DYNASERV
DrvMII DM/DR Series
Instruction Manual

(DeviceNet Option)

Table of Contents

Introduction

Thank you very much for your purchase of the DD servo actuator DYNASERV. The DINASERV is an outer rotor type servo actuator that achieves high torque, high speed and high precision. It can be used in a wide range of applications in the field of factory automation-related devices, including industrial robots and index units.
This instruction manual explains its compatibility of the DrvMII type driver with open field networks.
Please refer to this instruction manual thoroughly when you use the product.

Precautions for Using this Instruction Manual

1. Please make sure that this manual is handed out to the end user.
2. Please read this manual thoroughly and understand the contents fully before proceeding to the operation of the product.
3. Please note that the safety protection may be lost and the proper safety may not be guaranteed if the product is not used according to the instructions described in this manual.
4. Always make sure that this manual is handy for the operator when using this product. If it is stained or lost, we will distribute copies upon request, subject to charge.
5. This manual explains details of the features included in the product and does not guarantee to meet the specific purpose of the customers.
6. No part of this manual may be reprinted or reproduced in any form without permission.
7. The information in this document is subject to change without notice.
8. The information contained in this document is believed to be accurate at the time of publication, but if you notice any inaccuracies, errors, or omissions, please contact our sales or service staff.

Regarding the safe usage of this device

- This product has been marked with ⚠nd ⚠ Warnings signs so that it can be used safely. Ignoring precautions and prohibitions related to these signs and using this product in an incorrect way may cause danger to the life and body of the operator. Always follow the precautions and observe the prohibitions explained below.
- Please make sure to understand the information given below completely before you start reading the instruction manual.
- Please keep the instruction manual and this sheet handy while using the product. In addition, make sure that they are handed out to the operator of the product.

⚠ Warnings

● Operation warning:

The motor periphery part of this device rotates at a high speed. People and objects should be kept outside the rotation radius when a load is attached.

● Warning about electric shock:

Make sure to connect the device to ground to avoid electric shock.
Make sure to turn the power off when connecting cables to the driver part.
Make sure to turn the power off when removing the cover of the driver part while performing adjustment operations, etc.

● Fire and electric shock warning:

If any abnormalities such as abnormal noise, bad smell, or release of fumes that coming from the device are detected while it is in operation, turn the power off immediately, pull out the power supply plug, and contact us.
If the device is dropped or given a strong impact, stop the operation immediately, turn the power off, and contact us.
Do not operate at power supply voltages other than the one indicated on the device.

● Fire and electric shock warning:

Avoid dropping or inserting metal shards or combustible materials, or allowing water to get into the opening parts of the device (e.g., the clearance between the rotor and stator of the motor part, or the air vent of the driver part). In such an eventuality, turn the power off immediately and contact us.
The cables coming out from the motor part or the bottom of the index part should not be forcibly bent, twisted, pulled, heated, or placed under a heavy object.
Never try to remodel or repair the device by yourself.

- This product has been marked with ⚠nd ⚠Precautions signs so that it can be used safely. Ignoring precautions and prohibitions related to these signs and using this product in an incorrect way may cause danger to the life and body of the operator. Always follow the precautions and observe the prohibitions explained below.
- Please make sure to understand the information given below completely before you start reading the instruction manual.
- Please keep the instruction manual and this sheet handy while using the product. In addition, make sure that they are handed out to the operator of the product.

⚠ Precautions

- Make sure to read the instruction manual before using the device.
  Operational mistakes and faulty wiring may result in damages and failure of the device.
- Make sure to check the wiring once more before turning the power on.
  Faulty wiring may result in fire, electric shock, or damage of the device.
- Confirm that the proper combination of motor and driver parts is used. Using the device with an incorrect configuration may result in failure. (Be sure to confirm the model--MODEL--on the rating nameplates.)
- Make sure the conditions of temperature, humidity, dust, etc. are as specified for the installation and storage environments.
- Do not block the air vent of the device. Keep the specified open space around the device as well. Poor ventilation may cause overheating, leading to failure.
- Some of the motor parts are very heavy; please pay sufficient attention to this when carrying and installing the parts. If the weight is more than 10kg (22.04 lbs), carrying or lifting tools should be used as much as possible.
- Both the motor and driver parts should be installed in the specified orientation.
- Keep the protection cover (transparent plastic plate) attached on the power supply terminal part of the driver. It is provided to prevent inadvertent electric shock accidents.

Handling Precautions

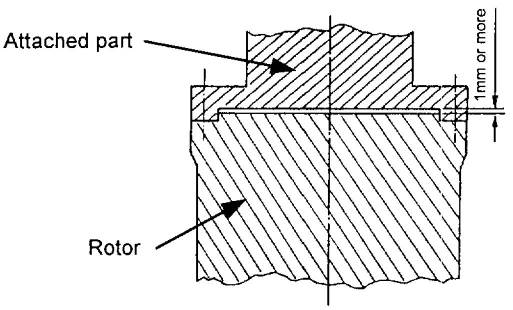
1. Do not install the motor in reverse direction in such a way that the rotor of the motor is fixed and the stator rotates.

2. Make sure to turn the power off before removing the side panel of the driver to set jumpers, etc. Touching the high voltage part inside the driver is dangerous.

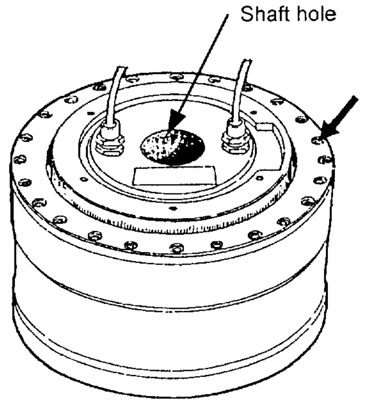
3. This motor rotates at a high speed and with a high torque. Take the rotation radius into consideration and pay special attention to the prevention of any dangerous situations that may occur during the operation when a load is attached to the motor.

4. Make sure to ground the ground terminal to earth.

5. When attaching a load to the rotor, make sure to keep a clearance of 1 mm or more between the load and the upper surface of the motor in order to maintain the surface precision. Furthermore, never push or squeeze an object into the shaft hole. (See the figure below.)



When feeding an object through the shaft hole, make sure to secure a clearance of at least 1mm on one side.



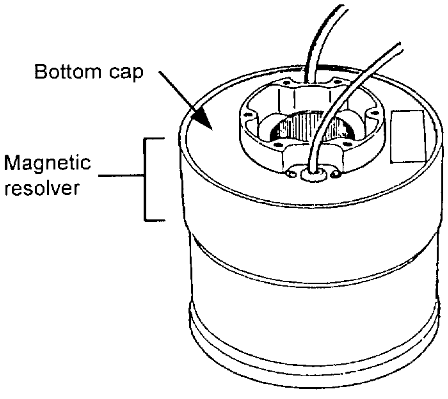
DM series motor

6. Do not touch the bolts (indicated by the arrow) that fix the bottom part of the rotor (see the figure to the right). If these bolts are loosened or tightened, the commutation angle will become inaccurate, which may result in uneven rotation (this applies only to the DM series).

7. The motor surface is magnetized; do not place things that can be affected by magnetism close to it.

8. The motor part shown in the figure to the right includes a magnetic resolver. Strong force, impacts, or magnetic fields should not be applied to the motor part (this applies only to the DR series).

9. Make sure to use load attachment screws that are shorter than the effective depth of the thread in the motor part. Depending on the model, if a screw exceeds the effective thread depth, the function may be impaired (this applies only to the DR series).



DR series motor

10. The motor is neither dust-, drip- nor water (oil)-proof; the motor should be installed in carefully chosen environments.

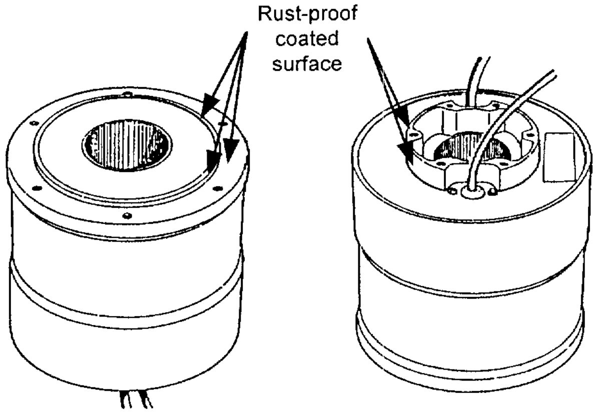
11. If the motor will be oscillating or rotating at small angles (50° or less), it should be allowed to oscillate at an angle of 90° or more for approximately 10 times (running-in operation) each time it has made 10,000 small-angle oscillations in order to prevent poor lubrication of the bearing.

12. In order for the motor and driver to be compatible with each other, they must be of the same model.

13. Never attempt to disassemble or remodel the motor and driver. If such service is necessary, please contact us. We assume no responsibility for products that have been disassembled or remodeled without permission.

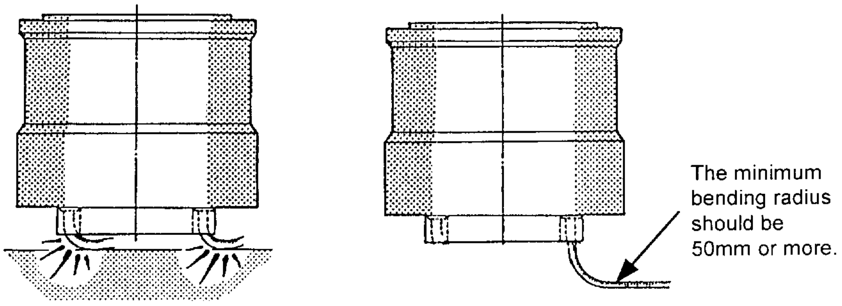
14. For the DYNASERV DR series motors, a coating has been applied on the load attachment surface of the upper surface of the motor and the stator on the lower surface in order to prevent rust.

    When starting to use the product, wipe off the coating completely with cloth or paper soaked in a petroleum or chlorine solvent before assembling. If any of the coating remains, it may affect the mechanical precision.

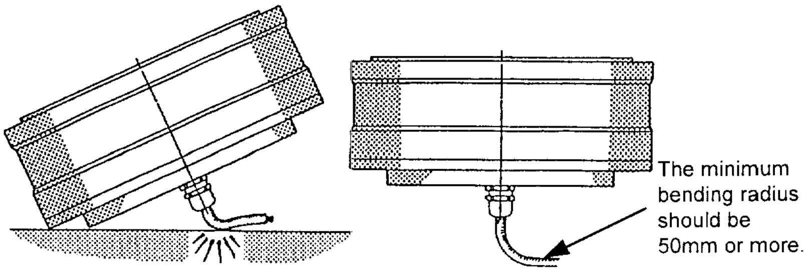


15. Do not place the motor on the floor and other surface in the manner shown in the figure below when carrying and installing the DYNASERV. The cables are crushed by the motor's own weight and the copper wires may be broken inside the cables.

    If it cannot be avoided to place the motor in such a manner, a support bench should always be placed so that the cables are lifted. Furthermore, if the cables need to be bent when installed in a device, etc., the minimum bending radius should be 50 mm or more. The cables are not strong enough to live up to robot cable specifications, so they should not be bent repeatedly.



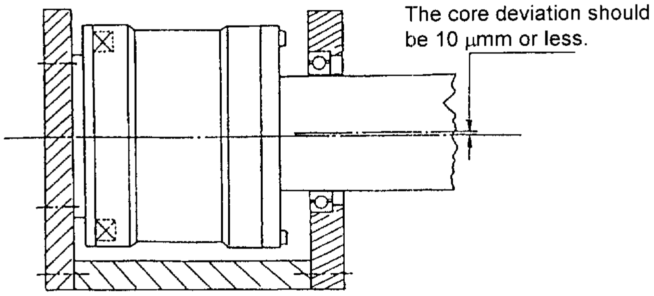
An example of a DR series motor



An example of a DM series motor

16. Do not perform a withstanding voltage test on this device. If such a test is performed without discretion, the circuits may be damaged. If such test must be conducted, make sure to contact us.

17. When connecting the motor with a load, the centerlines of both cores should be aligned to a sufficient degree. Please note that if the deviation between the two cores becomes 10 μm or more, the bearings inside the motor may be damaged.



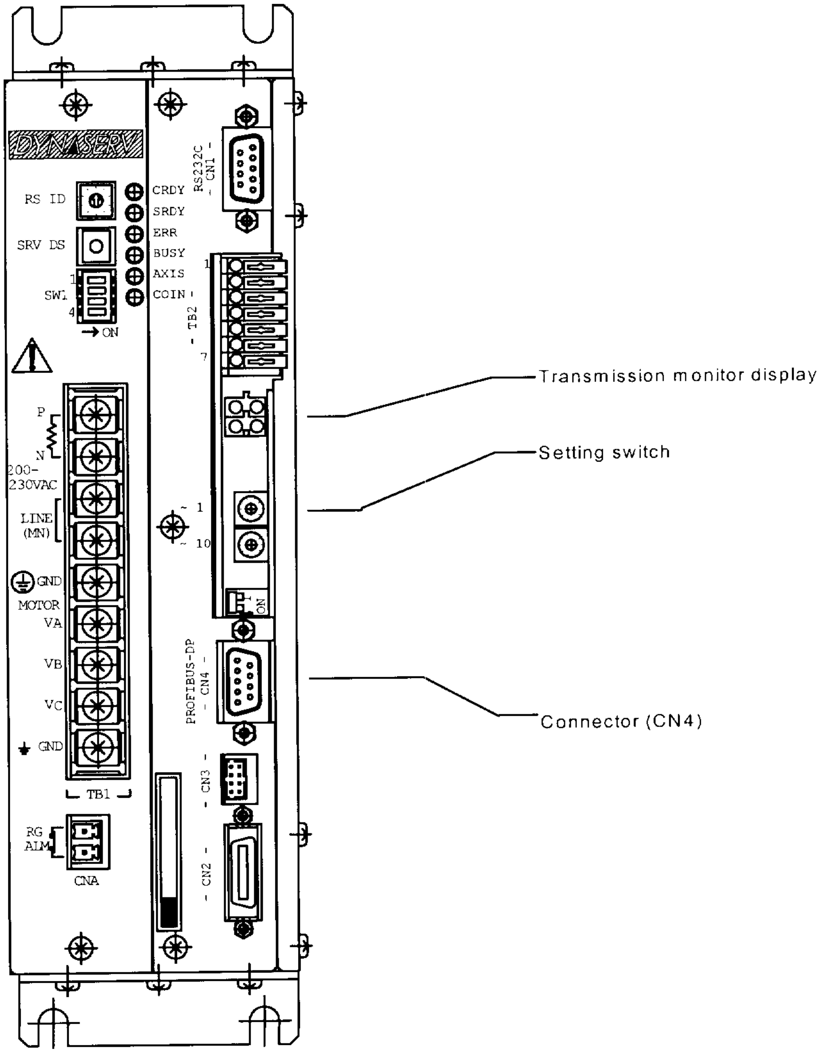
Chapter 1
DeviceNet Interface

1.1 Overview

The PLC interface for this driver can be chosen from a list of selectable options. It is possible to select a contact I/O interface, a CC-Link interface (under development), a PROFIBUS-DP interface, and a DeviceNet interface (under development).
The contact I/O interface is an I/O interface with 32 input points (sink type output support photo coupler inputs) and 20 output points (sink type open collector output).
CC-Link (Control & Communication Link) interface, PROFIBUS-DP interface, and DeviceNet interface are multi-vendor compatible field network interfaces. They allow for a reduction of wiring, high-speed data communication, and communication with various intelligent devices.
In this instruction manual, only the DeviceNet interface is explained.

1.1.1 Details of the Front Panel of the Driver

The figure below shows a detailed layout of the front panel of the driver, at which the DeviceNet interface is mounted. The interface part is identical for all models, although there may be slight differences depending on the driver capacity and availability of optional equipment.

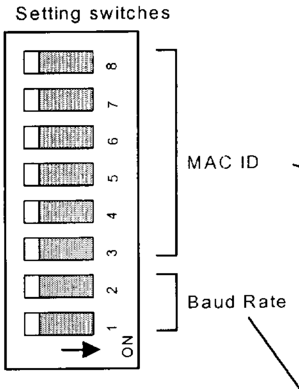


1.2 Connection and Hardware Setting

1.2.1 Setting Switches

These switches are used to select the communication speed and set the node address. Note that the communication speed varies depending on the communication distance. Refer to the DeviceNet user's manual, etc. of the host device for the details.

SW8	AD0
SW7	AD1
SW6	AD2
SW5	AD3
SW4	AD4
SW3	AD5
SW2	BD0
SW1	BD0



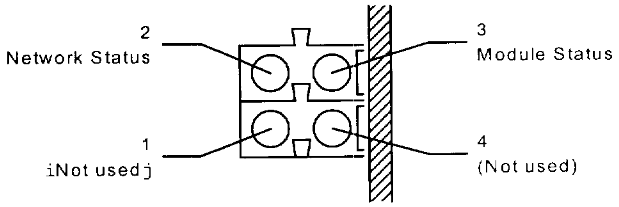
Address	AD5` AD0
0	000000
1	000001
2	000010
3	000011
F	F
62	111110
63	111111

0Fn ee
1Fn m

BaudRate bit^ sec	BD1` BD0
125k	00
250k	01
500k	10
reserved	11

1.2.2 Transmission Monitor Display

The LEDs of the transmission monitor display show the communication status of the DeviceNet. The table below lists the details for each LED.

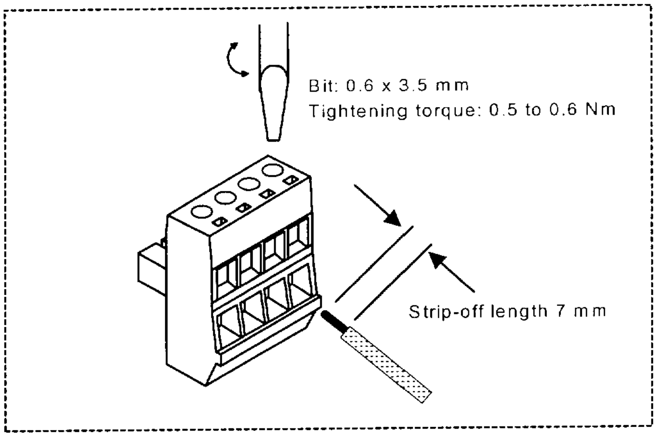


Name	Status		Description
Module Status	Off		Power off
	Red	On	Error (fatal)
	Green	On	Operating
	Red	Flashing	Error (minor)
Network Status	Off		Power off or not online
	Green	On	Online status, connected
	Red	On	Critical link error
	Green	Flashing	Online status, not connected
	Red	Flashing	Connection timeout

1.2.3 Connector

[CN4]

05	V (	V (
04	C_H	CAN_H
03	SLD	SHIELD
02	C_L	CAN_L
01	V |	V |



Wiring of communication cable

1.2.4 Communication Cable and Terminating Resistor

Refer to the user's manuals, etc. of the DeviceNet master station and the DeviceNet interface for the types of communication cable and terminating resistor that should be used.

1.2.5 Unit Connection and Link Method

It depends on your system configuration how each node must be connected. Refer to the user's manuals of the master station and the DeviceNet interface.

1.3 I/O Map

[Input signal]

Master station → Remote station (this driver)

	7	6	5	4	3	2	1	0
Block0	IN _MODE.3	IN _MODE.2	IN _MODE.1	IN _MODE.0	IN _MODE _STOP	IN _MODE _START	IN _SERVO	IN _EMG
Block1	IN _CODE.7	IN _CODE.6	IN _CODE.5	IN _CODE.4	IN _CODE.3	IN _CODE.2	IN _CODE.1	IN _CODE.0
Block2	IN _POS_INH	reserve	reserve	IN _M_ANS	IN _ERR _RESET	IN _ABORT	IN _INTER LOCK	IN _PRG _REWIND
Block3	reserve	IN _ABS _STR_OPT	IN_ ROTDIR_ STR_OPT1	IN_ ROTDIR_ STR_OPT0	IN _SIGN _INDEX	IN _OVER RIDE_SEL	IN _JOG_DN	IN _JOG_UP
Block4	reserve	IN_MON _B_CHNG _REQ	reserve	IN_MON _A_CHNG _REQ	reserve	IN _PRM _RD_REQ	reserve	IN _PRM _WR_REQ
Block5	reserve	reserve	reserve	reserve	reserve	reserve	reserve	reserve
Block6	reserve	reserve	reserve	reserve	reserve	reserve	reserve	reserve
Block7	reserve	reserve	reserve	reserve	reserve	reserve	reserve	reserve
Block8	WR_PRM_NO(High)		Parameter write			Parameter number (HighByte)		
Block9	WR_PRM_NO(Low)		Parameter write			Parameter number (LowByte)		
Block10	RD_PRM_NO(High)		Parameter/monitor read			Parameter/monitor number (HighByte)		
Block11	RD_PRM_NO(Low)		Parameter/monitor read			Parameter/monitor number (LowByte)		
Block12	MON_A_PRM_NO(High)		Parameter/monitor display A			Parameter/monitor number (HighByte)		
Block13	MON_A_PRM_NO(Low)		Parameter/monitor display A			Parameter/monitor number (LowByte)		
Block14	MON_B_PRM_NO(High)		Parameter/monitor display B			Parameter/monitor number (HighByte)		
Block15	MON_B_PRM_NO(Low)		Parameter/monitor display B			Parameter/monitor number (LowByte)		
Block16	WR_PRM_DATA(High-High)		Parameter write			Data to be written (HighWord-HighByte)		
Block17	WR_PRM_DATA(High-Low)		Parameter write			Data to be written (HighWord-LowByte)		
Block18	WR_PRM_DATA(Low-High)		Parameter write			Data to be written (LowWord-HighByte)		
Block19	WR_PRM_DATA(Low-Low)		Parameter write			Data to be written (LowWord-LowByte)		
Block20	reserve							
Block21	reserve							
Block22	reserve							
Block23	reserve							

[Output signal]

Remote station (this driver) → Master station

	7	6	5	4	3	2	1	0
Block0	reserve	reserve	OUT _M_EN	OUT _WARN	OUT _ERR	OUT _MODE _EXE	OUT _SRDY	OUT _CRDY
Block1	OUT _CODE.7	OUT _CODE.6	OUT _CODE.5	OUT _CODE.4	OUT _CODE.3	OUT _CODE.2	OUT _CODE.1	OUT _CODE.0
Block2	reserve	reserve	reserve	reserve	OUT _AREASIG .1	OUT _AREASIG .0	reserve	OUT _COIN
Block3	OUT_MON _B_CHNG _OK	OUT_MON _B_CHNG _END	OUT_MON _A_CHNG _OK	OUT_MON _A_CHNG _END	OUT _PRM_RD _OK	OUT _PRM_RD _END	OUT _PRM_WR _OK	OUT _PRM_WR _END
Block4	reserve	reserve	reserve	reserve	reserve	reserve	reserve	reserve
Block5	reserve	reserve	reserve	reserve	reserve	reserve	reserve	reserve
Block6	reserve	reserve	reserve	reserve	reserve	reserve	reserve	reserve
Block7	reserve	reserve	reserve	reserve	reserve	reserve	reserve	reserve
Block8	RD_PRM_DATA(High-High)		Parameter/monitor read			Data to be read (HighWord-HighByte)		
Block9	RD_PRM_DATA(High-Low)		Parameter/monitor read			Data to be read (HighWord-LowByte)		
Block10	RD_PRM_DATA(Low-High)		Parameter/monitor read			Data to be read (LowWord-HighByte)		
Block11	RD_PRM_DATA(Low-Low)		Parameter/monitor read			Data to be read (LowWord-LowByte)		
Block12	MON_A_PRM_DATA(High-High)		Parameter/monitor display A			Data to be read (HighWord-HighByte)		
Block13	MON_A_PRM_DATA(High-Low)		Parameter/monitor display A			Data to be read (HighWord-LowByte)		
Block14	MON_A_PRM_DATA(Low-High)		Parameter/monitor display A			Data to be read (LowWord-HighByte)		
Block15	MON_A_PRM_DATA(Low-Low)		Parameter/monitor display A			Data to be read (LowWord-LowByte)		
Block16	MON_B_PRM_DATA(High-High)		Parameter/monitor display B			Data to be read (HighWord-HighByte)		
Block17	MON_B_PRM_DATA(High-Low)		Parameter/monitor display B			Data to be read (HighWord-LowByte)		
Block18	MON_B_PRM_DATA(Low-High)		Parameter/monitor display B			Data to be read (LowWord-HighByte)		
Block19	MON_B_PRM_DATA(Low-Low)		Parameter/monitor display B			Data to be read (LowWord-LowByte)		
Block20	ERR_CODE(Main)		Error code			Main code		
Block21	ERR_CODE(Sub)		Error code			Subcode		
Block22	reserve							
Block23	reserve							

1.4 Operation 1

Some of the operations commanded by input signals via the PLC interface may not function depending on the *operation mode* setting. Refer to Section *.*, "Operation Mode" for an explanation of the operation mode. In this section, the operations of the DeviceNet interface will be explained.

1.4.1 Explanation of Signals

[Input signals]

Abbreviated name	Signal name	Contact I/O
IN_EMG	Emergency stop input	DI_0
IN_SERVO	Servo command input	DI_1
IN_MODE_START	Operating action start command input (start)	DI_2
IN_MODE_STOP	Operating action end command input (stop)	DI_3
IN_MODE [3..0]	Operation mode number input (binary)	DI_7 to 4
IN_I_CODE [7..0]	Code input (BCD)	DI_15 to 8
IN_PRG_REWIND	Program auto-rewind input	DI_16
IN_INTERLOCK	Interlock command input	DI_17
IN_ABORT	Operating action abort command input (abort)	DI_18
IN_ERR_RESET	Error reset command input	DI_19
IN_M_ANS	M answer input	DI_20
IN_ERRCODE_REQ	Error code request input	---
IN_POS_INH	Integral position control operation disable input	DI_23
IN_JOG_UP	Jog + command input	DI_24
IN_JOG_DN	Jog – command input	DI_25
IN_OVERRIDE_SEL	Velocity override selection input	DI_26
IN_SIGN_INDEX	Index sign input during index operation	DI_27
IN_ROTDIR_STR_OPT [1..0]	Moving direction at rotational coordinate start-up option input (binary)	DI_29 to 28
IN_ABS_STR_OPT	ABS/INC start-up option input	DI_30
IN_PRM_WR_REQ	Parameter write request input	DI_32
IN_PRM_RD_REQ	Parameter/monitor read request input	DI_34
IN_MON_A_CHNG_REQ	Parameter/monitor display A change request input	DI_36
IN_MON_B_CHNG_REQ	Parameter/monitor display B change request input	DI_38
IN_WR_PRM_NO[15..8]	Parameter write number input (binary)	DI_71 to 64
IN_WR_PRM_NO[7..0]		DI_79 to 72
IN_RD_PRM_NO[15..8]	Parameter/monitor read number input (binary)	DI_87 to 80
IN_RD_PRM_NO[7..0]		DI_95 to 88
IN_MON_A_PRM_NO[15..8]	Parameter/monitor display A number input (binary)	DI_103 to 96
IN_MON_A_PRM_NO[7..0]		DI_111 to 104
IN_MON_B_PRM_NO[15..8]	Parameter/monitor display B number input (binary)	DI_119 to 112
IN_MON_B_PRM_NO[7..0]		DI_127 to 120
IN_WR_PRM_DATA[31..24]	Parameter write data input (binary)	DI_135 to 128
IN_WR_PRM_DATA[23..16]		DI_143 to 136
IN_WR_PRM_DATA[15..8]		DI_151 to 144
IN_WR_PRM_DATA[7..0]		DI_159 to 152

[Output signals]

Abbreviated name	Signal name	Contact I/O
OUT_CPURDY	CPU ready output	DO_0
OUT_SRDY	Servo ready output	DO_1
OUT_MODE_EXE	Operation under execution output	DO_2
OUT_ERR	Error status output	DO_3
OUT_ALARM	Alarm status output	DO_4
OUT_M_EN	M code enable output	DO_5
OUT_ERRCODE_OUT	Output during error/alarm code output	---
OUT_O_CODE [7..0]	Code output (BCD)	DO_15 to 8
OUT_COIN	Position settling status output	DO_16
OUT_AREA0	Area signal 0 output	DO_18
OUT_AREA1	Area signal 0 output	DO_19
OUT_PRM_WR_END	Parameter write end output	DO_20
OUT_PRM_WR_OK	Parameter write normal output	DO_21
OUT_PRM_RD_END	Parameter/monitor read end output	DO_22
OUT_PRM_RD_OK	Parameter/monitor read normal output	DO_23
OUT_MON_A_CHNG_END	Parameter monitor display A change end output	DO_24
OUT_MON_A_CHNG_OK	Parameter monitor display A change normal output	DO_25
OUT_MON_B_CHNG_END	Parameter monitor display B change end output	DO_26
OUT_MON_B_CHNG_OK	Parameter monitor display B change normal output	DO_27
OUT_RD_PRM_DATA[31..24]	Parameter/monitor read data output (binary)	DO_71 to 64
OUT_RD_PRM_DATA[23..16]		DO_79 to 72
OUT_RD_PRM_DATA[15..8]		DO_87 to 80
OUT_RD_PRM_DATA[7..0]		DO_95 to 88
OUT_MON_A_DATA[31..24]	Parameter/monitor display A data output (binary)	DO_103 to 96
OUT_MON_A_DATA[23..16]		DO_111 to 104
OUT_MON_A_DATA[15..8]		DO_119 to 112
OUT_MON_A__DATA[7..0]		DO_127 to 120
OUT_MON_B_DATA[31..24]	Parameter/monitor display B data output (binary)	DO_135 to 128
OUT_MON_B_DATA[23..16]		DO_143 to 136
OUT_MON_B_DATA[15..8]		DO_151 to 144
OUT_MON_B__DATA[7..0]		DO_159 to 152
OUT_ERR_CODE_MAIN[7..0]	Error code (main code) output	DO_167 to 160
OUT_ERR_CODE_SUB[7..0]	Error code (subcode) output	DO_175 to 168

1.4.2 I/O Logic Setting

In the PROFIBUS-DP interface, it is possible to set the logical relationship of the bit status on the interface and the driver signal status for all available input and output signals. These settings can be made in contact and bit units, respectively.
The input signals IN*** are processed via the I/O logical conversion and are then expressed as a logical input signal ***. If the signal status is reached, it is expressed as 1 and if the status is not reached, expressed as 0.
A logical output signal *** is expressed as 1 if the status is reached and 0 if the status is not reached. After conversion via the I/O logical setting, it becomes an output signal OUT_***.
The logical setting at shipping from our factories is set to *positive logic* for all input and output signals. This means that the internal input and output signals are in the status 1 when the corresponding bit is 1. It is possible, however, to use the RC232C interface to confirm the basic operation under the conditions at shipping without connecting the PLC interface by setting IN_SERVO to negative logic via the PC utility.
Refer to Chapter * "PC Utility" for a description of how to set the I/O logic.

1.4.3 Servo ON/OFF (SERVO)

The function of SERVO, the servo command input, changes depending on whether the main operation authority by the operation mode is given to the RS232C interface or the PLC interface.
If the operation mode is set to the RS232C interface, the servo command input functions as the Servo ON enable/disable function. Servo ON is enabled in the status 1. Servo ON is enabled at contact OFF and bit 0 by negative logic setting for the input signal IN_SERVO (the setting at shipment). By positive logic setting, Servo ON is enabled at contact ON and bit 1. Refer to Section *.*, "@ Commands" for an explanation of how to use the Servo ON/OFF command in the RS232C interface operation mode.
If the operation mode is set to the PLC interface, the servo command input functions as the Servo ON/OFF command function. It becomes a Servo ON command when in the status 1. It becomes a Servo ON command at contact OFF and bit 0 by negative logic setting for the input signal IN_SERVO (the setting at shipment). By positive logic setting, it becomes a Servo ON command at contact ON and bit 1. Other than by this command, the actual Servo ON/OFF status is also affected by the setting of the Servo ON disable switch SRV DS on the front panel. See the table below.

Servo ON/OFF SERVO	SRV DS Servo ON disable on the front panel	Actual servo status
Status 0	Disabled	Servo OFF
	Enabled	
Status 1	Disabled	
	Enabled	Servo ON

1.4.4 Start and Stop (MODE_START MODE_STOP MODE [3..0])

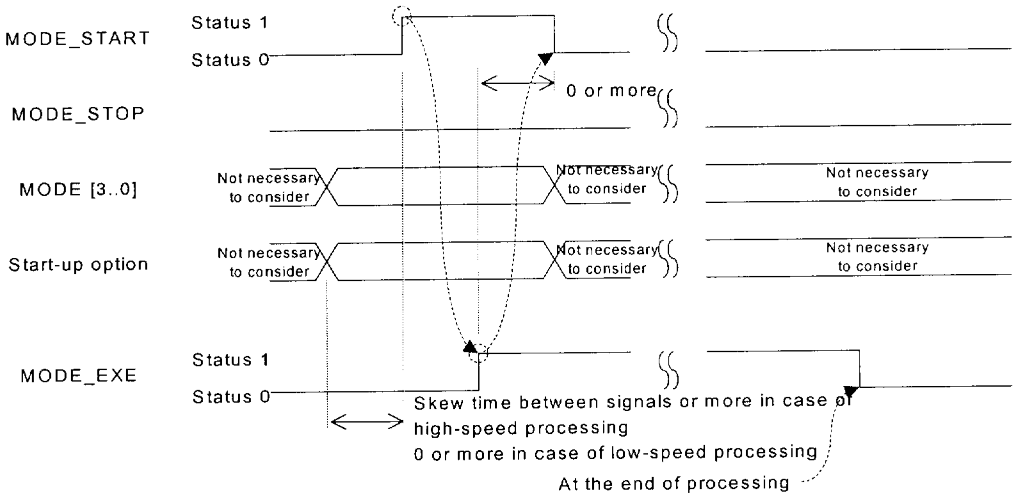
The MODE_START and MODE_STOP commands start and stop operating actions other than job moves. They can be issued via the PLC interface when the setting of the operation mode has given the main operation authority to the PLC interface rather than the RS232C interface.
The operating action start and stop commands are issued in the status 1.
When a start command is issued by MODE_START it depends on the operating action which signals (start-up options) that must be preset. When issuing a MODE [3..0] command, the number of the operating action you wish to perform must be always set. See the table below.
Moreover, if high-speed processing is not selected in the setting of the *#215 PLC operation: Start signal processing speed selection* parameter, the processing of start commands by MODE [3..0] and start-up options takes place at the next 10 ms scan. In this case, the content of each signal is read and processed after 10 ms has passed after the start command was issued. If the time delay between outputs by the PLC is in the order of several ms, the commands will function properly provided that they are set to the same time or before the start command is issued by MODE_START. However, the time wasted with respect to issuing the start command should be added (10 ms) instead. The total amount of wasted time until the motor start is thus 10 ms (scan time) + 10 ms (read delay) + internal delay time.
When high-speed processing is selected in the #215 parameter setting, the processing of the start command takes place at the next 2 ms scan. In this case, the content of each signal when the start command is issued is read and processed. The time lag between outputs must be taken into consideration on the PLC side and set that much earlier. On the other hand, the time wasted with respect to the start command itself can be eliminated. The total amount of wasted time until the motor start is the 2 ms (scan time) + internal delay time.
In MODE_STOP, the driver response to a stop command varies depending on the operating action. See the table below. Note furthermore that operations of the *self-end type*, which end automatically when the operation is completed, do not require a stop command. Operations of the *non-self-end type*, which cannot complete the operation by itself, must be ended by a stop command by MODE_STOP. In case of operations of the *non-end type*, which cannot be stopped once started, the stop command issued by MODE_STOP is invalid. Refer to Section *.*, "Operation Functions" for a listing of the end types of the different operations.
Outputs during MODE_EXE operation execution maintain their under-execution status during MODE_START command status (during status 1) after the operation is started, even after the operation is over.

No.	Name	Start-up option	End type	Response to stop command
1	Test operation	None	Non-self-end	End the operation upon returning to the start position.
2	Auto-tuning operation		Self-end	End the operation when the oscillation command to the motor is completed.
3	Homing move			Immediately decelerate and stop the move, and end the operation.
4	Program operations	I_CODE [7..0]		End the operation when execution of the current block is completed.
5	Signal search move	None		Immediately decelerate and stop the move, and end the operation.
7	MDI operation		Non-self-end	End the operation when execution of the current NC executable statement or parameter statement input via RS232C is completed.
8	Index Type A operation	I_CODE [7..0], SIGN_INDEX, ROTDIR_STR_OPT [1..0] (when necessary), ABS_STR_OPT (when necessary)	Self-end	Invalid (ignored)
9	Index Type B operation			
10	Table reference operation	I_CODE [7..0], ROTDIR_STR_OPT[1..0] (when necessary), ABS_STR_OPT (when necessary)		
15	Mechanical setting mode	None	Non-end	Invalid because the operation cannot be ended (ignored).

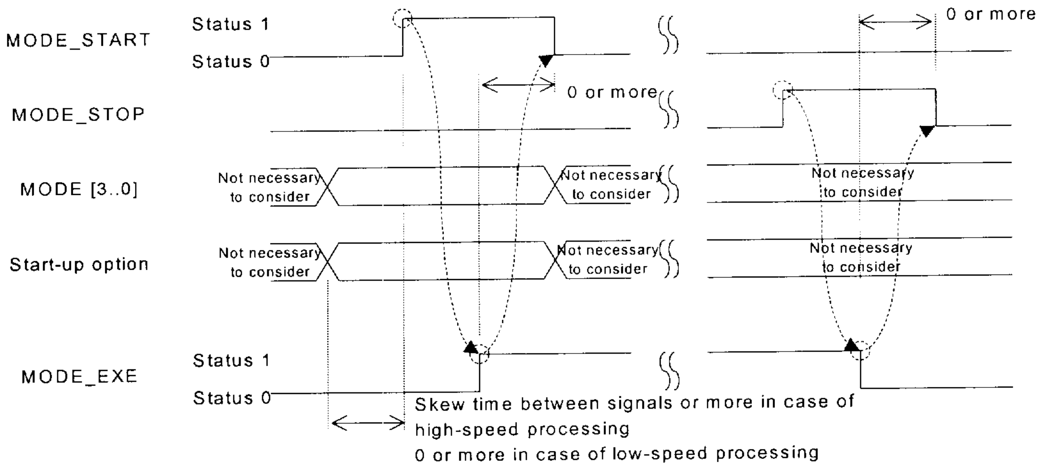
[Related parameters]

#215 PLC operation: Start signal processing speed selection

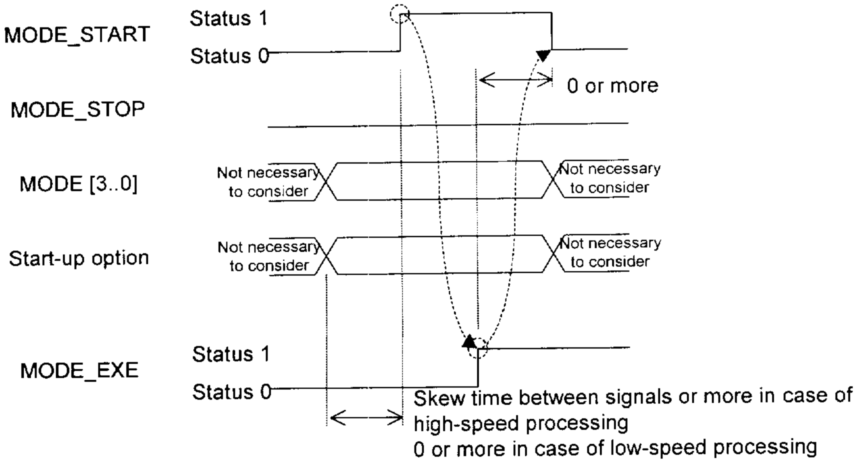
[Self-end type]


MODE_START
Status 1
Status 0
0 or more
MODE_STOP
MODE [3..0]
Not necessary to consider
Not necessary to consider
Not necessary to consider
Start-up option
Not necessary to consider
Not necessary to consider
Not necessary to consider
MODE_EXE
Status 1
Status 0
Skew time between signals or more in case of high-speed processing
0 or more in case of low-speed processing
At the end of processing

[Non-self-end type]


MODE_START
Status 1
Status 0
0 or more
0 or more
MODE_STOP
MODE [3..0]
Not necessary to consider
Not necessary to consider
Not necessary to consider
Start-up option
Not necessary to consider
Not necessary to consider
Not necessary to consider
MODE_EXE
Status 1
Status 0
Skew time between signals or more in case of high-speed processing
0 or more in case of low-speed processing

[Non-end type]


MODE_START
Status 1
Status 0
0 or more
MODE_STOP
MODE [3..0]
Not necessary to consider
Not necessary to consider
Start-up option
Not necessary to consider
Not necessary to consider
MODE_EXE
Status 1
Status 0
Skew time between signals or more in case of high-speed processing
0 or more in case of low-speed processing

[Start-up option at program operation]

The program number to be started at program operation is provided by the I-CODE [7..0] code entry.
In a setting where it is properly read as a BCD code, the program with the set number is executed from the top.
In a setting where it is not properly read as a BCD code, the program with the stored execution program number is executed from the block with the stored block number.

[Start-up options at index Type A operation and index Type B operation]

In index Type A and B operations, the index number is provided by the I_CODE [7..0] code entry and the SIGN_INDEX sign entry at index operation, the choice between absolute or incremental move is provided by the ABS_STR_OPT ABS/INC start-up option entry, and the moving direction option is provided by ROTDIR_STR_OPT [1..0].
I_CODE [7..0] must be set so that it can be read properly as a BCD code. When SIGN_INDEX is the in status 1, the value read by I_CODE [7..0] is treated as a negative value, and when in the status 0, treated as a positive value. In case of an incremental move the value is treated as a value relative to the current operation command value and in case of an absolute move, it is treated as the target operation command value.
ABS_STR_OPT indicates an absolute move in the status 1 and an incremental move in the status 0. When the *#104 ABS/INC setting during table index operation* parameter is set to *start-up option dependence*, the option specification given by ABS_STR_OPT is invalid.
ROTDIR_STR_OPT [1..0] indicates Type 0 when 0 or 3 is provided by binary code, Type 1 when 1 is provided, and Type 2 when 2 is provided. When the *#105 moving direction option for rotational coordinates* parameter is set to *start-up option dependence*, the option specification given by ROTDIR_STR_OPT [1..0] is invalid.

[Related parameters]

#104	ABS/INC setting during table index operation
#105	Moving direction option for rotational coordinates

[Start-up option at table reference operation]

In table parameter operations, the table entry is provided by the I_CODE [7..0] code entry, the choice between an absolute and incremental move is provided by the ABS_STR_OPT ABS/INC start-up option entry, and the moving direction option is provided by ROTDIR_STR_OPT [1..0].
I_CODE [7..0] must be set so that it can be read properly as a BCD code. The table data value and option are obtained based on the specified number. The value of the obtained table data is treated as a value relative to the current operation command value in case of an incremental move and in case of an absolute move it is treated as the target operation command value.
ABS_STR_OPT indicates an absolute move in the status 1 and an incremental move in the status 0. When the *#104 ABS/INC setting during table index operation* parameter is set to *start-up option dependence*, the option specification given by ABS_STR_OPT is invalid.
ROTDIR_STR_OPT [1..0] indicates Type 0 when 0 or 3 is provided by binary code, Type 1 when 1 is provided, and Type 2 when 2 is provided. When the *#105 Moving direction option for rotational coordinates* parameter is set to *start-up option dependence*, the option specification given by ROTDIR_STR_OPT [1..0] is invalid.

[Related parameters]

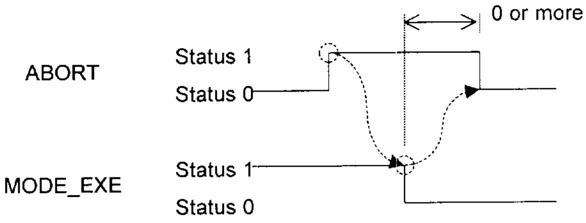
#104	ABS/INC setting during table index operation
#105	Moving direction option for rotational coordinates

Note: Regarding the moving direction in rotational coordinates

Type0	Proximity rotational move (multiple rotations are not allowed).
Type1	Does not cross the rotational coordinate's home position (multiple rotations are not allowed).
Type2	Calculates the target position relative to the current rotational coordinate's home position and determines the direction (multiple rotations are allowed).

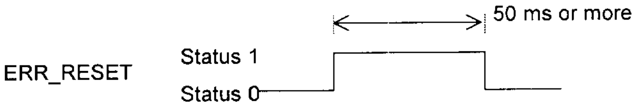
1.4.5 Abort (ABORT)

An operation abort command, ABORT, stops any operation other than a jog move. It functions irrespectively of the operation mode.
The operation abort command is issued as the status 1.
Unlike MODE_STOP, the operating action stop command, the motor immediately decelerates and stops and ends the operation even during an operation involving movement. In case the M function is being executed, the execution is aborted and the operation is stopped.



1.4.6 Error Reset (ERR_RESET)

The error reset command, ERR_RESET, cancels an error status in the driver. It functions irrespectively of the operation mode. It can only be executed while in the idle status.
The error reset command is issued as the status 1.
Depending on the error content, there are errors that cannot be canceled or errors that cause identical errors immediately after canceling. Avoid creating a program that maintains ERR_RESET and waits until ERR, the error status output, is canceled.



1.4.7 Interlock and Velocity Override Selection (INTERLOCK OVERRIDE_SEL)

Both the interlock command INTERLOCK and the velocity override selection OVERRIDE_SEL select a velocity override value of the driver. They function irrespectively of the operation mode.
The relationship between the INTERLOCK and OVERRIDE_SEL status and the velocity override value can be seen in the table below.

Interlock INTERLOCK	Velocity override selection OVERRIDE_SEL	Selected velocity override value
Status 0	Status 0	#16 Velocity override percentage 1
	Status 1	#17 Velocity override percentage 2
Status 1	Status 0	0
	Status 1	

[Related parameters]

#16 Velocity override percentage 1
#17 Velocity override percentage 2

1.4.8 Program Auto-Rewind (PRG_REWIND)

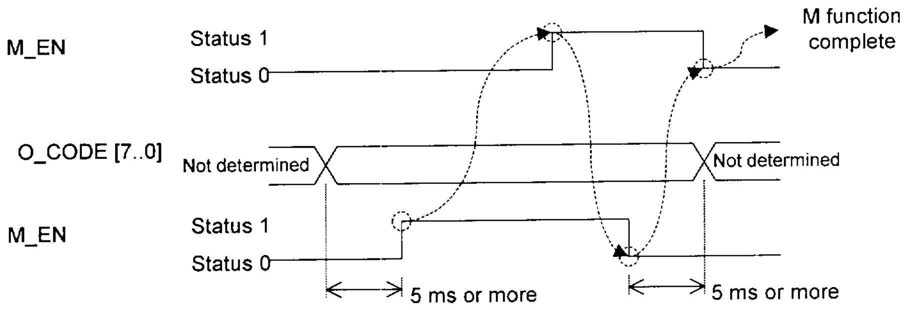
The program auto-rewind PRG_REWIND is a signal that specifies whether or not to repeat the program from the top block when the execution of the last block of the program is completed in program operation. If the program is repeated, it also monitors the signal status each time execution of the last block is completed and judges whether or not to repeat again. It functions irrespectively of the operation mode.
The program is repeated if the status of the signal is 1.

1.4.9 Integral Position Control Operation Inhibition (POS_INH)

The integral position control operation inhibition POS_INH is a signal that prohibits the integral operation at the position control part. It can be issued via the PLC interface when the setting of the operation mode has given the main operation authority to the PLC interface.
Integral operation is inhibited in the status 1 and allowed in the status 0.

1.4.10 Function (M_ANS, M_EN, O_CODE [7..0])

The *M function* operates under the PLC interface when the *#102 Enabling the selection of RS232C for the M function interface* parameter is set in such a way that the M function communicates with the host device via the PLC interface. See Section *.*, "M Function" for details about the M function.
The notification from the driver takes place through the use of M_EN and O_CODE [7..0]. In O_CODE [7..0], the M function is expressed by a two-digit BCD code. It notifies that an M code is issued by setting M_EN to the status 1. The host device connected via the PLC interface, upon detecting the notification of the M code from the driver, performs the necessary processing on its own side, then responds to the driver by setting M_ANS to the status 1. The driver sets M_EN to 0 when receiving this response. Hereafter, the driver detects when M_ANS is set to the status 0 and completes the rest of the M function interface.



1.4.11 Jog Move Command (JOG_UP and JOG_DN)

A jog move command via the PLC interface functions in the idle status when the *#217 Jog move operation: RS232C selection* parameter is set so that operations are performed via the PLC interface.
A jog move is executed by JOG_UP and JOG_DN as explained in the table below.
In the idle status, a jog move is executed according to the command as it is. If a start command is issued during the jog move, the jog move is immediately decelerated and stopped, after which the operation is started. After that, when the operation ends, the jog move is performed according to the jog move command after stopping.

Jog (–) command JOG_DN	Jog (+) command JOG_UP	Jog move command
Status 0	Status 0	Stop command
	Status 1	(+) direction move command
Status 1	Status 0	(-) direction move command
	Status 1	Stop command

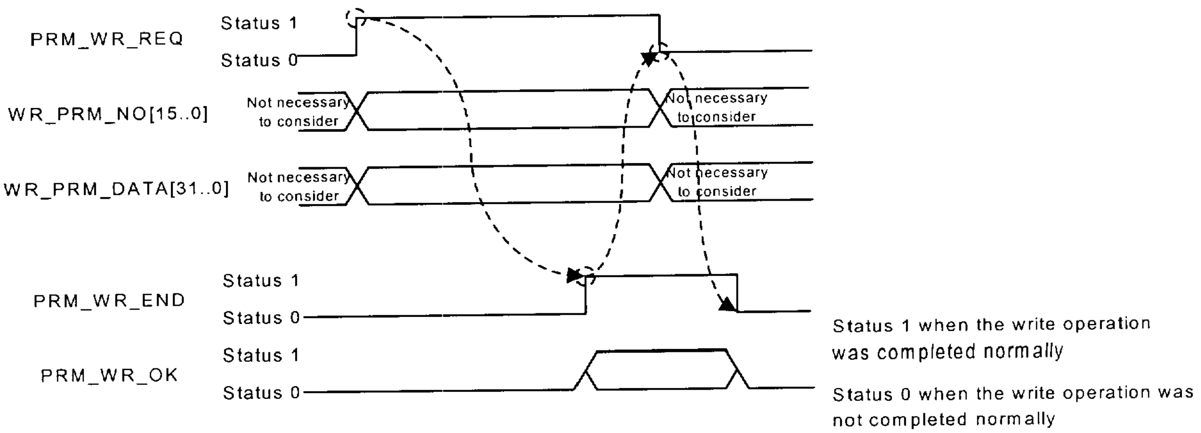
1.5 Operation 2

1.5.1 Error Code Request Function ERR_CODE_MAIN [7..0], ERR_CODE_SUB [7..0]

The error code request function operates irrespectively of the operation mode.
The driver notifies about the presence of an error code using ERR_CODE_MAIN [7..0] and ERR_CODE_SUB [7..0]. In ERR_CODE_MAIN [7..0], the error code is expressed as a two-digit BCD code. In ERR_CODE_SUB [7..0], the details about the error are expressed as a two-digit BCD code.
If no error has occurred, 0 is output as the error code.

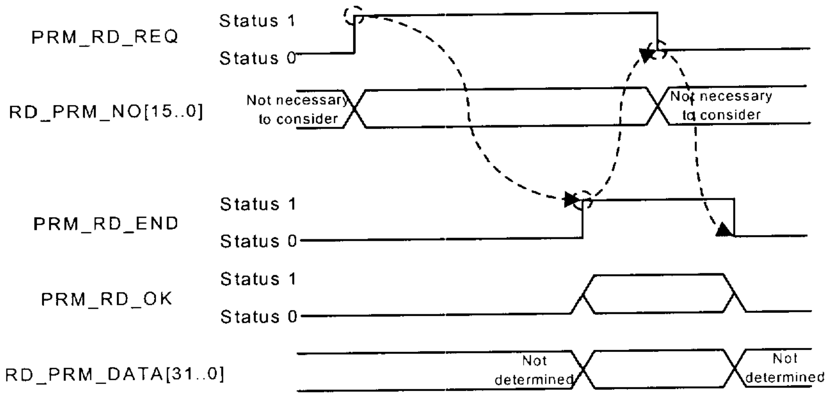
1.5.2 Parameter Write Function PRM_WR_REQ, WR_PRM_NO [15..0], WR_PRM_DATA [31..0], PRM_WR_END, PRM_WR_OK

The parameter write function operates irrespectively of the operation mode.
First, after storing the parameter number to be written in WR_PRM_NO and the parameter data to be written in WR_PRM_DATA, the host device connected to the PLC interface sets PRM_WR_REQ to the status 1. Then the driver processes the parameter write operation and sets PRM_WR_END to the status 1 to notify that the processing is completed. If the write operation is completed normally, it sets both PRM_WR_END and PRM_WR_OK to the status 1. If the write operation is not completed normally, PRM_WR_OK remains at the status 0 when PRM_WR_END is set to the status 1.


PRM_WR_REQ
Status 1
Status 0
WR_PRM_NO[15..0]
Not necessary to consider
Not necessary to consider
WR_PRM_DATA[31..0]
Not necessary to consider
Not necessary to consider
PRM_WR_END
Status 1
Status 0
PRM_WR_OK
Status 1
Status 0
Status 1 when the write operation was completed normally
Status 0 when the write operation was not completed normally

1.5.3 Parameter/Monitor Read Function PRM_RD_REQ, RD_PRM_NO [15..0], PRM_RD_END, PRM_RD_OK, RD_PRM_DATA [31..0]

The parameter/monitor read function operates irrespectively of the operation mode.
First, after storing the parameter/monitor number to be read in RD_PRM_NO, the host device connected to the PLC interface sets PRM_RD_REQ to the status 1. Then the driver processes the parameter/monitor read operation and sets PRM_RD_END to the status 1 to notify that the processing is completed. If the read operation is completed normally, it sets both PRM_RD_END and PRM_RD_OK to the status 1 and stores the read data in RD_PRM_DATA. If the read operation is not completed normally, PRM_RD_OK remains at the status 0 and RD_PRM_DATA becomes 0 when PRM_RD_END is set to the status 1.

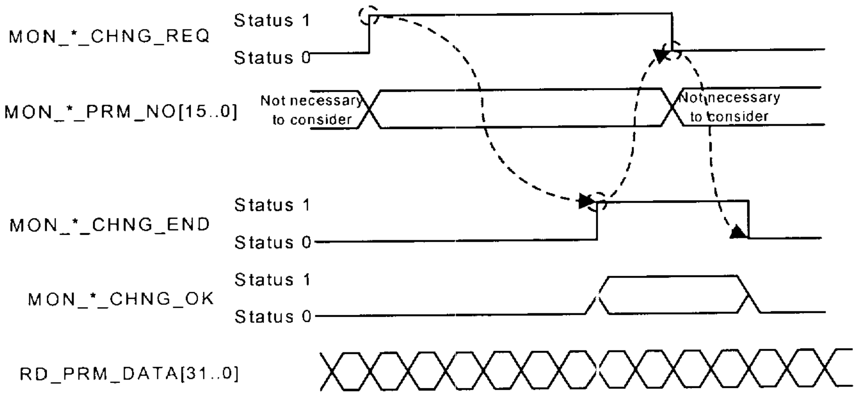


Status 1 when the read operation was completed normally

Status 0 when the read operation was not completed normally

1.5.4 Parameter/Monitor Display A and B Functions MON_A_CHNG_REQ, MON_A_PRM_NO [15..0], MON_A_CHNG_END, MON_A_CHNG_OK, MON_A_DATA [31..0] MON_B_CHNG_REQ, MON_B_PRM_NO [15..0], MON_B_CHNG_END, MON_B_CHNG_OK, MON_B_DATA [31..0]

The parameter/monitor display A and B functions operate irrespectively of the operation mode.
The purpose of these functions is to display set parameter/monitor values periodically. The refresh interval is approximately 10 msec. The parameter/monitor numbers are set to 320 for A and 321 for B by default when the power is turned on.
When changing a parameter/monitor number to be displayed, the host device connected to the PLC interface first stores the new parameter/monitor number in either MON_A_PRM_NO or MON_B_PRM_NO, and then sets MON_A_CHNG_REQ or MON_B_CHNG_REQ to the status 1 accordingly. The driver processes the parameter/monitor number change and sets either MON_A_CHNG_END or MON_B_CHNG_END to the status 1 to notify that the change processing is completed. If the number is changed normally, it sets MON_A_CHNG_END and MON_A_CHNG_OK or MON_B_CHNG_END and MON_B_CHNG_OK to the status 1. If the number is not changed normally, MON_A_CHNG_OK or MON_B_CHNG_OK remains at the status 0 when MON_A_CHNG_END or MON_B_CHNG_END is set to the status 1.



Status 1 when the write operation was completed normally

Status 0 when the write operation was not completed normally

Notifies the value corresponding to the parameter/monitor value before the change.

Notifies the value corresponding to the parameter/monitor value after the change.

1.6 Reference

1.6.1 ODVA JAPAN

Kyoto Research Park, Chudoji-Minami-Cho, Shimogyo-Ku, Kyoto-Shi 600-8813 Japan
c/o Advanced Software Technology & Mechatronics Research Institute of KYOTO
TEL: 075-315-9175, FAX: 075-315-2898
E-mail: odva_jp@odva.astem.or.jp
URL: http://www.odva.astem.or.jp/