</tr>
</tbody>
</table>
4 Stand-alone Function

4.1 Outline

In stand-alone mode, the file data transmission function is available. By the operations on the FS100 programming pendant, file data can be sent from the FS100 to a host computer such as personal computer to be saved, and from a host computer to the FS100 memory to be loaded.

Load : Transmits file data from a host computer to the FS100.
Save : Transmits file data from the FS100 to a host computer.
Verify : Verifies data between the FS100 and the host computer and informs if some parts are not matched.

The following data can be transmitted between the FS100 and a host computer.

System information can be saved but not loaded.

- Job data
- Condition data/General data
- System information
4.2 Operation Flow

Transmission of file data is performed in the following manner.

Diagram:

1. Select a data group
2. Select a data
3. Press "EXECUTE"
4. End
4.3 Operation

4.3.1 Selecting External Memory Unit

1. Select {EX. MEMORY} under the main menu.
2. Select (DEVICE).
   – The device selection display is shown.
3. Select “DEVICE”.
   – The selection dialog is shown.
4. Select the device to be changed.
   – The device is changed.
4.3.2 Save

The operation to transmit data from the FS100 to the external memory unit is explained in the following.

4.3.2.1 Saving Job

1. Select {EX. MEMORY} under the main menu.
2. Select {SAVE}.
   – The external memory menu display is shown.
3. Select “JOB”.
   – The external memory job list display is shown.
4. Select the job to be saved.
   - The select job is marked with “★”.

5. Press [ENTER].
   - The confirmation dialog is shown.

6. Select “YES”.
   - The job starts to be saved, and the transmission display is shown.
   - To interrupt the saving, press [SELECT].
   When the saving is completed or interrupted, the job content display appears.
4.3.2.2 Saving File

1. Select {EX. MEMORY} under the main menu.

2. Select {SAVE}.
   - The external memory menu display is shown.

3. Select the file group to be saved.
   - The file selection display is shown.
4. Select the file to be saved.
   - The select file is marked with “★”.

5. Press [ENTER].
   - The confirmation dialog is shown.

6. Select “YES”.
   - The file starts to be saved, and the transmission display is shown.
   - To interrupt the saving, press [SELECT].
   When the saving is completed or interrupted, the file selection display reappears.
4.3.3 Load

The operation to transmit data from the external memory unit to the FS100 is explained in the following.

4.3.3.1 Loading Job

1. Select {EX. MEMORY} under the main menu.
2. Select {LOAD}.
   - The external memory menu display is shown.
3. Select “JOB”.
   - The display to input the job name to be loaded is shown.
4. Enter the job to be loaded.
5. Select “EXEC”.

![External Memory Menu Display](image_url)

![Job Input Display](image_url)
4.3.2 Loading File

1. Select {EX. MEMORY} under the main menu.
2. Select {LOAD}.
   - The external memory menu display is shown.

3. Select the file group to be loaded.
   - The file selection display is shown.

   ![External Memory Menu Display](image)
   ![File Selection Display](image)
4. Select the file to be loaded.
   - The selected file is marked with "★".

5. Press [ENTER].
   - The confirmation dialog is shown.

6. Select "YES".
   - The file is starts to be loaded, and the transmission display is shown.
   - To interrupt the loading, press [SELECT].
     When the loading is completed or interrupted, the file selection display reappears.
4.3.4 Job Selection Mode

To select a job to save, load, or verify, the following selection modes are available.

4.3.4.1 Single Selection Mode

Only the selected job is loaded, saved, or verified.

4.3.4.2 Related Selection Mode

The selected job and the related jobs and data files are loaded, saved, or verified.
4.3.4.3 Switching Selection Mode

1. Press the page key in the external memory job list display.
   - Each time the page key is pressed, the displays in "single selection mode" and in "related selection mode" appears alternately.
4.3.5 Selecting Job and Data File

There are two ways to select a job and various data files to be loaded, saved, verified, or deleted.

4.3.5.1 EACH Selection

Selects job and data file one by one.

4.3.5.2 BATCH Selection

Selects all the jobs and data files at once.

For BATCH selection, proceed the following operation.

1. Select (EDIT) of the menu in the external memory job list display or the file selection display.
   - The pull down menu is displayed.
   2. Select (SELECT ALL).

4.4 Transmission Procedure

The transmission procedure is the same as for DCI function.

Refer to section 3.7 “Transmission Procedure” on page 3-13.
5 Host Control Function of FS100

The FS100 supports the host control function which carries out the following file data transfer or robot control according to the commands given by the host computer.

- File data transfer function
- Robot control function

To use the host control function, the following settings should be made.

- The “COMMAND REMOTE” described in chapter 7 “Remote Function Setting”, should be set valid (marked with “●”).
- The parameter RS000 should be set to “2”.
- The host control function should be validated. Whether the host control function is validated, can be verified in the “remote display” described in section 2.1.3 “Display in Command Remote Mode” on page 2-4.

5.1 File Data Transmission Function

According to commands from a host computer, the host control function sends the stored data of user memory of the FS100 to the host computer or receives data from the host computer.

The following data can be transmitted between the FS100 and a host computer.

The system information can be sent only to a host computer.

- Job data
- Condition file/General data
- System information
5.1.1 Transmission Procedure

5.1.1.1 Load

The transmission from a host computer to the FS100 proceeds as follows.

**Host computer → FS100**

1. The ENQ code is sent from the host computer to establish a data link.
2. After the data link is established, the data is sent from the host computer.
3. After the transmission is completed, the host computer should get ready to receive.
4. After the data link is established, a response to the data sent from the host computer is returned from the FS100 to terminate the transmission.

The data type is distinguished by the header number and the subcode number.

Refer to the header number list.

*Fig. 5-1: Loading File Data (Host Control Function)*

---

*1* ACK0 or ACK1

*2* Normal completion: 0000CR (ASCII code)
Abnormal completion: "Integer except 0000"CR (ASCII code)

*3* File name: CR (File name does not include extension)
The transmission from the FS100 to a host computer proceeds as follows.

**FS100 → Host computer**

1. The ENQ is sent from the host computer to establish a data link.
2. After the data link is established, a request to send is sent from the host computer.
3. The request to send consists of a header number and a subcode number. Refer to the header number list.
4. After the request to send is accepted, the host computer should get ready to receive data. The FS100 sends the ENQ code to establish a data link.
5. After the data link is established, receive the data sent from the FS100. The transmission terminates at completion of reception.

If the data requested to send are not found, or the header of the request to send has an error, the FS100 sends the following response message instead of data.

Check the header and take an appropriate action.

```
SOH 90,000 STX DATA CR ETX BCC
```

---

*1 ACK0 or ACK1

*2 File name: CR (File name does not include extension)
### 5.1.2 Data Management

The jobs for the FS100 may refer to another job or condition data according to instructions. When saving a single job or condition data to the host computer, the correspondence between job and files should be controlled.

To reduce this labor, the related jobs and condition data can be transmitted in a batch as the related job data.

When specification of “related job data” is made, the master job, the related job, and the related condition data are transmitted sequentially.

The header number and the subcode number indicate that the related job data are added.

Refer to the header number list.
5.2 Robot Control Function

To control manipulators by a host computer, the host control function can execute the commands listed in the outline.

5.2.1 Command Transmission

The command transmission proceeds as follows.

1. The ENQ code is sent from the host computer to establish a data link.
2. After the data link is established, commands are sent. Commands and file data are distinguished by the header number. Refer to the header number list. The transmission of a command should be completed in a single block. The FS100 cannot receive divided single command, nor receive to execute more than one command in a single block.
3. After the sending is completed, the host computer should get ready to receive. The FS100 sends the ENQ code to establish a data link.
4. After the data link is established, the FS100 sends the response for the command and terminates the transmission. The command format and the response format are explained in the following.

For the command that requires returning data as a response, the response format at normal completion of transmission is as shown in (2).

- **Command Format**
  
  SOH 01,000 STX COMMAND Data1, Data2, Data3 CR ETX BCC

- **Response Format**
  
  (1) SOH 90,000 STX {0000 or Error code} CR ETX BCC

  0000: Normal completion
  Error code: Number with 4 digits other than 0000. In case of smaller than 1000, 0 is added before the number.

  (2) SOH 90,001 STX Data1, Data2, ⋮ DataN CR ETX BCC

If the FS100 cannot execute the sent command, the FS100 returns an interpreter message.

An example of DELETE command (delete a job) is shown.
5 Host Control Function of FS100

5.2 Robot Control Function

Fig. 5-3: Sending Command from Host Computer

![Diagram showing the data transmission process between the host computer and FS100. The diagram includes command and acknowledgment signals with specific byte sequences.](image-url)
### List of Interlock for Commands of Host Control Function

The executable of each command differs depending on the status of the FS100 as shown in the following table.

<table>
<thead>
<tr>
<th>Command Name</th>
<th>Read/Write Enabled</th>
<th>Only Read Enabled</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Read or Monitor</td>
<td></td>
</tr>
<tr>
<td>Read or Monitor</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RALARM</td>
<td>i</td>
<td>i</td>
</tr>
<tr>
<td>RPOSC</td>
<td>i</td>
<td>i</td>
</tr>
<tr>
<td>RPOSJ</td>
<td>i</td>
<td>i</td>
</tr>
<tr>
<td>RSTATS</td>
<td>i</td>
<td>i</td>
</tr>
<tr>
<td>RSTATE</td>
<td>i</td>
<td>i</td>
</tr>
<tr>
<td>RJSEQ</td>
<td>i</td>
<td>i</td>
</tr>
<tr>
<td>JWAIT</td>
<td>i</td>
<td>i</td>
</tr>
<tr>
<td>RGROUP</td>
<td>i</td>
<td>i</td>
</tr>
<tr>
<td></td>
<td>Read or Data</td>
<td></td>
</tr>
<tr>
<td>Read or Data Access</td>
<td>Access</td>
<td></td>
</tr>
<tr>
<td>RJDIR</td>
<td>i</td>
<td>i</td>
</tr>
<tr>
<td>RUDIR</td>
<td>i</td>
<td>i</td>
</tr>
<tr>
<td>UPLOAD</td>
<td>i</td>
<td>i</td>
</tr>
<tr>
<td>SAVEV</td>
<td>i</td>
<td>i</td>
</tr>
<tr>
<td></td>
<td>Operation</td>
<td></td>
</tr>
<tr>
<td>HOLD</td>
<td>i</td>
<td>i</td>
</tr>
<tr>
<td>RESET</td>
<td>i</td>
<td>i</td>
</tr>
<tr>
<td>CANCEL</td>
<td>i</td>
<td>i</td>
</tr>
<tr>
<td>MODE</td>
<td>i</td>
<td>i</td>
</tr>
<tr>
<td>CYCLE</td>
<td>i</td>
<td>i</td>
</tr>
<tr>
<td>SVON 0 (OFF)</td>
<td>i</td>
<td>i</td>
</tr>
<tr>
<td>SVON 1 (ON)</td>
<td>i</td>
<td>i</td>
</tr>
<tr>
<td>HLOCK</td>
<td>i</td>
<td>i</td>
</tr>
<tr>
<td>MDSP</td>
<td>i</td>
<td>i</td>
</tr>
<tr>
<td>CGROUP</td>
<td>i</td>
<td>i</td>
</tr>
<tr>
<td>CTASK</td>
<td>i</td>
<td>i</td>
</tr>
<tr>
<td></td>
<td>Activation</td>
<td></td>
</tr>
<tr>
<td>START</td>
<td>M</td>
<td>i</td>
</tr>
<tr>
<td>MOVJ</td>
<td>M</td>
<td>i</td>
</tr>
<tr>
<td>MOVJ</td>
<td>M</td>
<td>i</td>
</tr>
<tr>
<td>MOVJ</td>
<td>M</td>
<td>i</td>
</tr>
<tr>
<td>IMOV</td>
<td>M</td>
<td>i</td>
</tr>
<tr>
<td>PMOVJ</td>
<td>M</td>
<td>i</td>
</tr>
<tr>
<td>PMOVJ</td>
<td>M</td>
<td>i</td>
</tr>
<tr>
<td></td>
<td>Editing</td>
<td></td>
</tr>
<tr>
<td>DELETE</td>
<td>i</td>
<td>i</td>
</tr>
<tr>
<td>DELETE</td>
<td>i</td>
<td>i</td>
</tr>
<tr>
<td>CVTRJ</td>
<td>i</td>
<td>i</td>
</tr>
<tr>
<td>CVTSJ</td>
<td>i</td>
<td>i</td>
</tr>
<tr>
<td>WUFRAME</td>
<td>i</td>
<td>i</td>
</tr>
<tr>
<td>DOWNLOAD</td>
<td>i</td>
<td>i</td>
</tr>
<tr>
<td>LOADV</td>
<td>i</td>
<td>i</td>
</tr>
<tr>
<td>Job selection</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SETMJ</td>
<td>i</td>
<td>i</td>
</tr>
<tr>
<td>SETMJ</td>
<td>i</td>
<td>i</td>
</tr>
<tr>
<td>JSEQ</td>
<td>i</td>
<td>i</td>
</tr>
</tbody>
</table>

Notes:
- "i" indicates that the command is executable.
- "A" indicates that the command is only readable.
Data Transmission Function

5 Host Control Function of FS100
5.2 Robot Control Function

<Interpreter message>

O : Possible to execute
A : Alarm/error occurring 2060
M : Incorrect mode 2080
H : Hold 2020 to 2050
MOVE : Manipulator moving 2010
C : No command remote setting 2100

*1 "O" if not being held ; "H" if being held
*2 "MOVE" if the manipulator is moving by operation other than command ; "O" if the manipulator is moving by command since a single command can be accepted.
*3 "O" during an alarm ; "A" during error
*4 Only a single job can be executed.
5.2.3 Command that Handle Axis Data

The data transmission function of the FS100 has restrictions on handling control axis data.

1. For the following commands, the order of response data varies whether the number of robot's axes is 6 or 7.
   - Object commands: RPOSJ, RPOSC, MOVJ, MOVL, IMOV, PMOVJ, PMOVL

2. Since the manipulator axes are fixed to a six-axis set, any manipulator having more than seven axes cannot use the following commands.
   - Object commands: RUFRAME, WUFRAME

5.2.4 Response to MOV-type Command

The responses to MOV-type command are as follows.

- If the manipulator is moving by operations other than commands, the interpreter message 2010 (manipulator moving) is returned and the manipulator does not move.
- If the manipulator is in stop status, it turns ON the start lamp and moves according to the command, and returns a response immediately.
- If the manipulator is moving according to the previous commands, only a single command is accepted and the response is held up. After completing execution of the preceding commands, when starting execution of the suspended command, the manipulator returns a response.

This applied to the following commands.

MOVJ, MOVL, IMOV, PMOVJ, PMOVL
5.2.5 Status Read Function

The details of each command are described.

5.2.5.1 Read/Monitor Command

**RALARM**

Reads the error alarm code.

Although the FS100 has the subcode to error code, it cannot read by RALARM because the command has no argument of the subcode.

Command format : RALARM

Response format : Data-1, Data-2, ⋅⋅⋅, Data-10 or Error code

Data-1 = Error code (0 to 9999)
Data-2 = Error data (0 to 256)
Data-3 = Alarm code (0 to 9999)
Data-4 = Alarm data (0 to 256)
Data-5 = Alarm code (0 to 9999)
Data-6 = Alarm data (0 to 256)
Data-7 = Alarm code (0 to 9999)
Data-8 = Alarm data (0 to 256)
Data-9 = Alarm code (0 to 9999)
Data-10 = Alarm data (0 to 256)

**<Example>**

Command  RALARM
Response  0, 1234, 12, 0, 0, 0, 0, 0, 0
5 Host Control Function of FS100
5.2 Robot Control Function

**RPOSJ**
Reads the current position in joint coordinate system.

Command format : RPOSJ

Response format : Data-1, Data-2, ..., Data-12 or Error code

Data-1 = Number of S-axis pulses
Data-2 = Number of L-axis pulses
Data-3 = Number of U-axis pulses
Data-4 = Number of R-axis pulses
Data-5 = Number of B-axis pulses
Data-6 = Number of T-axis pulses
Data-7 = Number of E-axis (7th axis) pulses
Data-8 = Number of 8th axis pulses
Data-9 = Number of 9th axis pulses
Data-10 = Number of 10th axis pulses
Data-11 = Number of 11th axis pulses
Data-12 = Number of 12th axis pulses

<Example>

Command    RPOSJ
Response     500, 2600, 1250, 10789, 624, 36, 0, 0, 0, 0, 0, 0
5.2 Robot Control Function

- **RPOSC**
  Reads the current position in a specified coordinate system. Whether there is an external axis or not can be specified.

  **Command format**: RPOSC Data-1, Data-2  
  Data-1 = Specification of coordinate system  
  0 : Base coordinate  
  1 : Robot coordinate  
  2 : User coordinate 1  
  
  17 : User coordinate 16  
  Data-2 = With or Without external axis  
  0 : Without external axis  
  1 : With external axis

  **Response format**: Data-1, Data-2, ..., Data-14  
  * The order of response data varies depending on the number of robot's axes.

<table>
<thead>
<tr>
<th>6-axis robot</th>
<th>7-axis robot</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data-1</td>
<td>X coordinate value (unit : mm, significant 3 decimal points)</td>
</tr>
<tr>
<td></td>
<td>X coordinate value (unit : mm, significant 3 decimal points)</td>
</tr>
<tr>
<td>Data-2</td>
<td>Y coordinate value (unit : mm, significant 3 decimal points)</td>
</tr>
<tr>
<td></td>
<td>Y coordinate value (unit : mm, significant 3 decimal points)</td>
</tr>
<tr>
<td>Data-3</td>
<td>Z coordinate value (unit : mm, significant 3 decimal points)</td>
</tr>
<tr>
<td></td>
<td>Z coordinate value (unit : mm, significant 3 decimal points)</td>
</tr>
<tr>
<td>Data-4</td>
<td>Wrist angle Rx (unit : degree (°), significant 4 decimal points)</td>
</tr>
<tr>
<td></td>
<td>Wrist angle Rx (unit : degree (°), significant 4 decimal points)</td>
</tr>
<tr>
<td>Data-5</td>
<td>Wrist angle Ry (unit : degree (°), significant 4 decimal points)</td>
</tr>
<tr>
<td></td>
<td>Wrist angle Ry (unit : degree (°), significant 4 decimal points)</td>
</tr>
<tr>
<td>Data-6</td>
<td>Wrist angle Rz (unit : degree (°), significant 4 decimal points)</td>
</tr>
<tr>
<td></td>
<td>Wrist angle Rz (unit : degree (°), significant 4 decimal points)</td>
</tr>
<tr>
<td>Data-7</td>
<td>Type</td>
</tr>
<tr>
<td></td>
<td>Elbow angle Re (unit : degree (°), significant 4 decimal points)</td>
</tr>
<tr>
<td>Data-8</td>
<td>Tool number (0 to 15)</td>
</tr>
<tr>
<td></td>
<td>Tool number (0 to 15)</td>
</tr>
<tr>
<td>Data-9</td>
<td>Number of 7th axis pulses (for travel axis, mm)</td>
</tr>
<tr>
<td></td>
<td>Number of 7th axis pulses (for travel axis, mm)</td>
</tr>
<tr>
<td>Data-10</td>
<td>Number of 8th axis pulses (for travel axis, mm)</td>
</tr>
<tr>
<td></td>
<td>Number of 8th axis pulses (for travel axis, mm)</td>
</tr>
<tr>
<td>Data-11</td>
<td>Number of 9th axis pulses (for travel axis, mm)</td>
</tr>
<tr>
<td></td>
<td>Number of 9th axis pulses (for travel axis, mm)</td>
</tr>
<tr>
<td>Data-12</td>
<td>Number of 10th axis pulses</td>
</tr>
<tr>
<td></td>
<td>Number of 10th axis pulses</td>
</tr>
<tr>
<td>Data-13</td>
<td>Number of 11th axis pulses</td>
</tr>
<tr>
<td></td>
<td>Number of 11th axis pulses</td>
</tr>
<tr>
<td>Data-14</td>
<td>Number of 12th axis pulses</td>
</tr>
<tr>
<td></td>
<td>Number of 12th axis pulses</td>
</tr>
<tr>
<td>Data-15</td>
<td>-</td>
</tr>
</tbody>
</table>

- "Number of 7th axis pulses" and after are added only when "With external axis" is specified.
- If the specified user coordinate system is undefined, an error occurs.
- The data of type is represented by the following bit data coded into a decimal number.
5. Host Control Function of FS100

5.2 Robot Control Function

Example

Command: RPOSC 2, 0
Response: 100.0, 50, 34, 12.34, 180.0, 0, 0, 0, 0, 0, 0
### RSTATS

Reads the status of mode, cycle, operation, alarm error, and servo.

Command format: RSTATS

Response format: Data-1, Data-2 or Error code

**Data-1**

<table>
<thead>
<tr>
<th>MSB</th>
<th>LSB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step</td>
<td>1 cycle</td>
</tr>
<tr>
<td>Auto</td>
<td>Running</td>
</tr>
<tr>
<td>Safety speed operation</td>
<td>Teach</td>
</tr>
<tr>
<td>Play</td>
<td>Command remote</td>
</tr>
</tbody>
</table>

**Data-2**

<table>
<thead>
<tr>
<th>MSB</th>
<th>LSB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hold (programming pendant)</td>
<td>Hold (external)</td>
</tr>
<tr>
<td>Hold (command)</td>
<td>Alarm occurring</td>
</tr>
<tr>
<td>Error occurring</td>
<td>Servo ON</td>
</tr>
</tbody>
</table>

**Example**

Command: RSTATS
Response: 1, 0
5 Host Control Function of FS100
5.2 Robot Control Function

- **RJSEQ**
  Reads the current job name, line No. and step No.
  
  Command format : RJSEQ

  Response format : Data-1, Data-2, Data-3 or Error code
  Data-1 = Read job name
  Data-2 = Read line No. (0 to 9999)
  Data-3 = Read step No. (0 to 999)

  <Example>
  
  Command  RJSEQ
  Response   WORK-A, 10, 5
Data Transmission Function

5 Host Control Function of FS100
5.2 Robot Control Function

- **JWAIT**
  - JWAIT is for checking operations (job) of the manipulator.
  - If a response is returned immediately after the job is started, in such a case with START command, completion of the job cannot be known.
  - Specify a waiting time as an operand for JWAIT command.
  - No response is sent out until the operation of manipulator is completed or the specified waiting time has elapsed.
  - JWAIT returns as a response, the information whether the operation has completed or not.
  
  **Command format:** JWAIT Time
  - Time = Waiting time (-1.0 to 32767 sec.)
  - -1.0 indicates infinite time.

  **Response format:** Data or Error code
  - Data = Operation status (0: completed, -1: not completed)

  Waits for stop of job execution.
  The response varies depending on the following status.

<table>
<thead>
<tr>
<th>Status</th>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>END or PAUSE execution during waiting time</td>
<td>Completed</td>
</tr>
<tr>
<td>Step execution during waiting time</td>
<td>Completed</td>
</tr>
<tr>
<td>Stopped by hold, alarm, emergency stop, servo OFF during waiting time</td>
<td>Not completed</td>
</tr>
<tr>
<td>Stopped by changing mode during waiting time</td>
<td>Not completed</td>
</tr>
<tr>
<td>Test run is interrupted during waiting time</td>
<td>Not completed</td>
</tr>
<tr>
<td>Waiting timeup</td>
<td>Not completed</td>
</tr>
<tr>
<td>Stopped (including when the control power ON)</td>
<td>Completed</td>
</tr>
<tr>
<td>Stopped (hold)</td>
<td>Interpreter message 2020 to 2050</td>
</tr>
<tr>
<td>Stopped (Alarm/error occurring)</td>
<td>Interpreter message 2060</td>
</tr>
<tr>
<td>Stopped (servo OFF)</td>
<td>Interpreter message 2070</td>
</tr>
</tbody>
</table>

**<Example>**

Command  JWAIT 10
Response   0000
**Data Transmission Function**

5 Host Control Function of FS100

5.2 Robot Control Function

### RGROUP

Reads the current control group set by CGROUP command or CTASK command, and the task selection status.

**Command format:** RGROUP

**Response format:** Data-1, Data-2, Data-3 or Error code

Data-1 = Robot control group information.
The control group information differs depending on the number of manipulators in the system.

Data-2 = Station control group information.
The control group information differs depending on the number of manipulators in the system.

Data-3 = Task information

In a system where independent control is not allowed, “0” is returned.

**Example**

Command: RGROUP
Response: 2, 1, 0

The above example shows that the current control group is robot1, robot 2 and station 1, and the task selection status is master task.
5.2.5.2 Read/Data Access System Commands

- **RJDIR**

  Reads all job names, or the names of jobs related to the parent job.

  **Command format**:
  
  RJDIR Job-Name
  
  Job-Name
  
  All the job names currently registered
  
  Parent job name
  
  If a parent job name is specified, RJDIR reads the name of related jobs excluding the parent job.
  
  If there is no related child job, the command returns the null list.
  
  If the parent job has related child jobs but they are not registered in the system, an error occurs.

  **Response format**:
  
  Name-1, Name-2, ..., Name-N or Error code
  
  Name-1  =  Job name-1 (32 characters)
  
  Name-2  =  Job name-2 (32 characters)
  
  Name-N  =  Job name-N (32 characters)

  **Example**

  **Command**: RJDIR MASTER-1
  
  **Response**: WORK-A, WORK-B, SAMPLE-1
RUFRAME
Reads a specified user coordinate data.

Command format: RUFRAME Data-1
Data-1 = User coordinate No.
0 : Reserved
1 : Reserved
2 : User coordinate 1
:
:
17 : User coordinate 16

Response format: Data-1, Data-2, ..., Data-28
Data-1 = ORG X coordinate value (unit: mm, significant 3 decimal points)
Data-2 = ORG Y coordinate value (unit: mm, significant 3 decimal points)
Data-3 = ORG Z coordinate value (unit: mm, significant 3 decimal points)
Data-4 = ORG wrist angle TX (unit: degree (°), significant 4 decimal points)
Data-5 = ORG wrist angle TY (unit: degree (°), significant 4 decimal points)
Data-6 = ORG wrist angle TZ (unit: degree (°), significant 4 decimal points)
Data-7 = ORG type
Data-8 = XX X coordinate value (unit: mm, significant 3 decimal points)
Data-9 = XX Y coordinate value (unit: mm, significant 3 decimal points)
Data-10 = XX Z coordinate value (unit: mm, significant 3 decimal points)
Data-11 = XX wrist angle TX (unit: degree (°), significant 4 decimal points)
Data-12 = XX wrist angle TY (unit: degree (°), significant 4 decimal points)
Data-13 = XX wrist angle TZ (unit: degree (°), significant 4 decimal points)
Data-14 = XX type
Data-15 = XY X coordinate value (unit: mm, significant 3 decimal points)
Data-16 = XY Y coordinate value (unit: mm, significant 3 decimal points)
Data-17 = XY Z coordinate value (unit: mm, significant 3 decimal points)
Data-18 = XY wrist angle TX (unit: degree (°), significant 4 decimal points)
Data-19 = XY wrist angle TY (unit: degree (°), significant 4 decimal points)
Data-20 = XY wrist angle TZ (unit: degree (°), significant 4 decimal points)
Data-21 = XY type
Data-22 = Tool No. (0 to 15)
Data-23 = Number of 7th axis pulses (for travel axis, mm)
Data-24 = Number of 8th axis pulses (for travel axis, mm)
Data Transmission Function

5 Host Control Function of FS100

5.2 Robot Control Function

- Data-25 = Number of 9th axis pulses (for travel axis, mm)
- Data-26 = Number of 10th axis pulses
- Data-27 = Number of 11th axis pulses
- Data-28 = Number of 12th axis pulses
  • ORG, XX, XY coordinates are read in the base coordinate system.
  • In a system having no external axis, Data-23 to Data-28 are “0”.
  • If the specified user coordinate system is not registered, an error occurs.
  • If the group axis of the specified user coordinate system is not R1, an error occurs.
  • If ORG, XX, and XY have different base axis data, an error occurs.
  • For 7-axis robots, this command cannot be used.

Example

Command  RUFRAME 2
Response  600.0, 12.34, 500.0, 180.0, 0.0, 0.0, 0, ..., 0
5. Host Control Function of FS100

5.2 Robot Control Function

### SAVEV
Sends variable data to a host computer.

**Command format:** `SAVEV Data-1, Data-2`  
**Data-1:** Type of variables  
0: Byte type variables  
1: Integer type variables  
2: Double precision type variables  
3: Real number type variables  
4: Robot axis position type variables  
5: Base axis position type variables  
6: Station axis position type variables (only pulse type)  
7: String variables  
**Data-2:** Variable No.

**Response format:** `Data-1, Data-2, Data-3, ..., Data-11` or Error code  
**Data-1 =** Byte value / Integer value / Double precision integer value / Real number value / Position data type / String

Position data type = 0: Pulse type  
1: Cartesian type

*(When the position data type is “0”)*  
*The order of response data varies depending on the number of robot's axes.*

<table>
<thead>
<tr>
<th>6-axis robot</th>
<th>7-axis robot</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data-2</td>
<td>Number of S-axis pulses / Number of base 1st axis pulses / Number of station 1st axis pulses</td>
</tr>
<tr>
<td>Data-3</td>
<td>Number of L-axis pulses / Number of base 2nd axis pulses / Number of station 2nd axis pulses</td>
</tr>
<tr>
<td>Data-4</td>
<td>Number of U-axis pulses / Number of base 3rd axis pulses / Number of station 3rd axis pulses</td>
</tr>
<tr>
<td>Data-5</td>
<td>Number of R-axis pulses / Number of base 4th axis pulses / Number of station 4th axis pulses</td>
</tr>
<tr>
<td>Data-6</td>
<td>Number of B-axis pulses / Number of base 5th axis pulses / Number of station 5th axis pulses</td>
</tr>
<tr>
<td>Data-7</td>
<td>Number of T-axis pulses / Number of base 6th axis pulses / Number of station 6th axis pulses</td>
</tr>
<tr>
<td>Data-8</td>
<td>Tool No. (0 to 15)</td>
</tr>
<tr>
<td>Data-9</td>
<td>-</td>
</tr>
<tr>
<td>Data-10</td>
<td>-</td>
</tr>
<tr>
<td>Data-11</td>
<td>-</td>
</tr>
</tbody>
</table>
(When the position data type is “1”)
Data-2 = Coordinate data
0 : Base coordinate
1 : Robot coordinate
2 : User coordinate 1
3 : User coordinate 2
:
17 : User coordinate 16
18 : Tool coordinate
19 : Master tool coordinate
* The order of response data varies depending on the number of robot’s axes.
When the system contains multiple robots, the order is that of the robot with the maximum number of axes.

<table>
<thead>
<tr>
<th>6-axis robot</th>
<th>7-axis robot</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data-3</td>
<td>Data-3</td>
</tr>
<tr>
<td>X coordinate value / Base 1st Cartesian value (unit : mm, significant 3 decimal points)</td>
<td>X coordinate value / Base 1st Cartesian value (unit : mm, significant 3 decimal points)</td>
</tr>
<tr>
<td>Data-4</td>
<td>Data-4</td>
</tr>
<tr>
<td>Y coordinate value / Base 2nd Cartesian value (unit : mm, significant 3 decimal points)</td>
<td>Y coordinate value / Base 2nd Cartesian value (unit : mm, significant 3 decimal points)</td>
</tr>
<tr>
<td>Data-5</td>
<td>Data-5</td>
</tr>
<tr>
<td>Z coordinate value / Base 3rd Cartesian value (unit : mm, significant 3 decimal points)</td>
<td>Z coordinate value / Base 3rd Cartesian value (unit : mm, significant 3 decimal points)</td>
</tr>
<tr>
<td>Data-6</td>
<td>Data-6</td>
</tr>
<tr>
<td>Wrist angle Rx coordinate value (unit : degree (°), significant 4 decimal points)</td>
<td>Wrist angle Rx coordinate value (unit : degree (°), significant 4 decimal points)</td>
</tr>
<tr>
<td>Data-7</td>
<td>Data-7</td>
</tr>
<tr>
<td>Wrist angle Ry coordinate value (unit : degree (°), significant 4 decimal points)</td>
<td>Wrist angle Ry coordinate value (unit : degree (°), significant 4 decimal points)</td>
</tr>
<tr>
<td>Data-8</td>
<td>Data-8</td>
</tr>
<tr>
<td>Wrist angle Rz coordinate value (unit : degree (°), significant 4 decimal points)</td>
<td>Wrist angle Rz coordinate value (unit : degree (°), significant 4 decimal points)</td>
</tr>
<tr>
<td>Data-9</td>
<td>Data-9</td>
</tr>
<tr>
<td>Form</td>
<td>Elbow angle Re coordinate value (unit : degree (°), significant 4 decimal points)</td>
</tr>
<tr>
<td>Data-10</td>
<td>Data-10</td>
</tr>
<tr>
<td>Tool No. (0 to 15)</td>
<td>Form</td>
</tr>
<tr>
<td>Data-11</td>
<td>Data-11</td>
</tr>
<tr>
<td>-</td>
<td>Tool No. (0 to 15)</td>
</tr>
</tbody>
</table>
Data of the form is a value obtained by converting the following bit data to decimal notation.

<Example>
Command  SAVEV 0, 0  
Response  123

In the above example, 123, the value of byte type variable B000, is sent to the host computer.
5.2.6 System Control Function

5.2.6.1 Operation System Commands

- **HOLD**
  Turns HOLD ON/OFF.
  Command format: HOLD [Data]
  Data = Specification of HOLD ON/OFF status (0 : OFF, 1 : ON)
  Response format: 0000 or Error code

  <Example>
  Command  HOLD 1
  Response  0000

- **RESET**
  Resets an alarm of manipulator.
  The transmission alarms can be reset only by the programming pendant.
  Command format: RESET
  Response format: 0000 or Error code

  <Example>
  Command  RESET
  Response  0000

- **CANCEL**
  Cancels an error.
  Command format: CANCEL
  Response format: 0000 or Error code

  <Example>
  Command  CANCEL
  Response  0000
### 5.2 Robot Control Function

#### MODE
Selects a mode.

- **Command format**: MODE Mode-No
- **Mode-No.** = 1 or 2
  - 1 : Teach mode
  - 2 : Play mode

- **Response format**: 0000 or Error code

**<Example>**

- **Command**: MODE 2
- **Response**: 0000

**NOTE** This function can be used when the external mode switch is permitted on the OPERATING CONDITION window.

#### CYCLE
Selects cycle.

- **Command format**: CYCLE Cycle-No
- **Cycle-No** = Cycle specification (1 to 3)
  - 1 : Step
  - 2 : 1 cycle
  - 3 : Auto

- **Response format**: 0000 or Error code

**<Example>**

- **Command**: CYCLE 2
- **Response**: 0000

#### SVON
Turns servo power supply ON/OFF.

To turn the servo ON/OFF by this command, connect the external servo ON (EXSVON) signal 29 of the input terminal block for the manipulator, to 30.

- **Command format**: SVON Data
  - **Data** = Specification of servo power supply ON/OFF status
  - (0 : OFF, 1 : ON)

- **Response format**: 0000 or Error code

**<Example>**

- **Command**: SVON 1
- **Response**: 0000
HLOCK
Sets an interlock between the programming pendant and I/O operation signals.

While the interlock is ON, all operations except the followings are prohibited.

- Emergency stop from the programming pendant
- Input signals except I/O mode change, external start, external servo ON, cycle change, I/O prohibited, P.P/PANEL prohibited, and master call

HLOCK is invalid while the programming pendant is in edit mode or accessing to a file for other function.

Command format : HLOCK Data
Data = Interlock status setting (0 : OFF, 1 : ON)

Response format : 0000 or Error code

<Example>
Command  HLOCK 1
Response  0000
MDSP
Receives message data and displays the message in the remote display of the programming pendant.

If the currently shown display is not the remote display, it is changed forcibly to the remote display to display the MDSP command message.

Command format: MDSP Data
Data = Message to be displayed (Max. 30 characters)

Response format: 0000 or Error code

<Example>
Command: MDSP auto running
Response: 0000
### CGROUP
Changes an objective control group of various commands used in the host control function.

The FS100 can support multiple number of manipulators and stations. In this case, CGROUP is used when any control group for commands such as RPOSC is to be changed.

When the power supply is started up, robot 1, base 1, and station 1 (when a base and a stations exist) are specified.

Command format: CGROUP Data-1, Data-2
- Data-1 = Robot control group specification.
- A control group can be specified according to the following data.
  - Selection of control axis which does not exist
  - Specification of multiple number of manipulators

In a system with a base axis (such as travel axis), when the manipulator with this base axis is specified, this base axis is automatically specified.
Data-2 = Station control group specification. A control group can be specified according to the following data. However, the following settings cannot be made.

- Selection of control axis which does not exist
- Specification of multiple number of stations

Response format: 0000 or Error code

**<Example>**

Command CGROUP 2, 1  
Response 0000

In the above example with two manipulators, robot 2 and station 1 are validated. By issuing RPOS after this command is issued, the current positions of robot 2 and station 1 can be read.
CTASK (Optional)

Changes the task for control in the host control function.

When the power supply is started up or in a system where an independent control is not allowed, this command is to be used as follows.

For details, refer to section 5.3 “Commands for Multi-control Group and Independent Control Functions” on page 5-50.

- When the power supply is started up, a master task is selected as a task to be controlled.
- CTASK cannot be used in a system where an independent control is not allowed.

Command format : CTASK Data-1

Data-1 = Specified task

0 : Master task
1 : Sub 1 task
2 : Sub 2 task
3 : Sub 3 task
4 : Sub 4 task
5 : Sub 5 task

Response format : 0000 or Error code

<Example>

Command  CTASK 1
Response  0000
5.2.6.2 Start-up System Commands

- **START**
  Starts a job.

  If a job name is specified for an operand, the relation between the job and the master job is checked and the execution is started from the beginning of the job.

  If no job name is specified, the execution is started from the current line number of the set execution job.

  Command format: START [Job-Name]
  Job-Name = Starting job name (32 characters)
  Can be omitted.

  Response format: 0000 or Error code

  <Example>
  Command: START WORK-A
  Response: 0000
### 5.2 Robot Control Function

#### MOVJ

Moves a manipulator to a specified coordinate position in joint motion.

- **Command format:** MOVJ Data-1, Data-2, ⋯, Data-17
  - **Data-1** = Motion speed (0.01 to 100.0%)
  - **Data-2** = Coordinate specification
    - 0 : Base coordinate
    - 1 : Robot coordinate
    - 2 : User coordinate 1
    - ⋯
    - 17 : User coordinate 16
  - The order varies depending on the number of robot's axes.

* In a system without external axis, Data-11 to Data-16 (for 7-axis robots, Data-12 to Data-17) should be set to “0”.

* If a specified user coordinate is not defined, an error occurs.

**Response format:** 0000 or Error code

**Example**

**Command**  MOVJ 50.0, 2, 123.1, 50.34, 10.8, 180.0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0

**Response**  0000

---

<table>
<thead>
<tr>
<th>6-axis robot</th>
<th>7-axis robot</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data-3</td>
<td>X coordinate value (unit : mm, significant 3 decimal points)</td>
</tr>
<tr>
<td>Data-4</td>
<td>Y coordinate value (unit : mm, significant 3 decimal points)</td>
</tr>
<tr>
<td>Data-5</td>
<td>Z coordinate value (unit : mm, significant 3 decimal points)</td>
</tr>
<tr>
<td>Data-6</td>
<td>Wrist angle Rx (unit : degree (°), significant 4 decimal points)</td>
</tr>
<tr>
<td>Data-7</td>
<td>Wrist angle Ry (unit : degree (°), significant 4 decimal points)</td>
</tr>
<tr>
<td>Data-8</td>
<td>Wrist angle Rz (unit : degree (°), significant 4 decimal points)</td>
</tr>
<tr>
<td>Data-9</td>
<td>Type</td>
</tr>
<tr>
<td>Data-10</td>
<td>Tool No. (0 to 15)</td>
</tr>
<tr>
<td>Data-11</td>
<td>Number of 7th axis pulses (for travel axis, mm)</td>
</tr>
<tr>
<td>Data-12</td>
<td>Number of 8th axis pulses (for travel axis, mm)</td>
</tr>
<tr>
<td>Data-13</td>
<td>Number of 9th axis pulses (for travel axis, mm)</td>
</tr>
<tr>
<td>Data-14</td>
<td>Number of 10th axis pulses</td>
</tr>
<tr>
<td>Data-15</td>
<td>Number of 11th axis pulses</td>
</tr>
<tr>
<td>Data-16</td>
<td>Number of 12th axis pulses</td>
</tr>
<tr>
<td>Data-17</td>
<td>Number of 13th axis pulses</td>
</tr>
</tbody>
</table>

---

- In a system without external axis, Data-11 to Data-16 (for 7-axis robots, Data-12 to Data-17) should be set to “0”.
- If a specified user coordinate is not defined, an error occurs.
Data Transmission Function

5.2 Robot Control Function

**MOVL**
Moves a manipulator to a specified coordinate position in linear motion.

Command format: MOVL Data-1, Data-2, ..., Data-18

- **Data-1** = Motion speed selection (0 : V (speed), 1 : VR (posture speed))
- **Data-2** = Motion speed (0.1 to \(9999\) mm/s, 0.1 to \(9999\) \(^\circ\)/s)
- **Data-3** = Coordinate specification
  - 0 : Base coordinate
  - 1 : Robot coordinate
  - 2 : User coordinate 1
  - ....
  - 17 : User coordinate 16

* The order varies depending on the number of robot's axes.

<table>
<thead>
<tr>
<th>6-axis robot</th>
<th>7-axis robot</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data-4</td>
<td>X coordinate value (unit : mm, significant 3 decimal points)</td>
</tr>
<tr>
<td>Data-5</td>
<td>Y coordinate value (unit : mm, significant 3 decimal points)</td>
</tr>
<tr>
<td>Data-6</td>
<td>Z coordinate value (unit : mm, significant 3 decimal points)</td>
</tr>
<tr>
<td>Data-7</td>
<td>Wrist angle Rx (unit : degree ((^\circ)), significant 4 decimal points)</td>
</tr>
<tr>
<td>Data-8</td>
<td>Wrist angle Ry (unit : degree ((^\circ)), significant 4 decimal points)</td>
</tr>
<tr>
<td>Data-9</td>
<td>Wrist angle Rz (unit : degree ((^\circ)), significant 4 decimal points)</td>
</tr>
<tr>
<td>Data-10</td>
<td>Type</td>
</tr>
<tr>
<td>Data-11</td>
<td>Tool No. (0 to 15)</td>
</tr>
<tr>
<td>Data-12</td>
<td>Number of 7th axis pulses (for travel axis, mm)</td>
</tr>
<tr>
<td>Data-13</td>
<td>Number of 8th axis pulses (for travel axis, mm)</td>
</tr>
<tr>
<td>Data-14</td>
<td>Number of 9th axis pulses (for travel axis, mm)</td>
</tr>
<tr>
<td>Data-15</td>
<td>Number of 10th axis pulses</td>
</tr>
<tr>
<td>Data-16</td>
<td>Number of 11th axis pulses</td>
</tr>
<tr>
<td>Data-17</td>
<td>Number of 12th axis pulses</td>
</tr>
<tr>
<td>Data-18</td>
<td>-</td>
</tr>
</tbody>
</table>

* In a system without external axis, Data-12 to Data-17 (for 7-axis robots, Data-13 to Data-18) should be set to "0".
* If a specified user coordinate is not defined, an error occurs.
Response format: 0000 or Error code

<Example>
Command: MOVL 0, 500.0, 2, 123.1, 50.34, 10.8, 180.0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
Response: 0000
IMOV
Moves a manipulator from the current position for a specified coordinate incremental value in linear motion.

Command format : IMOV Data-1, Data-2, ..., Data-18
Data-1 = Motion speed selection (0 : V (speed), 1 : VR (posture speed))
Data-2 = Motion speed (0.1 to 10.0 mm/s, 0.1 to 10.0 °/s)
Data-3 = Coordinate specification
0 : Base coordinate
1 : Robot coordinate
2 : User coordinate 1
: 
17 : User coordinate 16
18 : Tool coordinate
* The order varies depending on the number of robot's axes.

<table>
<thead>
<tr>
<th>6-axis robot</th>
<th>7-axis robot</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data-4</td>
<td>X coordinate incremental value (unit : mm, significant 3 decimal points)</td>
</tr>
<tr>
<td>Data-5</td>
<td>X coordinate incremental value (unit : mm, significant 3 decimal points)</td>
</tr>
<tr>
<td>Data-6</td>
<td>Y coordinate incremental value (unit : mm, significant 3 decimal points)</td>
</tr>
<tr>
<td>Data-7</td>
<td>Y coordinate incremental value (unit : mm, significant 3 decimal points)</td>
</tr>
<tr>
<td>Data-8</td>
<td>Z coordinate incremental value (unit : mm, significant 3 decimal points)</td>
</tr>
<tr>
<td>Data-9</td>
<td>Z coordinate incremental value (unit : mm, significant 3 decimal points)</td>
</tr>
<tr>
<td>Data-10</td>
<td>Wrist angle Rx incremental value (unit : degree (°), significant 4 decimal points)</td>
</tr>
<tr>
<td>Data-11</td>
<td>Wrist angle Rx incremental value (unit : degree (°), significant 4 decimal points)</td>
</tr>
<tr>
<td>Data-12</td>
<td>Wrist angle Ry incremental value (unit : degree (°), significant 4 decimal points)</td>
</tr>
<tr>
<td>Data-13</td>
<td>Wrist angle Ry incremental value (unit : degree (°), significant 4 decimal points)</td>
</tr>
<tr>
<td>Data-14</td>
<td>Wrist angle Rz incremental value (unit : degree (°), significant 4 decimal points)</td>
</tr>
<tr>
<td>Data-15</td>
<td>Wrist angle Rz incremental value (unit : degree (°), significant 4 decimal points)</td>
</tr>
<tr>
<td>Data-16</td>
<td>Elbow angle Re incremental value (unit : degree (°), significant 4 decimal points)</td>
</tr>
<tr>
<td>Data-17</td>
<td>Elbow angle Re incremental value (unit : degree (°), significant 4 decimal points)</td>
</tr>
<tr>
<td>Data-18</td>
<td>Tool No. (0 to 15)</td>
</tr>
<tr>
<td>Data-19</td>
<td>Tool No. (0 to 15)</td>
</tr>
<tr>
<td>Data-20</td>
<td>Number of 7th axis pulses (for travel axis, mm)</td>
</tr>
<tr>
<td>Data-21</td>
<td>Number of 7th axis pulses (for travel axis, mm)</td>
</tr>
<tr>
<td>Data-22</td>
<td>Number of 8th axis pulses (for travel axis, mm)</td>
</tr>
<tr>
<td>Data-23</td>
<td>Number of 8th axis pulses (for travel axis, mm)</td>
</tr>
<tr>
<td>Data-24</td>
<td>Number of 9th axis pulses (for travel axis, mm)</td>
</tr>
<tr>
<td>Data-25</td>
<td>Number of 9th axis pulses (for travel axis, mm)</td>
</tr>
<tr>
<td>Data-26</td>
<td>Number of 10th axis pulses (for travel axis, mm)</td>
</tr>
<tr>
<td>Data-27</td>
<td>Number of 10th axis pulses (for travel axis, mm)</td>
</tr>
<tr>
<td>Data-28</td>
<td>Number of 11th axis pulses</td>
</tr>
<tr>
<td>Data-29</td>
<td>Number of 11th axis pulses</td>
</tr>
<tr>
<td>Data-30</td>
<td>Number of 12th axis pulses</td>
</tr>
<tr>
<td>Data-31</td>
<td>Number of 12th axis pulses</td>
</tr>
</tbody>
</table>

• In a system without external axis, Data-12 to Data-17 (for 7-axis robots, Data-13 to Data-18) should be set to “0”.
• If a specified user coordinate is not defined, an error occurs.
5 Host Control Function of FS100
5.2 Robot Control Function

Response format: 0000 or Error code

<Example>
Command IMOV 0, 100.0, 2, 10.0, 10.0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
Response 0000
### 5.2 Robot Control Function

**PMOVJ**
Moves a manipulator to a specified pulse position in joint motion.

Command format: PMOVJ Data-1, Data-2, …, Data-15

* The order varies depending on the number of robot’s axes.

<table>
<thead>
<tr>
<th>6-axis robot</th>
<th>7-axis robot</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data-1 Motion speed (0.01 to 100.0 %)</td>
<td>Motion speed (0.01 to 100.0 %)</td>
</tr>
<tr>
<td>Data-2 Number of S-axis pulses</td>
<td>Number of S-axis pulses</td>
</tr>
<tr>
<td>Data-3 Number of L-axis pulses</td>
<td>Number of L-axis pulses</td>
</tr>
<tr>
<td>Data-4 Number of U-axis pulses</td>
<td>Number of U-axis pulses</td>
</tr>
<tr>
<td>Data-5 Number of R-axis pulses</td>
<td>Number of R-axis pulses</td>
</tr>
<tr>
<td>Data-6 Number of B-axis pulses</td>
<td>Number of B-axis pulses</td>
</tr>
<tr>
<td>Data-7 Number of T-axis pulses</td>
<td>Number of T-axis pulses</td>
</tr>
<tr>
<td>Data-8 Tool No. (0 to 15)</td>
<td>Number of E-axis pulses</td>
</tr>
<tr>
<td>Data-9 Number of 7th axis pulses</td>
<td>Tool No. (0 to 15)</td>
</tr>
<tr>
<td>Data-10 Number of 8th axis pulses</td>
<td>Number of 7th axis pulses</td>
</tr>
<tr>
<td>Data-11 Number of 9th axis pulses</td>
<td>Number of 8th axis pulses</td>
</tr>
<tr>
<td>Data-12 Number of 10th axis pulses</td>
<td>Number of 9th axis pulses</td>
</tr>
<tr>
<td>Data-13 Number of 11th axis pulses</td>
<td>Number of 10th axis pulses</td>
</tr>
<tr>
<td>Data-14 Number of 12th axis pulses</td>
<td>Number of 11th axis pulses</td>
</tr>
<tr>
<td>Data-15 -</td>
<td>Number of 12th axis pulses</td>
</tr>
</tbody>
</table>

* In a system without external axis, Data-9 to Data-14 (for 7-axis robots, Data-10 to Data-15) should be set to “0”.

Response format: 0000 or Error code

**<Example>**

Command PMOVJ 20.0, 100, 200, 300, 400, 500, 0, 0, 0, 0, 0, 0, 0
Response 0000
5 Host Control Function of FS100

5.2 Robot Control Function

PMOVL
Moves a manipulator to a specified pulse position in linear motion.

Command format: PMOVL Data-1, Data-2, ..., Data-16
Data-1 = Motion speed selection (0: V (speed), 1: VR (posture speed))
Data-2 = Motion speed (0.1 to 999.9 mm/s, 0.1 to 999.9 °/s)
* The order varies depending on the number of robot's axes.

Response format: 0000 or Error code

<Example>
Command: PMOVL 0, 123.0, 10, 200, 300, 400, 500, 0, 0, 0, 0, 0, 0, 0, 0
Response: 0000

<table>
<thead>
<tr>
<th>6-axis robot</th>
<th>7-axis robot</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data-3</td>
<td>Number of S-axis pulses</td>
</tr>
<tr>
<td>Data-4</td>
<td>Number of L-axis pulses</td>
</tr>
<tr>
<td>Data-5</td>
<td>Number of U-axis pulses</td>
</tr>
<tr>
<td>Data-6</td>
<td>Number of R-axis pulses</td>
</tr>
<tr>
<td>Data-7</td>
<td>Number of B-axis pulses</td>
</tr>
<tr>
<td>Data-8</td>
<td>Number of T-axis pulses</td>
</tr>
<tr>
<td>Data-9</td>
<td>Tool No. (0 to 15)</td>
</tr>
<tr>
<td>Data-10</td>
<td>Number of 7th axis pulses</td>
</tr>
<tr>
<td>Data-11</td>
<td>Number of 8th axis pulses</td>
</tr>
<tr>
<td>Data-12</td>
<td>Number of 9th axis pulses</td>
</tr>
<tr>
<td>Data-13</td>
<td>Number of 10th axis pulses</td>
</tr>
<tr>
<td>Data-14</td>
<td>Number of 11th axis pulses</td>
</tr>
<tr>
<td>Data-15</td>
<td>Number of 12th axis pulses</td>
</tr>
<tr>
<td>Data-16</td>
<td>-</td>
</tr>
</tbody>
</table>

* In a system without external axis, Data-10 to Data-15 (for 7-axis robots, Data-11 to Data-16) should be set to "0".

Response format: 0000 or Error code
5.2.6.3 Editing System Commands

- **DELETE**
  Deletes a specified job.
  
  Command format: DELETE Job-Name
  
  Job-Name = Job name to be deleted
  = * : Delete all jobs

  Response format: 0000 or Error code

  <Example>
  
  Command: DELETE WORK-B
  Response: 0000

- **CVTRJ (Optional)**
  Converts a specified job to a relative job of a specified coordinate.
  
  Command format: CVTRJ Data-1, Data-2
  
  Data-1 = Name of job to be converted
  Data-2 = Conversion coordinate system specification
  0 : Base coordinate
  1 : Robot coordinate
  2 : User coordinate 1
  :  
  17 : User coordinate 16
  18 : Reserved

  If the specified user coordinate is not defined, an error occurs.

  Response format: 0000 or Error code

  <Example>
  
  Command: CVTRJ TESTJOB, 2
  Response: 0000

**NOTE**

This function requires the relative job function of the FS100.
CVTSJ (Optional)
Converst a specified job to a standard job (pulse job) in a specified convering method.

Command format: CVTSJ Data-1, Data-2, Data-3
- Data-1 = Name of job to be converted
- Data-2 = Converting method specification
  - 0: Previous step regarded (B-axis sign same)
  - 1: Form regarded
  - 2: Previous step regarded (R-axis travel amount minimum)
- Data-3 = Reference position variable.
  - Position variable No. indicating the first step conversion reference position when the previous step is regarded.

Response format: 0000 or Error code

<Example>
Command: CVTSJ SAMPLE01, 1, 0
Response: 0000

In the above example, P000 is to be the reference point and the job “SAMPLE01” is converted to a standard job with the form regarded.

NOTE
This function required the relative job function of the FS100.
Data Transmission
Function

5 Host Control Function of FS100
5.2 Robot Control Function

**WUFRAME**

 Writes a user coordinate data to a specified user coordinate system.

Command format: WUFRAME Data-1, Data-2, ···, Data-29

Data-1 = User coordinate No.

0 : Reserved
1 : Reserved
2 : User coordinate 1
   ...
17 : User coordinate 16

Data-2 = ORG X coordinate value (unit: mm, significant 3 decimal points)

Data-3 = ORG Y coordinate value (unit: mm, significant 3 decimal points)

Data-4 = ORG Z coordinate value (unit: mm, significant 3 decimal points)

Data-5 = ORG wrist angle TX (unit: degree (°), significant 4 decimal points)

Data-6 = ORG wrist angle TY (unit: degree (°), significant 4 decimal points)

Data-7 = ORG wrist angle TZ (unit: degree (°), significant 4 decimal points)

Data-8 = ORG type

Data-9 = XX X coordinate value (unit: mm, significant 3 decimal points)

Data-10 = XX Y coordinate value (unit: mm, significant 3 decimal points)

Data-11 = XX Z coordinate value (unit: mm, significant 3 decimal points)

Data-12 = XX wrist angle TX (unit: degree (°), significant 4 decimal points)

Data-13 = XX wrist angle TY (unit: degree (°), significant 4 decimal points)

Data-14 = XX wrist angle TZ (unit: degree (°), significant 4 decimal points)

Data-15 = XX type

Data-16 = XY X coordinate value (unit: mm, significant 3 decimal points)

Data-17 = XY Y coordinate value (unit: mm, significant 3 decimal points)

Data-18 = XY Z coordinate value (unit: mm, significant 3 decimal points)

Data-19 = XY wrist angle TX (unit: degree (°), significant 4 decimal points)

Data-20 = XY wrist angle TY (unit: degree (°), significant 4 decimal points)

Data-21 = XY wrist angle TZ (unit: degree (°), significant 4 decimal points)

Data-22 = XY type

Data-23 = Tool No. (0 to 15)

Data-24 = Number of 7th axis pulses (for travel axis, mm)

Data-25 = Number of 8th axis pulses (for travel axis, mm)

Data-26 = Number of 9th axis pulses (for travel axis, mm)

Data-27 = Number of 10th axis pulses
5 Host Control Function of FS100

5.2 Robot Control Function

Data Transmission Function

Data-28 = Number of 11th axis pulses
Data-29 = Number of 12th axis pulses

Response format: 0000 or Error code

- ORG, XX, and XY coordinate are written in the base coordinate system.
- In a system without external axis, Data-24 to Data-29 should be set to "0".
- If the group axis of the specified user coordinate system is not R1, an error occurs.
- For base axis data of ORG, XX, and XY, the same data should be used.
- For 7-axis robots, this command cannot be used.

<Example>

Command  WUFRAME 2, 600.0, 12.34, 500.0, 180.0, 0.0, 0.0, 0, ..., 0
Response   0000
### LOADV

Receives variable data from a host computer and write it in a specified variable.

Command format: LOADV Data-1, Data-2, ..., Data-13

Data-1 = Type of variables
0: Byte type variables
1: Integer type variables
2: Double precision type variables
3: Real number type variables
4: Robot axis position type variables
5: Base axis position type variables
6: Station axis position type variables (only pulse type)
7: String variables

Data-2 = Variable No.

Data-3 = Byte value / Integer value / Double precision type integer value / Real number value / Position data type / String

Position data type = 0: Pulse type
1: Cartesian type

*The order varies depending on the number of robot's axes.*

<table>
<thead>
<tr>
<th>6-axis robot</th>
<th>7-axis robot</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data-4</strong></td>
<td>Number of S-axis pulses / Number of base 1st axis pulses / Number of station 1st axis pulses</td>
</tr>
<tr>
<td><strong>Data-5</strong></td>
<td>Number of L-axis pulses / Number of base 2nd axis pulses / Number of station 2nd axis pulses</td>
</tr>
<tr>
<td><strong>Data-6</strong></td>
<td>Number of U-axis pulses / Number of base 3rd axis pulses / Number of station 3rd axis pulses</td>
</tr>
<tr>
<td><strong>Data-7</strong></td>
<td>Number of R-axis pulses / Number of base 4th axis pulses / Number of station 4th axis pulses</td>
</tr>
<tr>
<td><strong>Data-8</strong></td>
<td>Number of B-axis pulses / Number of base 5th axis pulses / Number of station 5th axis pulses</td>
</tr>
<tr>
<td><strong>Data-9</strong></td>
<td>Number of T axis pulses / Number of base 6th axis pulses / Number of station 6th axis pulses</td>
</tr>
<tr>
<td><strong>Data-10</strong></td>
<td>Tool No. (0 to 15)</td>
</tr>
<tr>
<td><strong>Data-11</strong></td>
<td>Tool No. (0 to 15)</td>
</tr>
<tr>
<td><strong>Data-12</strong></td>
<td>-</td>
</tr>
<tr>
<td><strong>Data-13</strong></td>
<td>-</td>
</tr>
</tbody>
</table>
5 Host Control Function of FS100
5.2 Robot Control Function

(When the position data type is 1)
Data-4 = Coordinate data
Coordinate data = 0 : Base coordinate
1 : Robot coordinate
2 : User coordinate 1
3 : User coordinate 2
: 
17 : User coordinate 16
18 : Tool coordinate
19 : Master tool coordinate
* The order of response data varies depending on the number of robot’s axes.
When the system contains multiple robots, the order is that of the robot with the maximum number of axes.

<table>
<thead>
<tr>
<th>Data</th>
<th>6-axis robot</th>
<th>7-axis robot</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>X coordinate value / Base 1st axis</td>
<td>X coordinate value / Base 1st axis</td>
</tr>
<tr>
<td></td>
<td>Cartesian value (unit : mm,</td>
<td>Cartesian value (unit : mm,</td>
</tr>
<tr>
<td></td>
<td>significant 3 decimal points)</td>
<td>significant 3 decimal points)</td>
</tr>
<tr>
<td>6</td>
<td>Y coordinate value / Base 2nd axis</td>
<td>Y coordinate value / Base 2nd axis</td>
</tr>
<tr>
<td></td>
<td>Cartesian value (unit : mm,</td>
<td>Cartesian value (unit : mm,</td>
</tr>
<tr>
<td></td>
<td>significant 3 decimal points)</td>
<td>significant 3 decimal points)</td>
</tr>
<tr>
<td>7</td>
<td>Z coordinate value / Base 3rd axis</td>
<td>Z coordinate value / Base 3rd axis</td>
</tr>
<tr>
<td></td>
<td>Cartesian value (unit : mm,</td>
<td>Cartesian value (unit : mm,</td>
</tr>
<tr>
<td></td>
<td>significant 3 decimal points)</td>
<td>significant 3 decimal points)</td>
</tr>
<tr>
<td>8</td>
<td>Wrist angle Rx coordinate value</td>
<td>Wrist angle Rx coordinate value</td>
</tr>
<tr>
<td></td>
<td>(unit : degree (°), significant 4</td>
<td>(unit : degree (°), significant 4</td>
</tr>
<tr>
<td></td>
<td>decimal points)</td>
<td>decimal points)</td>
</tr>
<tr>
<td>9</td>
<td>Wrist angle Ry coordinate value</td>
<td>Wrist angle Ry coordinate value</td>
</tr>
<tr>
<td></td>
<td>(unit : degree (°), significant 4</td>
<td>(unit : degree (°), significant 4</td>
</tr>
<tr>
<td></td>
<td>decimal points)</td>
<td>decimal points)</td>
</tr>
<tr>
<td>10</td>
<td>Wrist angle Rz coordinate value</td>
<td>Wrist angle Rz coordinate value</td>
</tr>
<tr>
<td></td>
<td>(unit : degree (°), significant 4</td>
<td>(unit : degree (°), significant 4</td>
</tr>
<tr>
<td></td>
<td>decimal points)</td>
<td>decimal points)</td>
</tr>
<tr>
<td>11</td>
<td>Form</td>
<td>Elbow angle Re coordinate value</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(unit : degree (°), significant 4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>decimal points)</td>
</tr>
<tr>
<td>12</td>
<td>Tool No. (0 to 15)</td>
<td>Form</td>
</tr>
<tr>
<td>13</td>
<td>-</td>
<td>Tool No. (0 to 15)</td>
</tr>
</tbody>
</table>

Data of the form is a value obtained by converting the following bit data to decimal notation.

![Bit Data Diagram]

Response format : 0000 or Error code
5 Host Control Function of FS100

5.2 Robot Control Function

<Example>

Command LOADV 0, 0, 123
Response 0000

In the above example, 123 is stored in the FS100 byte type variable B000.
5.2.6.4 Job Selection System Commands

- **SETMJ**
  Sets a specified job as a master job.
  At the same time, the specified job is set as a execution job.
  
  Command format: SETMJ Job-Name
  Job-Name = Job name to be set
  
  Response format: 0000 or Error code
  
  **Example**
  
  Command: SETMJ WORK-C
  Response: 0000

- **JSEQ**
  Sets a job name and a line No.
  
  Command format: JSEQ Data-1, Data-2
  Data-1 = Job name to be set
  Data-2 = Line No. to be set (0 to 9999)
  
  Response format: 0000 or Error code
  
  **Example**
  
  Command: JSEQ WORK-A, 10
  Response: 0000
5.2.7 I/O Read/Write Function

The host control function can read out or write in (change) I/O signal status using the host computer.

The following table shows the number of signals and the types of signals to be sent or received by the host control function.

<table>
<thead>
<tr>
<th>Signal</th>
<th>Signal Range (Qty)</th>
<th>Classification</th>
<th>Read-out</th>
<th>Write-in</th>
</tr>
</thead>
<tbody>
<tr>
<td>0xxxx</td>
<td>00010 to 01287 (1024)</td>
<td>General input signal</td>
<td>i</td>
<td></td>
</tr>
<tr>
<td>1xxxx</td>
<td>10010 to 11287 (1024)</td>
<td>General output signal</td>
<td>i</td>
<td></td>
</tr>
<tr>
<td>2xxxx</td>
<td>20010 to 21287 (1024)</td>
<td>External input signal</td>
<td>i</td>
<td></td>
</tr>
<tr>
<td>3xxxx</td>
<td>30010 to 31287 (1024)</td>
<td>External output signal</td>
<td>i</td>
<td></td>
</tr>
<tr>
<td>4xxxx</td>
<td>40010 to 41607 (1280)</td>
<td>Specific input signal</td>
<td>i</td>
<td></td>
</tr>
<tr>
<td>5xxxx</td>
<td>50010 to 52007 (1600)</td>
<td>Specific output signal</td>
<td>i</td>
<td></td>
</tr>
<tr>
<td>7xxxx</td>
<td>70010 to 79997 (7992)</td>
<td>Auxiliary relay</td>
<td>i</td>
<td></td>
</tr>
<tr>
<td>8xxxx</td>
<td>80010 to 80647 (512)</td>
<td>Control status signal</td>
<td>i</td>
<td></td>
</tr>
<tr>
<td>82xxx</td>
<td>82010 to 82207 (160)</td>
<td>Pseudo input signal</td>
<td>i</td>
<td></td>
</tr>
<tr>
<td>25xxx</td>
<td>25010 to 26287 (1024)</td>
<td>Network input</td>
<td>i</td>
<td></td>
</tr>
</tbody>
</table>

5.2.7.1 Transmission Procedure

The transmission from the host computer proceeds as follows.

1. The ENQ code is sent from the host computer to establish a data link.
2. After the data link is established, the data is sent from the host computer. The data transmission should be completed in a single block.
3. After the request to send is accepted, the host computer should be ready to receive. The FS100 sends the ENQ code to establish the data link.
4. After the data link is established, the data sent from the FS100 is received to terminate the transmission at completion of receipt.

The read/write function can be distinguished by the header number. Refer to the header number list.
5.2.7.2 Read-out of I/O Signal Status

DATA-1

Command format: Data-1, Data-2
Data-1 = Start No.
Data-2 = Number of data points

DATA-2

Response format (at normal completion): Data-1, Data-2, ..., Data-256
Data-1 = First 8 points of data
Data-2 = Second 8 points of data
:  
Data-256 = Last (up to 256th) 8 points of data
Response format (at abnormal completion):

```
SOH 90,000 STX  Error code CR ETX  BCC
```

Error code: Number with 4 digits other than 0000
Number smaller than 1000, 0 is added before the number.

<Example> When 3 points are read out from 70010

Command 70010, 3
Response 2, 0, 5
5.2.7.3 Write-in of I/O Signal Status

**DATA-3**

Command format: Data-1, Data-2, Data-3, Data-4, …, Data-258
- Data-1 = Start No.
- Data-2 = Number of data points
- Data-3 = First 8 points of data
- Data-4 = Second 8 points of data
  :
  :
- Data-258 = Last (up to 256th) 8 points of data

**DATA-4**

Response format (at normal completion): 0000
Response format (at abnormal completion): Number with 4 digits other than 0000
Number smaller than 1000, 0 is added before the number.

*Example* When status of 3 points is changed from 22010
- Command 22010, 3, 4, 3, 12
- Response 0000
5.3 Commands for Multi-control Group and Independent Control Functions

5.3.1 Commands for Multi-control Group

The FS100 can control more than one manipulator or station simultaneously.

The following commands are available for this multi-control function.

- **CGROUP**: Changing the control group
- **RGROUP**: Reading the control group and task selected status

The following table shows the combination which can be set by using the above commands.

<table>
<thead>
<tr>
<th>R1 (robot 1)</th>
<th>R2 (robot 2)</th>
<th>S1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>2</td>
</tr>
<tr>
<td> 2)</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1 Either one station among S1 to S3 can be selected in a system having several stations.
2 Base axes is included in robot axes.

The following commands have influence when the above commands are used.

The operations of these commands are applicable to the set control group.

<table>
<thead>
<tr>
<th>Read System Commands</th>
<th>Startup System Commands</th>
<th>Editing System Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>RPOSJ</td>
<td>MOVJ</td>
<td>WUFRAME</td>
</tr>
<tr>
<td>RPOSC</td>
<td>MOVL</td>
<td></td>
</tr>
<tr>
<td>RUFRAME</td>
<td>IMOV</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PMOVJ</td>
<td></td>
</tr>
<tr>
<td></td>
<td>PMOVL</td>
<td></td>
</tr>
</tbody>
</table>
5.3.2 Commands for Independent Control Function

The FS100 supports the independent control function which can execute more than one job simultaneously.

For this independent function, the following commands are available.

- **CTASK**: Changing the tasks
- **RGROUP**: Reading the control group or task selected status

By using the above commands, a task to be controlled can be changed.

The following commands have influence when the independent control function is used.

1. **Job startup (START)**
   
   Starts up a job.
   
   When a job name specification is provided for operand, execution of that job is started from the head of job as a task that is currently selected.
   
   When a job name is not specified, all tasks that are currently set are executed from the current line No.

2. **Waiting for completion of startup (JWAIT)**
   
   As a response, returns the information whether the currently selected task operation has been completed.

3. **Master job registration (SETMJ)**
   
   Sets a specified job as a master job, to the currently selected task.

4. **Job selection (JSEQ)**
   
   Sets a job name, a line No. to the currently selected task.

5. **Read of selected job (RJSEQ)**
   
   Reads the job name, line No., and step No. of the currently selected task.

6. **Read of status (RSTATS)**
   
   Returns the system status disregarding the selected task status.
   
   However, the “running” status differs from the conventional status; the “running” is entered even if only one task was operating.
## 5.4 Alarm Codes

<table>
<thead>
<tr>
<th>Code</th>
<th>Contents</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>4112</td>
<td>Data sending error</td>
<td>The EOT code is sent out and the data link is canceled.</td>
</tr>
<tr>
<td></td>
<td>1 : NAK retry over</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2 : Timer A timeup retry over</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3 : Alternating response error retry over</td>
<td></td>
</tr>
<tr>
<td>4113</td>
<td>Data receiving error</td>
<td>For 3 to 7, the EOT code is sent out and the data link is canceled.</td>
</tr>
<tr>
<td></td>
<td>1 : Receiving timeup (Timer A)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2 : Receiving timeup (Timer B)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3 : Short heading length</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4 : Long heading length</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5 : Illegal header No.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>6 : Text longer than 256 characters</td>
<td></td>
</tr>
<tr>
<td></td>
<td>7 : Receiving other than expected control code</td>
<td></td>
</tr>
<tr>
<td>4114</td>
<td>Transmission hardware error</td>
<td>The EOT code is not sent.</td>
</tr>
<tr>
<td></td>
<td>1 : Overrun error</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2 : Parity error</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3 : Framing error</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4 : Sending timeup (Timer A)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5 : Sending timeup (Timer B)</td>
<td></td>
</tr>
<tr>
<td>4115</td>
<td>Transmission system block</td>
<td>For 4, the EOT code is sent out and the data link is canceled.</td>
</tr>
<tr>
<td></td>
<td>This alarm notifies that the transmission procedure is correct but the received contents makes inconsistency in the system. Usually this alarm is resulted from violation of rules on the other party or illegal notification.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 : EOT was received while waiting for ACK</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2 : EOT was received while waiting for ENQ</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3 : EOT was received before receiving the last block</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4 : Code other than EOT was received after receiving the last block</td>
<td></td>
</tr>
<tr>
<td>4206</td>
<td>Transmission system error</td>
<td>The EOT code is not sent.</td>
</tr>
<tr>
<td></td>
<td>This alarm notifies an error on processing of transmission system. This alarm occurs in the following cases. 100 Error in transmission task</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• A job containing position type variable of which the value is not set, was to be saved.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• A job which does not exist on the memory, was to be saved.</td>
<td></td>
</tr>
</tbody>
</table>
5.5 Interpreter Message List

The interpreter messages are classified into the following categories.

- 1xxx : Command text general error
- 2xxx : Command execution mode error
- 3xxx : Command execution error
- 4xxx : Job registration error
- 5xxx : File contents error

**Table 5-1: Interpreter Message List (Sheet 1 of 2)**

<table>
<thead>
<tr>
<th>Code</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>1010</td>
<td>Command error</td>
</tr>
<tr>
<td>1011</td>
<td>Error in number of command operands</td>
</tr>
<tr>
<td>1012</td>
<td>Command operand value range over</td>
</tr>
<tr>
<td>1013</td>
<td>Command operand length error</td>
</tr>
<tr>
<td>1020</td>
<td>Disk full of files</td>
</tr>
<tr>
<td>2010</td>
<td>Manipulator operating</td>
</tr>
<tr>
<td>2020</td>
<td>Hold by programming pendant</td>
</tr>
<tr>
<td>2030</td>
<td>Hold by playback panel</td>
</tr>
<tr>
<td>2040</td>
<td>External hold</td>
</tr>
<tr>
<td>2050</td>
<td>Command hold</td>
</tr>
<tr>
<td>2060</td>
<td>Error/alarm occurring</td>
</tr>
<tr>
<td>2070</td>
<td>Servo OFF</td>
</tr>
<tr>
<td>2080</td>
<td>Incorrect mode</td>
</tr>
<tr>
<td>2090</td>
<td>File accessing by other function</td>
</tr>
<tr>
<td>2100</td>
<td>Command remote not set</td>
</tr>
<tr>
<td>2110</td>
<td>This data cannot be accessed.</td>
</tr>
<tr>
<td>2120</td>
<td>This data cannot be loaded</td>
</tr>
<tr>
<td>2130</td>
<td>Editing</td>
</tr>
<tr>
<td>3010</td>
<td>Turn ON the servo power</td>
</tr>
<tr>
<td>3040</td>
<td>Perform home positioning</td>
</tr>
<tr>
<td>3050</td>
<td>Confirm positions</td>
</tr>
<tr>
<td>3070</td>
<td>Current value not made</td>
</tr>
<tr>
<td>3220</td>
<td>Panel lock ; mode/cycle prohibit signal is ON.</td>
</tr>
<tr>
<td>3230</td>
<td>Panel lock ; start prohibit signal is ON.</td>
</tr>
<tr>
<td>3350</td>
<td>User coordinate not taught</td>
</tr>
<tr>
<td>3360</td>
<td>User file destroyed</td>
</tr>
<tr>
<td>3370</td>
<td>Incorrect control group</td>
</tr>
<tr>
<td>3380</td>
<td>Incorrect base axis data</td>
</tr>
<tr>
<td>3390</td>
<td>Relative job conversion prohibit (at CVTRJ)</td>
</tr>
<tr>
<td>3400</td>
<td>Master call prohibit (parameter)</td>
</tr>
<tr>
<td>3410</td>
<td>Master call prohibit (lamp On during operation)</td>
</tr>
<tr>
<td>3420</td>
<td>Master call prohibit (teach lock)</td>
</tr>
<tr>
<td>3430</td>
<td>Robot calibration data not defined</td>
</tr>
<tr>
<td>3450</td>
<td>Robot calibration data not defined</td>
</tr>
<tr>
<td>3460</td>
<td>Coordinate system cannot be set.</td>
</tr>
<tr>
<td>4010</td>
<td>Insufficient memory capacity (job registered memory)</td>
</tr>
<tr>
<td>Code</td>
<td>Content</td>
</tr>
<tr>
<td>-------</td>
<td>--------------------------------------------------------------</td>
</tr>
<tr>
<td>4012</td>
<td>Insufficient memory capacity (position data registered memory)</td>
</tr>
<tr>
<td>4020</td>
<td>Job editing prohibit</td>
</tr>
<tr>
<td>4030</td>
<td>Same job name exists</td>
</tr>
<tr>
<td>4040</td>
<td>No specified job</td>
</tr>
<tr>
<td>4060</td>
<td>Set a execution job.</td>
</tr>
<tr>
<td>4120</td>
<td>Position data destroyed</td>
</tr>
<tr>
<td>4130</td>
<td>Position data not exist</td>
</tr>
<tr>
<td>4140</td>
<td>Incorrect position variable type</td>
</tr>
<tr>
<td>4150</td>
<td>END instruction for job which is not master job</td>
</tr>
<tr>
<td>4170</td>
<td>Instruction data destroyed</td>
</tr>
<tr>
<td>4190</td>
<td>Invalid character in job name</td>
</tr>
<tr>
<td>4200</td>
<td>Invalid character in label name</td>
</tr>
<tr>
<td>4230</td>
<td>Invalid instruction in this system</td>
</tr>
<tr>
<td>4420</td>
<td>No step in job to be converted</td>
</tr>
<tr>
<td>4430</td>
<td>Already converted</td>
</tr>
<tr>
<td>4480</td>
<td>Teach user coordinate.</td>
</tr>
<tr>
<td>4490</td>
<td>Relative job/Independent control function not permitted</td>
</tr>
<tr>
<td>5110</td>
<td>Syntax error (syntax of instruction)</td>
</tr>
<tr>
<td>5120</td>
<td>Position data error</td>
</tr>
<tr>
<td>5130</td>
<td>No NOP or END instruction</td>
</tr>
<tr>
<td>5170</td>
<td>Format error (incorrect format)</td>
</tr>
<tr>
<td>5180</td>
<td>Incorrect number of data</td>
</tr>
<tr>
<td>5200</td>
<td>Data range over</td>
</tr>
<tr>
<td>5310</td>
<td>Syntax error (except instruction)</td>
</tr>
<tr>
<td>5340</td>
<td>Error in pseudo instruction specification</td>
</tr>
<tr>
<td>5370</td>
<td>Error in condition data record</td>
</tr>
<tr>
<td>5390</td>
<td>Error in job data record</td>
</tr>
<tr>
<td>5430</td>
<td>System not matched</td>
</tr>
<tr>
<td>5480</td>
<td>Incorrect welding function type</td>
</tr>
</tbody>
</table>
# Data List

## 6.1 Header Number List

<table>
<thead>
<tr>
<th>Contents</th>
<th>File Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>01, 000 Command from a external computer</td>
<td>xxxxxxxx. JBI</td>
</tr>
<tr>
<td>02, 001 Single job data</td>
<td>xxxxxxxx. JBR</td>
</tr>
<tr>
<td>02, 051 Request for single job data</td>
<td></td>
</tr>
<tr>
<td>02, 052 Request for related job data</td>
<td></td>
</tr>
<tr>
<td>02, 200 Tool data</td>
<td>TOOL. CND</td>
</tr>
<tr>
<td>02, 202 User coordinate data</td>
<td>UFRAME. CND</td>
</tr>
<tr>
<td>02, 232 Variable data</td>
<td>VAR. DAT</td>
</tr>
<tr>
<td>02, 244 Shock detection level</td>
<td>SHOCKLVL.CND</td>
</tr>
<tr>
<td>02, 240 System information</td>
<td>SYSTEM. SYS</td>
</tr>
<tr>
<td>02, 241 Alarm history data</td>
<td>ALMHIST. DAT</td>
</tr>
<tr>
<td>02, 300 Request for tool data</td>
<td>TOOL.CND</td>
</tr>
<tr>
<td>02, 302 Request for user coordinate data</td>
<td>UFRAME.CND</td>
</tr>
<tr>
<td>02, 332 Request for variable data</td>
<td>VAR.DAT</td>
</tr>
<tr>
<td>02, 344 Request for shock detection level</td>
<td>SHOCKLVL.CND</td>
</tr>
<tr>
<td>02, 340 Request for system information</td>
<td>SYSTEM.SYS</td>
</tr>
<tr>
<td>02, 341 Request for alarm history data</td>
<td>ALMHIST.DAT</td>
</tr>
<tr>
<td>03, 001 Byte type variable</td>
<td></td>
</tr>
<tr>
<td>03, 002 Integer type variable</td>
<td></td>
</tr>
<tr>
<td>03, 003 Double precision type variable</td>
<td></td>
</tr>
<tr>
<td>03, 004 Real number type variable</td>
<td></td>
</tr>
<tr>
<td>03, 005 Robot axis position type variable (pulse type)</td>
<td></td>
</tr>
<tr>
<td>03, 006 Robot axis position type variable (XYZ type)</td>
<td></td>
</tr>
<tr>
<td>03, 007 External axis position type variable (pulse type)</td>
<td></td>
</tr>
<tr>
<td>03, 008 External axis position type variable (XYZ type)</td>
<td></td>
</tr>
<tr>
<td>03, 051 Request for byte type variable</td>
<td></td>
</tr>
<tr>
<td>03, 052 Request for integer type variable</td>
<td></td>
</tr>
<tr>
<td>03, 053 Request for double precision type variable</td>
<td></td>
</tr>
<tr>
<td>03, 054 Request for real number type variable</td>
<td></td>
</tr>
<tr>
<td>03, 055 Request for robot axis position type variable (pulse type)</td>
<td></td>
</tr>
<tr>
<td>03, 056 Request for robot axis position type variable (XYZ type)</td>
<td></td>
</tr>
<tr>
<td>03, 057 Request for external axis position type variable (pulse type)</td>
<td></td>
</tr>
<tr>
<td>03, 058 Request for external axis position type variable (XYZ type)</td>
<td></td>
</tr>
<tr>
<td>04, 001 Request for write-in of I/O signals</td>
<td></td>
</tr>
<tr>
<td>04, 051 Request for read-out of I/O signals</td>
<td></td>
</tr>
<tr>
<td>90, 000 Command or data response (normal/error)</td>
<td></td>
</tr>
<tr>
<td>001 Command or data response (data)</td>
<td></td>
</tr>
</tbody>
</table>
### 6.2 Parameter List

#### Table 6-1: Parameter for Transmission

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Contents and Set Value</th>
<th>Initial Value</th>
</tr>
</thead>
</table>
| S2C230    | Programming pendant operation (in remote) specification  
0 : Invalid  
1 : Valid  
   | D0 D1 D2 D3 D4 D5 D6 D7  
 Programming pendant servo ON ([SERVO ON READY] key)  
 Programming pendant servo ON (Enable Switch)  
 Mode change  
 Master call  
 Cycle change  
 Start  
 Reserved  | 0  |
| RS000     | Standard port protocol specification  
0 : NON  
1 : System reserved  
2 : BSC LIKE  
3 : FC1  | 2  |
| RS029     | Loading permission specification of jobs and variables during playback  
0 : Invalid  
1 : Valid  | 1  |

#### Table 6-2: Parameter for Transmission (for BSC protocol)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Contents and Set Value</th>
<th>Initial Value</th>
</tr>
</thead>
</table>
| RS030     | Number of data bits  
7 : 7 (bit)  
8 : 8  | 8  |
| RS031     | Number of stop bits  
0 : 1 (bit)  
1 : 1.5  
2 : 2  | 0  |
| RS032     | Parity specification  
0 : No specification  
1 : Odd parity  
2 : Even parity  | 2  |
| RS033     | Transmission speed specification  
1 : 150 (baud rate)  
2 : 300  
3 : 600  
4 : 1200  
5 : 2400  
6 : 4800  
7 : 9600  
8 : 19200  | 7  |
| RS034     | Timer A : Sequence monitoring timer  
Serves as protection against invalid response or no response  
Unit : 0.1 sec. (Setting range : 0 to 100)  | 30  |
### Table 6-2: Parameter for Transmission (for BSC protocol)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Contents and Set Value</th>
<th>Initial Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>RS035</td>
<td>Timer B: Text reception monitoring timer &lt;br&gt;Serves as protection against no response of text end character &lt;br&gt;Unit: 0.1 sec. (Setting range: 0 to 255)</td>
<td>200</td>
</tr>
<tr>
<td>RS036</td>
<td>Retry 1: Number of resendings of a control character for invalid response or no response &lt;br&gt;Setting range: 0 to 30</td>
<td>10</td>
</tr>
<tr>
<td>RS037</td>
<td>Retry 2: Number of resendings of a text for a block check error (reception of NAK) &lt;br&gt;Setting range: 0 to 10</td>
<td>3</td>
</tr>
<tr>
<td>RS038</td>
<td>Block check method &lt;br&gt;0: Check sum</td>
<td>0</td>
</tr>
</tbody>
</table>
7 Remote Function Setting

Whether I/O remote control or command remote control should be enabled can be set in the pseudo input display when selecting the remote mode in the management mode.

1. Select [IN/OUT] under the main menu.
2. Select [PSEUDO INPUT SIG].
3. Select an item.
   - Select "INHIBIT IO" or "CMD REMOTE SEL".
     The item enabled is marked with "●" while the item disabled is marked with "○".

   - When INHIBIT IO is marked with ○ (disabled), the I/O remote function is enabled. When CMD REMOTE SEL is marked with ● (enabled), the command remote function is enabled.

   - When INHIBIT IO is marked with ○ (disabled), the I/O remote function is enabled so that the operation from external I/O is enabled with the programming pendant mode key set to [REMOTE].

   - When INHIBIT IO is marked with ● (enabled), the operation from external I/O is disabled.

   - When CMD REMOTE SEL is marked with ● (enabled), the host control function is enabled with the programming pendant mode key set to [REMOTE].

   - When CMD REMOTE SEL is marked with ○ (disabled), the host control function is disabled.

   - When INHIBIT P.P/PANEL is marked with ○ (disabled), the operation from P.P/PANEL is enabled even in remote mode. When INHIBIT P.P/PANEL is marked with ● (enabled), the operation from P.P/PANEL is disabled, except for the operations of emergency stop, hold, and remote key.
Specifications are subject to change without notice
for ongoing product modifications and improvements.