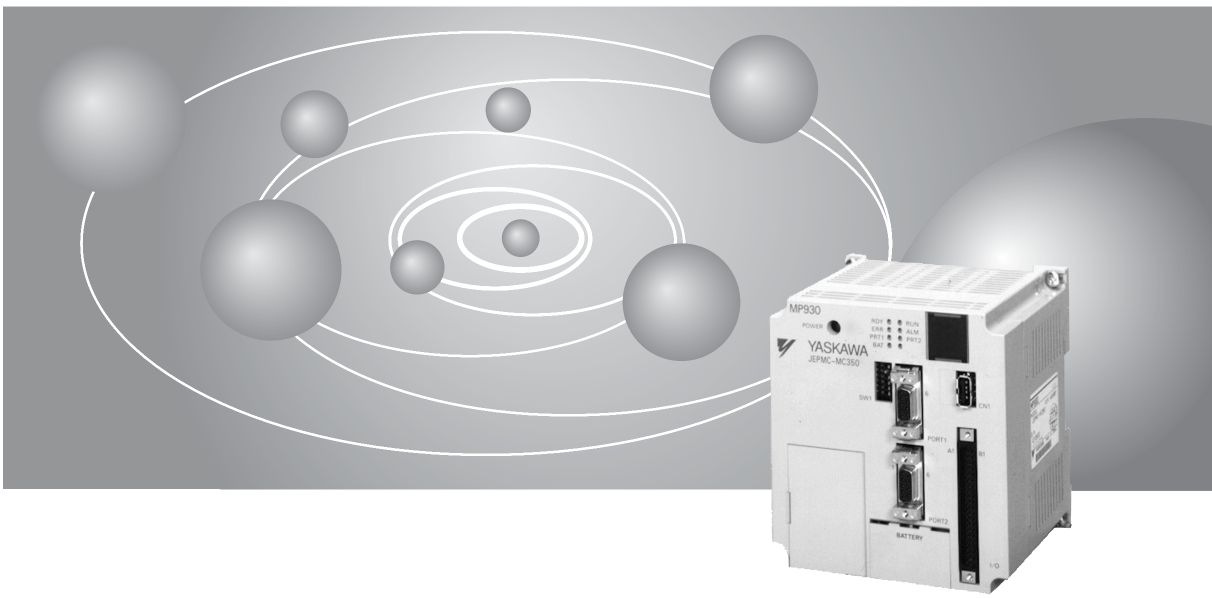
Machine Controller MP930

USER'S MANUAL
DESIGN AND MAINTENANCE



YASKAWA

MANUAL NO. SIEZ-C887-1.1

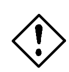
Safety Information

The following conventions are used to indicate precautions in this manual. Failure to heed precautions provided in this manual can result in serious or possibly even fatal injury or damage to the products or to related equipment and systems.

WARNING Indicates precautions that, if not heeded, could possibly result in loss of life or serious injury.

Caution Indicates precautions that, if not heeded, could result in relatively serious or minor injury, damage to the product, or faulty operation.

The warning symbols for ISO and JIS standards are different, as shown below.

ISO	JIS
	

The ISO symbol is used in this manual.

Both of these symbols appear on warning labels on Yaskawa products. Please abide by these warning labels regardless of which symbol is used.

Visual Aids

The following aids are used to indicate certain types of information for easier reference.

◀EXAMPLE▶ Indicates application examples.

INFO Indicates supplemental information.

IMPORTANT Indicates important information that should be memorized.

TERMS Describes technical terms that are difficult to understand, or appear in the text without an explanation being given.

TABLE OF CONTENTS

9 Troubleshooting

A External Dimensions

B Command List

C Parameter Lists

D Monitor Parameter Alarm List

E List of System Registers

Overview

■ About this Manual

This manual describes the design and maintenance for the MP930 Machine Controller, including the following information.

- Overview and component specifications
- Installation and wiring
- Examples of internal panel layout and drilling plan

Read this manual carefully to ensure the proper use of the MP930 Machine Controller. Also, keep this manual in a safe place so that it can be referred to whenever necessary.

■ Related Manuals

Refer to the following related manuals as required.

Thoroughly check the specifications, restrictions, and other conditions of the product before attempting to use it.

Manual Name	Manual Number	Contents
MP9□□ Machine Controller Ladder Programming User's Manual	SIEZ-C887-1.2	Describes the instructions used in MP9□□ ladder logic programming.
MP9□□ Machine Controller Motion Programming User's Manual	SIEZ-C887-1.3	Describes the motion instructions used by MP9□□ Machine Controllers.
MP9□□ Machine Controller Programming Software User's Manual	Part 1: SIEZ-C887-2.2–1 Part2: SIEZ-C887-2.2–2	Describes the installation and operating procedures for the CP-717 Engineering Tool Programming Software for MP9□□ Machine Controllers.

Using This Manual

■ Intended Audience

This manual is intended for the following users.

- Those responsible for estimating the MP930 system
- Those responsible for deciding whether to apply the MP930 system
- Those responsible for designing the MP930 system so that it can be mounted in the control and operating panels
- Those responsible for making, inspecting, testing, adjusting, and maintaining the control and operating panels in which the MP930 is mounted

■ Description of Technical Terms

In this manual, the following terms are defined as follows:

- **PP** = Programming Panel
- **PC** = Programmable Logic Controller
- "—" in "MOV [axis1]—..." represents numeric data for axis 1.

Safety Precautions

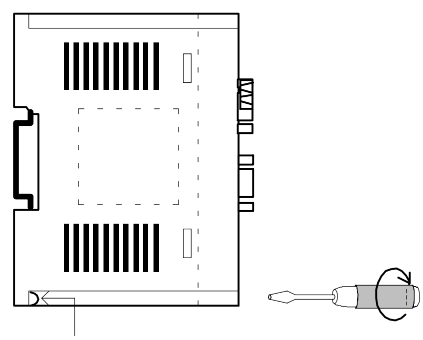
This section describes precautions that apply to ladder programming. Before programming, always read this manual and all other documents provided to ensure correct programming. Before using the equipment, familiarize yourself with equipment details, safety information, and all other precautions.

■ Installation

⚠ Caution

- Firmly tighten the Module mounting screws and terminal block mounting screws to prevent them from loosening during operation.

  Loose screws may result in a malfunction of the MP930.



Module mounting screw (M4, Phillips head)

- Be sure to turn OFF the MP930 before installing it.
- Insert the connectors of the cables that are to be connected to the MP930 and secure them well.

  Incorrect insertion of the connectors may result in a malfunction of the MP930.

■ Wiring

⚠ Caution

- Always connect a power supply that meets the given specifications.

  Connecting an inappropriate power supply may cause fires.
- Wiring must be performed by qualified personnel.

  Incorrect wiring may cause fires, product failure, or malfunctions.
- Do not accidentally leave foreign matter such as wire chips on the Mounting Base or in the Module when wiring.

  This may cause fires, failures, and malfunctions.

Mandatory

- Always use the ground the FG terminal to a ground resistance 100 Ω or less.

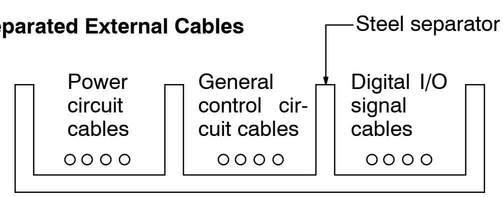
  Failure to ground the MP930 may result in electrical shocks or malfunctioning.

Select, separate, and lay external cables correctly.

- Consider the following items when selecting the I/O signal lines (external cables) to connect the MP930 Module to external devices.
  - Mechanical strength
  - Noise interference
  - Wiring distance
  - Signal voltage, etc.
- Separate the I/O signal lines from the power lines both inside and outside the control panel to reduce the influence of noise from the power lines.

  If the I/O signal lines and power lines are not separated properly, malfunctioning may result.

Example of Separated External Cables



■ Application Precautions

WARNING

- Do not touch any Module terminals when the system power is ON.

  There is a risk of electrical shock.

Caution

- Do not attempt to modify the MP930 programs, force outputs, switch between RUN and STOP, or performed other similar operations while the MP930 is operating without knowing the direct and indirect consequences of the operation.

  Incorrect programming or operation may damage the equipment or cause an accident.

■ Maintenance Precautions

⚠ WARNING

- Make sure that the polarity of the Module's built-in battery is correct. The battery must be installed correctly and must not be charged, disassembled, heated, thrown into fire, or short-circuited.

  Improper handling may cause the battery to explode or ignite.

⚠ Caution

- Do not attempt to disassemble or modify the MP930 Module in any way.

  Doing so can cause fires, product failure, or malfunctions.
- The customer must not replace the built-in fuse.

  If the customer replaces the built-in fuse, the MP930 Module may malfunction or break down.
  The built-in fuse must always be replaced by Yaskawa service staff.

■ General Precautions

Always note the following to ensure safe use.

- MP930 was not designed or manufactured for use in devices or systems directly related to human life. Users who intend to use the product described in this manual for special purposes such as devices or systems relating to transportation, medical, space aviation, atomic power control, or underwater use must contact Yaskawa Electric Corporation beforehand.
- MP930 has been manufactured under strict quality control guidelines. However, if this product is to be installed in any location in which a failure of MP930 involves a life and death situation or in a facility where failure may cause a serious accident, safety devices MUST be installed to minimize the likelihood of any accident.
- Drawings in this manual show typical product examples that may differ somewhat from the product delivered.
- This manual may change without prior notice due to product improvements and specification changes or for easier use. We will update the manual number of the manual and issue revisions when changes are made. The revision number of the revised manual appears on the back of the manual.
- Contact your nearest Yaskawa sales representative or the dealer from whom you purchased the product and quote the manual number on the front page of the manual if you need to replace a manual that was lost or destroyed.
- Contact your nearest Yaskawa sales representative or the dealer from whom you purchased the product to order new nameplates whenever a nameplate becomes worn or damaged.
- Products modified by the customer are not covered by the Yaskawa warranty, nor does Yaskawa assume any liability for injury or damage that may result from such modifications.

1

MP930

This chapter describes the type of work that can be done by the MP930 system, and gives an easy-to-understand overview of the MP930.

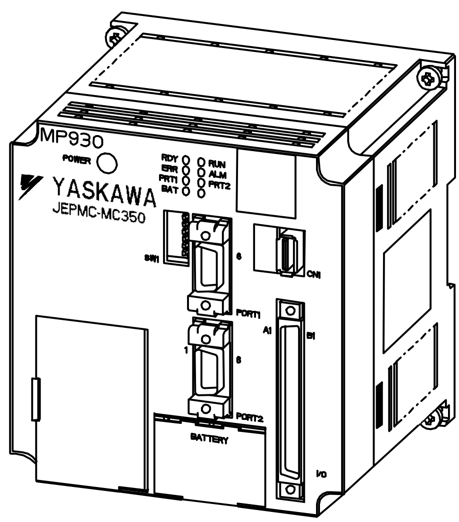
1.1 Overview of the MP930

This section gives an overview of the MP930.

1.1.1 Appearance of MP930 Units

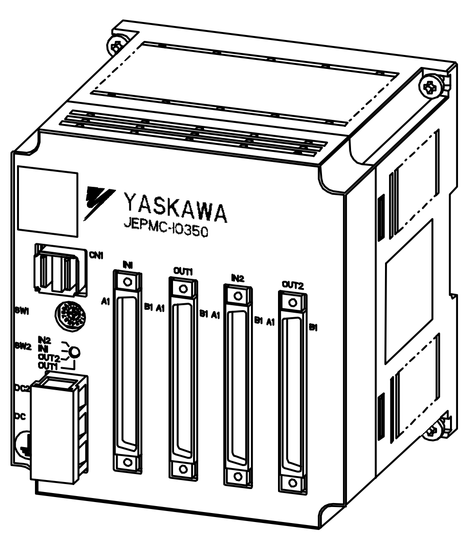
■ Appearance of the MC Units

MC Unit Model No.: The following illustration shows the appearance of the JEPMC-MC350 Machine Controller Unit.



■ Appearance of the I/O Units

I/O Unit Model No.: The following illustration shows the appearance of the JEPMC-IO350 I/O Unit.



1.1.2 Features of the MP930

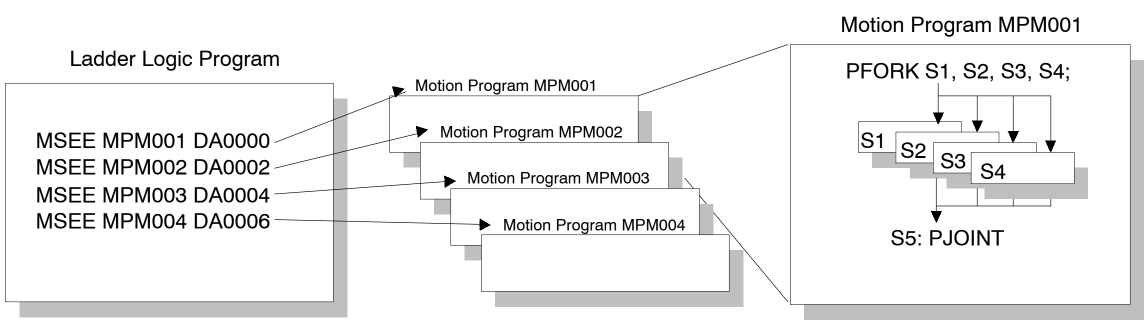
The MP930 has the following features:

■ The MP930 Machine Controller Unit is a micro machine controller with a one-piece construction that integrates the power supply, CPU, communications, and I/O.

The MP930 consists of an MC Unit, which provides both motion control functions and sequence control functions, and an I/O Unit. The servo amplifier and the I/O Unit are connected to the MC Unit by a high-speed field network called MECHATROLINK. One MC Unit can control a total of 14 servos and I/O Units.

■ Motion programs can be executed in parallel.

- Multiple motion programs can be executed in parallel.
- Four program blocks can be executed in parallel in one motion program.

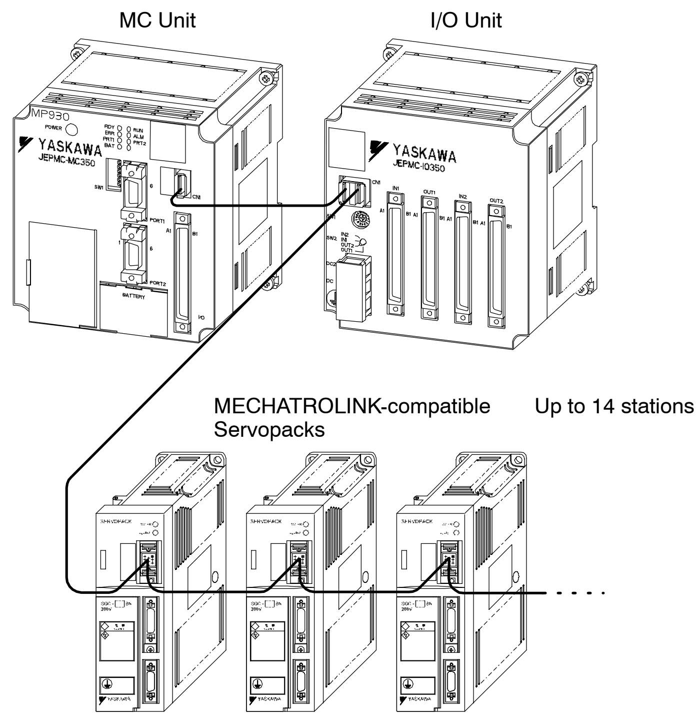


■ Reduced Startup Time and High Reliability

The use of the MECHATROLINK high-speed field network reduces startup time.

- The control panel startup time is reduced.
- High reliability is achieved.
  - **a)** Shielded twisted-pair cable
  - **b)** Only three lines to wire
  - **c)** Half the number of connectors

1



■ Wide Range of Motion Commands

Commands such as those shown in the following table can be used.

Complete Range of Move Commands	POSITIONING, LINEAR INTERPOLATION, CIRCULAR INTERPOLATION, HELICAL INTERPOLATION, HOME RETURN, SKIP, SET TIME POSITIONING, and EXTERNAL POSITIONING
Immediately Effective Acceleration/Deceleration Commands	ACCELERATION/DECELERATION TIME, S-CURVE TIME CONSTANT, FEED SPEED SETTING, INTERPOLATION FEED ACCELERATION/DECELERATION TIME SETTING, MAXIMUM INTERPOLATION FEED SPEED SETTING, and so on
Many Control Commands	I/O WAIT, IF statement, WHILE statement, Parallel Execution commands, Select Execution commands, and so on
Wide Range of Math and Logic Commands	Integer arithmetic commands: ADD, SUBTRACT, MULTIPLICATION, DIVISION Floating-point commands: Addition, Subtraction, Multiply, Divide Logic operations, SET BIT, RESET BIT BCD-TO-BINARY, BINARY-TO-BCD Trigonometric functions, inverse trigonometric functions, SQUARE ROOT, and so on

1

1.2 Control Using the MP930

The MP930 is a machine controller with fully integrated sequence control and motion control. One Controller simultaneously performs motion control and sequence control.

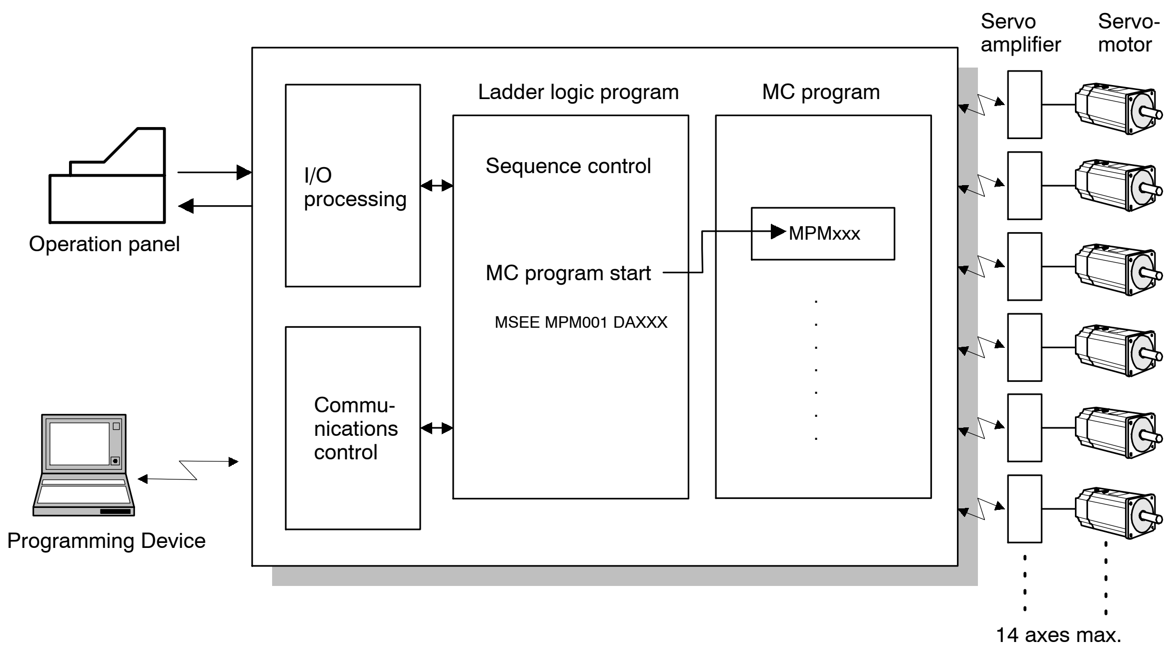


Figure 1.1 MP930 System Concept

1.2.1 Types of Program

Programs consist of ladder logic programs, which are designed mainly for sequence control, and motion programs (called MC programs), which are designed mainly for servo control.

1.2.2 Ladder Logic Programs

A ladder logic program is a program used for coding the sequence logic for conditional control and sequence control, and for coding the sequence logic that starts an MC program. The ladder logic program is created as the basic unit called drawings (DWGs).

■ Types of Drawing

The following types of drawing are provided: Start drawing, high-speed scan drawings, low-speed scan drawings, and user functions.

- Startup Drawings

  Startup drawings are executed once when the power is turned ON. The logic used to set constants and initialize operation is normally coded in these drawings.

- High-speed Scan Drawings

  High-speed scan drawings are executed at regular intervals. The scan time is within the range of 2 to 32 ms, and scan times can be set at 2-ms intervals. The circuits used to start the MC program are coded in the high-speed scan drawings.

- Low-speed Scan Drawings

  Low-speed scan drawings are executed at regular intervals. The scan time is within the range of 2 to 300 ms, and scan times can be set at 2-ms intervals. Sequence logic that does not require high-speed processing, such as lamp output and display circuits, should be coded in low-speed scan drawings.

- User Functions

  User functions are defined as user commands in the coded drawings, using the commands provided for the MP930. User functions can be used in startup drawings, high-speed scan drawings, and low-speed scan drawings.

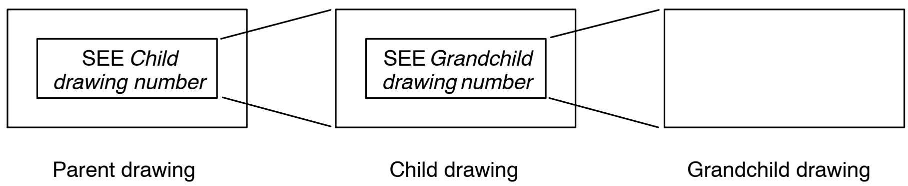
<table>
<tr><th>Types of Drawing</th><th>Maximum Number of Drawings</th><th>Drawing and Function Notation</th><th>Remarks</th></tr>
<tr><td>Startup Drawings</td><td>64</td><td>DWG.A</td><td rowspan="3">• 500 steps max. per drawing
• Equivalent to 20 Ksteps max. of ladder logic program memory
• Security function can be set separately for each drawing.</td></tr>
<tr><td>High-speed Scan Drawings</td><td>100</td><td>DWG.H</td></tr>
<tr><td>Low-speed Scan Drawings</td><td>100</td><td>DWG.L</td></tr>
<tr><td>User Functions</td><td>200</td><td>FUNC-xxx</td><td>• Separate revision history or each drawing.</td></tr>
</table>

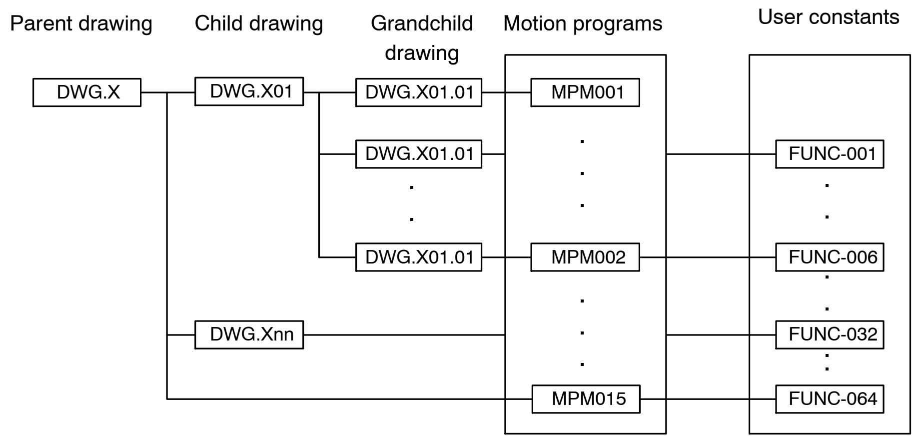
1

■ Configuration of Drawings

Drawings can be arranged in up to three hierarchical levels: parent, child, and grandchild drawings. These drawings are developed downward using the SEE instruction.

Functions can be started from any drawing.





INFO

1. X is replaced by A, H, or L.
2. A motion program can be started only from an H drawing.
3. Motion programs are started with the MSEE instruction.

1.2.3 MC Programs

An MC program codes the logic used for servo control in a motion control language. An MC program is started using the MOTION PROGRAM CALL instruction (MSEE) in the ladder logic program. There are two methods of designating an MC program: Direct designation of the program number, and indirect designation of the number of the register in which the MPM number is to be stored.

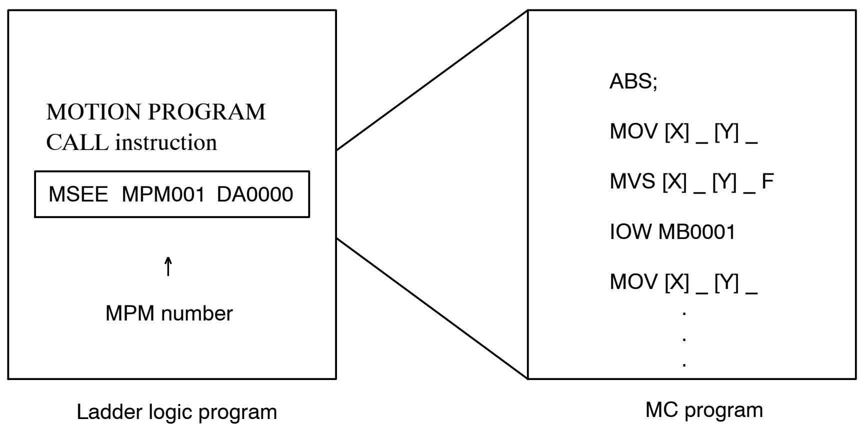


Figure 1.2 Starting an MC Program by Direct Designation

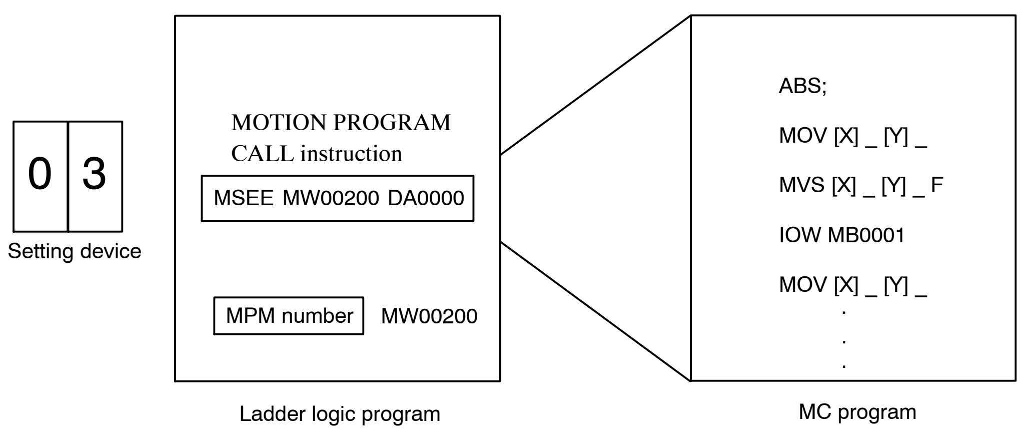
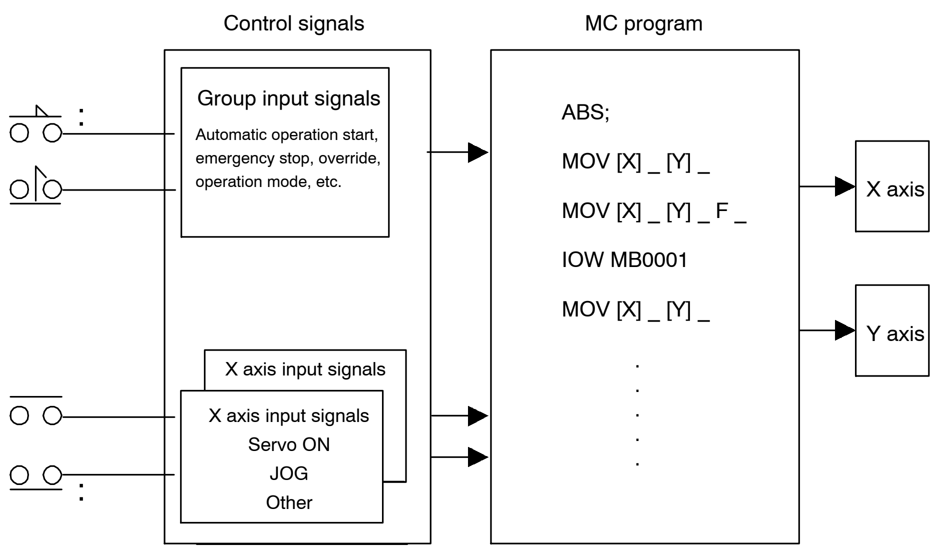


Figure 1.3 Starting an MC Program by Indirect Designation

1.2.4 Control Signals

There are two types of control signal: 1) Group input control signals, which are shared by groups, such as automatic operation start signals and emergency stop signals, and 2) Axis input signals, which are used by designated axes only, such as servo ON signals and JOG signals. The signals (variables) to be used as control signals are allocated on the Group Definition Screen.



1.3 Operation from Programming Devices

This section gives an overview of the types of operation that can be performed using peripheral devices.

A computer running the CP-717 programming software for the ladder logic programs and motion programs for the MP930 is called a “Programming Device.”

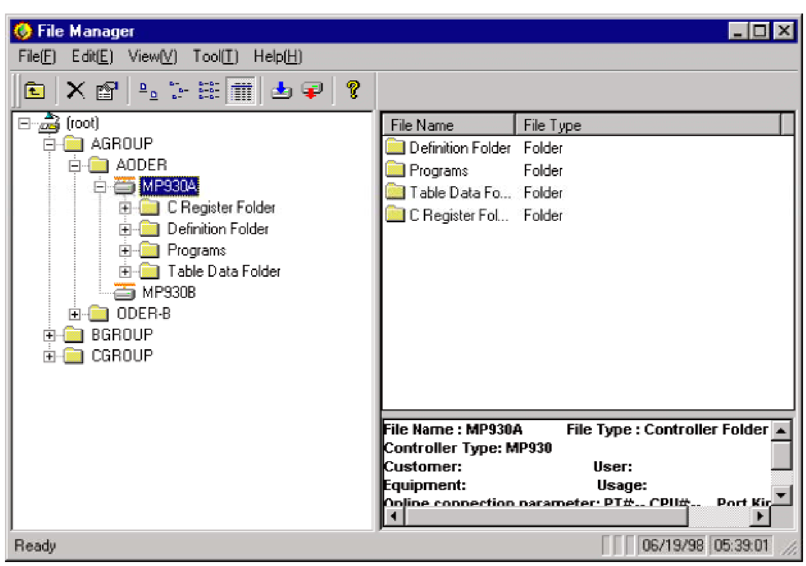
Communications with the Unit are enabled by connecting a Programming Device to the MEMO-BUS port of the MP930 MC Unit using a special cable.

The following operations can be performed from the Programming Device.

1.3.1 File Manager

The following folder and file management functions and file transfer functions are performed.

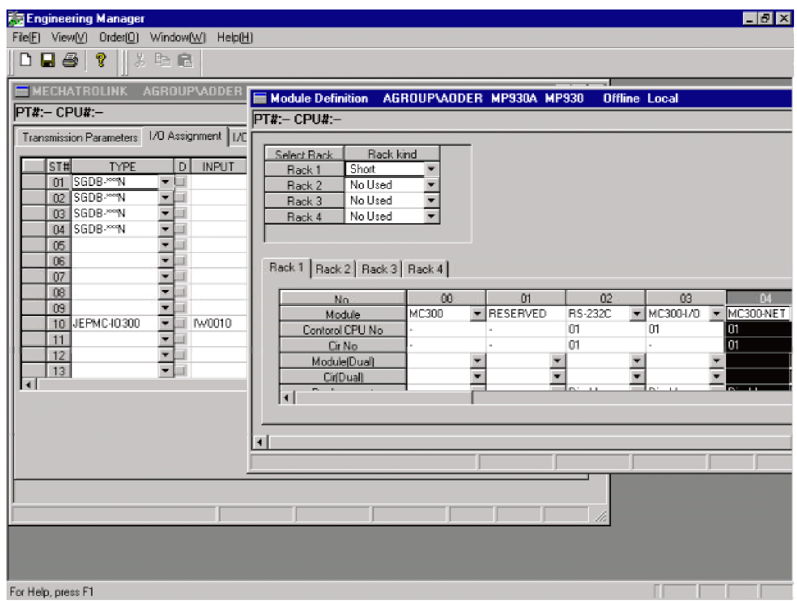
- File management
- User management
- File transfer
- Online/offline
- Logging off
- CPU control



1.3.2 System Information Definitions

The following information definitions are set.

- System definitions
- Scan time settings
- Application settings
- Module configurations
- Failure monitoring
- Data traces
- Group definitions
- Motion parameters

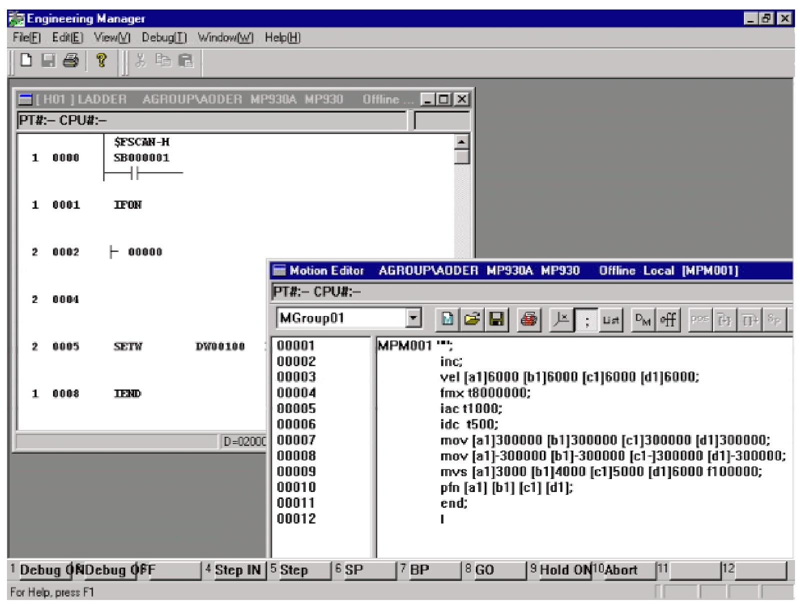


1

1.3.3 Programming

The following ladder logic programs and motion programs are edited.

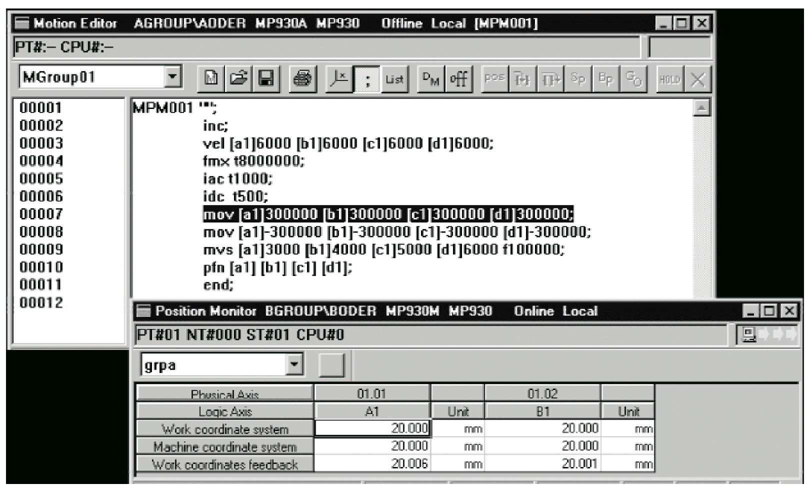
- Main program creation
- Table format program creation
- Adjustment panel creation
- C registers
- Table data definitions
- Motion editor



1.3.4 Debugging and Monitoring

The following functions are provided for debugging.

- Register list
- Adjustment panel
- Program monitor
- Position monitor
- Task monitor
- Failure monitor
- Data trace monitor



1.3.5 Printing

The following data created for definitions and programming can be printed.

- Definitions
- Drawings and functions
- Motions
- Table data
- Registers

2

MP930 Specifications and System Configuration

This chapter explains the MP930 Unit specifications, together with the products used in the system configuration of the MP930.

2.1 Specifications

This section gives an overview of the specifications and functions of the MP930 Units.

2.1.1 General Specifications

■ General Specifications of the MP930 Units

Table 2.1 lists the general specifications of the MP930 Units.

Table 2.1 General Specifications of the MP930 Units

Item		Specifications
Environmental Conditions	**Ambient Operating Temperature**	0 to 55 °C
	Storage Temperature	–25 to 85 °C
	Ambient Operating Humidity	30% to 95% RH (with no condensation)
	Ambient Storage Humidity	5% to 95% RH (with no condensation)
	Pollution Level	Pollution level 1 (conforming to JIS B 3501)
	Corrosive Gas	There must be no combustible or corrosive gas.
	Operating Altitude	2,000 m above sea level or lower
Electrical Operating Conditions	**Noise Resistance**	Conforming to JIS B 3502: 1,500 V (p-p) in either normal or common modes with a pulse width of 100 ns/1 µs and a rise time of 1 ns (tested with impulse noise simulator)
Mechanical Operating Conditions	**Vibration Resistance**	Conforming to JIS B 3502: 10 to 57 Hz with single-amplitude of 0.075 mm 57 to 150 Hz with fixed acceleration of 9.8 m/s^2 (1G) 10 sweeps each in X, Y, and Z directions (sweep time: 1 octave/min)
	Shock Resistance	Conforming to JIS B 3502: Peak acceleration of 147 m/s^2 (15G) twice for 11 ms each in the ±X, ±Y, and ±Z directions
Installation Requirements	**Ground**	Ground to 100 Ω max.
	Cooling Method	Natural cooling

2.1.2 Hardware Specifications

■ MP930 MC Unit Hardware Specifications

Table 2.2 lists the hardware specifications of the MP930 MC Unit.

Table 2.2 MC Unit Hardware Specifications

Item	Specifications
Name	MC Unit
Model Number	JEPMC-MC350
Memory	Flash: 1 MB RAM: 2 MB (battery backup)
Communications Ports	RS-232C × 2 ports Baud rate: 19.2 kbps Female 9-pin D-sub connector (special pin assignments) • MEMOBUS • No protocol (custom) • MELSEC
I/O Signals *1, *2	Inputs: 16 points 24 VDC (20.4 to 28.8 V) 5 mA, combined sinking/sourcing Outputs: 16 points 24 VDC, 50 mA, sinking outputs
Field Bus	MECHATROLINK (high-speed field network) Up to 14 servos and I/O stations can be connected.
Display Switch	Unit status indicators I/O signal indicators DIP switch for mode setting
Power Supply	24 VDC (20.4 to 28.8 V) Rated current: 1 A; In-rush current: 50 A
Dimensions	120 (W) × 130 (H) × 105 (D) mm

*** 1.** The I/O signal functions can be allocated.

*** 2.** The Expansion I/O Unit can be used with MECHATROLINK communications.

2

■ I/O Unit Hardware Specifications

Table 2.3 lists the hardware specifications of the I/O Unit.

Table 2.3 I/O Unit Hardware Specifications

Item	Specifications
Name	I/O Unit
Model Number	JEPMC-IO350
I/O Signals	Inputs: 64 points 24 VDC, 5 mA, combined sinking/sourcing Outputs: 64 points 24 VDC, 50 mA, sinking outputs (all points ON) * Signal connection method: Connector (FCN360 Series)
MC Unit Interface	MECHATROLINK (high-speed field network)
Unit Power Supply	24 VDC (20.4 to 28.8 V) Rated current: 0.5 A; inrush current: 1 A
Dimensions	120 (W) × 130 (H) × 105 (D) mm

* The maximum rating per point is 100 mA (depending on derating conditions).

2.1.3 Function Lists

■ MP930 Motion Control Function Specifications

Table 2.4 lists the motion control function specifications for the MP930.

Table 2.4 MP930 Motion Control Function Specifications

Item		Specification
Number of Controlled Axes		1 to 14 axes
Control Specifications	**PTP Control**	Linear, rotary, infinite-length, and independent axes
	Interpolation	Up to 14 linear axes, 2 circular axes, and 3 helical axes
	Speed Control	None
	Torque Limit	Yes (According to parameter setting only)
Reference Unit		mm, inch, deg, pulse
Reference Unit Minimum Setting		1, 0.1, 0.01, 0.001, 0.0001, 0.00001
Maximum Programmable Value		–2147483648 to +2147483647 (signed 32-bit value)
Speed Reference Unit		mm/min, inch/min, deg/min, pulse/min
Acceleration/Deceleration Type		Linear, asymmetric, S-curve Asymmetric acceleration/deceleration is not possible with POSITIONING (MOV).
Override Function		Positioning: 0.01% to 327.67% by axis Interpolation: 0.01% to 327.67% by group

<table>
<tr><th colspan="2">Item</th><th>Specification</th></tr>
<tr><td colspan="2">Coordinate System</td><td>Rectangular coordinates</td></tr>
<tr><td colspan="2">Zero Point Return</td><td>Four types
Dog + phase C, zero point limit switch, dog + zero point limit switch, phase C

Home position setting function provided.</td></tr>
<tr><td rowspan="4">Programs</td><td>Language</td><td>Special motion language</td></tr>
<tr><td>Number of Tasks</td><td>Multiple programs can be executed in parallel.</td></tr>
<tr><td>Number of Programs</td><td>Up to 256</td></tr>
<tr><td>Program Capacity</td><td>Equivalent to 80 Kbytes (characters)
(Can be increased or decreased according to the size of ladder logic program used; maximum of 100 Kbytes.)</td></tr>
<tr><td colspan="2">Applicable Servopack</td><td>SGD-□□□N/SGDB-□□AN</td></tr>
<tr><td colspan="2">Encoder</td><td>Incremental or absolute</td></tr>
<tr><td colspan="2">Command Words</td><td>Axis Move Commands: 8 commands
MOV, MVS, MCW, MCC, ZRN, SKP, MVT, EXM

Basic Control Commands: 6 commands
ABS, INC, POS, PLN, MVM, PLD

Speed and Acceleration/Deceleration Commands: 7 commands
ACC, SCC, VEL, IAC, IDC, IFP, FMX

High-level Control Commands: 4 commands
PFN, INP, SNG, UFC

Control Commands: 10 commands
MSEE, TIM, IOW, END, RET, EOX, IF ELSE IEND, WHILE WEND, PFORK JOINTO PJOINT, SFORK JOINTO SJOINT

Math and Sequence Control Commands: 32 commands
=, +, −, *, /, MOD, |, ^, &, !, (), S{}, R{}, SIN, COS, TAN, ASN, ACS, ATN, SQRT, BIN, BCD, ==, <>, >, <, >=, <=, SFR, SFL, BLK, CLR</td></tr>
</table>

■ PLC Function Specifications

Table 2.5 lists the PLC function specifications.

Table 2.5 PLC Function Specifications

Item	Specifications
Program Capacity	Equivalent to 4 Ksteps (varies according to amount of motion program used; 20 Ksteps max.)
Control Method	Sequence: High-speed and low-speed scan methods
Programming Language	CP language Ladder logic diagram: Relay circuit Text-type language: Numeric operations, logic operations, etc.

2

Item	Specifications
Scanning	Two scan levels: High-speed scan and low-speed scan High-speed scan time setting: 2 to 32 ms (2 ms units) Low-speed scan time setting: 2 to 300 ms (0.1 ms units)
User Drawings and Functions Motion Programs	Start drawings (DWG.A): 64 drawings max. Up to three hierarchical drawing levels High-speed scan process drawings (DWG.H): 100 drawings max. Up to three hierarchical drawing levels Low-speed scan process drawings (DWG.L): 100 drawings max. Up to three hierarchical drawing levels Number of steps: Up to 500 steps per drawing User functions: Up to 200 functions Motion programs: Up to 256 Revision history of drawings and motion programs Security function for drawings and motion programs
Data Memory	Common data (M) registers: 32 Kwords System (S) registers: 1 Kwords Drawing local (D) registers: Up to 16 Kwords per drawing Drawing constant (#) registers: Up to 16 Kwords per drawing Input (I) registers: 2 Kwords (including internal input registers) Output (O) registers: 2 Kwords (including internal output registers) Constant (C) registers: 4 Kwords
Trace Memory	Data trace: 128 Kwords (32 Kwords × 4 groups), 16 points defined Failure trace: 32 Kwords, 500 items defined
Memory Backup	User memory: CMOS battery backup
Data Types	Bit (relay): ON/OFF Integer: −32768 to +32767 Double integer: –2147483648 to +2147483647 Real number: ± (1.175E – 38 to 3.402E + 38)
Register Designation Method	Register number: Direct designation of register number Symbolic designation: Up to 8 alphanumeric characters (up to 200 symbols per drawing) With automatic number or symbol assignment
Instructions	Program control instructions: 14 instructions Direct I/O instructions: 2 instructions Relay circuit instructions: 14 instructions (including set and reset coils) Logic operation instructions: 3 instructions Numeric operation instructions: 16 instructions Numeric conversion instructions: 9 instructions Numeric comparison instructions: 7 instructions Data manipulation instructions: 14 instructions Basic function instructions: 10 instructions Table data manipulation instructions: 11 instructions DDC instructions: 13 instructions System functions: 14 instructions

■ Motion Commands

Table 2.6 lists the motion commands.

Table 2.6 Motion Command List

Classification	Command	Function
Axis Move Commands	**MOV**	Positioning
	MVS	Linear interpolation
	MCC	Counterclockwise circular interpolation, Helical circular interpolation (counterclockwise)
	MCW	Clockwise circular interpolation, Helical circular interpolation (clockwise)
	ZRN	Zero point return
	SKP	Skip
	MVT	Set time positioning
	EXM	External positioning
Basic Control Commands	**ABS**	Absolute mode
	INC	Incremental mode
	POS	Current position set
	PLN	Coordinate plane setting
	MVM	Move on machine coordinate
	PLD	Program current position updating
Speed and Acceleration/ Deceleration Commands	**ACC**	Acceleration time change
	SCC	S-curve time constant change
	VEL	Set velocity
	IAC	Interpolation acceleration time change
	IDC	Interpolation deceleration time change
	IFP	Interpolation feed speed ratio setting
	FMX	Maximum interpolation feed speed setting
High-Level Control Commands	**PFN**	In-position check
	INP	Second in-position check
	SNG	Ignore single block signal
	UFC	User function call

Classification	Command	Function
Control Commands	**MSEE**	Subroutine call
	TIM	Dwell time
	IOW	I/O wait
	END	Program end
	RET	Subroutine end
	EOX	One scan wait
	IF **ELSE** **IEND**	Branching commands
	WHILE **WEND**	Repeat commands
	PFORK **JOINTO** **PJOINT**	Parallel execution commands
	SFORK **JOINTO** **SJOINT**	Selective execution commands
Sequence Commands	**=**	Substitution
	+, –, *, /, MOD	Arithmetic operations
	|, ^, &, !	Logic operations
	SIN, COS, TAN, ASN, ACS, ATN, SQRT, BIN BCD	Function commands
	==, <>, >, <, >=, <=	Numeric comparison commands
	SFR, SFL, BLK, CLR	Data operation
	( ), S{ }, R{ }	Others

■ Motion Command Descriptions

Table 2.7 describes the motion commands.

Table 2.7 Motion Command Description

Classification	Command	Name	Programming Format	Function/Meaning
Axis Move Commands	**MOV**	POSITIONING	MOV [*axis1*] – [*axis2*] – ···; (Up to 14 axes can be designated.)	Executes positioning at rapid traverse speed for up to 14 axes simultaneously. In programming, replace "–" with the numerical data for each axis.
	MVS	LINEAR INTERPOLATION	MVS [*axis1*] – [*axis2*] – ···F–; (Up to 14 axes can be designated.)	Executes linear travel at interpolation feed speed F for up to 14 axes simultaneously.
	MCW **MCC**	CLOCKWISE CIRCULAR INTERPOLATION COUNTERCLOCKWISE CIRCULAR INTERPOLATION	MCW [*axis1*] – [*axis2*] – R–F–; MCC [*axis1*] – [*axis2*] – U– V– T– F–;	Executes circular interpolation at tangential speed F for two axes simultaneously following radius R (or designated center point coordinates). With the center point coordinate designation, multiple circles can be designated with T–. (T– can also be omitted.)
	MCW **MCC**	CLOCKWISE HELICAL INTERPOLATION COUNTERCLOCKWISE HELICAL INTERPOLATION	MCW [*axis1*] – [*axis2*] –U–V–[*axis3*] –T– F–; MCC [*axis1*] – [*axis2*] –R–[*axis3*] –F–;	Moves three axes simultaneously in a combination of circular interpolation and linear interpolation outside of the circular interpolation plane. Speed F will be the circular interpolation tangential speed. With the center point coordinate designation, the number of turns can be designated with T–. (T– can also be omitted.)
	ZRN	ZERO POINT RETURN	ZRN [*axis1*] – [*axis2*] – ···; (Up to 14 axes can be designated.)	Returns each axis to its zero point.
	SKP	SKIP	SKP [*axis1*]– [*axis2*]– ··· SS–; (Up to 14 axes can be designated.)	If the SKIP signal turns ON during a linear interpolation operation, skips the remaining movement and proceeds to the next block.
	MVT	SET TIME POSITIONING	MVT [*axis1*]– [*axis2*]– ··· T–; (Up to 14 axes can be designated.)	Executes positioning by clamping the feed speed so that travel can be completed at the designated time.
	EXM	EXTERNAL POSITIONING	EXM [*axis1*]– D–;	When an external positioning signal is input while external positioning is being executed, only the travel distance designated by "D–" is positioned with an incremental value, and then the next command is executed.

2

Classification	Command	Name	Programming Format	Function/Meaning
Basic Control Commands	**ABS**	ABSOLUTE MODE	ABS;	Treats all subsequent coordinate words as absolute values.
	INC	INCREMENTAL MODE	INC;	Treats all subsequent coordinate words as incremental values.
	POS	CURRENT POSITION SET	POS [*axis1*] – [*axis2*] – ⋯;	Changes the current values to the desired coordinate values for up to 14 axes simultaneously. Subsequent move commands use this new coordinate system.
	PLN	COORDINATE PLANE SETTING	PLN [*axis1*] [*axis2*]	Designates the coordinate plane to be used for a command requiring a plane designation command.
	MVM	MOVE ON MACHINE COORDINATE	MVM MOV [*axis1*]– [*axis2*]–; or MVM MVS [*axis1*]– [*axis2*]–;	Goes to the target position on the machine coordinate system. The coordinate system set automatically on completion of the zero point return is called a machine coordinate system. This coordinate system is not affected by the POS command.
	PLD	PROGRAM CURRENT POSITION UPDATE	PLD [*axis1*] – [*axis2*] – ⋯;	Updates the program current position for axes shifted by manual intervention. Up to 14 axes can be designated.
Speed and Acceleration/ Deceleration Commands	**ACC**	ACCELERATION TIME CHANGE	ACC [*axis1*] – [*axis2*] – ⋯;	Sets the acceleration time for linear acceleration/deceleration for up to 14 axes simultaneously.
	SCC	S-CURVE TIME CONSTANT CHANGE	SCC [*axis1*] – [*axis2*] – ⋯;	Sets the time constant for moving average acceleration/deceleration for up to 14 axes simultaneously.
	VEL	SET VELOCITY	VEL [*axis1*] – [*axis2*] – ⋯;	Sets the feed speed for up to 14 axes.
	IAC	INTERPOLATION ACCELERATION TIME CHANGE	IAC T–;	Sets the acceleration time for linear acceleration/deceleration for interpolation travel.
	IDC	INTERPOLATION DECELERATION TIME CHANGE	IDC T–;	Sets the deceleration time for linear acceleration/deceleration for interpolation travel.
	IFP	INTERPOLATION FEED SPEED RATIO SETTING	IFP P–;	Designates the maximum feed % for the speed designation during an interpolation feed.
	FMX	MAXIMUM INTERPOLATION FEED SPEED SETTING	FMX T–;	Sets the maximum speed during an interpolation feed. The interpolation acceleration time is the time from “0” until this speed is reached.

Classification	Command	Name	Programming Format	Function/Meaning
High-Level Control Commands	**PFN**	IN-POSITION CHECK	MVS [*axis1*] – [*axis2*] – ··· PFN; or PFN [*axis1*] [*axis2*] ;	Proceeds to the next block after the positioning commanded by the interpolation travel command in the same block or a previous block enters the positioning completion range (parameter setting).
	INP	SECOND IN-POSITION CHECK	INP [*axis1*] – [*axis2*] – ···;	Proceeds to the next block after the positioning subsequently commanded by the interpolation travel command with PFN enters the second positioning completion range.
	SNG	IGNORE SINGLE BLOCK SIGNAL	SNG MVS [*axis1*] 100. [*axis2*] 200. F1000;	A block with this command will be executed continuously, even in single-block operation mode. SNG cannot be designated on its own.
	UFC	USER FUNCTION CALL	UFC *Function_name* *Input_data*, *Input_address*, *Output_data*;	Calls a function created by the user.
Sequence Commands	**=**	SUBSTITUTE	(Result) = (Arithmetic expression)	Substitutes operation results. Performs calculations from left to right (with no order of priority).
	+	ADD	MW– = MW– + MW–; MW– = MW– + 123456; MW– = 123456 + MW–;	Performs integer and real number addition. Calculates combinations of integers and real numbers as real numbers.
	–	SUBTRACT	MW– = MW– – MW–; MW– = MW– – 123456; MW– = 123456 – MW–;	Performs integer and real number subtraction. Calculates combinations of integers and real numbers as real numbers.
	✻	MULTIPLY	MW– = MW– ✻ MW–; MW– = MW– ✻ 123456; MW– = 123456 ✻ MW–;	Performs integer and real number multiplication. Calculates combinations of integers and real numbers as real numbers.
	/	DIVIDE	MW– = MW–/MW–; MW– = MW–/123456; MW– = 123456/MW–;	Performs integer and real number division. Calculates combinations of integers and real numbers as real numbers.
	MOD	REMAINDER	MW– = MW–/MW–; MW– = MOD;	When programmed in the next block after a division, MOD stores the remainder in the designated register.

<table>
<tr><th>Classification</th><th>Command</th><th>Name</th><th>Programming Format</th><th>Function/Meaning</th></tr>
<tr><td rowspan="13">Sequence Commands</td><td>|</td><td>OR (logical OR)</td><td>MB– = MB– | MB–;
MB– = MB– | 1;
MW– = MW– | MW–;
MW– = MW– | H00FF;</td><td>Performs bit/integer logical OR.</td></tr>
<tr><td>^</td><td>XOR (logical exclusive OR)</td><td>MW– = MW– ^ MW–;
MW– = MW– ^ H00FF;</td><td>Performs integer logical exclusive OR.</td></tr>
<tr><td>&</td><td>AND (logical AND)</td><td>MB– = MB– & MB–;
MB– = MB– & 1;
MW– = MW– & MW–;
MW– = MW– & H00FF;</td><td>Performs bit/integer logical AND.</td></tr>
<tr><td>!</td><td>NOT (logical complement)</td><td>MB– = !MB–;
MB– = !1;
MW– = !MW–;
MW– = !H00FF;</td><td>Performs bit/integer logical complement (inverts bits).</td></tr>
<tr><td>( )</td><td>PARENTHESES</td><td>MW– = MW– & (MW– | MW–);</td><td>The logical arithmetic expression inside parentheses is calculated first.</td></tr>
<tr><td>S{ }</td><td>SET BIT</td><td>S{MB–} = MB– & MB–;</td><td>If the logical operation result is “true,” the designated bit turns ON. The designated bit does not turn OFF, even if the logical operation result is “false.”</td></tr>
<tr><td>R{ }</td><td>RESET BIT</td><td>R{MB–} = MB– & MB–;</td><td>If the logical operation result is “true,” the designated bit turns OFF. The designated bit does not turn ON, even if the logical operation result is “false.”</td></tr>
<tr><td>SIN</td><td>SINE</td><td>SIN (MW–);
SIN (90);</td><td>Obtains the sine of the integer or real number (deg), and returns a real value.</td></tr>
<tr><td>COS</td><td>COSINE</td><td>COS (MW–);
COS (90);</td><td>Obtains the cosine of the integer or real number (deg), and returns a real value.</td></tr>
<tr><td>TAN</td><td>TANGENT</td><td>TAN (MF–);
TAN (45.0);</td><td>Obtains the tangent of the real number (deg), and returns a real value.</td></tr>
<tr><td>ASN</td><td>ARC SINE</td><td>ASN (MF–);
ASN (45.0);</td><td>Obtains the arc sine of the real number (deg), and returns a real value.</td></tr>
<tr><td>ACS</td><td>ARC COSINE</td><td>ACS (MF–);
ACS (90.0);</td><td>Obtains the arc cosine of the real number (deg), and returns a real value.</td></tr>
</table>

Classification	Command	Name	Programming Format	Function/Meaning
Sequence Commands	**ATN**	ARC TANGENT	ATN (MW–); ATN (45);	Obtains the arc tangent of the integer or real number (deg), and returns a real value.
	SQRT	SQUARE ROOT	SQT (MW–); SQT (100);	Obtains the square root of the integer or real number, and returns a real value.
	BIN	BCD-TO-BINARY	BIN (MW–);	Converts BCD data to binary data.
	BCD	BINARY-TO-BCD	BCD (MW–);	Converts binary data to BCD data.
	= =	MATCH	IF MW– = = MW–; WHILE MW– = = MW–;	Used in an IF or WHILE conditional expression. If the left side and right side match, the condition is “true.”
	< >	MISMATCH	IF MW– < > MW–; WHILE MW– < > MW–;	Used in an IF or WHILE conditional expression. If the left side and right side do not match, the condition is “true.”
	>	GREATER THAN	IF MW– > MW–; WHILE MW– > MW–;	Used in an IF or WHILE conditional expression. If the left side is greater than the right side, the condition is “true.”
	<	LESS THAN	IF MW– < MW–; WHILE MW– < MW–;	Used in an IF or WHILE conditional expression. If the left side is less than the right side, the condition is “true.”
	> =	GREATER THAN OR EQUAL TO	IF MW– >= MW–; WHILE MW– >= MW–;	Used in an IF or WHILE conditional expression. If the left side is greater than or equal to the right side, the condition is “true.”
	< =	LESS THAN OR EQUAL TO	IF MW– <= MW–; WHILE MW– <= MW–;	Used in an IF or WHILE conditional expression. If the left side is less than or equal to the right side, the condition is “true.”
	SFR	RIGHT SHIFT	SFR MB– N– W–;	Shifts only the designated number of word variables to the right.
	SFL	LEFT SHIFT	SFL MB– N– W–;	Shifts only the designated number of word variables to the left.
	BLK	BLOCK MOVE	BLK MW– MW– W–;	Moves the block (constant designation) beginning with the designated bit (word) variable.
	CLR	CLEAR	CLR MB– W–;	Sets the number of constants specified in the variable group beginning with the designated bit (word) variable to OFF (“0”).

2

Classification	Command	Name	Programming Format	Function/Meaning
Control Commands	**MSEE**	SUBROUTINE CALL	MSEE MPS– ;	Executes the MPS– subroutine.
	TIM	DWELL TIME	TIM T–;	Waits for the period of time specified by T, and then proceeds to the next block.
	IOW	I/O WAIT	IOW MB– = = ***;	Stops execution of the motion program until the conditional expression given in the command is satisfied.
	END	PROGRAM END	END;	Ends the motion program.
	RET	SUBROUTINE RETURN	RET;	Ends the subroutine.
	EOX	ONE SCAN WAIT	EOX;	Separates continuous sequence instructions and forces a wait of one scan before continuing execution.

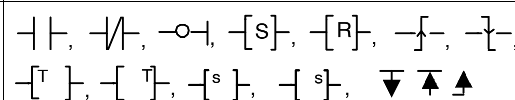
	IF
	ELSE
	IEND
		Branching commands	IF (conditional expression) ;
			(process 1)
			ELSE;
			(process 2)
			IEND;
				Executes process 1 if the conditional expression is satisfied, and executes process 2 if the conditional expression is not satisfied.

	WHILE **WEND**	Repeat commands	WHILE (conditional expression) ; ... WEND;	Repeatedly executes WHILE to WEND processing for as long as the conditional expression is satisfied.
	PFORK **JOINTO** **PJOINT**	Parallel execution commands	PFORK label 1, label 2,...; Label 1: Process 1 JOINTO label X Label 2: Process 2 JOINTO label X Label • • Label X: PJOINT;	Executes the blocks designated by the labels in parallel. With a subroutine, a maximum of two labels can be designated. Also, a motion command cannot be used in the block designated by the second label. END and RET cannot be used during parallel execution processing.
	SFORK **JOINTO** **SJOINT**	Selective execution commands	SFORK conditional expression 1? label 1, Conditional expression 2? label 2,...; Label 1: Process 1 JOINTO label X Label 2: Process 2 JOINTO label X Label • • Label X: SJOINT;	Executes process 1 if conditional expression 1 is satisfied, and executes process 2 if the conditional expression 2 is satisfied.

■ Ladder Instructions

Table 2.8 lists the ladder instructions.

Table 2.8 Ladder Instructions

Type of Instruction Word	Symbols
Instructions with [ ]	–
Program Control Instructions	SEE, MSEE, FOR FEND, WHILE ON/OFF WEND, IFON/ IFOFF ELSE IEND, FSTART, FIN, FOUT, DEND, COM-MENT, XCALL
Direct I/O Instructions	INS, OUTS
Relay Circuit Instructions	
Logic Operation Instructions	AND (∧), OR (∨), XOR (⊕)
Numeric Operation Instructions	⊢ ,⊩ , ⇒, +, −, + +, − −, ×, ÷, MOD, REM, INC, DEC, TMADD, TMSUB, SPEND
Numeric Conversion Instructions	INV, COM, ABS, BIN, BCD, PARITY, ASCII, BINASC, ASCBIN
Number Comparison Instructions	<, ≦, =, ≠, ≧, >, RCHK
Data Manipulation Instructions	ROTL, ROTR, MOVB, MOVW, XCHG, SETW, BEXTD, BPRESS, BSRCH, SORT, SHFTL, SHFTR, COPYW, BSWAP
Basic Function Instructions	SQRT, SIN, COS, TAN, ASIN, ACOS, ATAN, EXP, LN, LOG
DDC Instructions	DZA, DZB, LIMIT, PI, PD, PID, LAG, LLAG, FGN, IFGN, LAU, SLAU, PWM
Table Data Manipulation Instructions	TBLBR, TBLBW, TBLSRL, TBLSRC, TBLCL, TBLMV, QTBLR, QTBLRI, QTBLW, QTBLWI, QTBLCL
System Functions	COUNTER, FINFOUT, TRACE, DTRC-RD, FTRC-RD, ITRC-RD, INC-WR, ICNS-RD, MSG-SND, MSG-RCV

■ Ladder Instructions and Standard System Functions

Table 2.9 lists the ladder instructions and standard system functions.

Table 2.9 Ladder Instructions and Standard System Functions

Type	Name	Symbol	Abbreviated Instructions	Description
Program Control Instructions	Instructions with [ ]	–	–	–
	CHILD DRAWING CALL	SEE	SEE	Designate the child drawing number or the grandchild drawing number to be referenced after SEE. SEE H01
	DRAWING END	DEND	END	End of drawing (DWG)
	MOTION PROGRAM CALL	MSEE	MSEE	Designate the motion program number and the MSEE work register address to be referenced after MSEE. MSEE MPM001 DA00000
	FOR Structure	FOR : : FEND	FOR	Repeats execution statement 1 FOR V = a to b by c V: Can designate any integer register I or J. a, b, c: Can designate an any integer value (b > a > 0, c > 0). FEND: End of FOR instruction.
	WHILE Structure	WHILE : ON/OFF : WEND	WHILE ON OFF	Repeats execution statement 2 WEND: End of WHILE-ON/OFF instruction
	IF Structure -1, -2	IFON/IFOFF : ELSE : IEND	IFON IFOFF ELSE	Conditional execution statement IEND: End of IFON/IFOFF instruction
	FUNCTION CALL FUNCTION INPUT FUNCTION OUTPUT	FSTART	FSTART	Calls a function.
		FIN	FIN	Function input instruction Stores input data from the designated input register in the function input register.
		FOUT	FOUT	Function output instruction Stores output data from the function output register in the designated output register.
	COMMENT	"nnnnnnn"	"	A character string enclosed in quotation marks is treated as a comment.
	EXTENSION PROGRAM CALL	XCALL	XCALL	Calls an extension program.

Type	Name	Symbol	Abbreviated Instructions	Description
Direct I/O Instructions	INPUT STRAIGHT	INS	INS	INS MA00100 Executes the input and storage of data with interrupts disabled.
	OUTPUT STRAIGHT	OUTS	OUTS	OUTS MA00100 Executes the setting and output of data with interrupts disabled.
Relay Circuit Instruction	NO CONTACT		][	No limit in a series circuit. Bit designation of any register as a relay number is possible.
	NC CONTACT		]/	No limit in a series circuit. Bit designation of any register as a relay number is possible.
	COIL		@	MW0200 = 0001 MB000000 MB000000 IFON
	SET COIL	[S]	@S	MB000000 MB000010 [S]
	RESET COIL	[R]	@R	MB000020 MB000010 [R]
	RISING PULSE		]P	No limit in a series circuit. Bit designation of any register as a relay number is possible.
	FALLING PULSE		]N	No limit in a series circuit. Bit designation of any register as a relay number is possible.
	10-MS ON-DELAY TIMER	[T]	[ON	Set value: Timer register [T] Set value = any register or constant (setting unit: 10 ms) Timer register = M or D register
	10-MS OFF-DELAY TIMER	[T]	[OFF	
	1-S ON-DELAY TIMER	[s]	[SON	Set value: Timer register [S] Set value = any register or constant (setting unit: 10 ms) Timer register = M or D register
	1-S OFF-DELAY TIMER	[s]	[SOFF	
	Branching/convergence		,,.	A branching or convergence symbol can be connected to any of the above relay instructions.
Logic Operation Instructions	AND	<	&	Integer designation of any register or constant is possible.
	OR	>	|	Integer designation of any register or constant is possible.
	XOR	⊕	^	Integer designation of any register or constant is possible.

Type	Name	Symbol	Abbreviated Instructions	Description
Numeric Operation Instructions	INTEGER ENTRY	⊢	;	Starts an integer operation. ⊢ MW00280 + 00100 ⇒ MW00220
	REAL NUMBER ENTRY	⊩	;;	Starts a real number operation. ⊩MW00280 + 00100 ⇒ MW00220
	STORE	⇒	:	Stores the operation result in the designated register.
	ADDITION	+	+	Ordinary numeric addition (with operation error) ⊢ MW00280 +00100 ⇒ MW00220
	SUBTRACTION	–	–	Ordinary numeric subtraction (with operation error) ⊢ MW00280 –00100 ⇒ MW00220
	EXTENDED ADDITION	++	++	Closed numeric addition (without operation error) 0 → 32767 → –32768 → 0
	EXTENDED SUB-TRACTION	– –	– –	Closed numeric subtraction (without operation error) 0 → 32768 → –32767 → 0
	MULTIPLICA-TION	×	*	For integer and long integers, use × and ÷ in combination.
	DIVISION	÷	/	
	MOD	MOD	MOD	Gets the remainder of the division result. ⊢ MW00100 × 0100 ÷ 00121 MOD ⇒ MW00101
	REM	REM	REM	Gets the remainder of the division result. MF00200 REM 1.5 ⇒ MF00202
	INCREMENT	INC	INC	Adds 1 to the designated register. INC MW00100
	DECREMENT	DEC	DEC	Subtracts 1 from the designated register. DEC MW00100
	ADD TIME	TMADD	TMADD	Addition of hours, minutes, and seconds TMADD MW00000, MW00100
	SUBTRACT TIME	TMSUB	TMSUB	Subtraction of hours, minutes, and seconds TMSUB MW00000, MW00100
	SPEND TIME	SPEND	SPEND	Calculates the elapsed time between two times. SPEND MW00000, MW00100

Type	Name	Symbol	Abbreviated Instructions	Description
Numeric Conversion Instructions	SIGN INVERSION	INV	INV	⊦ MW00100 INV If MW00100 = 99, the operation result = –99.
	1'S COMPLE-MENT	COM	COM	⊦ MW00100 COM If MW00100 = FFFFH, the operation result = 0000H.
	ABSOLUTE VAL-UE CONVERSION	ABS	ABS	⊦ MW00100 ABS If MW00100 = –99, the operation result = 99.
	BINARY CON-VERSION	BIN	BIN	⊦ MW00100 BIN If MW00100 = 1234H (hexadecimal), the operation result = 1234 (decimal).
	BCD CONVER-SION	BCD	BCD	⊦ MW00100 BCD If MW00100 = 1234 (decimal), the operation result = 1234H (hexadecimal).
	PARITY CON-VERSION	PARITY	PARITY	Calculates the number of binary bits that are ON. If MW00100 = F0F0H, the operation result = 8.
	ASCII CONVER-SION 1	ASCII	ASCII	The designated character string is converted to ASCII code and substituted in the register. MW00200 "ABCDEFG"
	ASCII CONVER-SION 2	BINASC	BINASC	Converts 16-bit binary data to 4-digit hexadecimal ASCII code. BINASC MW00100
	ASCII CONVER-SION 3	ASCBIN	ASCBIN	Converts the numeric value indicated by a 4-digit hexadeci-mal ASCII code to 16-bit binary data. ASCBIN MW00100
Numeric Comparison Instructions	<	<	<	MB000010 ⊢ MW00000 < 10000 ——○⊣ MB000010 ⊢—┤├—— IFON
	≦	≦	< =	
	=	=	=	
	≠	≠	< >	
	≧	≧	> =	
	>	>	>	
	RANGE CHECK	RCHK	RCHK	Checks whether or not the value in the A register is in range. ⊦ MW00100 RCHK –1000, 1000

2

Type	Name	Symbol	Abbreviated Instructions	Description
Data Operation Instructions	BIT ROTATION LEFT and BIT ROTATION RIGHT	ROTR ROTL	ROTR ROTL	Example: ROTR Bit-addr Count Width ROTR MB00100A → N = 1 W = 20
	MOVE BITS	MOVB	MOVB	Source Desti. Width MOVB MB00100A → MB00200A W = 20
	MOVE WORD	MOVW	MOVW	Source Desti. Width MOVW MB00100 → MB00200 W = 20
	EXCHANGE	XCHG	XCHG	Source1 Source2 Width XCHG MB00100 → MB00200 W = 20
	SET WORDS	SETW	SETW	Desti. Data Width SETW MW00200 D = 00000 W = 20
	BYTE-TO-WORD EXPANSION	BEXTD	BEXTD	Expands the byte data stored in the word registers into words. BEXTD MW00100 to MW00200 B = 10
	WORD-TO-BYTE COMPRESSION	BPRESS	BPRESS	Collects the lower bytes of the word data stored in the word register area. BPRESS MW00100 to MW00200 B = 10
	BINARY SEARCH	BSRCH	BSRCH	Retrieves the register position that matches the data within the designated register range. BSRCH MW00000 W = 20 D = 100 R = MW00100
	SORT	SORT	SORT	Sorts registers within the designated register range. SORT MW00000 W = 100
	BIT SHIFT LEFT	SHFTL	SHFTL	Shifts the designated bit strings to the left. SHFTL MB00100A N = 1 W = 20
	BIT SHIFT RIGHT	SHFTR	SHFTR	Shifts the designated bit strings to the right. SHFTR MB00100A N = 1 W = 2
	COPY WORD	COPYW	COPYW	Copies the designated register range. COPYW MW00100 → MW00200 W = 20
	BYTE SWAP	BSWAP	BSWAP	The upper and lower bytes of the designated word are swapped. BSWAP MW00100

Type	Name	Symbol	Abbreviated Instructions	Description
Basic Function Instructions	SQUARE ROOT	SQRT	SQRT	Taking the square root of a negative number will result in the square root of the absolute value multiplied by −1. MF00100 SQRT |⊢
	SINE	SIN	SIN	Input = degrees |⊢MF00100 SIN
	COSINE	COS	COS	Input = degrees |⊢MF00100 COS
	TANGENT	TAN	TAN	Input = degrees |⊢MF00100 TAN
	ARC SINE	ASIN	ASIN	|⊢MF00100 ASIN
	ARC COSINE	ACOS	ACOS	|⊢MF00100 ACOS
	ARC TANGENT	ATAN	ATAN	|⊢MF00100 ATAN
	EXPONENT	EXP	EXP	|⊢MF00100 EXP e MF00100
	NATURAL LOG-ARITHM	LN	LN	|⊢MF00100 LN $\log_e$ (FM00100)
	COMMON LOG-ARITHM	LOG	LOG	|⊢MF00100 LOG $\log_{10}$ (FM00100)

2

Type	Name	Symbol	Abbreviated Instructions	Description
DDC Instructions	DEAD ZONE A	DZA	DZA	⊦ MW00100 DZA 00100
	DEAD ZONE B	DZB	DZB	⊦ MW00100 DZB 00100
	UPPER/LOWER LIMIT	LIMIT	LIMIT	⊦ MW00100 LIMIT –00100 00100
	PI CONTROL	PI	PI	⊦ MW00100 PI MA00200
	PD CONTROL	PD	PD	⊦ MW00100 PD MA00200
	PID CONTROL	PID	PID	⊦ MW00100 PID MA00200
	FIRST-ORDER LAG	LAG	LAG	⊦ MW00100 LAG MA00200
	PHASE LEAD/ LAG	LLAG	LLAG	⊦ MW00100 LLAG MA00200
	FUNCTION GENERATOR	FGN	FGN	⊦ MW00100 FGN MA00200
	INVERSE FUNCTION GENERATOR	IFGN	IFGN	⊦ MW00100 IFGN MA00200
	LINEAR ACCELERATOR/DECELERATOR 1	LAU	LAU	⊦ MW00100 LAU MA00200
	LINEAR ACCELERATOR/DECELERATOR 2	SLAU	SLAU	⊦ MW00100 SLAU MA00200
	PULSE WIDTH MODULATION	PWM	PWM	⊦ MW00100 PWM MA00200

Type	Name	Symbol	Abbreviated Instructions	Description
Table Data Operation Instructions	TABLE READ	TBLBR	TBLBR	TBLBR TBL1, MA00000, MA00100
	TABLE WRITE	TBLBW	TBLBW	TBLBW TBL1, MA00000, MA00100
	ROW SEARCH	TBLSRL	TBLSRL	TBLSRL TBL1, MA00000, MA00100
	COLUMN SEARCH	TBLSRC	TBLSRC	TBLSRC TBL1, MA00000, MA00100
	TABLE CLEAR	TBLCL	TBLCL	TBLCL TBL1, MA00000
	TABLE BLOCK MOVE	TBLMV	TBLMV	TBLMV TBL1, TBL2, MA00000
	QUEUE TABLE READ	QTBLR	QTBLR	QTBLR TBL1, MA00000, MA00100
	QUEUE TABLE READ AND IN-CREMENT	QTBLRI	QTBLRI	QTBLRI TBL1, MA00000, MA00100
	QUEUE TABLE WRITE	QTBLW	QTBLW	QTBLW TBL1, MA00000, MA00100
	QUEUE TABLE WRITE AND IN-CREMENT	QTBLWI	QTBLWI	QTBLWI TBL1, MA00000, MA00100
	QUEUE POINTER CLEAR	QTBLCL	QTBLCL	QTBLCL TBL1

Type	Name	Symbol	Abbreviated Instructions	Description
Standard System Functions	DATA TRACE READ	DTRC-RD	DTRC-RD	Data readout from data trace memory to user memory
	TRACE	TRACE	TRACE	Data trace execution control
	FAILURE TRACE READOUT	FTRC-RD	FTRC-RD	Data readout from failure trace memory to user memory
	SEND MESSAGE	MSG-SND	MSG-SND	Sending a message from a Communications Module
	RECEIVE MES-SAGE	MSG-RCV	MSG-RCV	Receiving a message from a Communications Module
	COUNTER	COUNTER	COUNTER	Increments or decrements a counter.
	FIRST-IN FIRST-OUT	FINFOUT	FINFOUT	First-in, first-out
	INVERTER TRACE READ	ITRC-RD	ITRC-RD	Reads inverter trace data to store it in user register.
	INVERTER CONSTANT WRITE	ICNS-WR	ICNS-WR	Writes inverter constant.
	INVERTER CONSTANT READ	ICNS-RD	ICNS-RD	Reads inverter constant to register.

■ Program Development Support Tool Function Specifications

Table 2.10 lists the program development support tool specifications.

Table 2.10 Program Development Support Tool Specifications

Item		Specifications
Basic Hardware	**Model**	IBM PC/AT or compatible
	CPU	Pentium 133 MHz or better, or equivalent
	Main Storage	64 MB min.
	Display Resolution	640 × 480 min. (800 × 600 min. recommended)
	HDD	200 Mbytes min. of unused capacity is required.
	Pointing Device	PS/2 interface
Basic Software	**Operating System**	Windows 95
Printer		Windows 95-compatible
Functions	**File Manager**	File management
		User management
		File transfer
	System Definitions	System definitions
		Scan time settings
		Application information settings
		Failure monitoring
		Data traces
		Group definitions
		Motion parameters
	Unit Configuration Definitions	Configuration definitions
		General-purpose serial definitions
		Local I/O definitions
		MECHATROLINK definitions

2

Item		Specifications
Functions	**Tools**	Register lists
		Cross references
		Disable lists
		Register number searches and replacements
		Comment lists
		Source conversions
	Document	Printing
	Sequence Programming	Property settings
		Main program creation
		Table format program creation
		Adjustment panel creation
	Adjustment Panel Creation	C register creation
	Table Data Definitions	Table data creation
	Motion Programming	Motion program editor
		Position monitor
		Task monitor
		Motion alarm

■ Program Development Support Tool Function List

Table 2.11 lists the program development support tool functions.

Table 2.11 Program Development Support Tool Functions

Lop-level Function	Intermediate Functions	Lower-level Functions
File Management	File management	Name change
		New PLC registration
		PLC information change
		Folder/file delete
		CPU logon/logoff
		Online/offline
		CPU status switching
		Drawing/function program editing
		Latest information update
	User management	User management
		Default user settings
	File transfer	Batch transfer
		Individual transfer
		Consecutive transfer
		Flash memory transfer
		Multiple CPU transfer
System Information Definitions	System definitions	System definitions
		Definition data save
	Scan time settings	PLC selection
		Scan time setting
		Setting data save
	Application information setting	Information setting
		Information save

2

Lop−level Function	Intermediate Functions	Lower−level Functions
System Information Definitions	Failure monitoring	Failure definitions
		Annunciator signal definitions
		Failure status display
		Failure occurrence display
		Failure trace display
		Definition data save
		Definition data delete
	Data traces	Trace data definitions
		Definition data save
		Definition data delete
		Trace data write
		Trace data read
		Data trace start
		Trace data list display
		Trend graph display
	Group definitions	Group definitions
		Save
		Delete
	Motion parameters	Parameter setting
		Save
		Delete
		Monitor
Unit Configuration Definitions	Configuration definitions	Configuration information setting
		Save
		Delete
	General-purpose serial port definitions	Serial port setting
		Save
		Delete

Lop-level Function	Intermediate Functions	Lower-level Functions
Unit Configuration Definitions	Local I/O definitions	Local I/O setting
		Save
		Delete
	MECHATROLINK definitions	MECHATROLINK setting
		Save
		Delete
Tools	Register list	Register value monitor
		Display format change
		Register value change
	Cross reference	Cross reference execution
	Disable list	Disable coil retrieval
	Register number retrieval and replacement	Data input
		Retrieval and replacement execution
	Comment list	Comment creation
		Comment information update
	Source conversion	Source conversion
		Conversion rules
Document	Printing	Print data setting
		Print execution
		Print status display
		Print setting file editing
		Print results sampling
Sequence Programming	Properties setting	Configuration information definitions
		I/O definitions
		Symbolic definitions
		# register list
		Change history
		Properties setting data save

Lop–level Function	Intermediate Functions	Lower–level Functions
Sequence Programming	Main program creation	Display mode switching
		Command input
		Program save
		Program printing
	Table format program creation	Input mode switching
		Display data input
		Table editing
		Table format program save
	Adjustment panel creation	Input mode switching
		Definition data input
		Adjustment panel editing
		Adjustment panel save
Adjustment Panel Creation	C register creation	C register table list display
		C register table creation
Table Data Definitions	Table data creation	Table data list display
		Column attribute setting
		Table data setting
Motion Programming	Motion editor	Motion properties
		Import/Export
		Text editing
		Debugging
		Position teaching
	Position monitor	Current position coordinate display
	Task monitor	Task status display
	Motion alarm	Error status display

2.2 Basic System Configuration

This section gives an overview of the system configuration used by the MP930, together with the various devices in this configuration.

2.2.1 List of Basic Units

Table 2.12 lists of the Units and other devices required in a system using the MP930.

Table 2.12 List of Basic Units and Equipment

No.	Name	Description	Outline
1	MC Unit	MC350	Performs sequence and motion control.
2	I/O Unit	IO350	Expansion I/O Unit IN: 64 points, OUT: 64 points
3	MECHATROLINK Cable 1	W6000	Used to expand the MC Unit. Connected to the I/O Unit.
4	MECHATROLINK Cable 2	W6010	Connects the Expansion I/O Unit (or MC Unit) and servo amp.
5	MC Unit I/O Cable	W5410	Connects the MC Unit and an external device.
6	Expansion I/O Unit I/O Cable	W5410	Connects the Expansion I/O Unit and an external device.
7	Servopack	SGD-□□□N SGDB-□□AN	Select a MECHATROLINK-compatible servo.
8	Servomotor	–	Select from a number of Series.
9	Motor Cable	DP320081	Power cable for motor
10	Encoder Cable	DB320089	Connects the motor encoder and the servo amp.
11	Software Package	–	Motion program and ladder logic program creation, etc.
12	Communications Cable	W5311	Connects the MC Unit and Programming Device.
13	DC power supply	–	24 VDC power supply for the MC Unit, Expansion I/O Unit, and external I/O signals

2.2.2 Basic System Configuration

The following illustration shows the basic system configuration of the MP930.

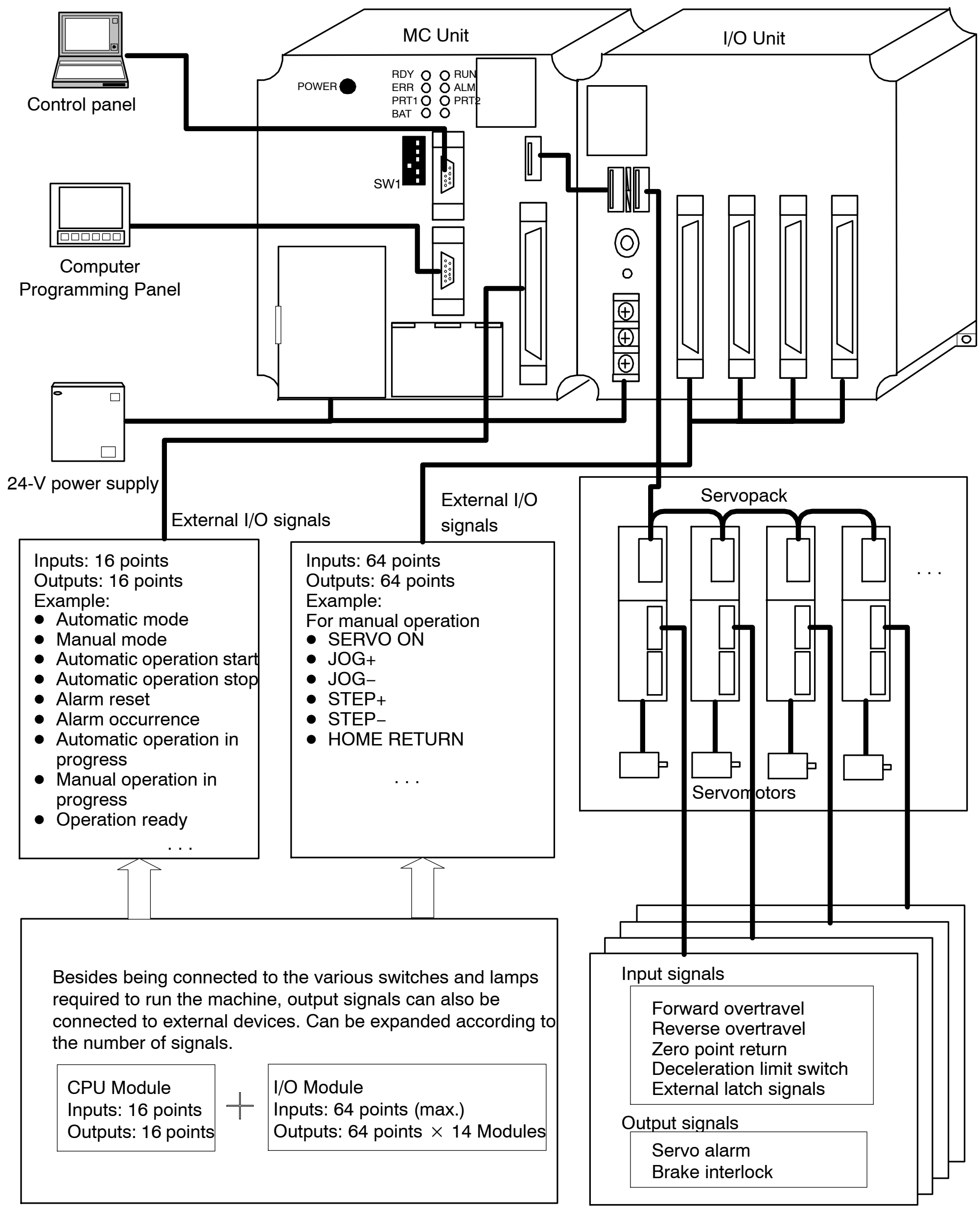


Figure 2.1 Basic System Configuration Diagram

2.2.3 Precautions on System Configuration

The following precautions are on designing the system using the MP930.

- Use the connection cables specified by Yaskawa.
  Various types of standard cable are provided by Yaskawa. When selecting cables, carefully check the equipment for which the cables are to be used to avoid making any mistake.
- The customer must prepare the connection cables between the Expansion I/O Unit and the servos.
- The Servopack that can be connected to the MP930 are the SGD-□□□N and the SGDB-□□AN.
- The customer must provide the 24 VDC power supply.
- With the MP930, the overtravel signals, zero point return deceleration limit switch signals, and external latch signals are connected to the servo amps.

2.2.4 Programming Device

Using the CP-717 Windows 95-compatible programming software increases the ease of programming, design efficiency, and the testing efficiency.

The CP-717 is provided with special functions, as support tools for the design, testing, and maintenance of the sequence and motion programs.

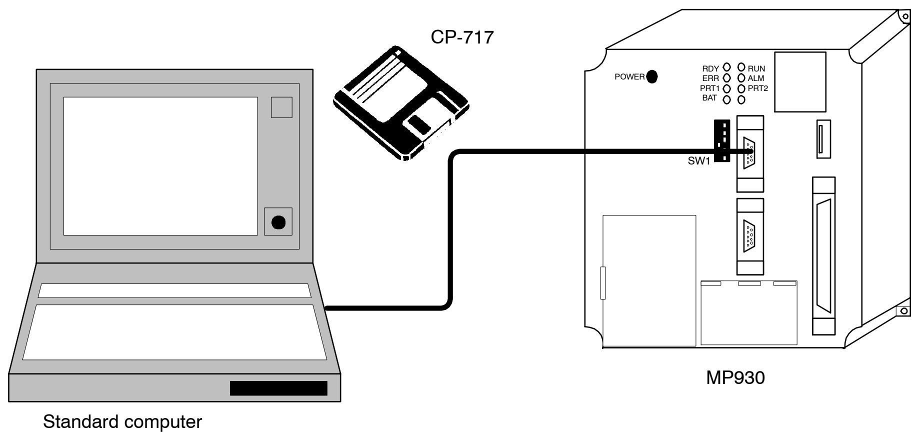


Figure 2.2 Programming Device

Basic Functions

The CP-717 has the five main functional areas, as shown in the following table.

Name	Functions
File Manager	Control functions for file management and online/offline control, logoff, CPU control, etc.
Definitions	Functions for defining system definitions, such as the Unit configuration
Programming	Programming functions for sequence programs (ladder logic programs) and motion programs
Debugging and Monitoring	Functions for program maintenance management: Register lists, adjustment panels, program monitoring, position monitoring, task monitoring, etc.
Printing	Printing functions for definitions, programming, debugging, monitoring, etc.

3

Basic System Operation

3

This chapter explains the basic operation of the MP930 system.

3.1 Operating Modes

This section explains the online operating mode and the offline stop mode, both of which indicate the MP930 operating status.

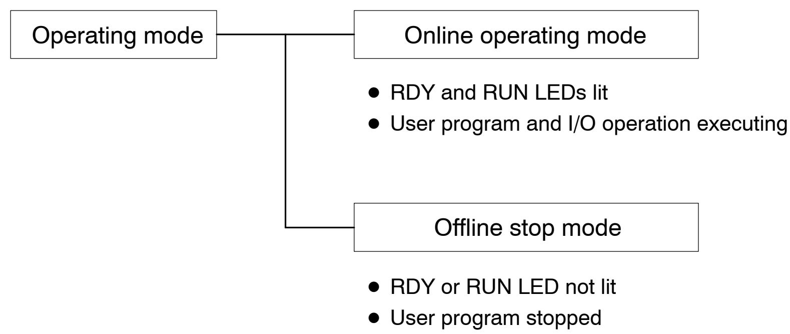


Figure 3.1 MP930 Operating modes

3.1.1 Online Operating Mode

When the power for the MP930 is turned ON, the RDY and RUN indicators will light (the ERR and ALM indicators will not light) and the Unit will enter the online operating mode. This means that the user program and I/O operations are being executed in the MP930 without any errors or failures. If an alarm does occur, such as for an I/O conversion error or a user calculation error, the execution of the user program will not stop, and the online operating mode will be maintained. The ALM indicator lights to indicate the occurrence of an error. For details on the error content and the action to be taken, see *Chapter 9 Troubleshooting*.

3.1.2 Offline Stop Mode

The execution of the user program is stopped, and all outputs are reset (i.e., “0” is output for all digital outputs). The RUN or RDY indicator will go OFF to indicate the status. Drawings (DWG.H and DWG.L) are not executed in this status.

The Controller will be in the offline stop mode in the following four cases:

- When a scan time has not been set (see note 1)
- When the program memory has not been initialized (see note 1)
- When a serious failure, such as watchdog timer error, has occurred (see note 2)
- When a STOP operation has been performed from the CP-717 (see note 2)
- When the RUN/STOP switch has been set to OFF (STOP) and the power has been turned ON (see note 3)

Note The above cases apply when a user program error occurs, or when there is a hardware fault in the MP930. For details on the error content and the action to be taken, see *Chapter 9 Troubleshooting.*

3.2 Start and Stop Sequences

This section explains the start and stop sequences of the MP930. The methods of setting the DIP switch, the types of self-diagnosis, and the indicator patterns are also explained.

3.2.1 DIP Switch Settings

The DIP switch on the CPU Module are used to control start and stop sequences. As shown in *Figure 3.2*, there are six pins on the DIP switch on the CPU Module. *Table 3.1* shows the function of each pin.

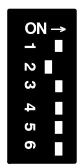


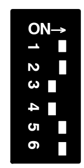
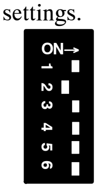
Figure 3.2

Table 3.1 DIP Switch Pin Functions

No.	Name	Setting	Function	Default Setting
1	FLASH	ON	Used by the system	Normally set to OFF.
		OFF	–	
2	RUN	ON	User program operating	Normally set to ON.
		OFF	User program stopped	
3	INIT	ON	Number 4 ON: Memory clear OFF: Programming Panel port default	Normally set to OFF.
		OFF	Online	
4	TEST	ON	Terminal (communications) mode	Normally set to OFF.
		OFF	Online	
5	Not used	ON	–	Normally set to OFF.
		OFF	–	
6	S.TST	ON	Used by the system	Normally set to OFF.
		OFF	Online	

Memory Initialization

When the DIP switch is set according to the following procedure and the power is turned ON or OFF, memory will be initialized, and the user programs and definition data will be deleted.

1	2	3	4	5
Turn OFF the MP930 power.	Turn ON DIP switch pins 3 and 4. 	Turn ON the power, and check that the RDY and RUN indicators flash (about 3 seconds).	Turn OFF the power, and return the DIP switch pins to their original settings. 	Turn ON the power again.

Always execute memory initialization if you remove the battery when the Module power supply is turned OFF.

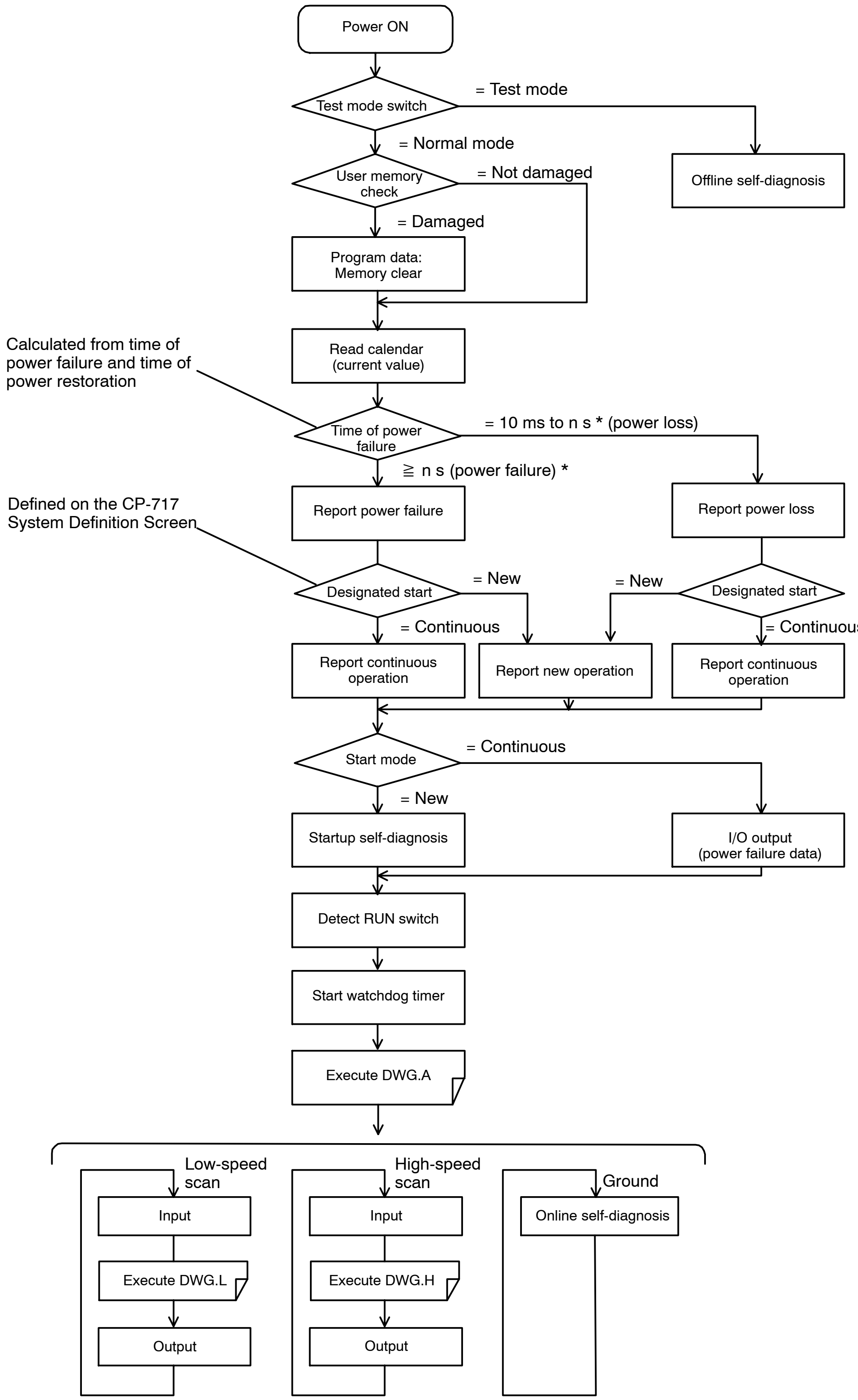
3.2.2 Start Sequence

The MP930 makes a number of determinations at startup. If an error is detected, the ERR indicator will flash and the error content will be indicated by the number of times the indicator flashes. When the indicator is flashing, the CP-717 cannot be operated. For details on the error content and the action to be taken, see *Chapter 9 Troubleshooting*. *Table 3.2* shows the MP930 indicators.

Table 3.2 Indicators and Indicator Patterns

<table>
<tr><th rowspan="2">Type</th><th colspan="5">Indicator</th><th rowspan="2">Meaning</th></tr>
<tr><th>RDY</th><th>RUN</th><th>ALM</th><th>ERR</th><th>BAT ALM</th></tr>
<tr><td rowspan="2">Normal</td><td>Lit</td><td>Unlit</td><td>Unlit</td><td>Unlit</td><td>Unlit</td><td>The user program is stopped.</td></tr>
<tr><td>Lit</td><td>Lit</td><td>Unlit</td><td>Unlit</td><td>Unlit</td><td>The user program is executing normally.</td></tr>
<tr><td rowspan="4">Error</td><td>Lit</td><td>Lit</td><td>Lit</td><td>Lit</td><td>Unlit</td><td>Hardware reset status (while display is continuing)</td></tr>
<tr><td>Unlit</td><td>Unlit</td><td>Unlit</td><td>Unlit</td><td>Unlit</td><td>During initial execution (while display is continuing)</td></tr>
<tr><td>Unlit</td><td>–</td><td>Unlit</td><td>Lit</td><td>Unlit</td><td>A serious error has occurred.</td></tr>
<tr><td>Unlit</td><td>Unlit</td><td>Unlit</td><td>Flashing</td><td>Unlit</td><td>1. Flashing once or twice: RAM error
2. Flashing two or three times: ROM error
3. Flashing three or four times: Peripheral LSI error</td></tr>
<tr><td rowspan="3">Alarm</td><td>–</td><td>–</td><td>–</td><td>–</td><td>Lit</td><td>Battery alarm</td></tr>
<tr><td>Lit</td><td>Lit</td><td>Lit</td><td>Unlit</td><td>Unlit</td><td>1. Calculation error
2. I/O error</td></tr>
<tr><td colspan="5">Reported to the system (S) register (no indicator display)</td><td>Hardware status (power loss, RUN/STOP, test mode, etc.)</td></tr>
<tr><td rowspan="4">Other</td><td>Flashing</td><td>Flashing</td><td>Unlit</td><td>Unlit</td><td>Unlit</td><td rowspan="2">Memory initialization has been completed for the DIP switch settings.</td></tr>
<tr><td colspan="5">RDY and RUN flash repeatedly at the same time.</td></tr>
<tr><td>Unlit</td><td>Unlit</td><td>Lit</td><td>Unlit</td><td>Unlit</td><td rowspan="2">Offline test mode</td></tr>
<tr><td colspan="5"></td></tr>
</table>

■ MP930 Start Sequence and Basic Operation



* The time for power loss is defined on the CP-717 System Definition Screen.

The MP930 start sequence and basic operations are as follows:

1. Startup Self-diagnosis

   The following operations are provided for startup self-diagnosis:

   - Memory (RAM) read/write diagnosis
   - System program (ROM) diagnosis
   - Main processor (CPU) function diagnosis
   - Numeric co-processor (FCPU) function diagnosis

   If an error occurs in the diagnostic result, the RDY indicator will flash the specified number of times.

2. Online Self-diagnosis

   The following operations are provided for online self-diagnosis:

   - System program (ROM) diagnosis
   - Main processor (CPU) function diagnosis
   - Numeric co-processor (FCPU) function diagnosis

   If an error occurs in the diagnostic result, the RDY indicator will flash the specified number of times.

3. New Operation Start

   Set the operation method for “New Operation” on the CP-717 System Definition Screen. When the system is next started, the new operation will be used to start. Unlike a continuous operation start, self-diagnostic processing is performed before DWG.A is executed.

4. Continuous Operation Start

   a) Set the operation method for “Continuous Operation” on the CP-717 System Definition Screen. When the system is next started, the continuous operation will be used to start. Unlike a new operation start, no self-diagnostic processing is performed. Therefore, the startup time for drawing execution is shortened.

   b) If the RUN switch is ON (RUN) or if it turns ON (RUN) from OFF (STOP), the CPU starts the watchdog timer and then executes DWG.A.

   c) Once the execution of DWG.A has been completed, the scan is started. The initial scan is executed only after the time for the high-speed or low-speed scan has elapsed following the completion of DWG.A. System inputs and outputs are executed from the first scan.

5. Operation Stop

   The MP930 stops operating in the following cases:

   - When the power supply is interrupted (see *1)
   - When a power failure has occurred (see *1)
   - When a fatal error has occurred (see *2)
   - When a STOP operation has been performed from the CP-717 (see *3)

*** 1.** The MP930 can be restarted only by turning ON the power again.

*** 2.** Restart the system by turning the power ON and OFF. The cause of the error can be ascertained by the indicator status.

*** 3.** Restart the system by performing a RUN operation from the CP-717.

3.3 Power Failures

This section explains the processing when an MP930 power failure occurs.

3.3.1 Power Failure Detection

Table 3.3 shows the start methods used when an MP930 power failure occurs.

The MP930 can select two types of startup: Continuous operation and new operation. The selection of continuous operation or new operation is made on the CP-717 System Definition Screen.

For details on the CP-717 operation method, refer to the *MP9□□ Machine Controller Programming Software User's Manuals* (SIEZ-C887-2.2-1, SIEZ-C887-2.2-2).

Table 3.3 MP930 Start Modes

Length of Power Failure	Continuous Operation/ New Operation	Start Method
0 to 20 ms	–	Operations continue.
20 ms to Ns* (Power loss)	When continuous operation is selected	After the CPU is reset, no self-diagnosis is performed, and operations continue.
	When new operation is selected	After the CPU is reset, self-diagnosis is performed, and the new operation starts.
Ns* or more	When continuous operation is selected	After the CPU is reset, no self-diagnosis is performed, and operations continue.
	When new operation is selected	After the CPU is reset, self-diagnosis is performed, and the new operation starts.

* Ns (power loss decision time) is defined on the CP-717 System Definition Screen.

3.4 User Programs

This section explains the basic operation of the MP930, such as the types of user program, the priority levels, and the execution processing methods.

3.4.1 Drawings (DWGs)

User programs are managed in units of programming called drawings, Each drawing is identified by a drawing number (DWG No.). These drawings serve as the basis of user programs.

The drawings include parent drawings, child drawings, grandchild drawings, and operation error drawings. Besides the drawings, there are functions that can be freely called from each drawing, and motion programs that can be called only from H drawings.

3

- Parent Drawings

  Parent drawings are executed automatically by the system program when the execution condition is established. See *Table 3.4* for execution conditions.

- Child Drawings

  Child drawings are executed by being called from a parent drawing using the SEE command.

- Grandchild Drawings

  Grandchild drawings are executed by being called from a child drawing using the SEE instruction.

- Operation Error Drawings

  Operation error drawings are executed automatically by the system program when an operation error occurs.

- Functions

  Functions are executed by being called from a parent, child, or grandchild drawing using the FSTART instruction.

- Motion Programs

  Motion programs can be called only from H drawings. They can be executed by being called from a parent, child, or grandchild drawing using the MSEE instruction.

Types and Priority Levels of Parent Drawings

Parent drawings are classified by the first character of the drawing number (A, H, L) according to the purpose of the process. The priority levels and execution conditions are as shown in *Table 3.4*.

Table 3.4 Types and Priority Levels of Parent Drawings

Type of Parent Drawing	Role of Drawing	Priority Level	Execution Condition	Number of Drawings
DWG.A	Startup process	1	Started when power is turned ON (executed once only when the power is turned ON)	64
DWG.H	High-speed scan process	2	Started at a fixed interval (executed during each high-speed scan)	100
DWG.L	Low-speed scan process	3	Started at a fixed interval (executed during each low-speed scan)	100

Table 3.5 gives details of the number of drawings for each type of drawing.

Table 3.5 Details of Drawings

Drawing	Number of Drawings		
	DWG.A	DWG.H	DWG.L
Parent Drawing	1 (A)	1 (H)	1 (L)
Operation Error Drawing	1 (A00)	1 (H00)	1 (L00)
Child Drawings	Maximum total of 62 drawings	Maximum total of 98 drawings	Maximum total of 98 drawings
Grandchild Drawings			

3.4.2 Execution Control of Parent Drawings

Each drawing is executed based on its priority level, as shown in *Figure 3.3*.

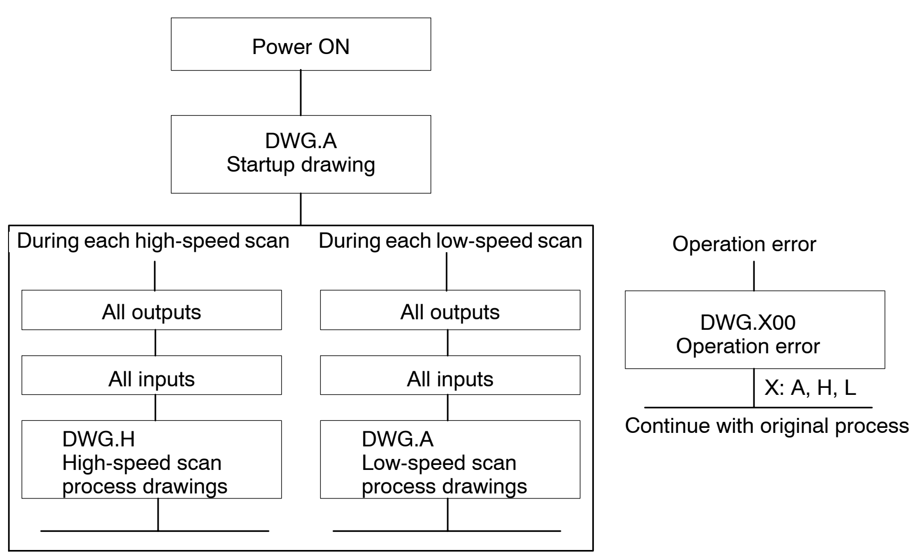


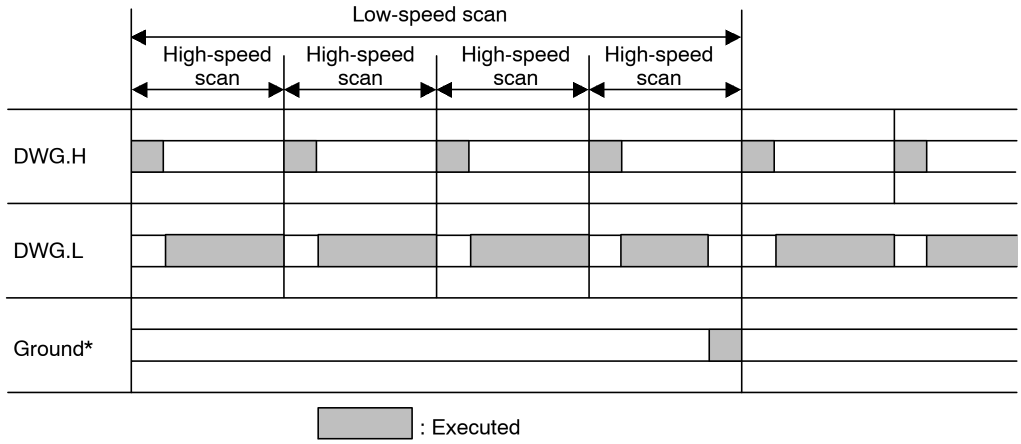
Figure 3.3 Execution Control of Parent Drawings

IMPORTANT

Interrupt drawings cannot be used with the MP930.

■ Execution Scheduling of Scan Process Drawings

The scan process drawings are not executed simultaneously. As shown in *Figure 3.4*, they are scheduled based on the priority level and are executed according to the schedule.



* Used for internal system processes, such as self diagnosis.

Figure 3.4 Execution Scheduling of Scan Process Drawings

IMPORTANT

1. Set the high-speed scan time as a multiple of 2 with a minimum of 2 ms.

2. The low-speed scan process is executed in the unused time of the high-speed scan process. Therefore, as a guideline, set a time that is twice the execution time of all the DWG.H drawings as the high-speed scan time.

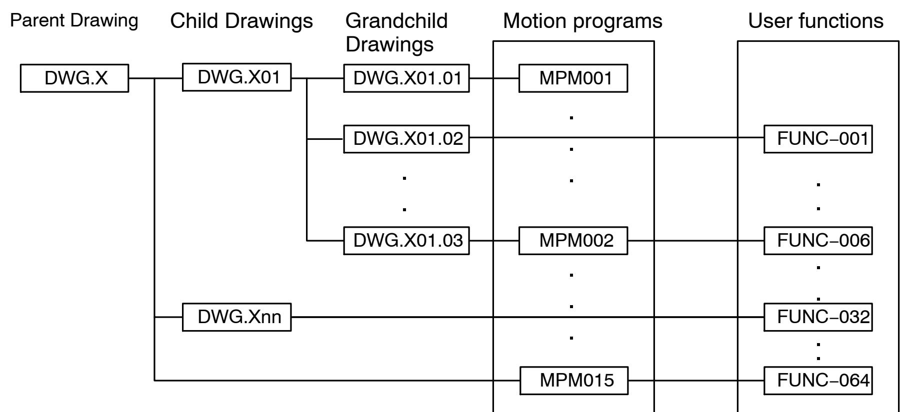
■ Hierarchical Arrangement of Drawings

Drawings are arranged in the following order: Parent drawing, child drawings, grandchild drawings. A parent drawing cannot call a child drawing of a different type, and a child drawing cannot call a grandchild drawing of a different type. A parent drawing also cannot directly call a grandchild drawing. A child drawing is called from a parent drawing, and a grandchild drawing is called from that child drawing. This is called the hierarchical arrangement of drawings.

3

■ Execution of Drawings

The user prepares each processing program with the parent drawing, child drawing, grandchild drawing hierarchy, as shown in *Figure 3.5*.



Note Substitute A, H, or L for X.

Figure 3.5 Hierarchical Arrangement of Drawings

Note 1. A parent drawing is executed automatically by the system, because the execution condition is determined for each one. In other words, a parent drawing is automatically called by the system. See *Table 3.4 Types and Priority Levels of Parent Drawings*. The user can execute any child or grandchild drawing by programming an instruction that calls a drawing (the SEE instruction) in a parent or child drawing.

2. Functions can be called from any drawing. A function can also be called from a function.

3. If an operation error occurs, the operation error drawing corresponding to the drawing will be called.

4. Motion programs must be called from H drawings using the MSEE instruction. The MSEE instruction can be used from any H drawing, i.e., from parent, child, or grandchild H drawings.

INFO

1. A parent drawing cannot call a child drawing of a different type, and a child drawing cannot reference a grandchild drawing of a different type.

2. A parent drawing cannot directly call a grandchild drawing. A child drawing must be referenced from a parent drawing, and a grandchild drawing must be called from that child drawing.

3. A parent drawing is automatically called (and executed) by the system. A child drawing is called from a parent drawing using the SEE instruction. Therefore, user programs containing only child and grandchild drawings cannot be executed.

Execution Processing Method of Drawings

Drawings in the hierarchy are executed by the lower-level drawings being called from upper-level drawings. *Figure 3.6* shows the hierarchical arrangement of drawings, using the example of DWG.A.

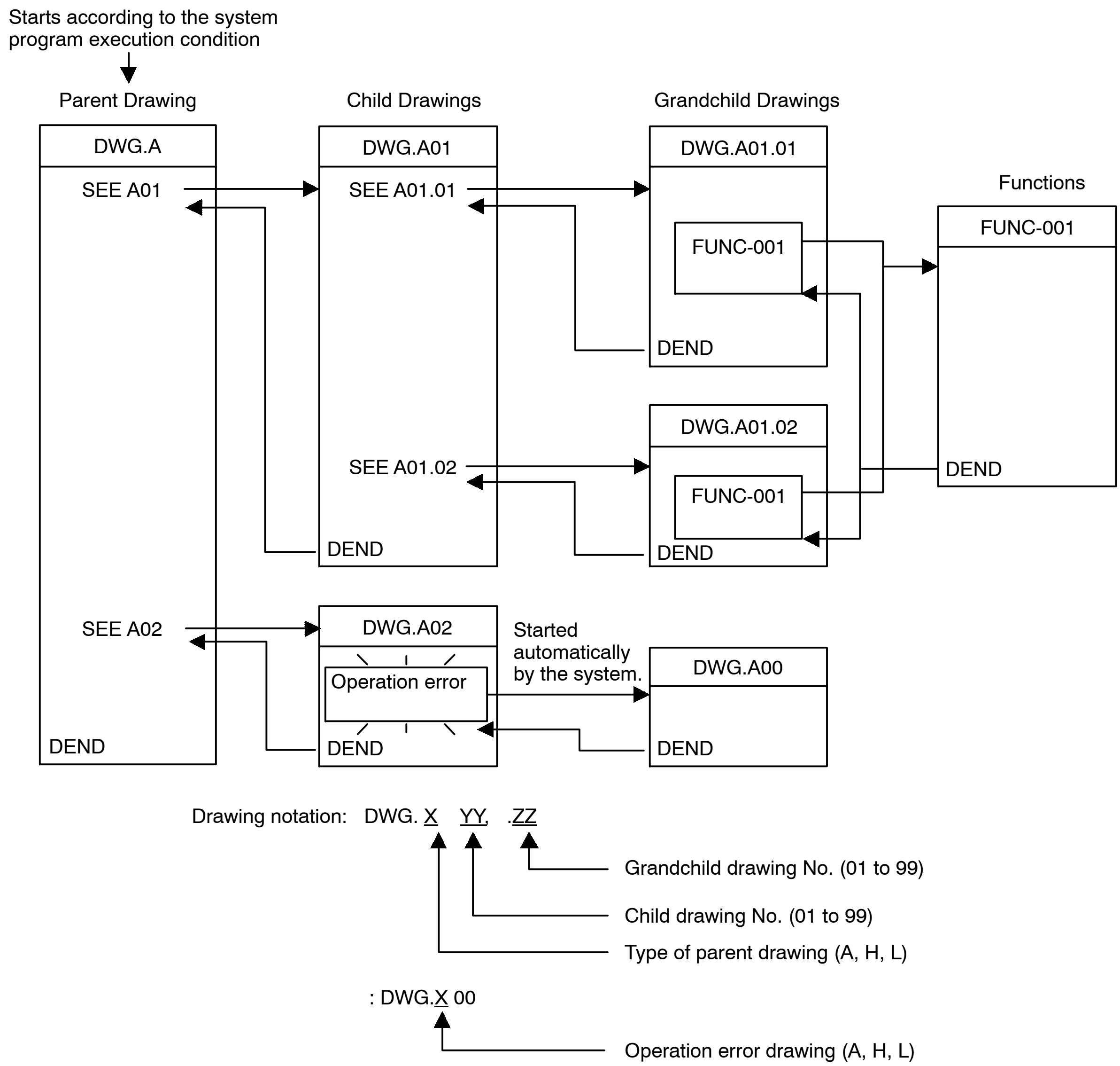


Figure 3.6 Hierarchical Arrangement of Drawings

3.4.3 Motion Programming

■ Overview of Motion Programs

Motion programming is a textual motion programming language. Motion programming can be used to create 256 programs separate from the ladder drawings.

Two types of motion program are provided: Main programs (MPM□□□) that can be called from DWG.H, and subroutines (MPS□□□) that can be called from the main programs.

Table 3.6 Types of Motion Program

Classification	Designation Method	Feature	Number of Programs
Main Programs	MPM□□□ 1 to 256	Can be called from DWG.H drawings.	A total of up to 256 main programs and subroutines can be created.
Subroutines	MPS□□□ 1 to 256	Can be called from the main programs.	

There are two methods of designating a motion program: Direct designation of the program number, and indirect designation of the number of the register in which the program number is stored.

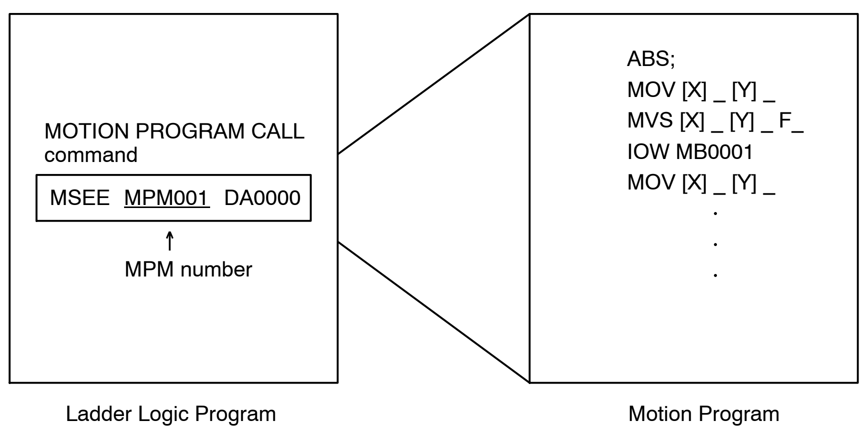


Figure 3.7 Starting a Motion Program by Direct Designation

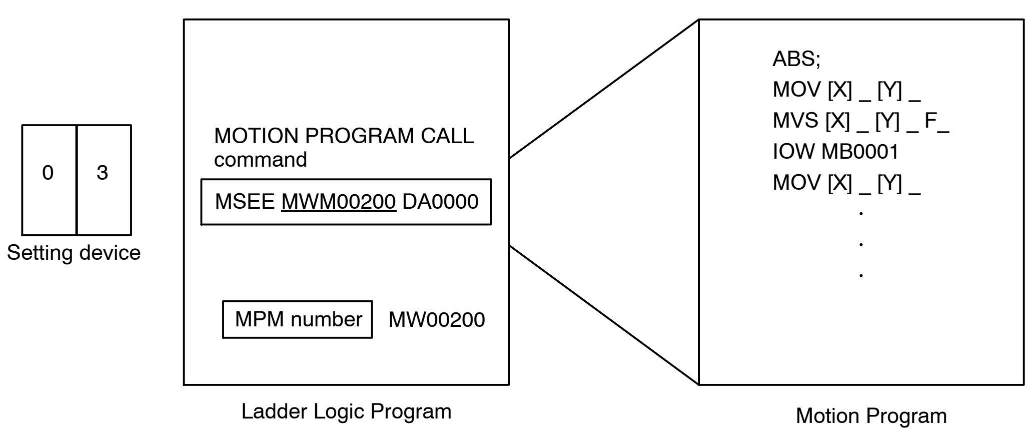


Figure 3.8 Starting an Motion Program by Indirect Designation

■ Groups

With the MP930, the axes can be grouped by operation so that multiple machines can be independently controlled by one Machine Controller. This enables programming to be done for each axis group. The axes to be included in a group are defined in the group definitions. For details, refer to the *MP9□□ Machine Controller Programming Software User's Manuals* (SIEZ-C887-2.2-1, SIEZ-C887-2.2-2).

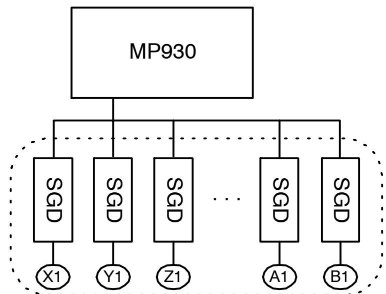


Figure 3.9 Operation as One Group

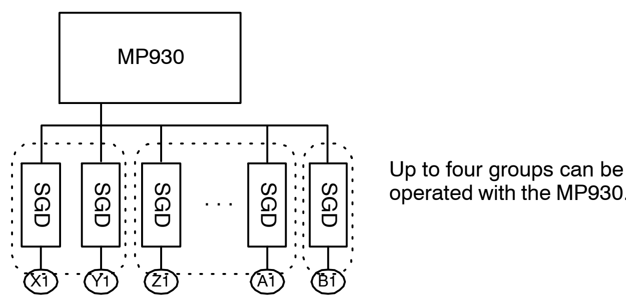


Figure 3.10 Operation with Multiple Groups

■ Motion Program Execution Processing Method

A motion program must be called from DWG.H using the MSEE instruction. Motion programs can be called from any H drawing, i.e., from parent, child, and grandchild H drawings.

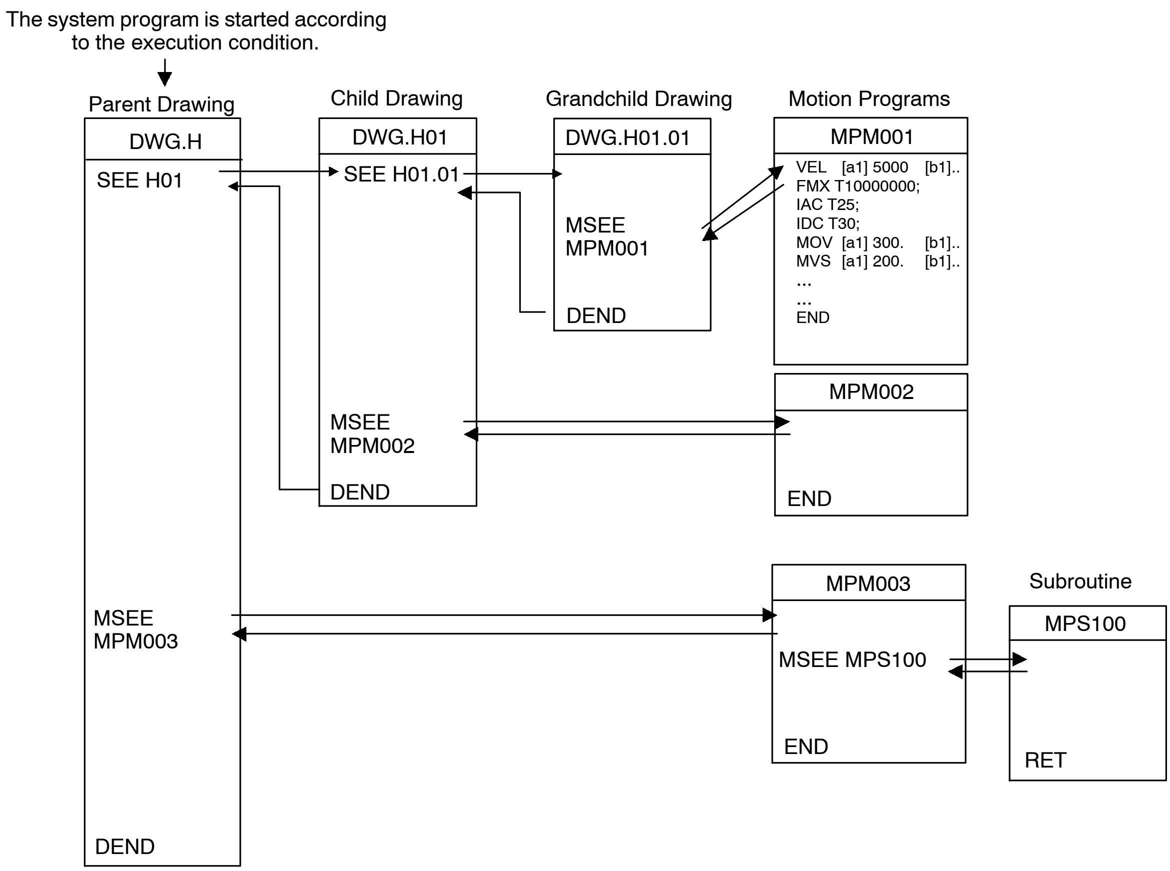


Figure 3.11 Execution Processing Method for Motion Programs

- In each high-speed scanning cycle, the ladder logic instructions for H drawings are executed in the following hierarchical order: Parent drawing - child drawing - grandchild drawing.
- Motion programs are called in the scanning cycle, but as with ladder logic programs, all programs cannot be executed in one scan. Motion programs are executed and controlled by special system motion management functions.
- Motion programs are called in the scanning cycle, but they are not executed only in this cycle. The control signals set on the Group Definition Screen must be input. (See the table on the next page.)

INFO

The following restrictions apply to calling motion programs. Call motion program with care.

1. More than one motion program with the same number cannot be called using the MSEE instruction.

2. Subroutines (MPSxxx) cannot be called from the ladder logic program MSEE instruction. They can be called only from within motion programs (MPMxxx and MPSxxx).

3. The same subroutine cannot be called from two different locations at the same time.

Executing Motion Programs

To execute a motion program called from a DWG.H drawing by the MSEE instruction, program control signals (such as program operation start requests and program stop requests) must be input. Operations are enabled by inputting the external control signals defined on the Group Definition Screen as the program control signals.

The signals used to control motion programs are shown in the following table.

Bit	Signal Name	Signal Type
b0:	Program operation start request	Differential input
b1:	Program pause request	N.O. contact (”A” contact)
b2:	Program stop request	N.O. contact
b3:	Program debugging mode selection	N.O. contact
b4:	Program debugging start request	Differential input
b5:	Alarm reset request	N.O. contact
b6:	Skip 1 information	N.O. contact
b7:	Skip 2 information	N.O. contact

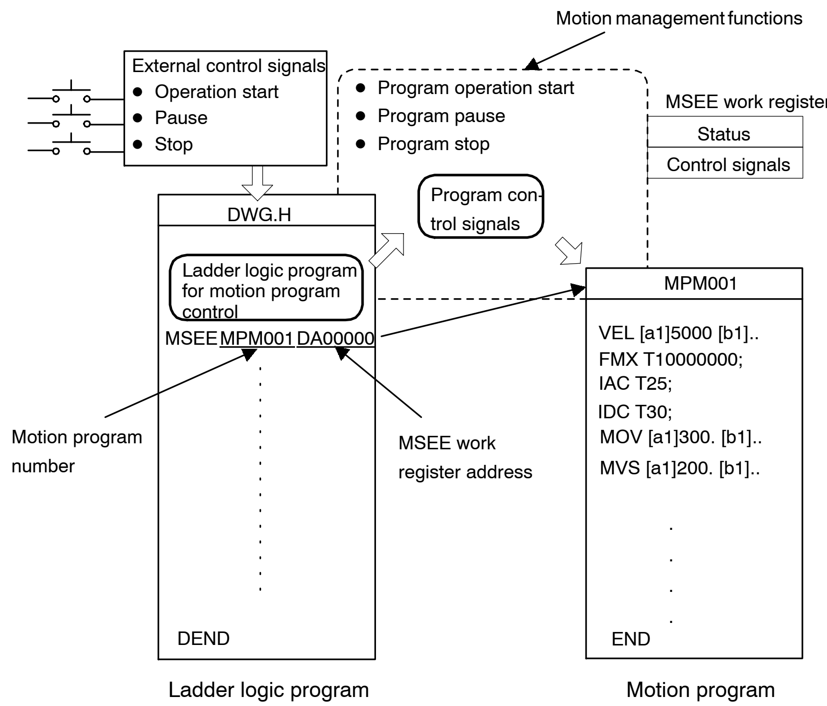
The motion program operation, stop, pause, and so on, can be controlled by using a ladder logic program to input these signals into the one register higher than the work register specified with the MSEE instruction.

For the ladder logic program inputs, make sure the signals are in accordance with the signal type.

IMPORTANT

When the start signal is input using an N.O. contact, the program is completed and then restarted. The program will not be executed if the start signal has been turned ON when the power is turned ON.

The following illustration shows the method of executing a motion program.



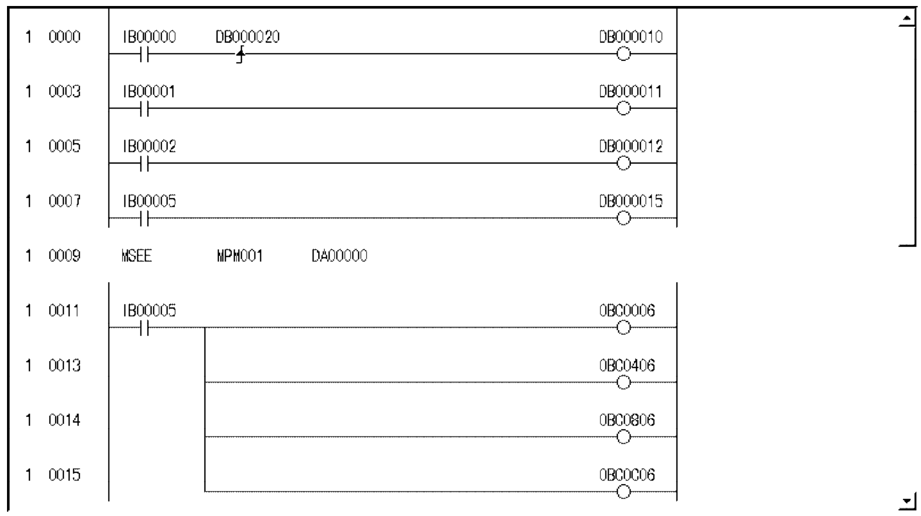
■ Status Flag of Motion Program

The 1st word of MSEE work resister is a motion program status flag that indicates the execution status of the motion program. The detailed contents of status flag are shown in the table below.

Bit	Status
b0	Program running
b1	Program pause
b2	Program pause by a program stop request
b3	(For system use)
b4	Program under debugging
b8	Program alarm occurring
bB	In debugging mode (EWS debugging)
bE	Duplicated main program error
bF	Main program over-numbered error

■ Example of a Ladder Logic Program for Motion Program Control

The minimum ladder logic program required to control a motion program is shown in the following illustration.



The contents of this ladder logic program are shown in the following table.

Step Number	Program Content
1 to 7	The signals connected to the MP930 external input signals are stored as the motion program control signals. IW0000 (external input signals) → DW00001 (second word of MSEE work registers) • Program operation start • Program pause • Program stop • Alarm reset
8	Calls motion program MP001 MSEE <u>MPM001</u> <u>DA00000</u> 1 2 1. Motion program number 2. MSEE work register address
11 to 15	Resets the alarm (bit 6 of OWxx00) using the alarm reset signal (IB00005), and clears the alarm for each axis.

When the external input signals (IB00000 to IB00007) connected to the MP930 are input to DW00001 (second word of MSEE work registers) as motion program control signals using the ladder logic program shown above, motion program operations such as run, stop and pause can be performed by the system motion management functions.

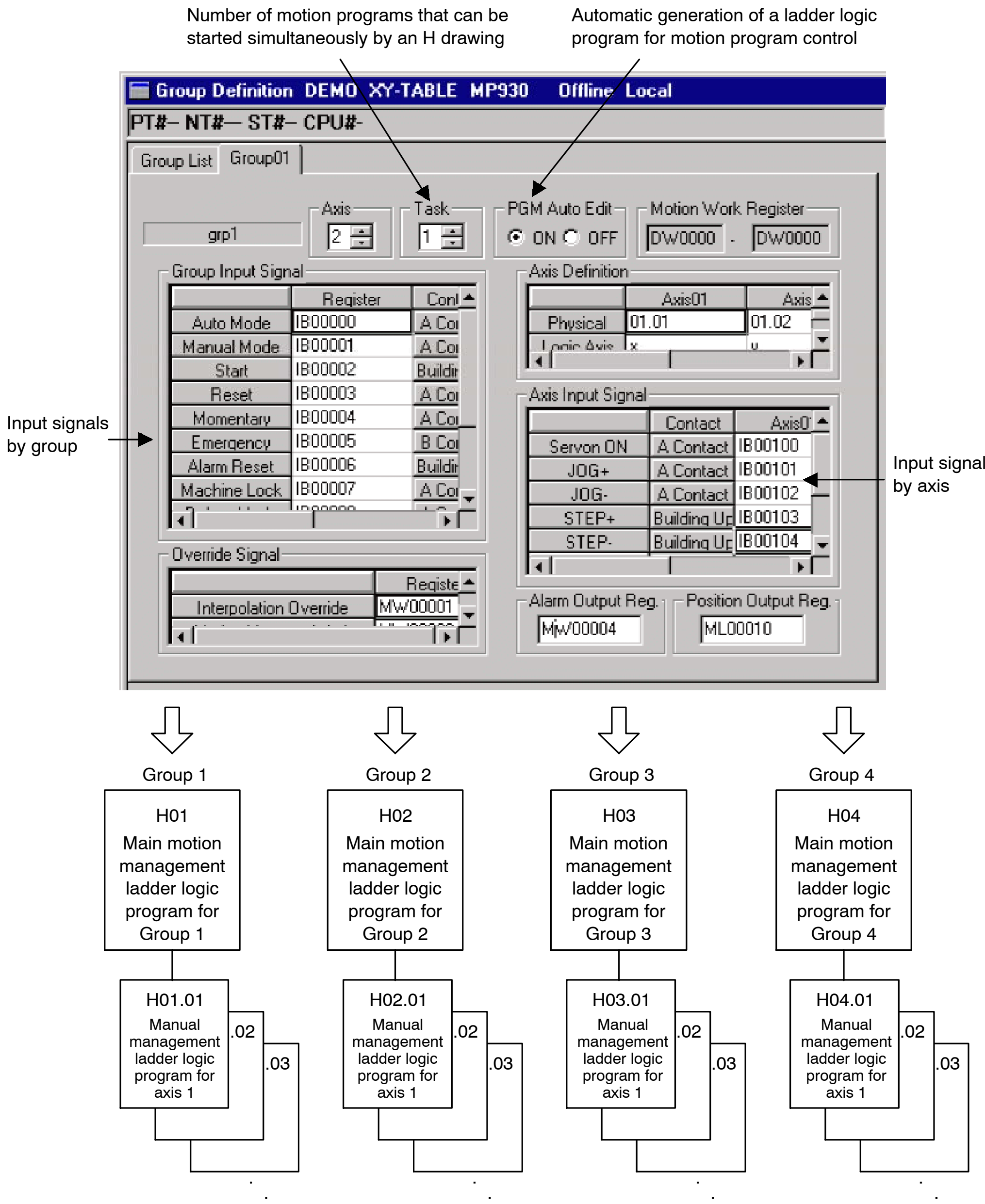
◀EXAMPLE▶ *Table 3.7* shows an example of the default external input signals that are allocated by the MP930 on the Group Definition Screen.

Table 3.7 External Input Signals and Motion Program Control Signals

External Signal Address	External Signal Name		BIT	Motion Program Control Signal
IB00000:	Program operation start	⇨	**B0:**	Program operation start request
IB00001:	Program pause		**B1:**	Program pause request
IB00002:	Program stop		**B2:**	Program stop request
IB00003:	Program debugging mode		**B3:**	Program debugging mode selection
IB00004:	Program debugging start		**B4:**	Program debugging start request
IB00005:	Alarm reset		**B5:**	Alarm reset request

■ Automatic Generation of Ladder Logic Programs for Motion Program Control

An automatic generation function for the ladder logic programs used to control motion programs is provided with the MP930. This function enables JOG operations and program operations to be performed without the need for special ladder logic programs to be created, and greatly reduces the system startup time.



INFO

1. The ladder logic programs that are generated for motion program control are created automatically using the external input signals that are allocated on the Group Definition Screen. These ladder logic programs can also be used as is. It is recommended, however, that they be used as templates to be optimized (changed) to suit individual system requirement.
2. When a ladder logic programs used for motion program control is created by automatic generation, up to four motion programs can be called simultaneously from the ladder logic program. In other words, when automatic generation is used, a maximum of four groups will be controlled.

3.5 Functions

This section explains the methods of using and the advantages of the MP930 functions.

Functions are executed by being called from a parent, child, or grandchild drawing using the FSTART instruction.

Unlike child and grandchild drawings, functions can be called from any drawing. The same function can also be called simultaneously from drawings of different types and different hierarchies. Moreover, a function can also be called from another function that was previously created.

The following advantages can be obtained by using functions:

- Programs can be easily divided into parts.
- Programs can be easily prepared and maintained.

Functions are divided into standard system functions, which are provided by the system, and user functions, which are defined by the user.

3.5.1 Standard System Functions

Seven functions, including the transfer function, are provided by the system as standard functions. See *Table 3.8*. The user cannot change the system functions.

Table 3.8 List of Standard System Functions

Type	Name	Symbol	Description
System Functions	Counter	COUNTER	Up/down counter
	First-in first-out	FINFOUT	First-in or first-out stack
	Trace function	TRACE	Data trace execution control
	Data trace read	DTRC-RD	Data readout from data trace memory to user memory
	Failure trace readout	FTRC-RD	Data readout from failure trace memory to user memory
	Inverter trace read	ITRC-RD	Reads inverter trace data to store it in user registers
	Inverter constant write	ICNS-WR	Writes inverter constants
	Inverter constant read	ICNS-RD	Reads inverter constants to registers
	Send message function	MSG-SND	Sending a message from a Communications Module
	Receive message function	MSG-RCV	Receiving a message from a Communications Module

3.5.2 Creating User Functions

The body of the function (program) and the function definitions can be set by the user. The maximum number of user functions is 500 per drawing.

For details on the CP-717 operation methods, refer to the *MP9□□ Machine Controller Programming Software User's Manuals* (SIEZ-C887-2.2-1, SIEZ-C887-2.2-2). For details on instructions, such as the FSTART instruction, refer to the *MP9□□ Machine Controller Ladder Programming User's Manual* (SIEZ-C887-1.2). The methods for creating user functions is explained according to the following procedure.

1. Determining I/O Specifications

   Determine the number of inputs and outputs and the data types.

2. Defining Function I/O

   Input using the CP-717.

3. Programming the Body of the Function

   Prepare in the same way as the drawings, except that different registers are used. Program according to the correspondence between the register numbers used in the body of the function program and the I/O data used when calling the function.

4. Preparing the Program that Calls the Function

   Input using the following procedure:

   a) Use the FSTART instruction to input the function name.

   b) Use the FIN instruction to connect the input data.

   c) Use the FOUT instruction to connect the output data.

■ Determining the I/O Specifications

When a user function is created, the function capabilities, the number of inputs and outputs required to satisfy the function, and other specifications must first be determined. Determine the four types of specification shown in *Table 3.9*.

Table 3.9 Overview of Function Definition Values

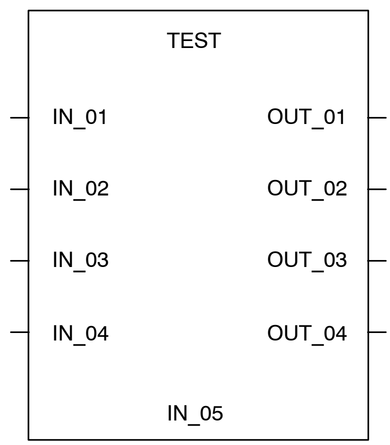
Specification to be Determined	Overview
Function Name	Up to eight characters can be input.
Number of Inputs	The number of arguments input into a function. Up to 16 can be input. Up to 17 arguments are possible if the address input is also counted.
Number of Address Inputs*	The designated number of addresses required by the function. A maximum of one value can be input.
Number of Outputs	The number of outputs from the function. Up to 16 can be input.

* Indicates the number of pointers to be provided for the external function registers used by the function.

■ Defining Function I/O

The function name and other specifications determined in the previous step are defined using the CP-717. For details on operation methods, refer to the *MP9□□ Machine Controller Programming Software User's Manuals* (SIEZ-C887-2.2-1, SIEZ-C887-2.2-2).

◀EXAMPLE▶ The following illustration shows the graphic representation of a function when the following function is defined: Function name = TEST, number of inputs = 4, number of address inputs = 1, and number of outputs = 4.



Note 1. After creating the graphic representation of the function, define the data types of the function inputs, outputs, and address inputs.

2. Three data types can be defined: Bit, integer, and long integer.

3. When the data types are defined, the system automatically allocates inputs to the X registers, outputs to the Y registers, and address inputs to the A registers.

Figure 3.12 Graphic Representation of a Function 1 (Example)

The following illustration shows an example of the I/O definitions of a function.

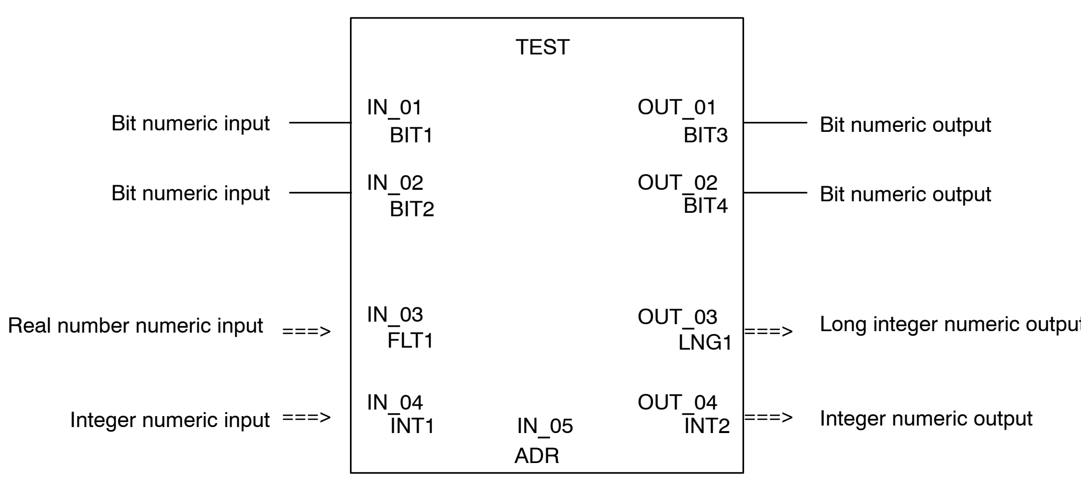


Figure 3.13 Graphic Representation of a Function 2 (Example)

I/O signal addresses are automatically allocated from the highest signal on the graphic representation. For the example given in *Figure 3.13*, the allocation of each I/O register will be as shown in *Table 3.10*.

Table 3.10 Allocation of I/O Registers

Name	Data Type	I/O Register
IN_01 (BIT1)	Bit	XB000000
IN_02 (BIT2)	Bit	XB000001
IN_03 (FLT1)	Real number	XF00001
IN_04 (INT1)	Integer	XW00003
IN_05 (ADR)	Address input	AW00000
OUT_01 (BIT3)	Bit	YB000000
OUT_02 (BIT4)	Bit	YB000001
OUT_03 (LNG1)	Double integer	YL00001
OUT_04 (INT2)	Integer	YW00003

Note XW00000 and YW00000 of the X and Y registers are used for bit data.

The function I/O registers shown in *Figure 3.13* are allocated automatically. The external framework of the function is completed at this stage.

■ Creating the Body of the Function

The body of the function is created in the same way as the drawings except that the types of register used are different. For details on the registers, see *3.6.3 Types of Register.*

■ Creating the Program that Calls the Function

The user function is completed when the graphic representation and body program of the function have been created. As with the standard system functions, user functions can be called from any parent, child, or grandchild drawing or any other user function.

Functions can be called from a drawing or from within the program of another user function by using the following procedure. For details on the operation methods, refer to the *MP9□□ Machine Controller Programming Software User's Manuals* (SIEZ-C887-2.2-1, SIEZ-C887-2.2-2) .

1. Input the function name using the FSTART instruction.

Example: Input "FSTART, *Enter Key*, TEST, *Enter Key*".

The previously defined graphic representation of the function will be displayed.

2. Use the FIN instruction to create the input data program.
Provide input data for the function inputs and address inputs.

3. Use the FOUT instruction to create the output data program.

Example: I/O data is provided for the graphic representation as shown in the following illustration.

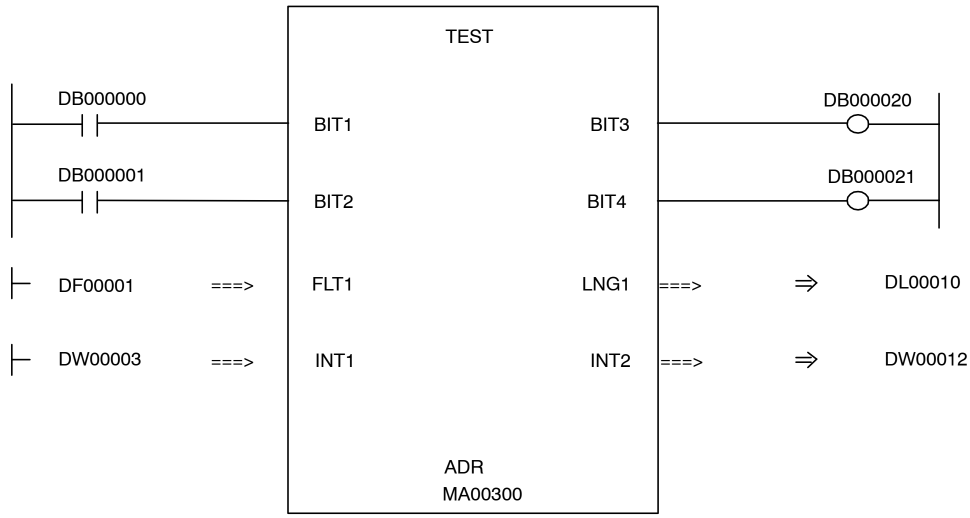


Figure 3.14 Graphic Representation for which Input Data is Provided (Example)

Table 3.11 Relationship Between I/O Data and Internal Function Registers

Name	I/O Data		Internal Function Register
BIT1	DB000000	→	XB000000
BIT2	DB000001	→	XB000001
FLT1	DF00001	→	XF00001
INT1	DW00003	→	XW00003
ADR	MA00300	↔	AW00000
BIT3	OB00020	←	YB000000
BIT4	OB00021	←	YB000001
LNG1	DL00010	←	YL00001
INT2	DW0012	←	YW00003

In the table, address input register AW00000 is allocated to MA00300. That is, registers AW00000, AW00001, and so on, used inside the TEST function correspond to external registers MA00300, MA00301, and so on. Therefore, if a given value in AW00000 is stored inside the function, this value will be stored in MA00300.

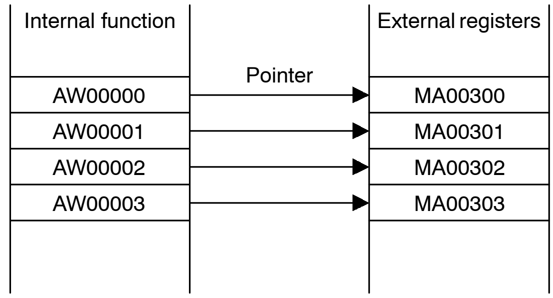


Figure 3.15 Pointer Designation for Address Input Registers

4. Creating a motion program that calls the function.

   User functions can also be called from motion programs.

   Example: The user functions shown in *Table 3.11* would be called from a motion program using the following coding:

```
UFC TEST DB000000 DB000001 DF00001 DW00003,MA00300,
DB000020 DB000021 DL00010 DW00012;
```


INFO

Conditions for referencing a user function from a motion program:

- The first item of output data must be bit data.

This output will be the completion signal for moving to the next motion command.

```
UFC DB000000..., MA00300, DB000020...; ↓ condition BB000020==ON
MVS [X]100. [Y]200. F10000;
          .
          .
          .
```

3.6 Registers

This section explains the types of register used by MP930 user programs and how these registers are used.

3.6.1 Register Designation Methods

Registers can be designated by direct designation of the register number or by symbolic designation. These two types of register designation can be used together in the same user program. When symbolic designation is used, the correspondence between the symbols and the register numbers is defined in the symbol table that is described later.

Table 3.12 shows the register designation methods.

Table 3.12 Register Designation Methods

Designation Type	Description
Direct Register Number Designation	Bit registers: MB00100Ax Integer registers: MW00100x Double integer registers: ML00100x Real # registers: MF00100x Address registers: MA00100x x: For subscripts, add the subscript i or j after the register number.
Symbol Designation	Bit registers: RESET-A.x Integer registers: STIME-H.x Double integer registers: POS-REF.x Real # registers: IN-DEF.x Address registers: PID-DATA.x Address registers are designated using up to 8 alphanumeric characters. x: For subscripts, add a period (.) and then the subscript i or j after the symbol.

■ Direct Register Number Designation

Register number: V T No. [Bit No.] [Subscript]

- [Subscript] → Can designate the subscript i or j.
- [Bit No.] → When T = B (bit) (hexadecimal, 0 to F)
- No. → Register No. for V (decimal or hexadecimal)
- T → Data type of V (T: B | W | L | F | A)
- V → Type of register
  Drawing: (V: S | M | I | O | C | # | D)
  Function: (V: S | M | I | O | C | # | D | X | Y | Z | A)

■ Symbol Designation

Symbol: [Symbol Name] [.] [Subscript]

Can designate the subscript i or j.

Required if a subscript is to be used
(symbol name and subscript delimiter)

(Name given to the register, 8 characters or less)
X XXXXXXX

Alphanumeric characters or symbols

Alphabetic character or symbol
(A numeral cannot be designated at the beginning of a symbol name.)

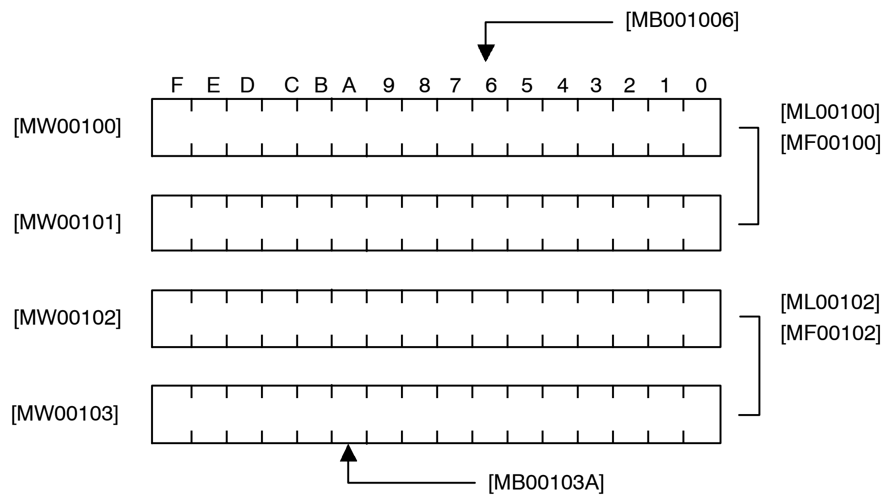
3.6.2 Data Types

There are five data types: Bit, integer, double integer, real number, and address. Use them as required. Address data is used only for pointer designations inside functions. For details, refer to the *MP9□□ Ladder Programming Manual* (SIEZ-C887-1.2). *Table 3.13* shows the data types.

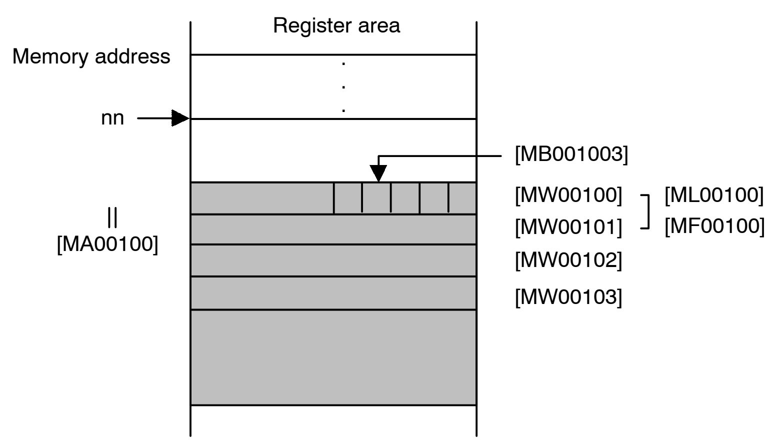
Table 3.13 Data Types and Numeric Range

Type	Data Type	Numeric Range	Remarks
B	Bit	ON, OFF	Used in relay circuits.
W	Integer	–32768 to +32767 (8000H) (7FFFH)	Used in numeric operations. The values in parentheses ( ) are used in logic operations.
L	Double integer	–2147483648 to +2147483647 (80000000H) (7FFFFFFFH)	Used in numeric operations. The values in parentheses ( ) are used in logic operations.
F	Real number	±(1.175E–38 to 3.402E+38), 0	Used in numeric operations.
A	Address	0 to 32767	Used only for pointer designations.

- Register Designations and Data Types



- Pointer Designations

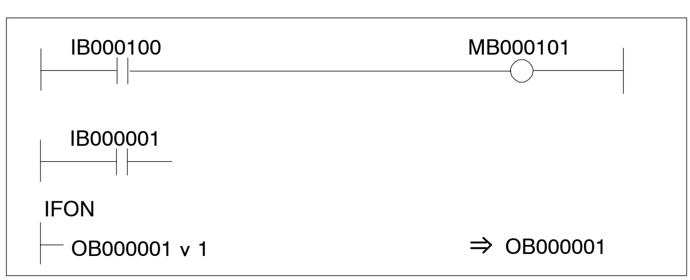


■ Examples of Use by Data Type

Some examples of use by data type are explained below.

Bits

Bits are used for relay circuit ON/OFF or for logic operations.



◀**EXAMPLE**▶

- Motion Program Example

```
MB000101=IB000100;
IF IB000001==1;
DB000001=DB000001|1
```

Words

Words are used for numeric operations and logic operations.

MW00100 v H00FF	⇒	MW00101
MW00102 + 12345	⇒	MW00103
MW00104 INV	⇒	MW00105

◀EXAMPLE▶

- Motion Program Example

MW00101=MW00100|00FFH;
MW00103=MW00102+12345;
MW00105=MW00104;

Double-length Integers

Double-length integers are used for numeric operations and logic operations.

ML00100 + ML00102	⇒	ML00104
ML00106 × ML00108 ÷ 18000	⇒	MW00110
ML00112 BIN	⇒	ML00114

◀EXAMPLE▶

- Motion Program Example

ML00104=ML00100+ML00102;
ML00110=ML00106∗ML00108/18000;
ML00114=BIN (ML00112);

Real Numbers

Real numbers are used for floating-point numeric operations.

1.23456	⇒	DF00100 (1.23456)
DF00102 SIN (30.0)	⇒	DF00104 (0.5)
DF00200 TAN (45.0)	⇒	DF00202 (1.0)

◀EXAMPLE▶

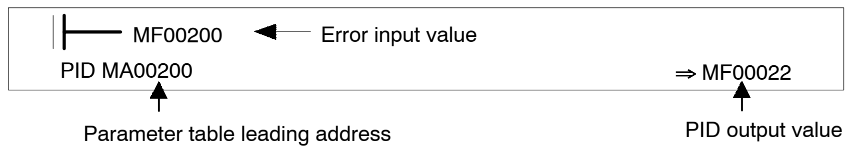
- Motion Program Example

DF00100=1.23456;
DF00104=SIN (DF00102);
DF00202=TAN (DF00200);

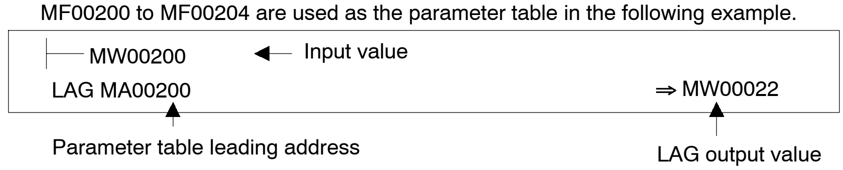
Addresses

Addresses are used only for pointer designations.

MF00200 to MF00228 are used as the parameter table in the following example



MF00200 to MF00204 are used as the parameter table in the following example.



3.6.3 Types of Register

Registers include drawing registers and function registers.

■ Registers in Drawings

The seven types of register shown in *Table 3.14* can be used in all drawings and motion programs.

Table 3.14 Types of Drawing Register

Type	Name	Designation Method	Range	Description	Charac-teristic
S	System registers	SB, SW, SL, SFnnnnn (SAnnnnn)	SW00000 to SW01023	System registers provided by the system. Register number nnnnn is expressed as a decimal number. When the system is started, SW00000 to SW00049 are cleared to 0.	Common to all drawings
M	Data registers	MB, MW, ML, MFnnnnn (MAnnnnn)	MW00000 to MW32767	Data registers are shared by all drawings. Used as interfaces between drawings. Register number nnnnn is expressed as a decimal number.	
I	Input registers	IB, IW, IL, IFhhhh (IAhhhh)	IW0000 to IW07FF	Registers used for input data. Register number hhhh is expressed as a hexadecimal number.	
O	Output registers	OB, OW, OL, Ofhhhh (OAhhhh)	OW0000 to OW07FF	Registers used for output data. Register number hhhh is expressed as a hexadecimal number.	
C	Constant registers	CB, CW, CL, CFnnnnn (CAnnnnn)	CW00000 to CW04095	Constant registers can be referenced only in the program. Register number nnnnn is expressed as a decimal number.	
#	# registers	#B, #W, #L, #Fnnnnn (#Annnnn)	#W00000 to #W16383	# registers can be referenced only in the program and can be used only in the corresponding drawing. The actual range used is specified by the user on the CP-717. Register number nnnnn is expressed as a decimal number.	Unique to each drawing
D	D registers	DB, DW, DL, DFnnnnn (DAnnnnn)	DW00000 to DW16383	D registers are unique to each drawing and can be used only in the corresponding drawing. The actual range used is specified by the user on the CP-717. Register number nnnnn is expressed as a decimal number.	

Note The servo parameter register number (input or output register number) depends on the axis number (axes 1 to 14). *Table 3.15* shows the servo parameter register numbers for each axis.

Table 3.15 Servo Parameter Register Numbers

Axis Number	IW (OW) Address	Axis Number	IW (OW) Address
1	C000 to C03F	8	C1C0 to C1FF
2	C040 to C07F	9	C200 to C23F
3	C080 to C0BF	10	C240 to C27F
4	C0C0 to C0FF	11	C280 to C2BF
5	C100 to C13F	12	C2C0 to C2FF
6	C140 to C17F	13	C300 to C33F
7	C180 to C1BF	14	C340 to C37F

INFO

registers cannot be used in motion programs.

■ Registers in Functions

The 11 types of register shown in *Table 3.16* can be used in functions.

Table 3.16 Types of Function Register

Type	Name	Designation Method	Range	Description	Charac-teristic
X	Function input registers	XB, XW, XL, XFnnnnn	XW00000 to XW00016	Input to a function. Bit input: XB000000 to XB0000F Integer input: XW00001 to XW00016 Double integer input: XL00001 to XL00015 Register number nnnnn is expressed as a decimal number.	Unique to each function
Y	Function output registers	YB, YW, YL, YFnnnnn	YW00000 to YW00016	Output from a function. Bit input: YB000000 to YB0000F Integer input: YW00001 to XW00016 Double integer input: YL00001 to YL00015 Register number nnnnn is expressed as a decimal number.	
Z	Internal function registers	ZB, ZW, ZL, ZFnnnnn	ZW0000 to ZW00063	Internal registers unique to each function. Can be used in the function for internal processes. Register number nnnnn is expressed as a decimal number.	
A	External function registers	AB, AW, AL, AFnnnnn	AW0000 to AW32767	External registers that use the address input value as the base address. For linking with S, M, I, O, #, and DAnnnnn registers. Register number nnnnn is expressed as a decimal number.	
#	# registers	#B, #W, #L, #Fnnnnn (Annnnn)	#W00000 to #W16383	Registers that can only be referenced by a function. Can be used only by the corresponding function. The actual range used is specified by the user on the CP-717. Register number nnnnn is expressed as a decimal number.	
D	D registers	DB, DW, DL, DFnnnnn (DAnnnnn)	DW00000 to DW16383	Internal registers unique to each function. Can be used only by the corresponding function. The actual range used is specified by the user on the CP-717. Register number nnnnn is expressed as a decimal number.	

Type	Name	Designation Method	Range	Description	Charac-teristic
S	System registers	SB, SW, SL, SFnnnnn (SAnnnnn)	Same as the registers for drawings. These registers can be referenced from any drawings or function. Use them carefully when the same function is referenced from drawings with different priority levels.		Common to all drawings
M	Data registers	MB, MW, ML, MFnnnnn (MAnnnnn)			
I	Input registers	IB, IW, IL, IFhhhh (IAhhhh)			
O	Output registers	OB, OW, OL, Ofhhhh (OAhhhh)			
C	Constant registers	CB, CW, CL, CFnnnnn (CAnnnnn)			

Note SA, MA, IA, OA, DA, #A, and CA can also be used inside functions.

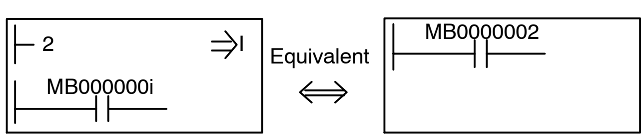
3.6.4 Using Subscripts I and J

Two types of register, I and J, are used exclusively for modifying relay numbers and register numbers. I and J have exactly the same function.

An example of each register data type is explained below.

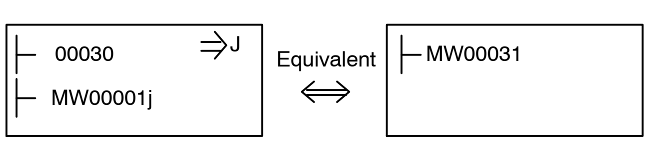
■ Subscripts Attached to Bit Data

When subscript i or j is attached to bit data, the value of I or J is added to the relay number. For example, if I = 2, MB000000i will be the same as MB000002. If J = 27, MB000000j will be the same as MB00001B.



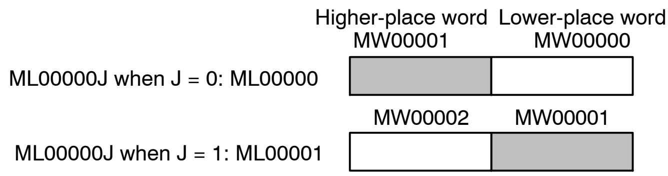
■ Subscripts Attached to Integer Data

When a subscript is attached to integer data, the value of I or J is added to the relay number. For example, if I = 3, MW00010i will be the same as MW00013. If J = 30, MW00001j will be the same as MW00031.



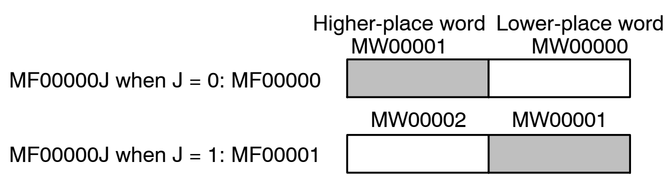
Subscripts Attached to Double Integer Data

When a subscript is attached to double integer data, the value of I or J is added to the relay number. For example, if I = 1, ML00000i will be the same as ML00001. ML00000j when J = 0, and ML00000j when J = 1 will be as follows:



Subscripts Attached to Real Number Data

When a subscript is attached to double integer data, the value of I or J is added to the relay number. For example, if I = 1, MF00000i will be the same as MF00001. MF00000j when J = 0, and MF00000j when J = 1 will be as follows:



Programming Example Using Subscripts

The programming code shown in *Figure 3.16* sets the sum of 100 registers from MW00100 to MW00199 in MW00200 using subscript J.

```
├ 00000                          ⇒ MW00200
FOR  J = 00000 to 00099 by 00001
├ MW00200 +  MW00100j            ⇒ MW00200
FEND
```

Figure 3.16 Programming Example Using a Subscript

INFO

Subscripts I and J cannot be used in motion programs.

3.6.5 I/O and Registers in Functions

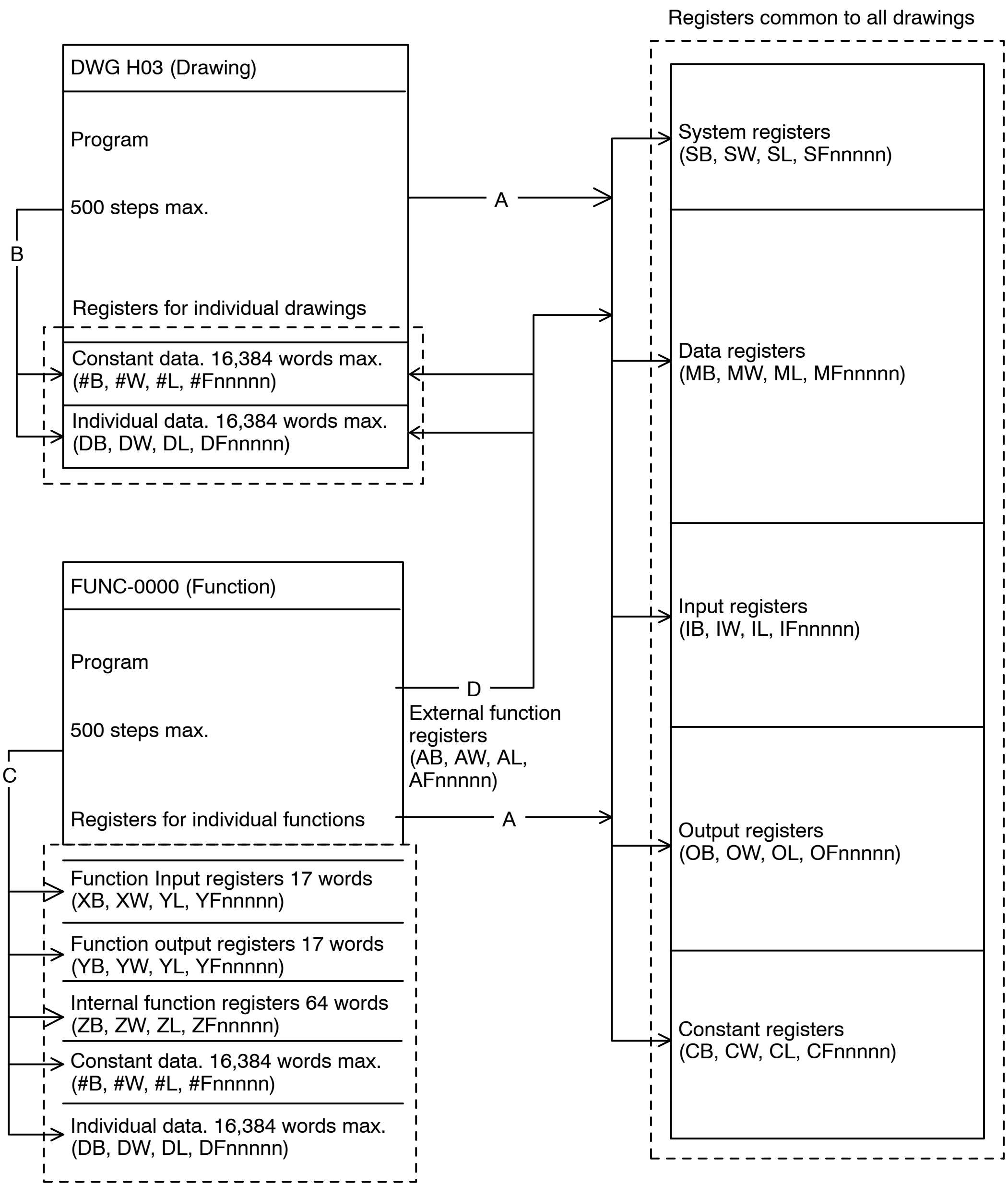
Table 3.17 shows the I/O and registers referenced in functions.

Table 3.17 Correspondence Between I/O and Registers in Functions

Function I/O	Function Register
Bit inputs	The bit numbers increase continuously from XB000000 in order of the bit inputs: XB000000, XB000001, XB000002, ……, XB00000F
Integer, double integer, and real number inputs	The register numbers increase continuously from XW, XL, and XF00001 in order of the integer, double integer, and real number inputs: XW00001, XW00002, XW00003, ……, XW00016 XL00001, XL00003, XL00005, ……, XL00015 XF00001, XF00003, XF00005, ……, XF00015
Address inputs	The address input values correspond to register numbers 0 of the external register: Input value = MA00100: MW00100 = AW00000, MW00100 = AW00001...
Bit outputs	The bit number increases consecutively from YB000000 in order of bit outputs: (YB000000, YB000001, YB000002, YB00000F)
Integer, double integer, and real number outputs	The register numbers increase continuously from YW, YL, and YF00001 in order of the integer, double integer, and real number outputs. YW00001, YW00002, YW00003, ..., YW00016 YL00001, YL00003, YL00005, ..., YL00015 YF00001, YF00003, FY00005, ..., YF00015

3.6.6 Register Ranges in Programs

Figure 3.17 shows the ranges that can be called for registers in programs.



A: Registers that are common to all drawings can be referenced from any drawing or function.

B: Registers that are unique to each drawing can be referenced only from within that drawing.

C: Registers that are unique to each function can be referenced only from within that function.

D: Registers that are common to all drawings and registers that are unique to each drawing can be referenced from a function using the external function registers.

Figure 3.17 Referencing Ranges for Registers in Programs

3.7 Managing Symbols

This section describes symbol management and upward linking, together with the automatic allocation of register numbers.

3.7.1 Symbols in Drawings

The symbols used in drawings are all managed with a symbol table, such as the one shown in *Table 3.18*. For details, refer to the *MP9□□ Ladder programming manual* (SIEZ-C887-1.2).

Table 3.18 Drawing Symbol Table (Example)

No.	Register No.	Symbol	Size *	Remarks
0	IB00000	STARTPBL	1	The register number is expressed as a hexadecimal number.
1	OB00000	STARTCOM	1	The register number is expressed as a hexadecimal number.
2	MW00000	SPDMAS	1	
3	MB000010	WORK-DB	16	
4	MW00010	PIDDATA	10	
5	MW00020	LAUIN	1	
6	MW00021	LAUOUT	1	
:				
N				

* If a program is written using data configurations such as arrays or indexed data, define the size to be used in the data configuration. For example, if the data is referenced as PIDDATA_1 and i varies in a range of 0 to 9, define the size as 10.

3.7.2 Symbols in Functions

All symbols used in the functions are managed with the function symbol table shown in *Table 3.19*. For the method of defining the function symbol table, refer to the *MP9□□ Ladder programming manual* (SIEZ-C887-1.2) .

Table 3.19 Function Symbol Table

No.	Register No.	Symbol	Size *	Remarks
0	XB000000		1	
1	XW00001		1	
2	AW00001		1	
3	AB00000F		1	
4	YB000000		1	
5	YW00001		1	
6	ZB000000		4	
7	ZW00001		1	
8	ZW00002		1	
:				
N				

* If a program is prepared using data configurations such as arrays or indexed data, define the size to be used in the data configuration. For example, if the data is referenced as PIDDATA_1 and i varies in a range of 0 to 9, define the size as 10.

3

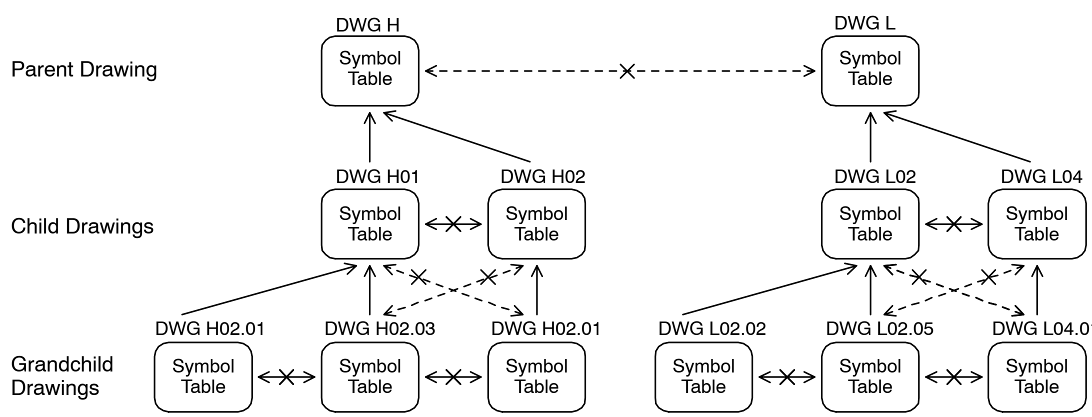
3.7.3 Upward Linking of Symbols

Table 3.20 shows the symbols that can be linked and the symbols tables that are subject to linking. For details on the upward linking of symbols, refer to the *MP9□□ Ladder Programming Manual* (SIEZ-C887-1.2) and the *MP9□□ Programming Software User's Manuals* (SIEZ-C887-2.2-1, SIEZ-C887-2.2-2) .

Table 3.20 Linkable Symbols and Symbol Table for Linking

Symbols \ Symbol Table	Parent Drawing	Child Drawing	Grandchild Drawing
Parent Drawing Symbols	No	No	No
Child Drawing Symbols	**Yes**	No	No
Grandchild Drawing Symbols	**Yes**	**Yes**	No
Symbols Within A Function	No	No	No

Yes: Possible
No: Not possible



3.7.4 Automatic Register Number Allocation

Table 3.21 shows the register numbers for which automatic allocation is possible and those for which it is not possible. For details on the automatic allocation of register numbers, refer to the *MP9□□ Ladder programming manuals* (SIEZ-C887-1.2) and the MP9□□ *Programming Software User's Manuals* (SIEZ-C887-2.2-1, SIEZ-C887-2.2-2) .

Table 3.21 Automatic Allocation of Register Numbers

Drawing Symbol Table	Automatic Number Allocation	Function Symbol Table	Automatic Number Allocation
System registers S	**Yes**	System registers S	**Yes**
Input registers I	**Yes**	Input registers I	**Yes**
Output registers O	**Yes**	Output registers O	**Yes**
Data registers M	**Yes**	Data registers M	**Yes**
# registers #	**Yes**	# registers #	**Yes**
C registers C	**Yes**	C registers C	**Yes**
D registers D	**Yes**	D registers D	**Yes**
–	–	Function input registers X	No
–	–	Function output registers Y	No
–	–	Internal function registers Z	**Yes**
–	–	External function registers A	No

Yes: Automatic number allocation possible
No: Automatic number allocation not possible

4

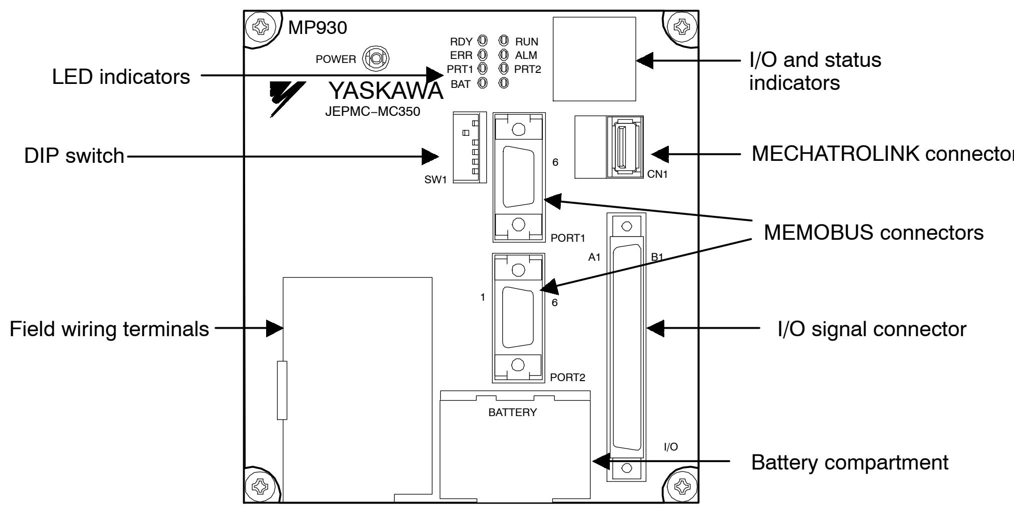
System Startup

This chapter explains the method of connecting the system and the startup procedure.

4.1 Part Names

This section gives the name of the parts of the MP930 and a general explanation of each part.

4.1.1 MC Unit



■ LED Indicators

Indicator Name	Indicator Color	Meaning When Indicator Is Lit	Appearance
RDY	Green	System operating normally.	RDY RUN ERR ALM PRT1 PRT2 BAT
RUN	Green	User program running.	
ERR	Red	System fault or failure	
ALM	Red	Minor system failure	
PRT1	Green/Red	Lights during port 1 transmission and reception	
PRT2	Green/Red	Lights up during port 2 transmission and reception	
BAT	Red	Battery needs replacing.	

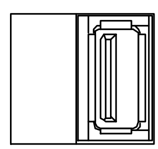
I/O and Status Indicators

Indicator Name	Indicator Color	Meaning When Indicator Is Lit	Appearance
R	Orange	Not used. (Not lit.)	R ACTIVE F 1 9 17 25 2 10 18 26 3 11 19 27 4 12 20 28 5 13 21 29 6 14 22 30 7 15 23 31 8 16 24 32
ACTIVE	Orange	Lights during MECHATROLINK transmission.	
F	Red	Broken fuse (24-V power is supplied to the I/O signal connector. The indicator lights even if no power is supplied.)	
1 to 16	Orange	Input signal monitor	
17 to 32	Orange	Output signal monitor	

MECHATROLINK Connector

An I/O Unit for MECHATROLINK communications (JEPMC-IO350) or a Servopack (SGD-□□□N or SGDB-□□AN) is connected by a MECHATROLINK Cable (JEPMC-W6000-A3).

For details on the connection methods, see *4.3.5 Connecting the Devices*.



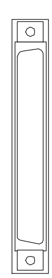
CN1

I/O Signal Connector

The MC Unit and external I/O signals are connected by an I/O Cable (JEPMC-W5410-05).

Number of signal points: 16 inputs and 16 outputs

For details on the connection methods, see *4.3.5 Connecting the Devices*.

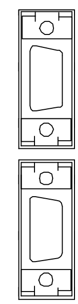


MEMOBUS Ports

- Using RS-232C, the MC Unit can communicate with transmission devices on the MEMOBUS network through the MEMOBUS ports.
- The following transmission devices can be connected to the MEMOBUS ports: Programming Device (computer equipped with an RS-232C interface).

- The MEMOBUS port connector is a D-sub connector (9-pin, female). *Table 4.1* shows the layout of the connector pins and the signal names.

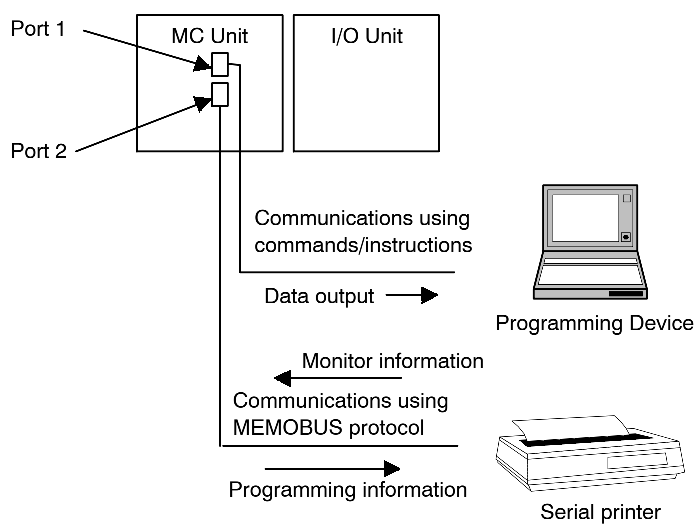
Table 4.1 MEMOBUS Port Layout and Signal Names

Pin Number	Abbreviation	Signal Name	Appearance
1	FG	Protective ground	
2	TXD	Transmission data	
3	RXD	Reception data	
4	RTS	Request to send	
5	CTS	Clear to send	
6	DSR	Data set ready	
7	GND	Signal ground	
8	–		
9	DTR	Data terminal ready	

◀**EXAMPLE**▶

- The following illustration shows an example of the use of the MEMOBUS ports.

  An ASCII Device (printer) can be connected to the port.

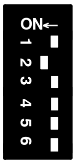


■ DIP Switch

- The DIP switch consists of six pins. The pins are numbered 1 to 6, as shown in the diagram in *Table 4.2*.
- Each pin is ON when it is moved to the left.

- *Table 4.2* shows the function of each pin.

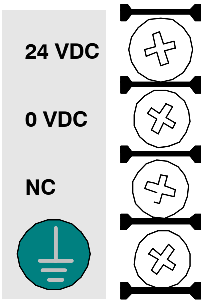
Table 4.2 DIP Switch Functions

Pin Number	Abbreviation	Setting	Function	Appearance
1	FLASH	ON	Used by the system.	
		OFF	–	
2	RUN	ON	User program executed.	
		OFF	User program stopped.	
3	INIT	ON	Pin Number 4 ON: Memory clear OFF: Programming Panel port defaults	
		OFF	Online	
4	TEST	ON	Terminal (communications) mode	
		OFF	Online	
5	Not used	ON	–	
		OFF	–	
6	S.TST	ON	Used by the system	
		OFF	Online	

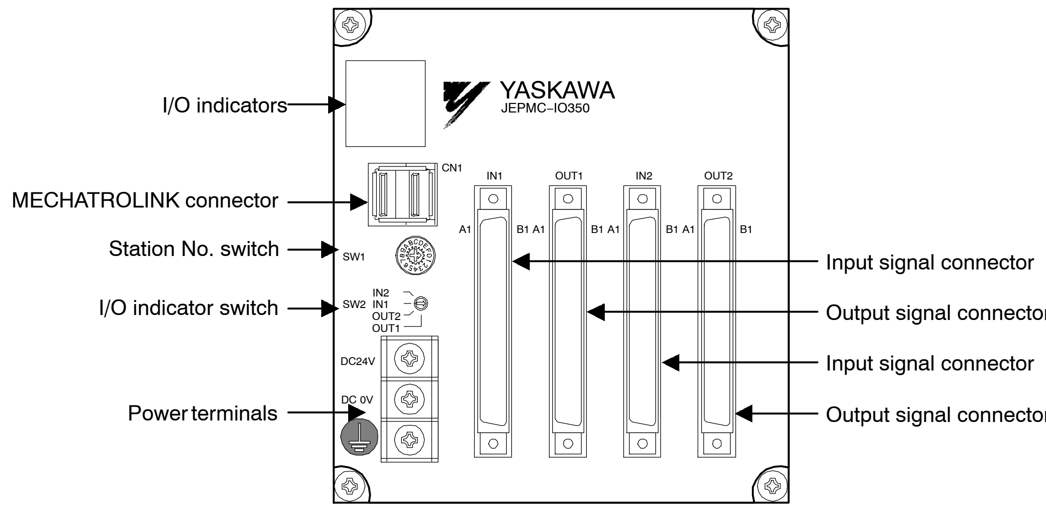
Note During normal use, pins 1, 4, 5, and 6 are always OFF.

■ Field Wiring Terminals

The field wiring terminal supplies 24 VDC to the CPU Module.

Terminal Name	Function	Appearance
24 VDC	+24 VDC	
0 VDC	0 VDC	
NC	Not used.	
FG	Protective ground terminal	

4.1.2 I/O Unit



4

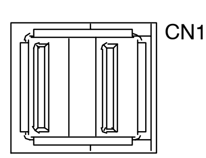
■ I/O and Status Indicators

Indicator Name	Indicator Color	Meaning When Indicator Is Lit	Appearance
R	Green	Lights when current is conducted.	R ACTIVE F 1 9 17 25 2 10 18 26 3 11 19 27 4 12 20 28 5 13 21 29 6 14 22 30 7 15 23 31 8 16 24 32
ACTIVE	Orange	Lights during MECHATROLINK transmission.	
F	Red	Broken fuse	
1 to 32	Orange	Input signal and output signal monitors. The indicator meaning is changed with the I/O indicator switch.	

■ MECHATROLINK Connector

The MC Unit and Servopack are connected by MECHATROLINK Cable (such as JEPMC-W6000-A3 or JEPMC-W6000-01).

For details on the connection methods, see *4.3.5 Connecting the Devices*.

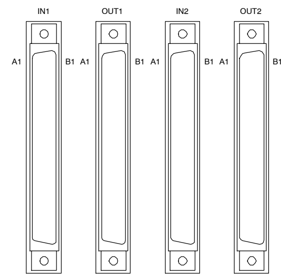


■ I/O Signal Connectors

The I/O Unit and external I/O signals are connected by I/O Cables (JEPMC-W5410-05).

Number of signal points: 16 inputs and 16 outputs

For details on the connection methods, see *4.3.5 Connecting the Devices*.



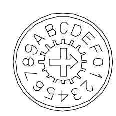
Station Number Switch

This switch sets the MECHATROLINK station number.

- Setting range: 1 to E

When multiple Units are connected, make sure the same station number is not used more than once.

SW1

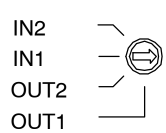


I/O Indicator Switch

The I/O signal monitor displays can be switched in 32-point units.

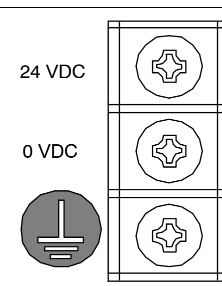
IN1: Input signals 1 to 32
IN2: Input signals 33 to 64
OUT1: Output signals 1 to 32
OUT2: Output signals 33 to 64

SW2



Field Wiring Terminals

The field wiring terminals supply 24-VDC to the I/O Unit.

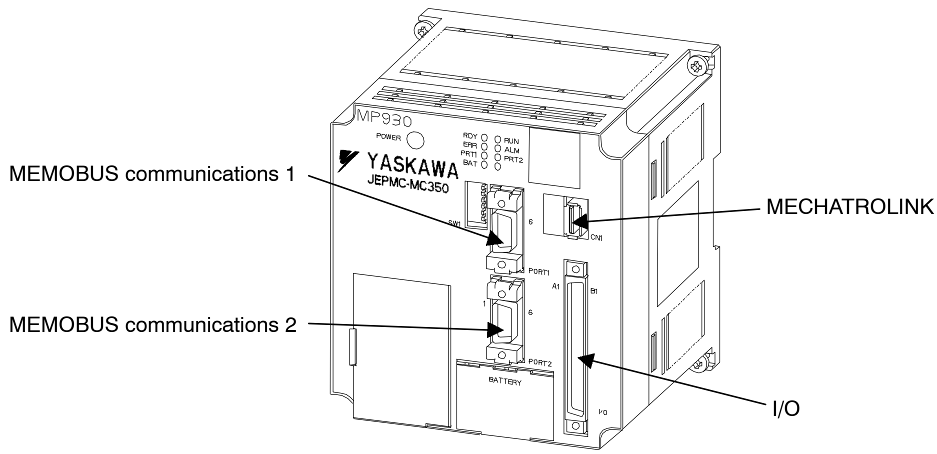
Terminal Name	Function	Appearance
24 VDC	+24 VDC	
0 VDC	0 VDC	
FG	Protective ground terminal	

4.2 Connection Methods

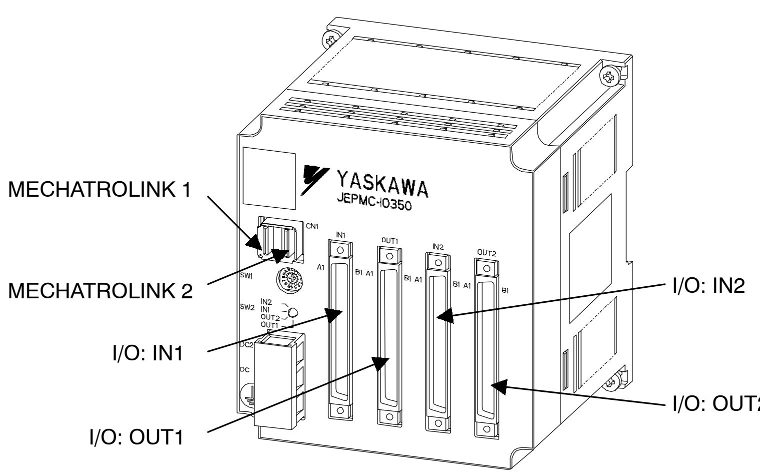
This section explains methods used to connect the MP930 Units to other devices.

4.2.1 Names and Locations of Connectors

The following illustration shows the names and locations of the connectors on the front of the MP930 MC Unit and I/O Unit.



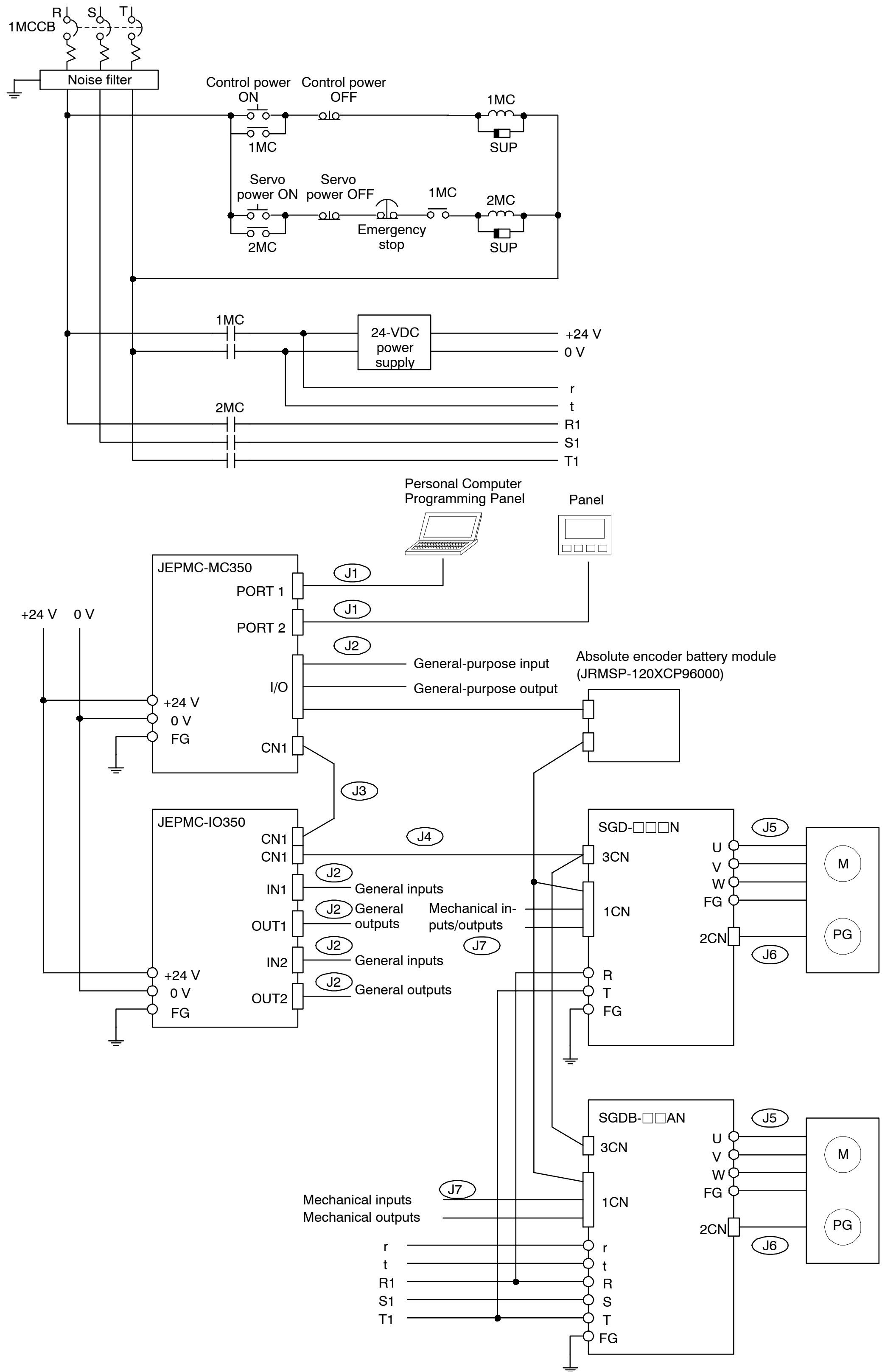
MC Unit



I/O Unit

4.2.2 System Connection Example

The following illustration shows an example of the system connections used for the MP930.



4.2.3 Standard Cables

YASKAWA provides the standard cables shown in the following table.

These cables are used to connect the Servo Amps and external I/O devices to the MP930 MC Unit and I/O Unit.

The following table shows a list of the cables required when the system is configured with the MP930.

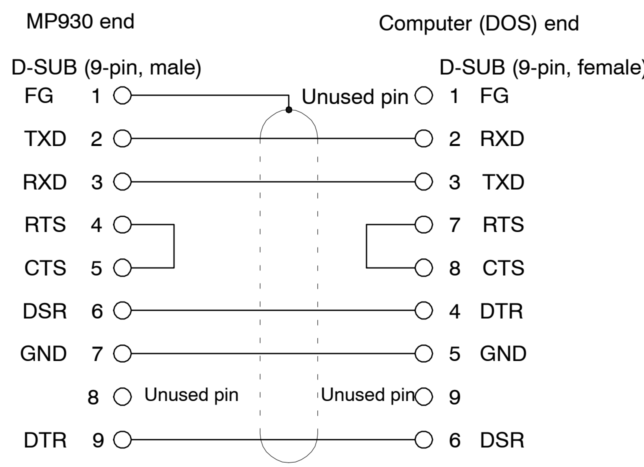
<table>
<tr><th>No.</th><th>Specifications</th><th>Model</th><th>Length</th></tr>
<tr><td rowspan="4">J1</td><td rowspan="2">MEMOBUS to Computer Connection Cable
D-SUB connector (9-pin, male) to D-SUB connector (25-pin, male)</td><td>JEPMC-W5310-03</td><td>2.5 m</td></tr>
<tr><td>JEPMC-W5310-15</td><td>15 m</td></tr>
<tr><td rowspan="2">MEMOBUS to Computer Connection Cable
D-SUB connector (9-pin, male) to D-SUB connector (9-pin, female)</td><td>JEPMC-W5311-03</td><td>2.5 m</td></tr>
<tr><td>JEPMC-W5311-15</td><td>15 m</td></tr>
<tr><td rowspan="3">J2</td><td rowspan="3">I/O Cable
FCN360 connector to loose wires</td><td>JEPMC-W5410-05</td><td>0.5 m</td></tr>
<tr><td>JEPMC-W5410-10</td><td>1 m</td></tr>
<tr><td>JEPMC-W5410-30</td><td>3 m</td></tr>
<tr><td>J3</td><td>MECHATROLINK Cable (MC Unit to I/O Unit)
USB connector to USB connector</td><td>JEPMC-W6000-A3</td><td>0.3 m</td></tr>
<tr><td rowspan="10">J4</td><td rowspan="3">MECHATROLINK Cable
(MC Unit to Servopack, I/O Unit to Servopack)
USB connector to loose wire</td><td>JEPMC-W6010-01</td><td>1 m</td></tr>
<tr><td>JEPMC-W6010-03</td><td>3 m</td></tr>
<tr><td>JEPMC-W6010-05</td><td>5 m</td></tr>
<tr><td>Servopack Connector Kit
MR connector (8-pin, female), for axis 1</td><td>DE9411357</td><td>–</td></tr>
<tr><td rowspan="6">MECHATROLINK Cable
Cable material</td><td>DE9411358-1</td><td>10 m</td></tr>
<tr><td>DE9411358-2</td><td>20 m</td></tr>
<tr><td>DE9411358-3</td><td>30 m</td></tr>
<tr><td>DE9411358-4</td><td>40 m</td></tr>
<tr><td>DE9411358-5</td><td>100 m</td></tr>
<tr><td>DE9411358-6</td><td>200 m</td></tr>
</table>

No.	Specifications	Model	Length
J5	Cable for SGD−□□□N Motor without Brake	DP9320081-1	1 m
		DP9320081-2	2 m
		DP9320081-3	3 m
		DP9320081-4	4 m
		DP9320081-5	5 m
	Cable SGD−□□□N Motor with Brake	DP9320083-1	1 m
		DP9320083-2	2 m
		DP9320083-3	3 m
		DP9320083-4	4 m
		DP9320083-5	5 m
J6	Incremental Encoder Cable	DP9320089-1	1 m
		DP9320089-2	2 m
		DP9320089-3	3 m
		DP9320089-4	4 m
		DP9320089-5	5 m
	Absolute Encoder Cable	DP9320088-1	1 m
		DP9320088-2	2 m
		DP9320088-3	3 m
		DP9320088-4	4 m
		DP9320088-5	5 m
J7	Servopack External I/O Signal Terminal Block Set Used for both SGD−N and SGDB−N Servopack 0.5−m Cable and Terminal Block Unit	JUSP-TA26P	0.5 m
	Servopack External I/O Signal Cable MDR connector (26−pin) to loose wires	DE9411355	1 m

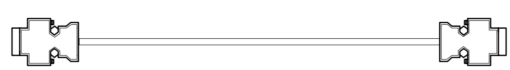
4.2.4 Connector Pin Layout and I/O Circuits

■ Connection of Communications Cables

The following illustration shows the internal connections and outline of the communications cables.

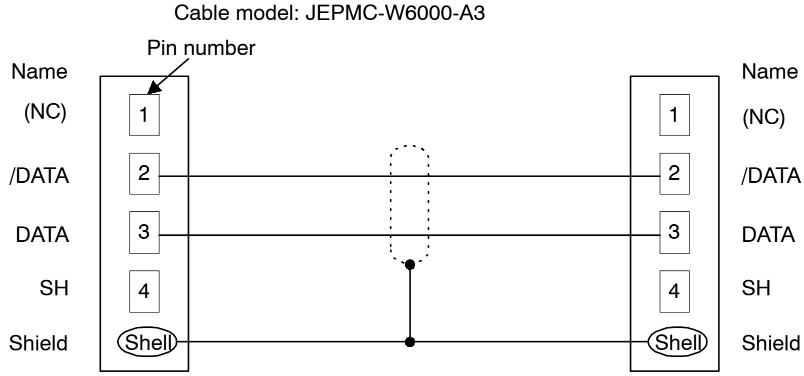


The following illustration shows the appearance of the MEMOBUS cable.

Model: JEPMC-W5311-□□


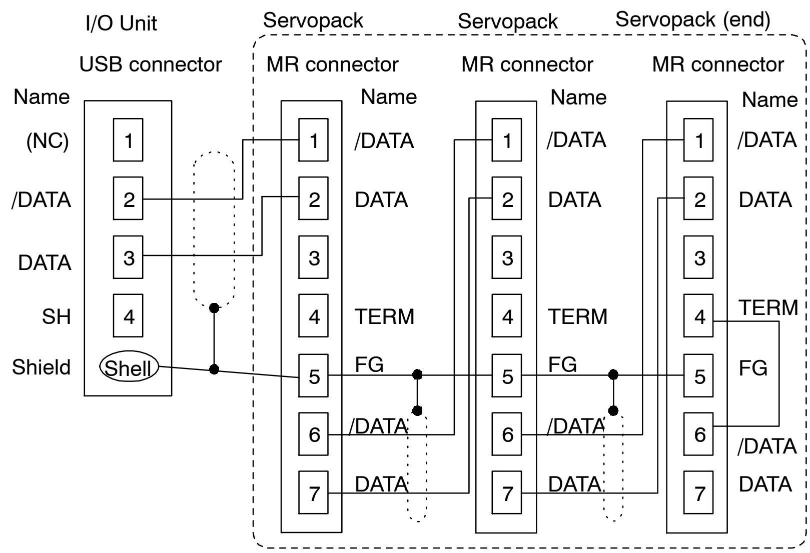
■ MECHATROLINK Cables

The following illustration shows the internal connections of the cable between the MC Unit and the I/O Unit.



The following illustration shows the internal connections of a 1:N MECHATROLINK cable.

Cable model: JEPMC-W6010-□□



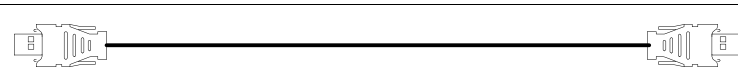
Note 1. The JEPMC-6010-□□ has loose wires on one end and a USB connector on the other. The customer must prepare the 1:N cable using an MR connector and wire material.

2. Red lead: DATA
Black lead: /DATA

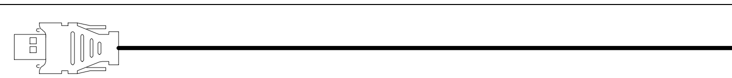
3. Sield wiring
In general, follow the instructions in Servopack Manuals. However, in the combination with MP9□□series, the connection shown in the above illustration is recommended.

Appearance of the MECHATROLINK Cables

Model: JEPMC-W6000-A3



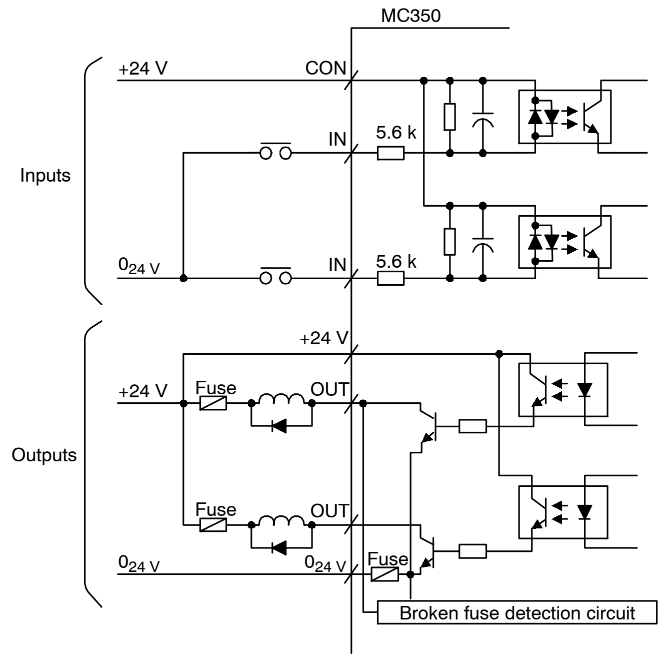
Model: JEPMC-W6010-□□



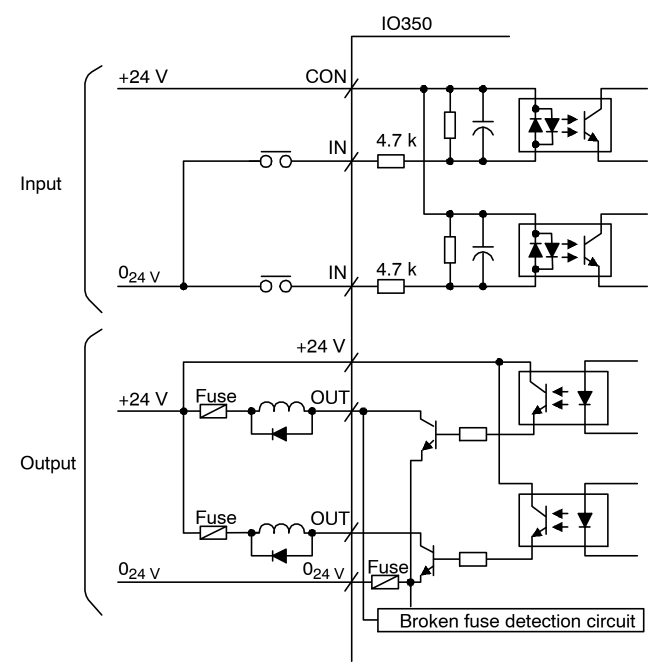
■ I/O Circuits

The following illustration shows the I/O circuits of the MC350 Unit and the IO350 Unit.

MC350 I/O Circuits



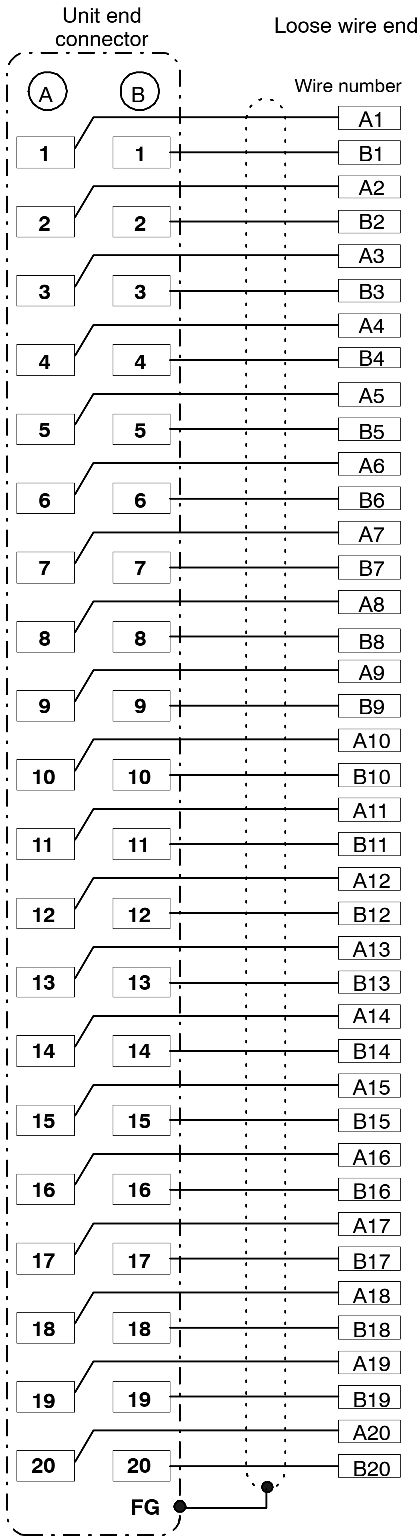
IO350 I/O Circuits



IMPORTANT A fuse is inserted in the common output line of the MC Unit and I/O Unit as a protective circuit. If the output layer shorts, there is a risk that the fuse may not break. Insert a protective element, such as a fuse, for each output, as shown in the above illustration.

■ MC Unit I/O Cables

The pin layout of the cables (model: JEPMC-W5410-□□) between the MC Unit I/O connectors and the external I/O signals is as shown in the following illustration.



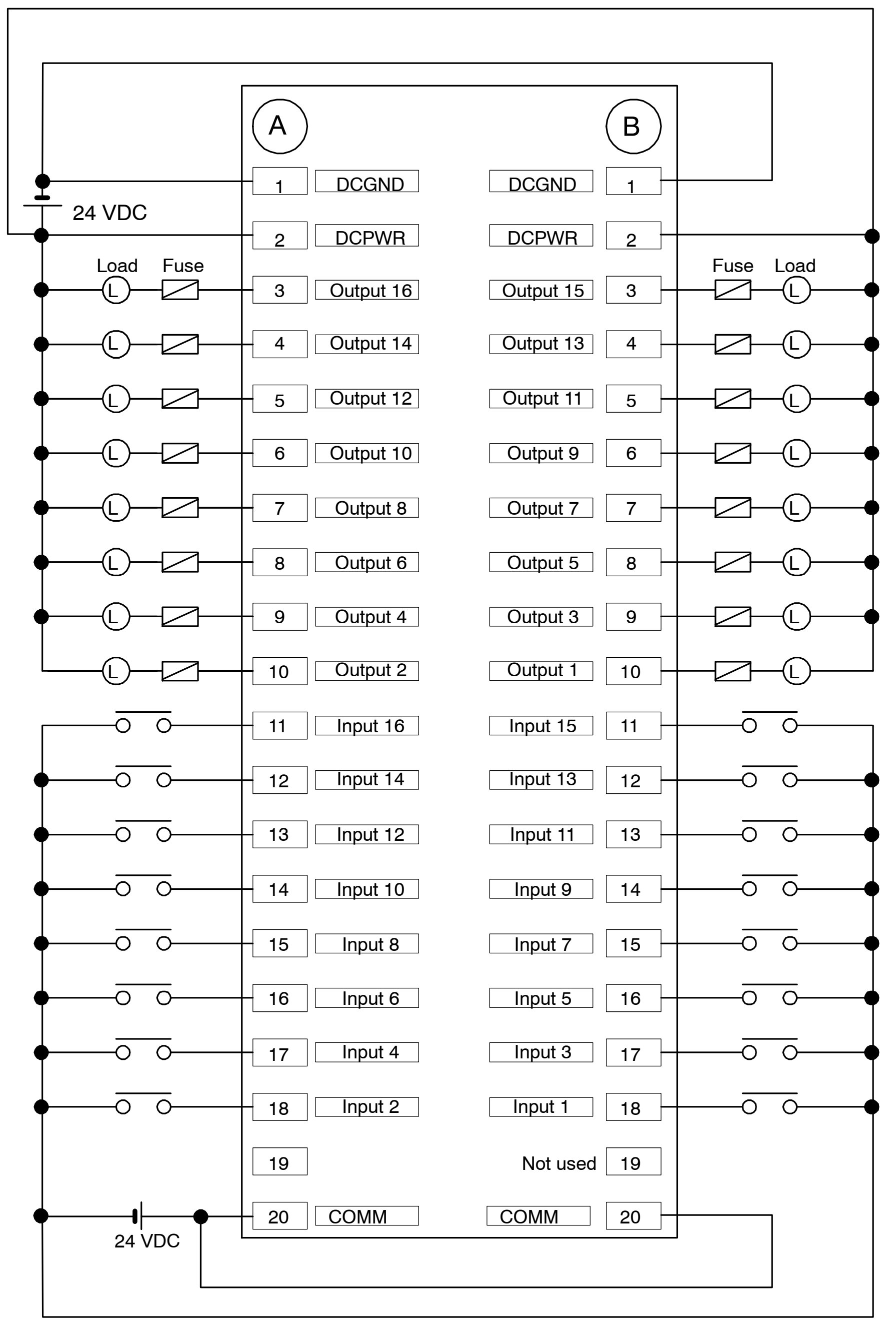
Signal Name	Covering Color	Mark: About 1 mm Space: About 1 mm
DCGND	Blue	Red –
DCGND	Blue	Black –
DCPWR	Pink	Red –
DCPWR	Pink	Black –
Output 16	Green	Red –
Output 15	Green	Black –
Output 14	Orange	Red –
Output 13	Orange	Black –
Output 12	Gray	Red –
Output 11	Gray	Black –
Output 10	Blue	Red – –
Output 9	Blue	Black – –
Output 8	Pink	Red – –
Output 7	Pink	Black – –
Output 6	Green	Red – –
Output 5	Green	Black – –
Output 4	Orange	Red – –
Output 3	Orange	Black – –
Output 2	Gray	Red – –
Output 1	Gray	Black – –
Input 16	Blue	Red – – –
Input 15	Blue	Black – – –
Input 14	Pink	Red – – –
Input 13	Pink	Black – – –
Input 12	Green	Red – – –
Input 11	Green	Black – – –
Input 10	Orange	Red – – –
Input 9	Orange	Black – – –
Input 8	Gray	Red – – –
Input 7	Gray	Black – – –
Input 6	Blue	Red – – – –
Input 5	Blue	Black – – – –
Input 4	Pink	Red – – – –
Input 3	Pink	Black – – – –
Input 2	Green	Red – – – –
Input 1	Green	Black – – – –
Not used	Orange	Red – – – –
Not used	Orange	Black – – – –
COMM	Gray	Red – – – –
COMM	Gray	Black – – – –

Note Cable end connector Model : FCN-361J-040-AU for soldering jack
FCN-360C-040-B for cover
(Made by Fujitsu Ltd.)

Appearance of the MC Unit I/O Cable

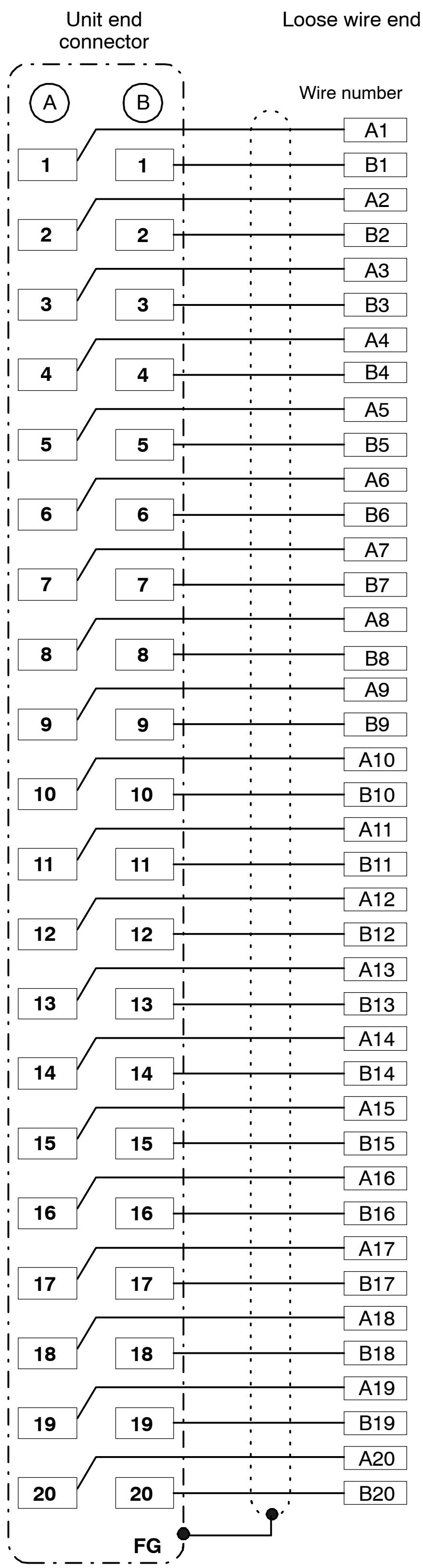
Model: JEPMC-W5410-□□

Connection Example of the MC Unit I/O Connector



■ I/O Unit Input Cables

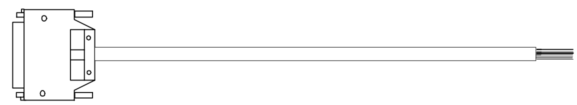
The pin layout of the cables (model: JEPMC-W5410-□□) between the I/O Unit input connectors and the external I/O signals is as shown in the following illustration.



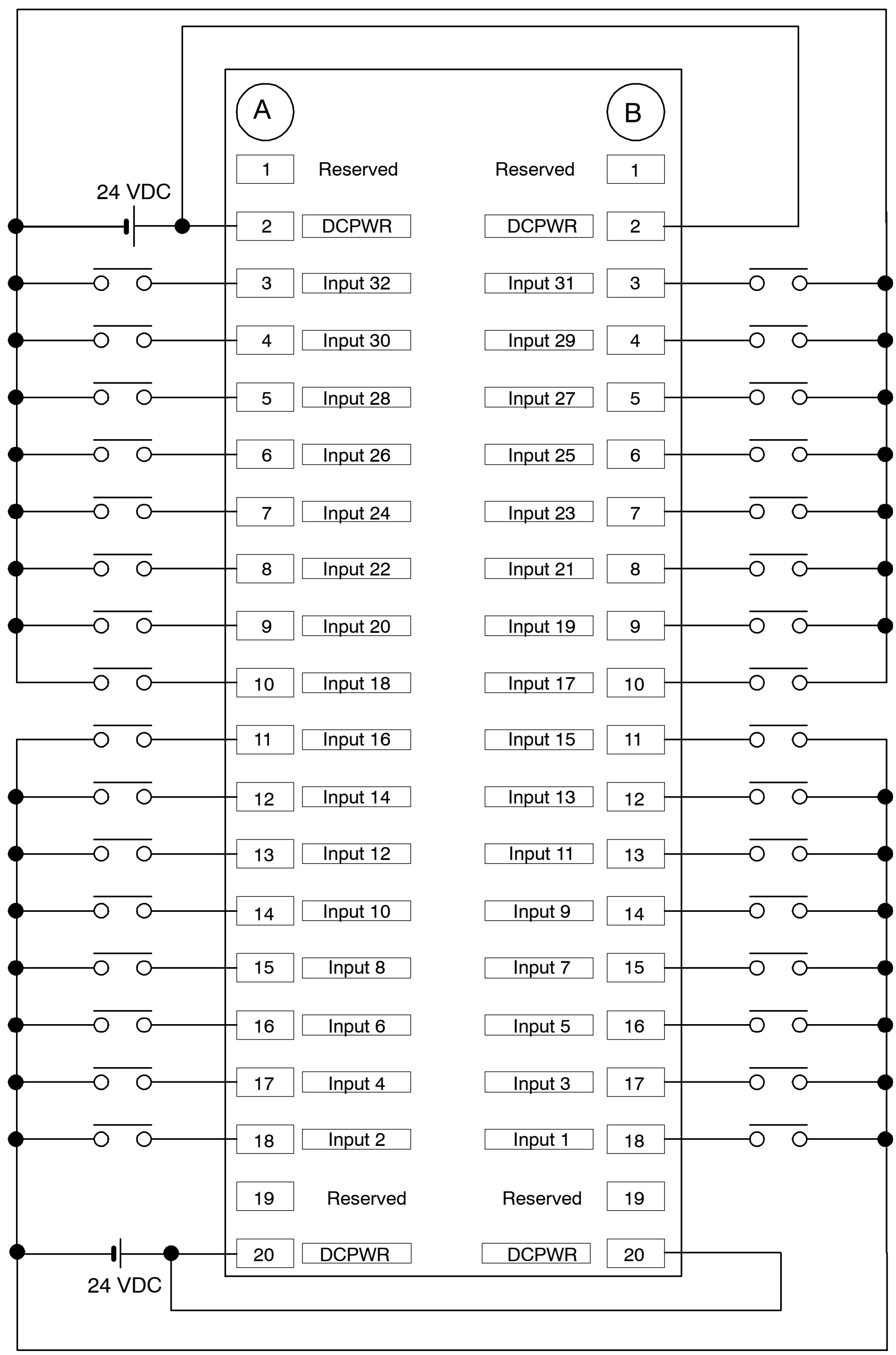
Signal Name	Covering Color	Mark: About 1 mm Space: About 1 mm
Not used	Blue	Red –
Not used	Blue	Black –
DCPWR	Pink	Red –
DCPWR	Pink	Black –
Input 32	Green	Red –
Input 31	Green	Black –
Input 30	Orange	Red –
Input 29	Orange	Black –
Input 28	Gray	Red –
Input 27	Gray	Black –
Input 26	Blue	Red – –
Input 25	Blue	Black – –
Input 24	Pink	Red – –
Input 23	Pink	Black – –
Input 22	Green	Red – –
Input 21	Green	Black – –
Input 20	Orange	Red – –
Input 19	Orange	Black – –
Input 18	Gray	Red – –
Input 17	Gray	Black – –
Input 16	Blue	Red – – –
Input 15	Blue	Black – – –
Input 14	Pink	Red – – –
Input 13	Pink	Black – – –
Input 12	Green	Red – – –
Input 11	Green	Black – – –
Input 10	Orange	Red – – –
Input 9	Orange	Black – – –
Input 8	Gray	Red – – –
Input 7	Gray	Black – – –
Input 6	Blue	Red – – – –
Input 5	Blue	Black – – – –
Input 4	Pink	Red – – – –
Input 3	Pink	Black – – – –
Input 2	Green	Red – – – –
Input 1	Green	Black – – – –
Not used	Orange	Red – – – –
Not used	Orange	Black – – – –
DCPWR	Gray	Red – – – –
DCPWR	Gray	Black – – – –

Appearance of the I/O Unit Input Cable

Model: JEPMC-W5410-□□

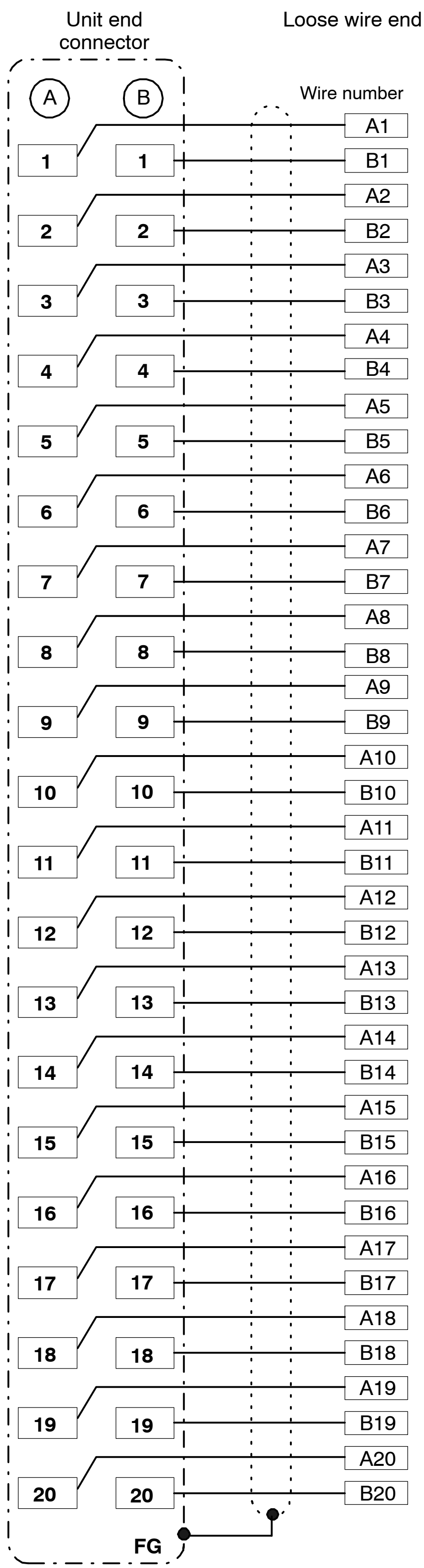


Connection Example of the I/O Unit Input Connector



■ I/O Unit Output Cables

The pin layout of the cables (model: JEPMC-W5410-□□) between the I/O Unit output connectors and the external I/O signals is as shown in the following illustration.

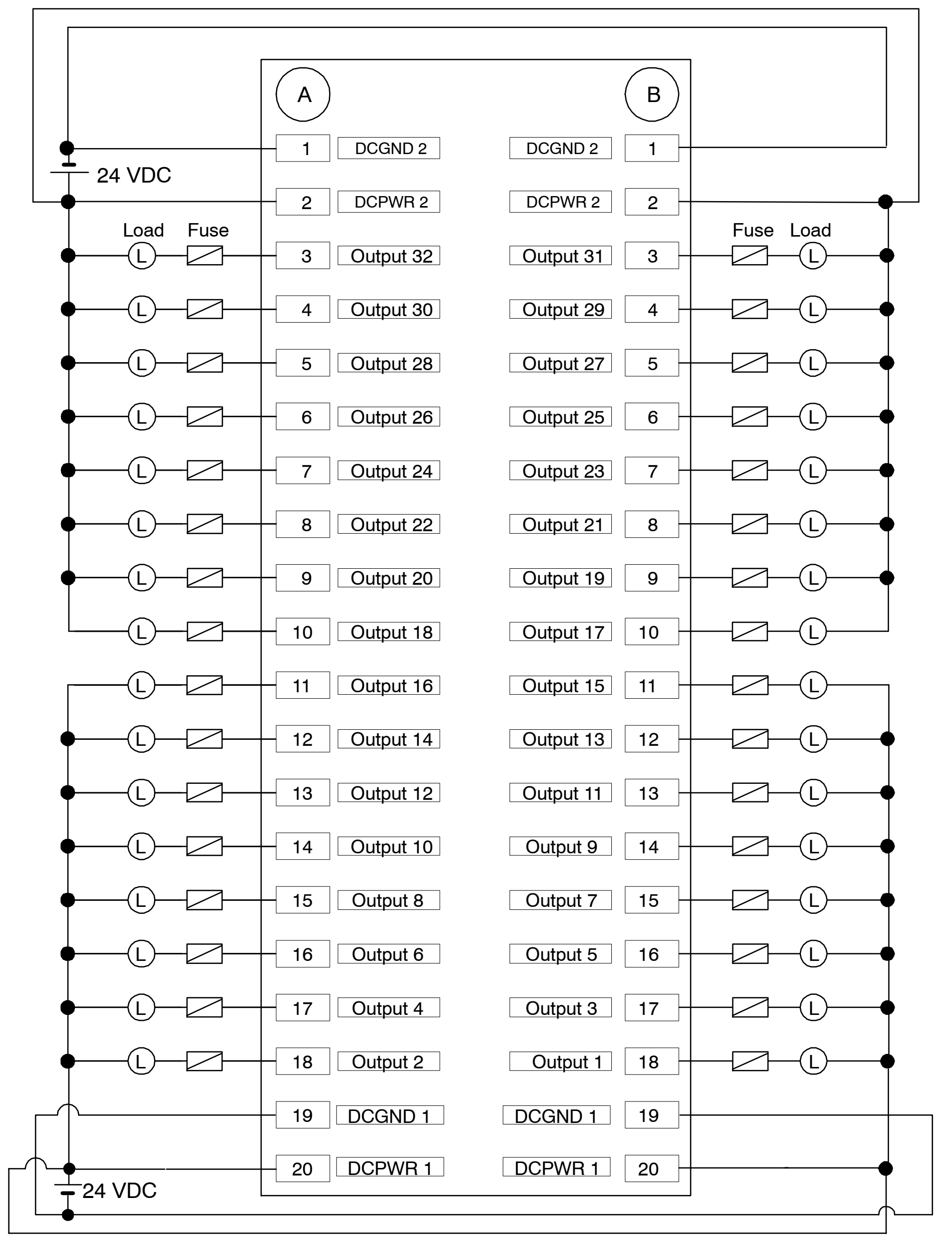


Signal Name	Covering Color	Mark: About 1 mm Space: About 1 mm
DCGND 2	Blue	Red –
DCGND 2	Blue	Black –
DCPWR 2	Pink	Red –
DCPWR 2	Pink	Black –
Output 32	Green	Red –
Output 31	Green	Black –
Output 30	Orange	Red –
Output 29	Orange	Black –
Output 28	Gray	Red –
Output 27	Gray	Black –
Output 26	Blue	Red – –
Output 25	Blue	Black – –
Output 24	Pink	Red – –
Output 23	Pink	Black – –
Output 22	Green	Red – –
Output 21	Green	Black – –
Output 20	Orange	Red – –
Output 19	Orange	Black – –
Output 18	Gray	Red – –
Output 17	Gray	Black – –
Output 16	Blue	Red – – –
Output 15	Blue	Black – – –
Output 14	Pink	Red – – –
Output 13	Pink	Black – – –
Output 12	Green	Red – – –
Output 11	Green	Black – – –
Output 10	Orange	Red – – –
Output 9	Orange	Black – – –
Output 8	Gray	Red – – –
Output 7	Gray	Black – – –
Output 6	Blue	Red – – – –
Output 5	Blue	Black – – – –
Output 4	Pink	Red – – – –
Output 3	Pink	Black – – – –
Output 2	Green	Red – – – –
Output 1	Green	Black – – – –
DCGND 1	Orange	Red – – – –
DCGND 1	Orange	Black – – – –
DCPWR 1	Gray	Red – – – –
DCPWR 1	Gray	Black – – – –

Appearance of the I/O Unit Output Cable

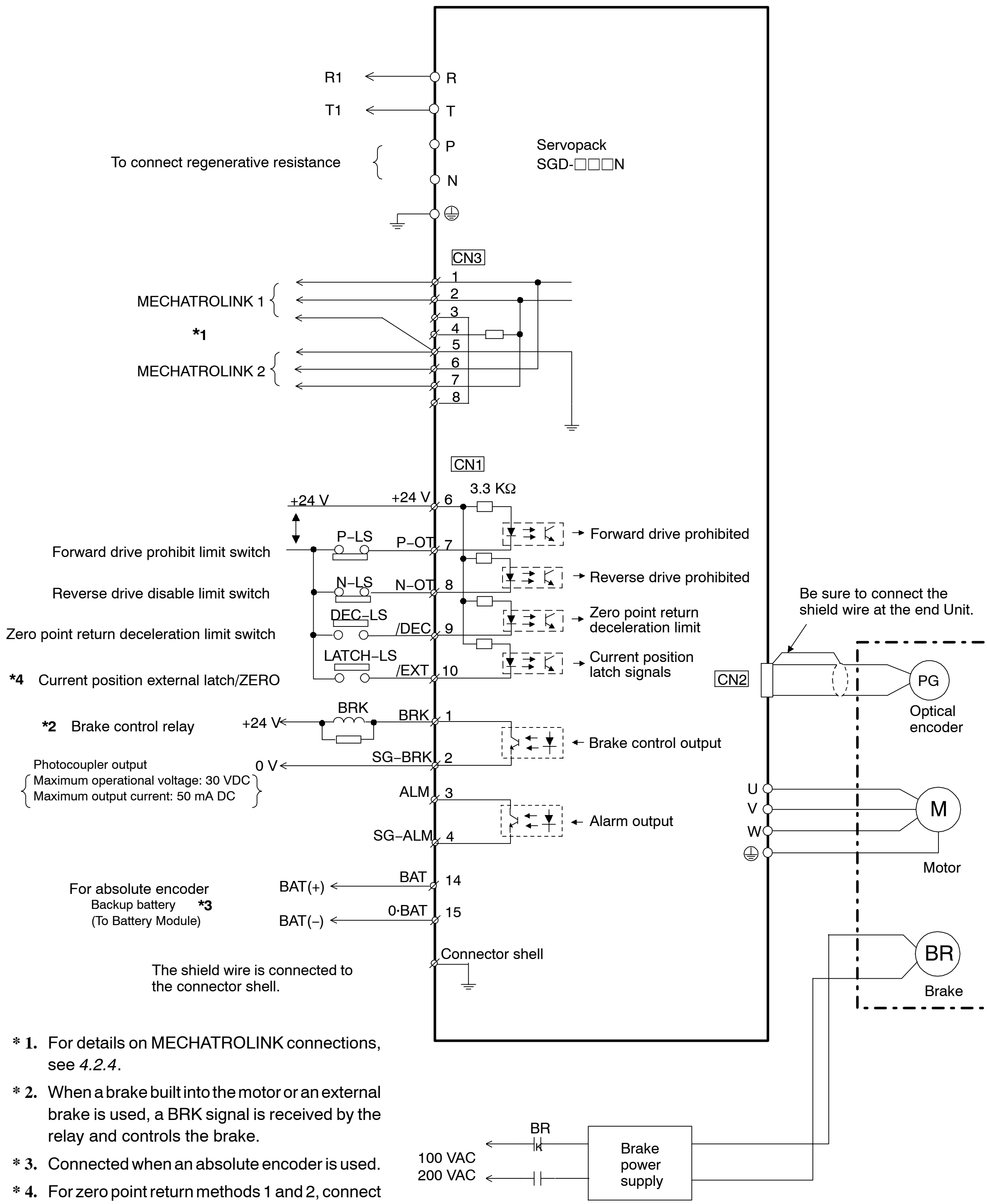
JEPMC-W5410-□□

Connection Example of the I/O Unit Output Connector



4.2.5 Connections to the Servopack and Motor

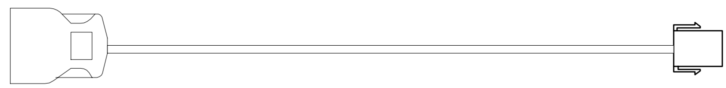
■ Connections to the SGD-□□□N Servopack



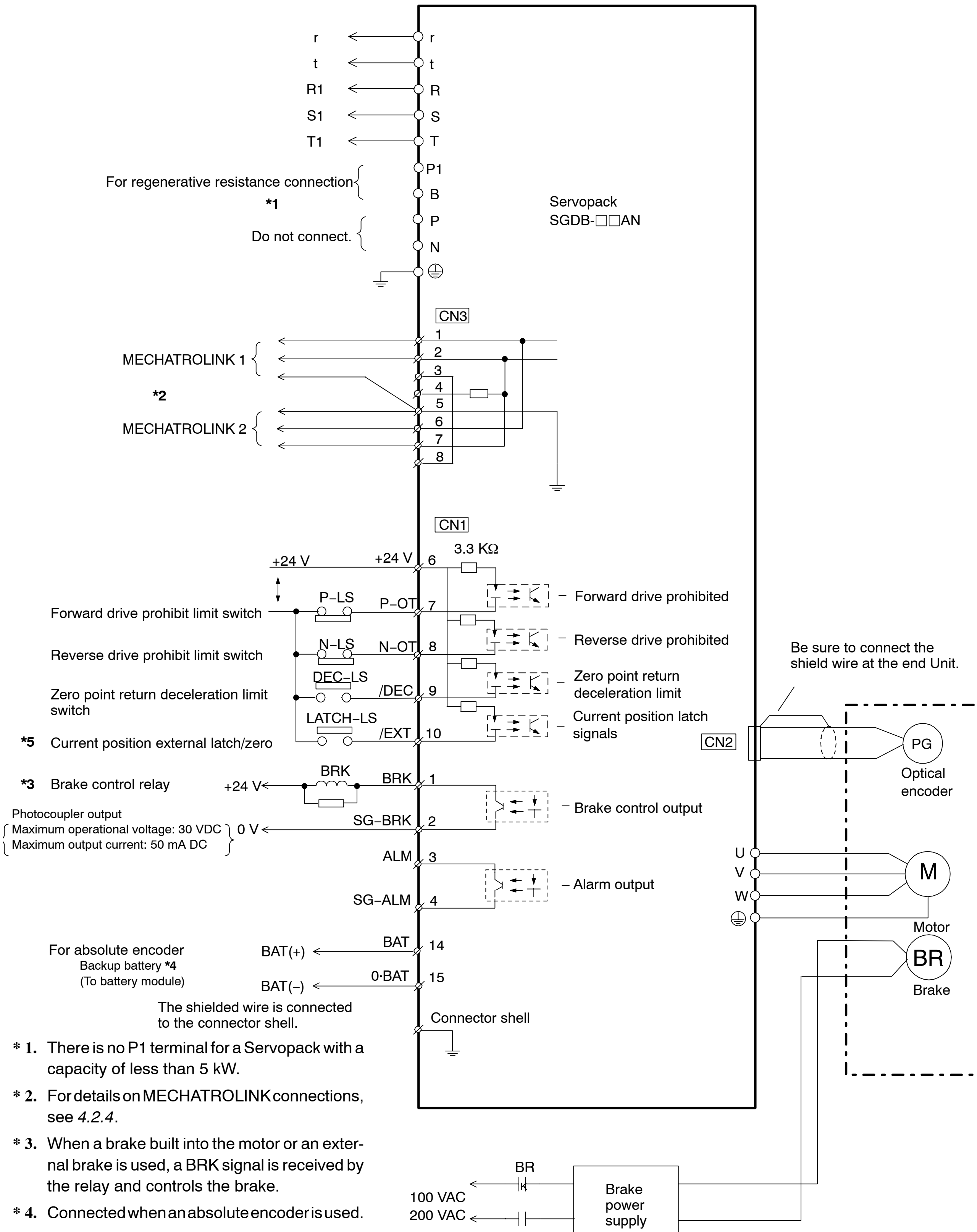
* **1.** For details on MECHATROLINK connections, see *4.2.4*.
* **2.** When a brake built into the motor or an external brake is used, a BRK signal is received by the relay and controls the brake.
* **3.** Connected when an absolute encoder is used.
* **4.** For zero point return methods 1 and 2, connect ZERO signal (zero point LS) to /EXT signal.

Servo Cable Models

Model: DP9320081-1, etc.


Model: DP9320089-1, etc.


■ Connections to the SGDB-□□AN Servopack



* **1.** There is no P1 terminal for a Servopack with a capacity of less than 5 kW.
* **2.** For details on MECHATROLINK connections, see *4.2.4*.
* **3.** When a brake built into the motor or an external brake is used, a BRK signal is received by the relay and controls the brake.
* **4.** Connected when an absolute encoder is used.
* **5.** For Zero point return methods 1 and 2, connect ZERO signal (Zero point LS) to /EXT signal.

4.3 System Startup Methods

This section explains the procedure when the Test Unit is used for positioning control. Consult the relevant reference manuals for the various procedures required. Details of the machine system design have been omitted here.

4.3.1 Overview of the Startup Procedure

The system startup procedure is as follows:

1. Prepare the equipment to be used.

Refer to *4.3.2 Test Unit Configuration* and *4.3.3 Equipment Preparations*.

2. Mount the Units.

Mount the MC Unit and the Expansion I/O Unit.

3. Connect and wire the system.

Connect the Programming Device, and wire the external I/O signals, servomotor, and Servopack.

4. Start the Programming Device.

Prepare the system definitions, such as the Module definitions, group definitions, and scan time settings.

5. Set, save, and transfer parameters.

Use the Programming Device to set the machine system parameters for the Test Unit.

6. Create, save, and transfer a motion program.

Use the Programming Device to create the motion program for testing.

7. Transfer definitions, parameters, and programs

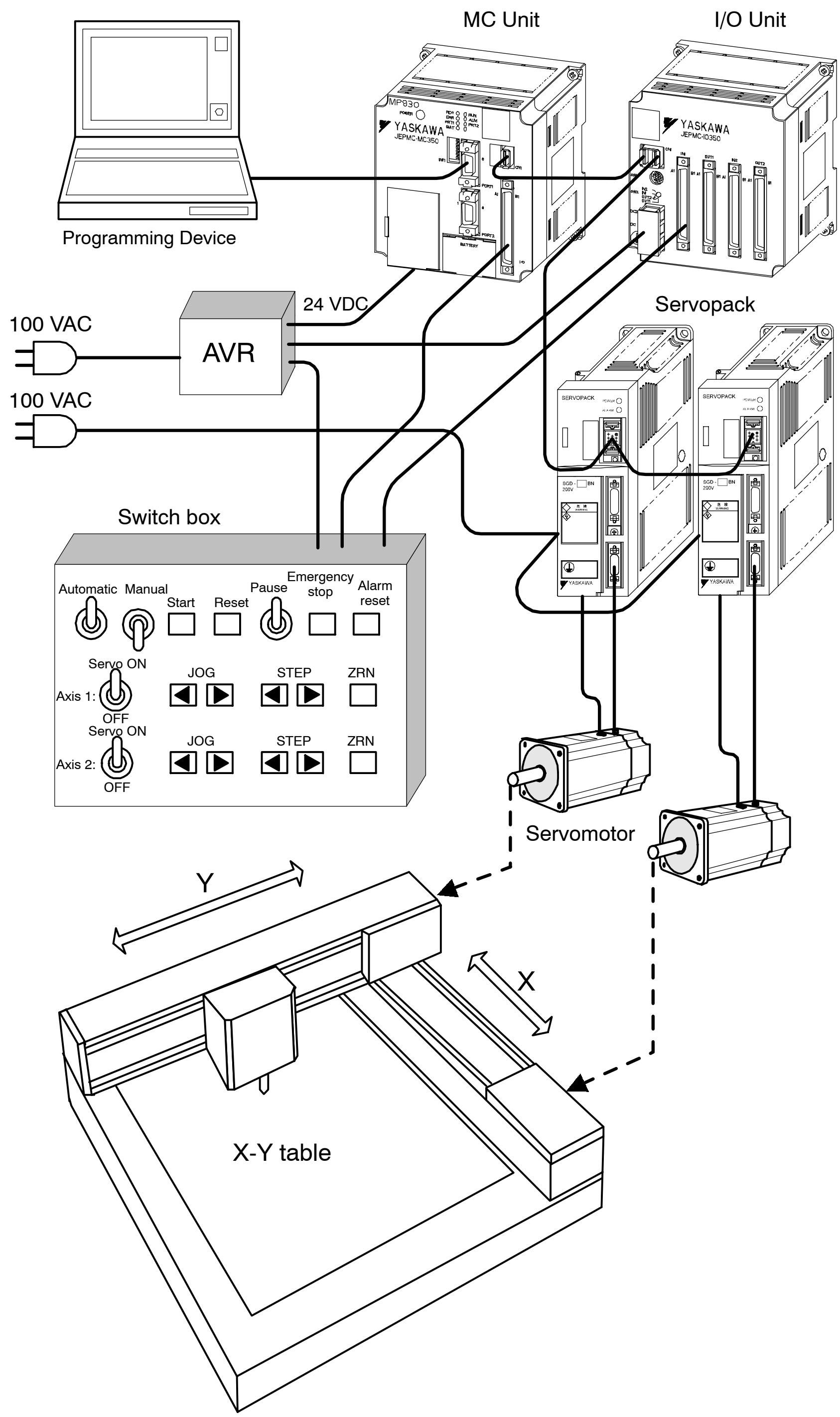
8. Check operation.

Execute the program and check the test operation.

4.3.2 Test Unit Configuration

The Test Unit is a simple Unit for explaining in simple terms the MP930 system startup. This Unit is different from the one that is used for actual applications.

The following illustration shows the Test Unit configuration.



IMPORTANT

Because this Unit is used for testing, there is no emergency stop circuit, and no servo amp power OFF circuit for an overtravel. For actual applications, be sure to insert the correct emergency stop circuits.

4.3.3 Equipment Preparations

Prepare the equipment shown in the following tables.

■ Controller-related Equipment

Name	Model
MC Unit	JEPMC-MC350
I/O Unit	JEPMC-IO350
MECHATROLINK Cables	JEPMC-W6000-A3, JEPMC-W6010-01, MR Connector
Servopack Connector Kit **MR connector (8-pin, female)**	DE9411357
I/O Signal Cable	JEPMC-W5410-05

■ Servo-related Equipment

Name	Model
Servopack	SGD-01BN
Servomotor	SGM-01B312
Motor Cable	DP9320081-1
PG Cable	DP9320089-1

■ Programming Device-related Equipment

Name	Model
Computer	Windows 95, Windows NT, or compatible
Software	CP-717
MEMOBUS Cable	JEPMC-W5311-03

■ Other Required Equipment

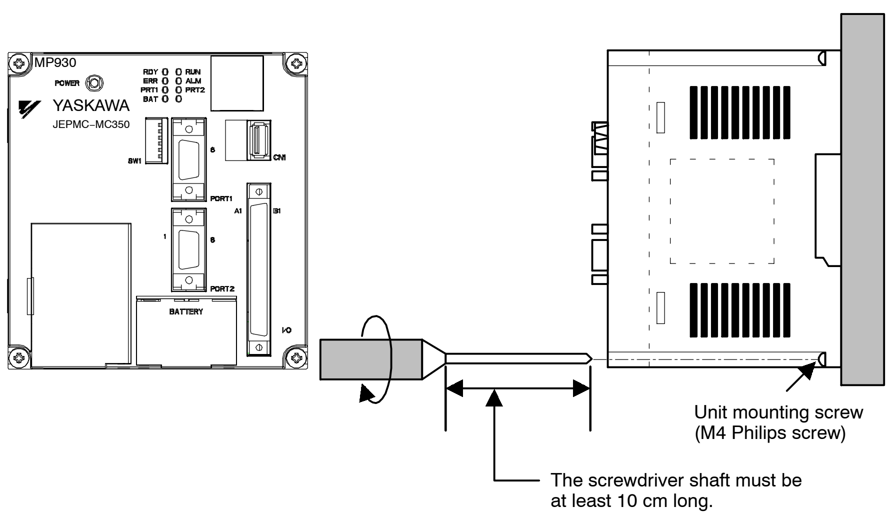
Name	Model
Switch box	–
24-VDC power supply (AVR)	–
No-fuse breaker	–
Switches	–
Wiring material	–

4.3.4 Mounting the Units

■ Mounting with Screws

Use the method shown in the following illustration to mount the MC Unit.

Be sure to securely tighten the Unit mounting screws (four places) to secure the MC350 Unit to the mounting surface.

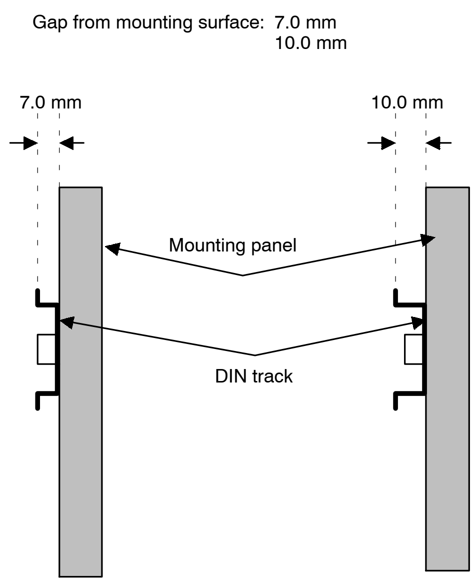


4

■ Mounting to DIN Track

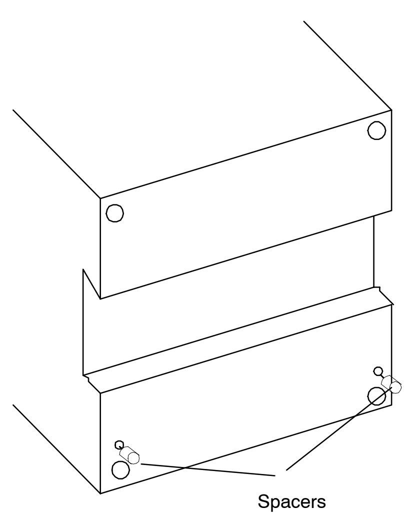
Before Mounting on the DIN Track

As shown in the following illustration, there are various types of DIN track, depending on the gap from the mounting surface.



- Mounting the MP930 Units on DIN Track with a 10-mm Gap

  To protect the MP930 from vibration, mount spacers at the bottom rear of the MP930, as shown in the illustration. The mounting parts, such as spacers, are sold separately.



IMPORTANT

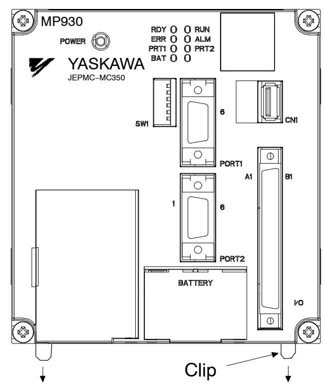
Parts required for mounting the Unit on the DIN track are sold separately, and are not provided with the Unit. Purchase the following part:

- Model: JEPMC-OP300

Mounting Procedure

The mounting procedure is as follows:

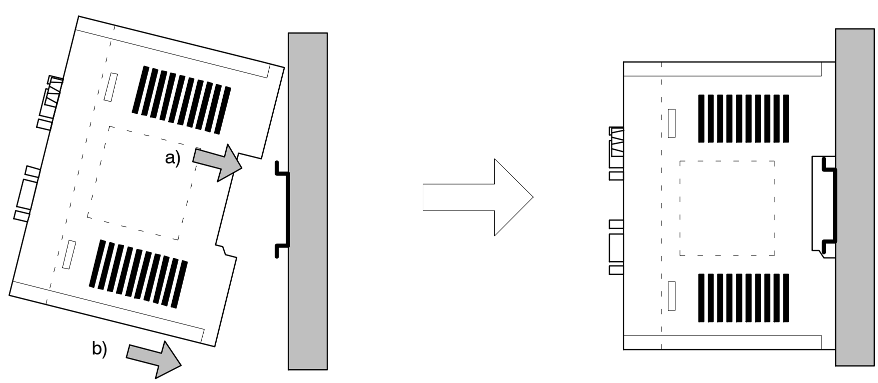
1. Release the mounting clips.
   Pull down the DIN track mounting clips to release them.



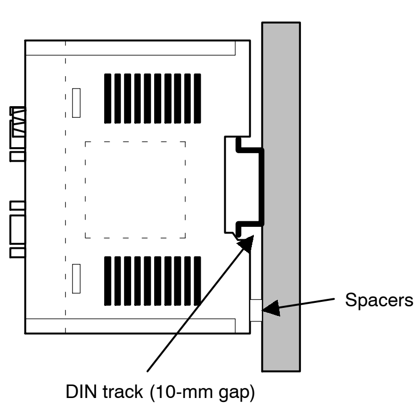
2. Mount the Unit on the DIN track.

   a) Hook the MP930 Unit on the DIN track.

   b) Push the Unit inward at the bottom until the MP930 comes into contact with the mounting surface.

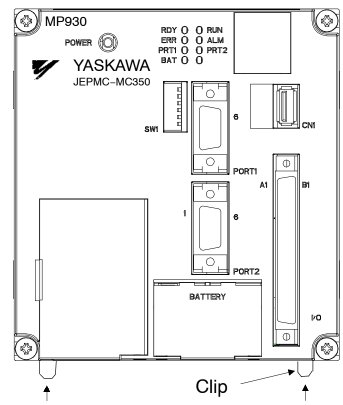


When a DIN track with a 10-mm gap is used, mount spacers on the mounting surface.

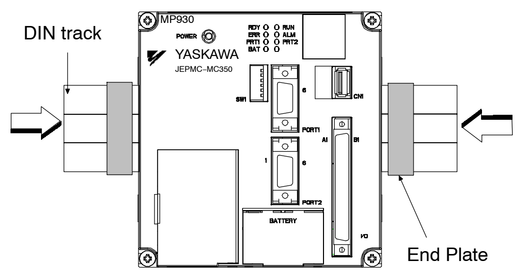


4

3. Lock the mounting clips.
Push in the DIN track mounting clips and lock the clips.



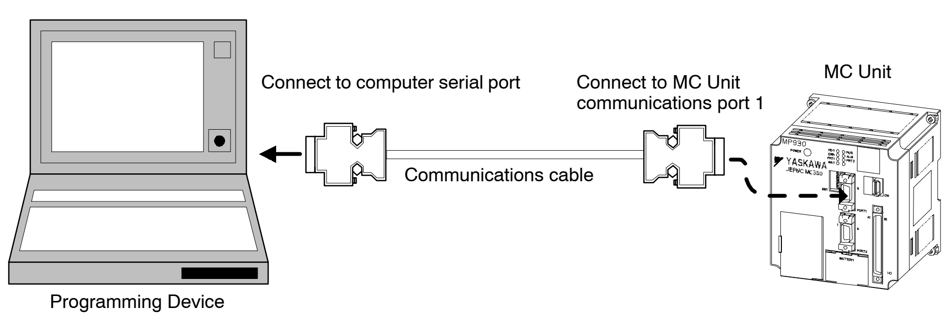
4. Attach the End Plate.
Attach End Plates on the DIN track on both sides of the MP930 Unit to secure it in place.



4.3.5 Connecting the Devices

■ Connecting the Programming Device

The following illustration shows the method of connecting the Programming Device and the MC Unit.



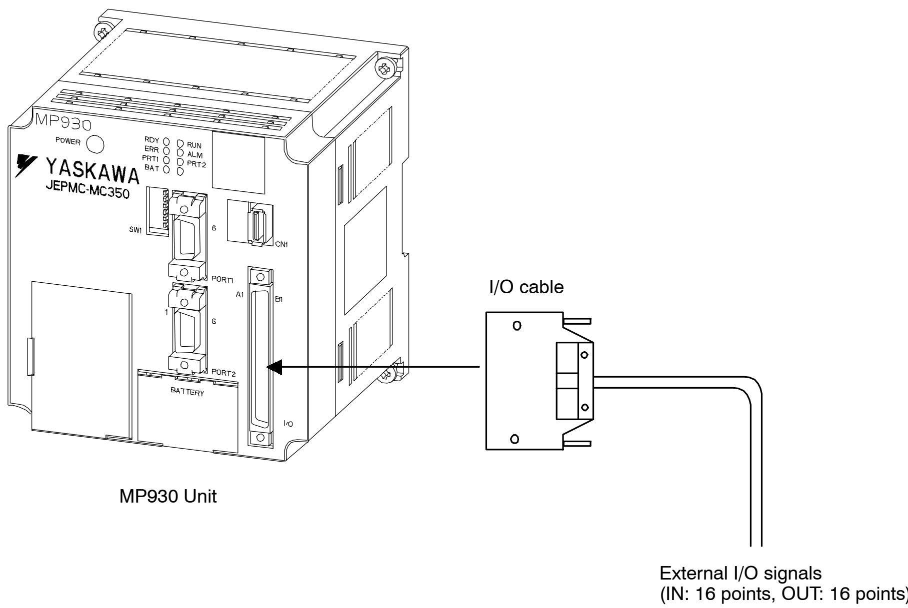
When connecting the Programming Device and the MC Unit communications port, use the following cables.

Figure 4.1 MEMOBUS Communications Cable Models

Cable Length	Model
2.5 m	JEPMC-W5311-03
15 m	JEPMC-W5311-15

■ Local I/O Connector Wiring

The following illustration shows the method of connecting the external I/O signals and the MC Unit local I/O connector.



When connecting the external I/O signals and the MC Unit local I/O connector, use the following cables.

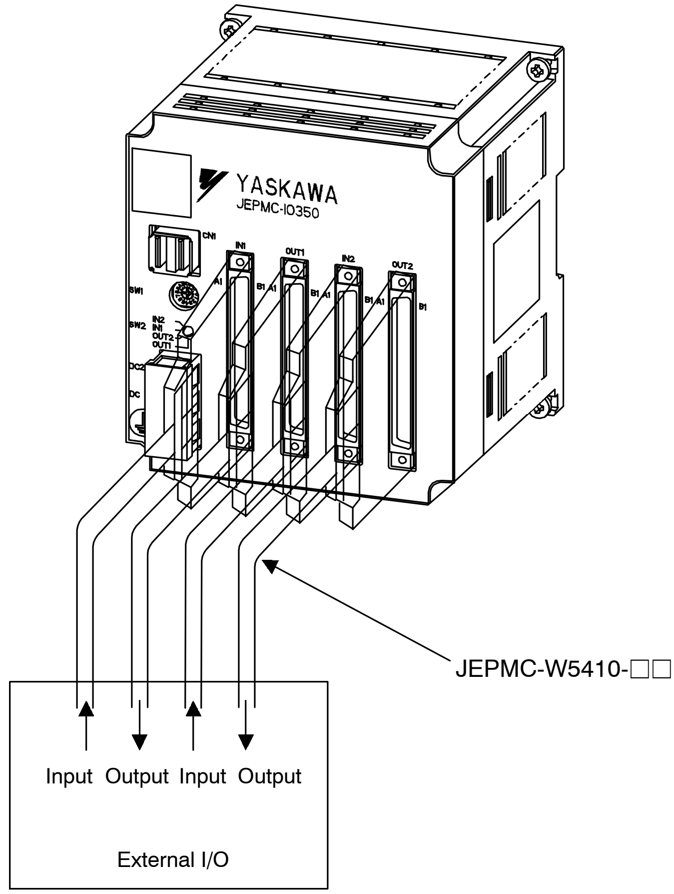
Table 4.3 I/O Cable Models

Cable Length	Model
0.5 m	JEPMC-W5410-05
1 m	JEPMC-W5410-10
3 m	JEPMC-W5410-30

INFO See *MC Unit I/O Cables* in *4.2.4 Connector Pin Layout and I/O Circuits* for the pin layout of the cables between the MC Unit I/O connectors and the external I/O signal lines.

4

■ Remote I/O Connector Wiring



When connecting the external I/O signals and the MC Unit remote I/O connectors, use the following cables.

Table 4.4 I/O Cable Models

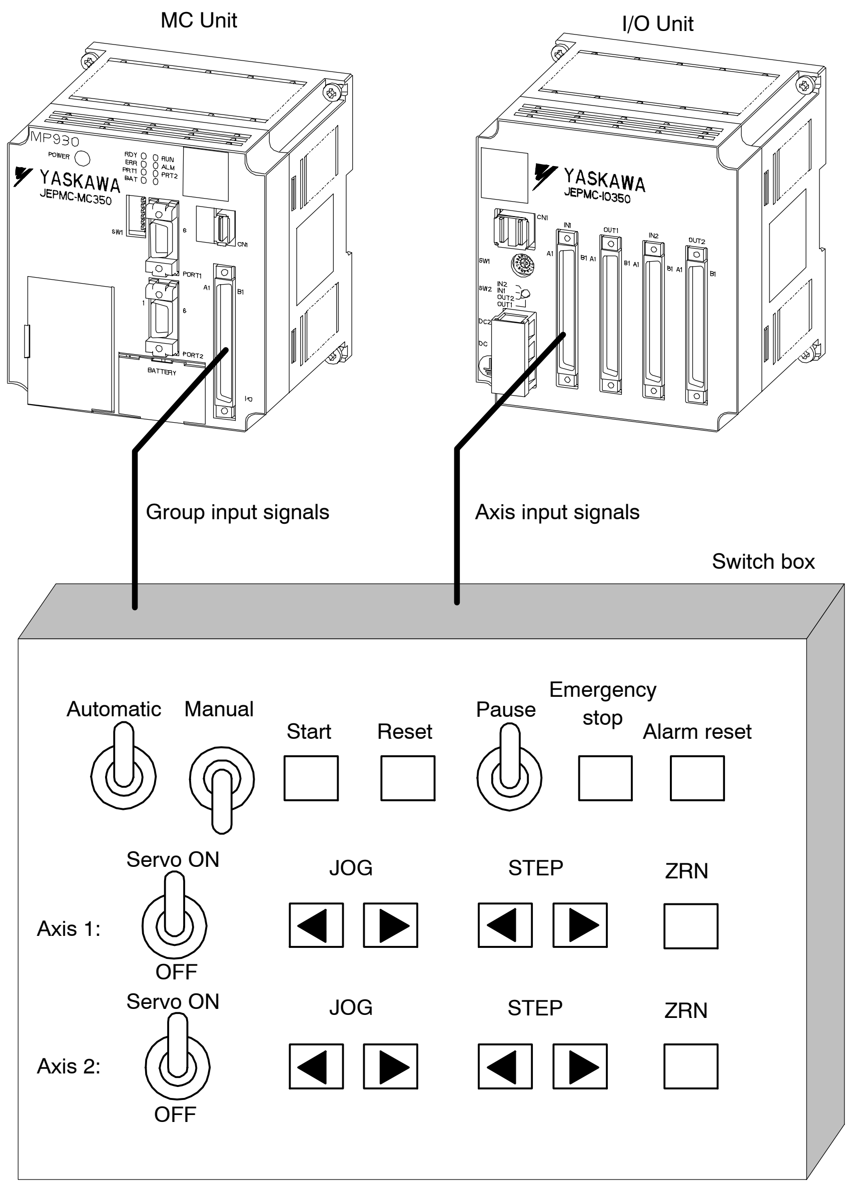
Cable Length	Model
0.5 m	JEPMC-W5410-05
1 m	JEPMC-W5410-10
3 m	JEPMC-W5410-30

INFO See *I/O Unit Input Cables* and *I/O Unit Output Cables* in *4.2.4 Connector Pin Layout and I/O Circuits* for the pin layout of the cables between the MC Unit I/O connectors and the external I/O signal lines.

■ Connecting the Switch Box

The switch box used by the ladder logic program that is automatically generated on the Group Definition Screen is connected as shown in the following illustration.

- Group input signals: MC Unit I/O connector
- Axis input signals: I/O Unit IN1 connector



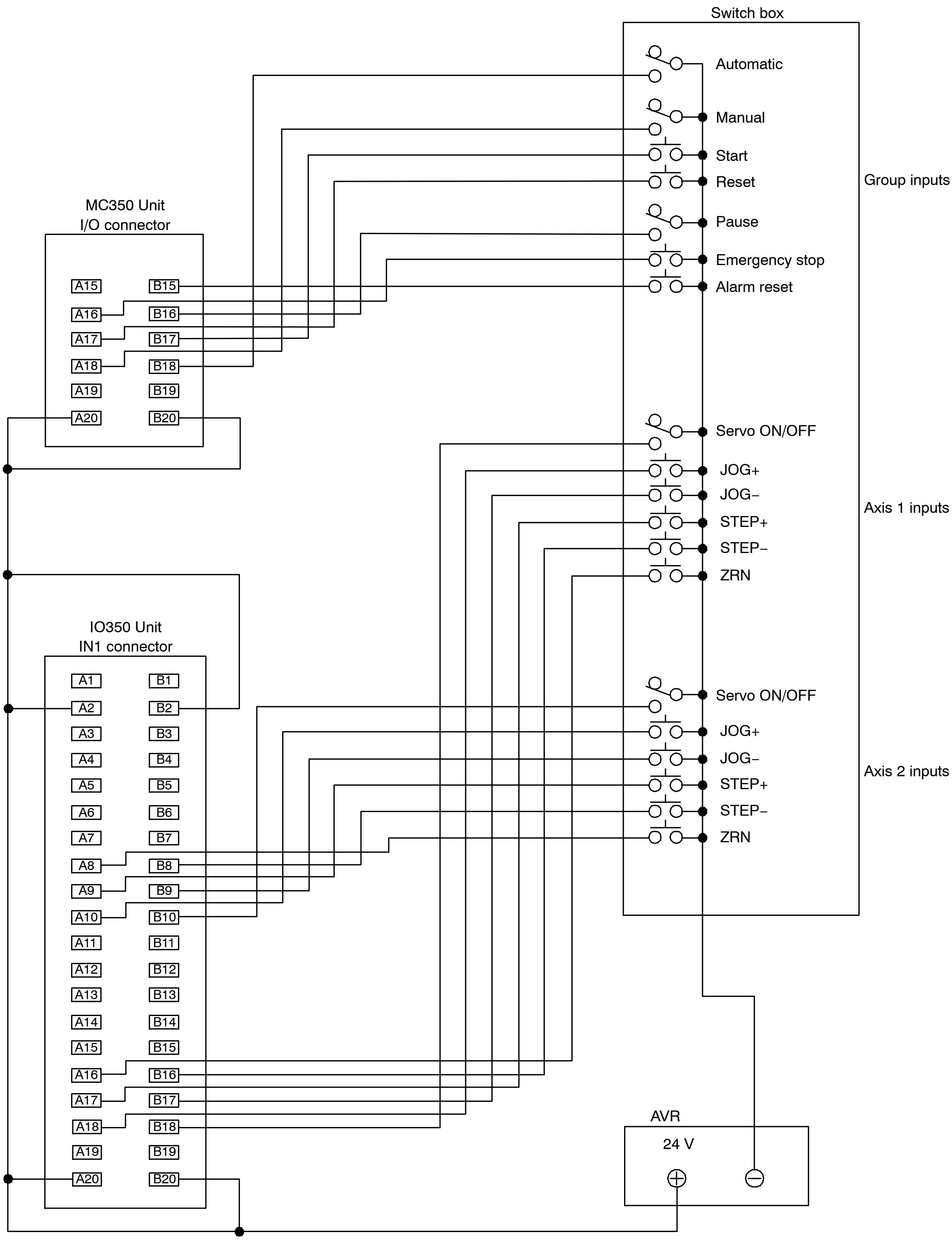
Switch Box External Signal Allocation

The switch box signals are allocated as shown in the following table.

Group Input Signals		Axis 1 Input Signals		Axis 2 Input Signals	
IB00000	Automatic mode	**IB00100**	Servo ON	**IB00110**	Servo ON
IB00001	Manual mode	**IB00101**	JOG+	**IB00111**	JOG+
IB00002	Start	**IB00102**	JOG−	**IB00112**	JOG−
IB00003	Reset	**IB00103**	STEP+	**IB00113**	STEP+
IB00004	Pause	**IB00104**	STEP−	**IB00114**	STEP−
IB00005	Emergency stop	**IB00105**	ZRN	**IB00115**	ZRN
IB00006	Alarm reset	**IB00106**	–	**IB00116**	–
IB00007	–	**IB00107**	–	**IB00117**	–
IB00008	–				
IB00009	–				
IB0000A	–				
IB0000B	–				

Switch Box Connection Diagram

The following illustration shows the switch box connection diagram.



■ Connection of Servopack and Servomotor

Use the special cable and encoder cable to connect the Servopack and Servomotor.

SGD-□□□N Servopack

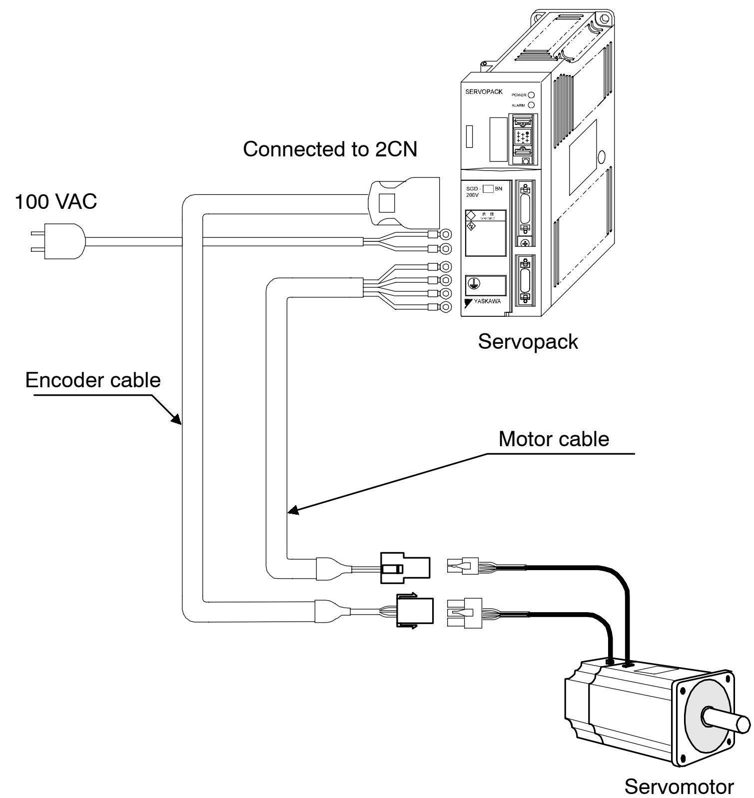


Table 4.5 Motor Cables

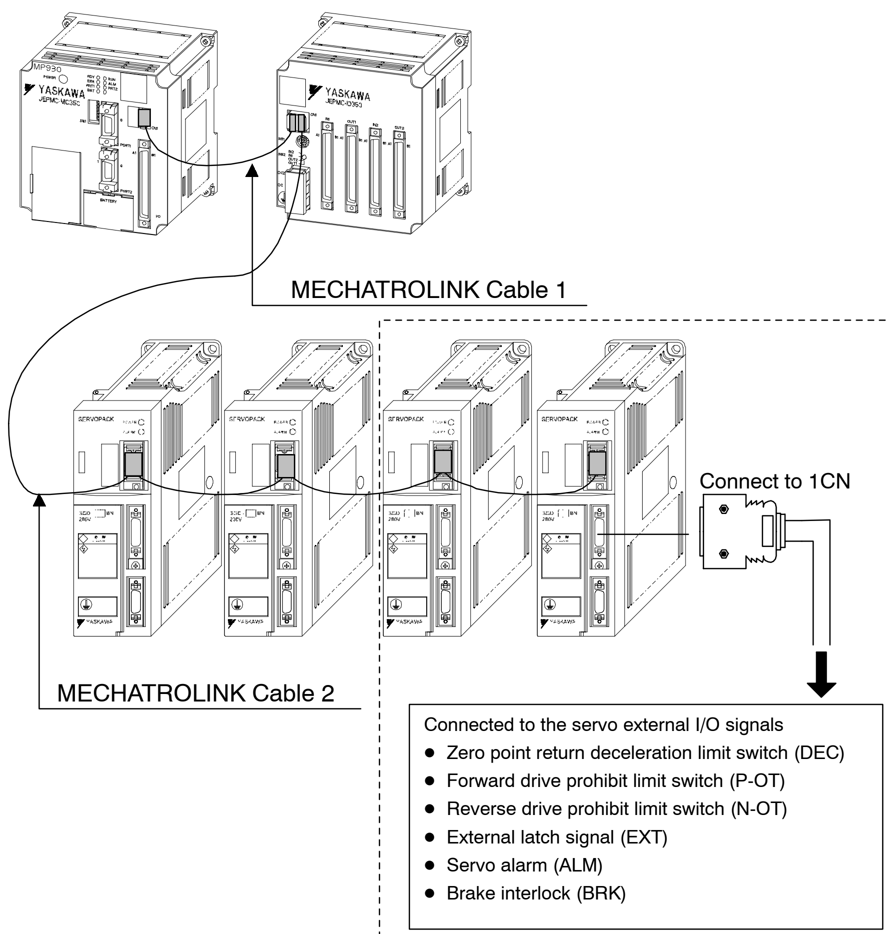
Cable Length	Model	Cable Length	Model
3 m	DP9320081-1	3 m	DP9320083-1
5 m	DP9320081-2	5 m	DP9320083-2
10 m	DP9320081-3	10 m	DP9320083-3
15 m	DP9320081-4	15 m	DP9320083-4
20 m	DP9320081-5	20 m	DP9320083-5

Table 4.6 Encoder Cables

Cable Length	Model
3 m	DP9320089-1
5 m	DP9320089-2
10 m	DP9320089-3
15 m	DP9320089-4
20 m	DP9320089-5

Connection of MECHATROLINK Cables

The MC Unit and I/O Unit, and the I/O Unit and Servopack, are connected using the MECHATROLINK Cables shown in the following illustration.



Note With the Test Unit, the signals enclosed by the dotted lines are not connected.

Table 4.7 MECHATROLINK Cable 1

Cable Length	Model
0.3 m	JEPMC-W6000-A3

Table 4.8 MECHATROLINK Cable 2

Cable Length	Model
1 m	JEPMC-W6010-01
3 m	JEPMC-W6010-03
5 m	JEPMC-W6010-05

Table 4.9 1CN Cable and 1CN Terminal Block

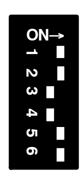
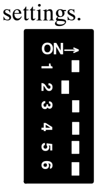
1CN Cable Length	Model	1CN Terminal Block Cable Length	Model
1 m	DE9411355	0.5 m	JUSP-TA26P

IMPORTANT

MECHATROLINK Cable 2 (JEPMC-W6010-□□) has loose wires on one end and a USB connector on the other. The customer must prepare this cable using an MR connector and wiring material. For the internal connections, see *MECHATROLINK Cables* in *4.2.4 Connector Pin Layout and I/O Circuits*.

■ Memory Initialization

After connecting the equipment, use the following procedure to initialize the memory of the MC Unit.

1	2	3	4	5
Turn OFF the MP930 power.	Turn ON DIP switch pins 3 and 4. 	Turn ON the power, and check that the RDY and RUN indicators flash (about 3 seconds).	Turn OFF the power, and return the DIP switch pins to their original settings. 	Turn ON the power again.

Always initialize memory if you remove the battery when the Unit power supply is turned OFF.

4.3.6 Starting the CP-717

This section explains the Units configuring the MP930, the I/O allocation module configuration definitions, and the methods of setting the group definitions for the number of axes and tasks. Be sure to set these when the system is first started up.

Make sure the CP-717 System Software is installed in advance.

The following is an overview of the CP-717 startup procedure.

1. Creation of an Order Folder
2. Creation of a PLC Folder
3. Offline Logon
4. Module Configuration Definitions
5. Group Definitions
6. Scan Time Settings

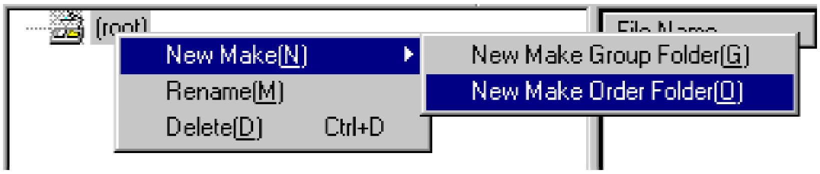
4

■ Creation of an Order Folder

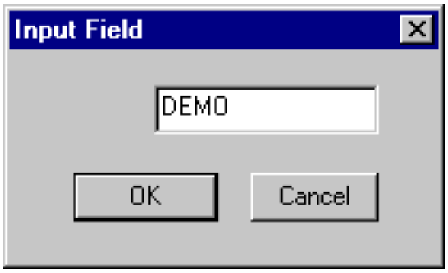
Start the CP-717, and create an order folder from the File Manager Screen.

- Example: File name: DEMO

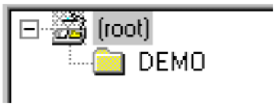
1. Point to ***root***, right click, and then click ***New Make (N) → New Make Order Folder (O)***.



2. In the dialog box, input the order folder name and click the **OK** button. The order folder name must be eight characters or less.



3. The new “DEMO” order folder will be created.



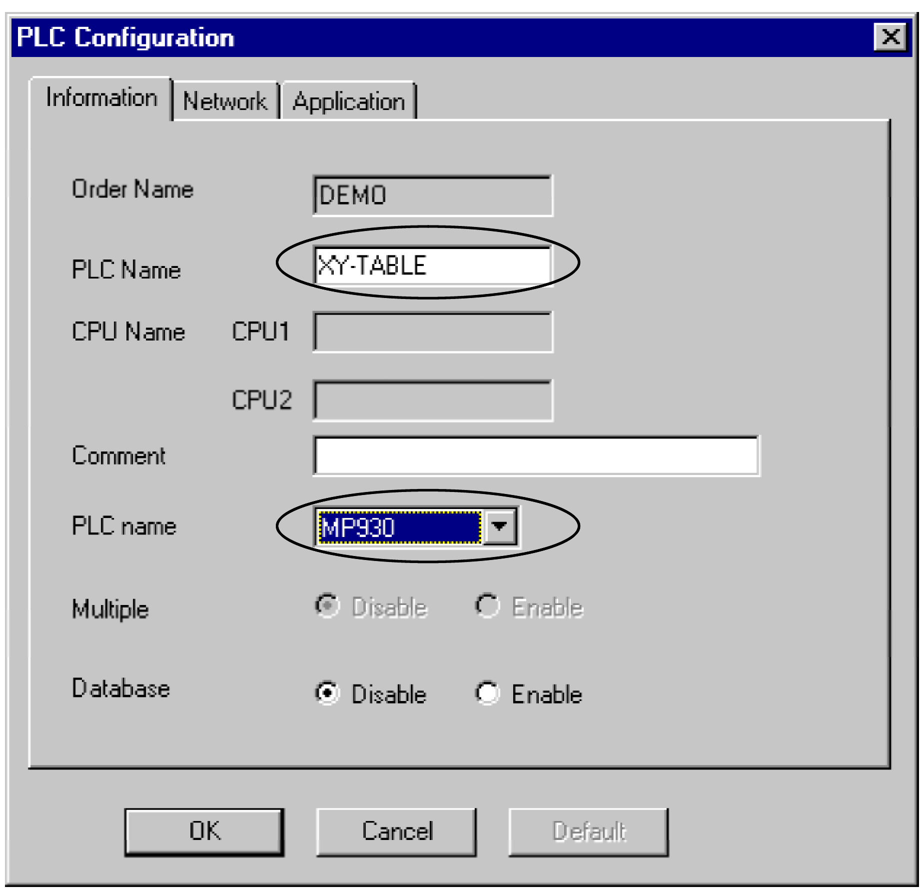
■ Creating a PLC Folder

Register the new PLC to be used to create the program.

- Example: PLC name: XY-TABLE
  Device name: MP930

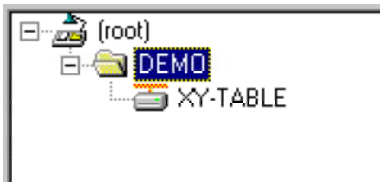
1. Point to the **DEMO** order folder, right click, and then click ***New Make (N)*** → ***PLC Folder (C)***.

2. In the PLC Configuration Window, set the "PLC Name" and "PLC name" (model), and click the **OK** button.



4

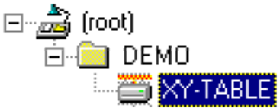
The new PLC folder will be created.



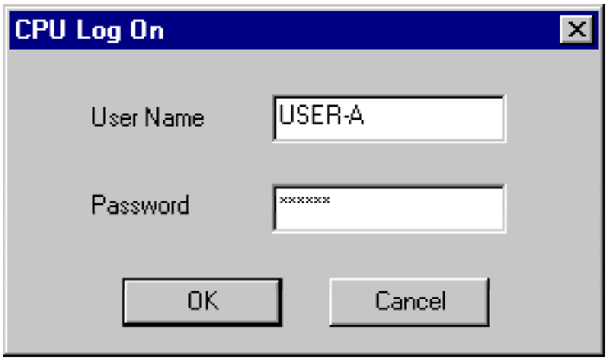
Offline Logon

When creating a PLC program or definition data, you must log on to the PLC.

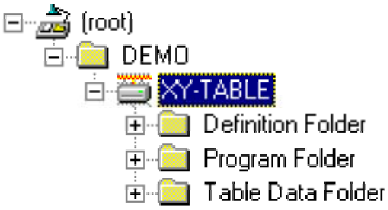
1. Double-click the ***XY-TABLE*** PLC folder.



2. Input the user name **USER-A** and password **USER-A**, and click the **OK** button.



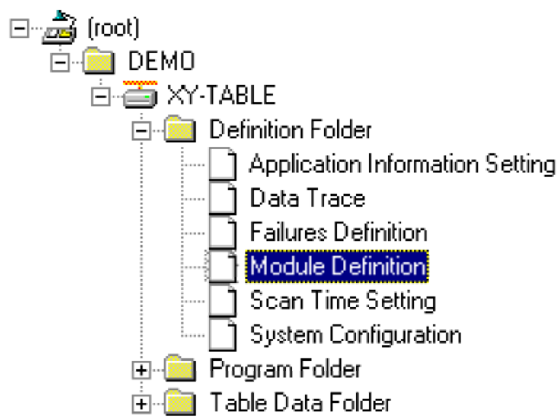
The PLC folder XY-TABLE definition, program, and table data folders will be displayed, and logon has been completed.



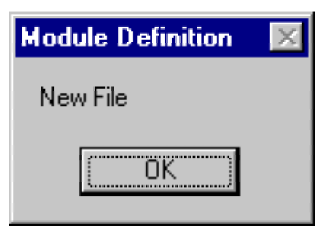
Module Definitions

Set the Servopack and I/O Module connected to the MP930 MC Module via the MECHATRO-LINK high-speed field network.

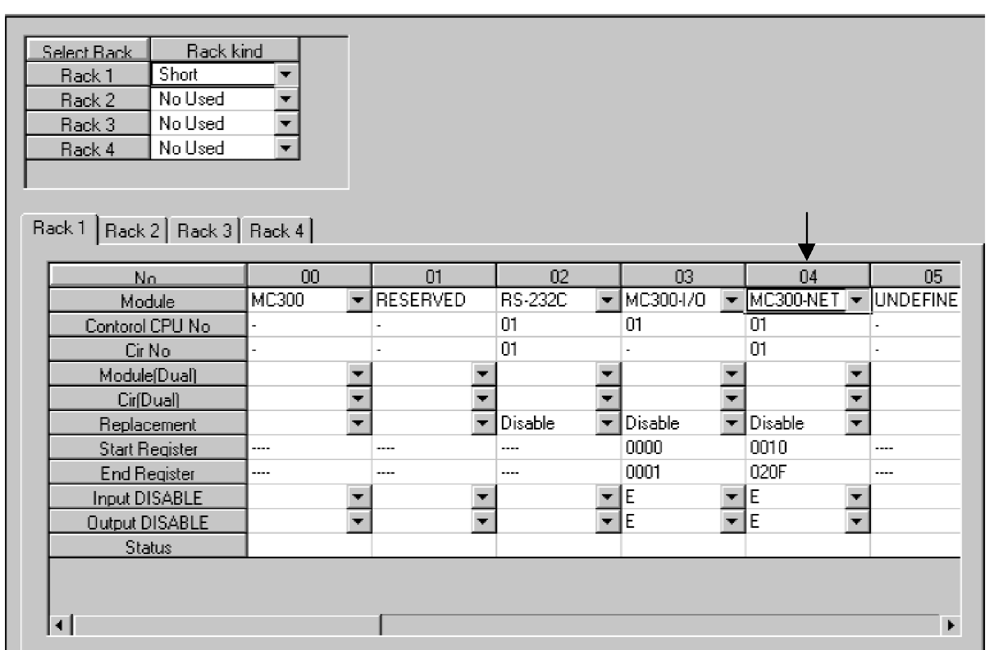
1. Double-click the ***Definition Folder*** under the ***XY-TABLE*** folder, and double-click ***Module Definition***.



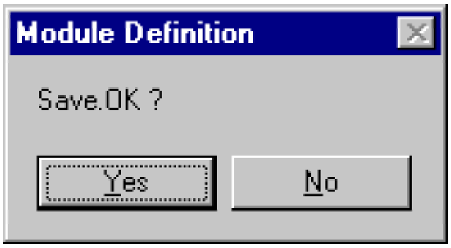
2. The Module Definition message box will be displayed. Click **OK**.



3. Double-click No. 04 on the Module Definition Screen.

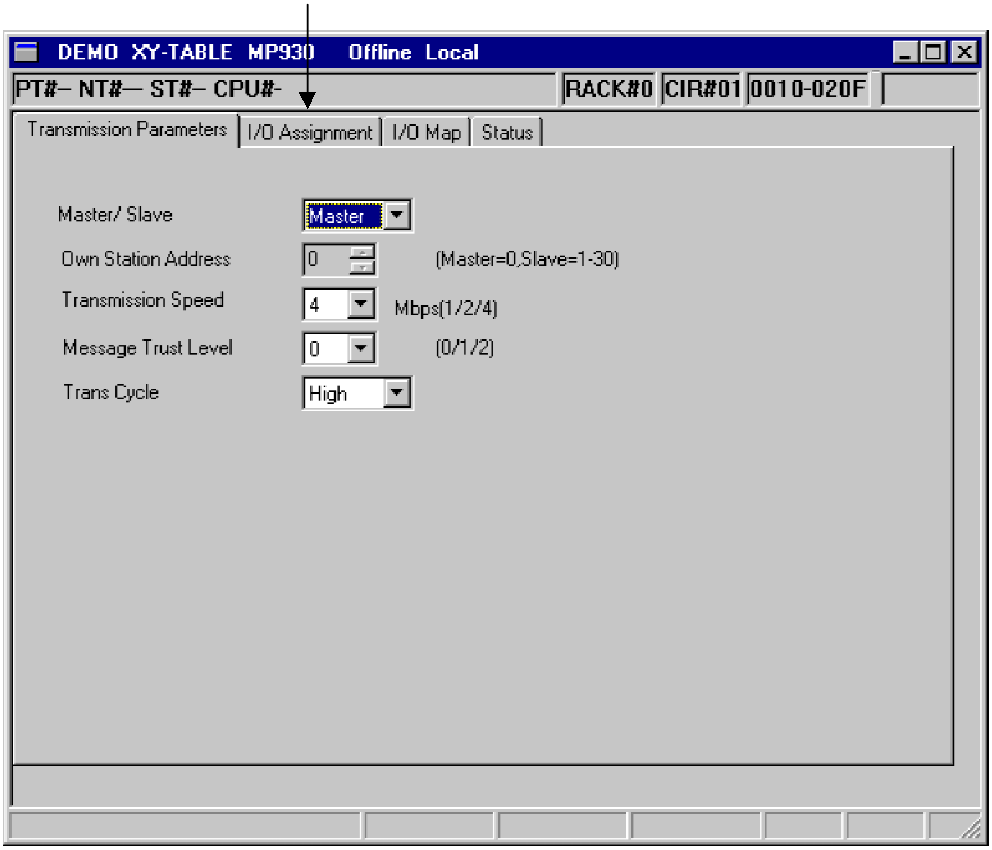


4. The Module Definition message box will be displayed. Click the **Yes** button.

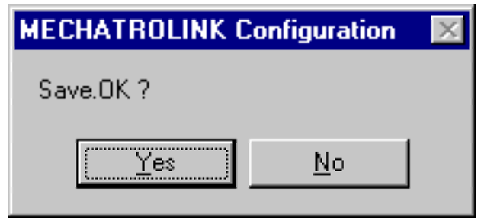


5. The MECHATROLINK Configuration message box will be displayed. Click OK.

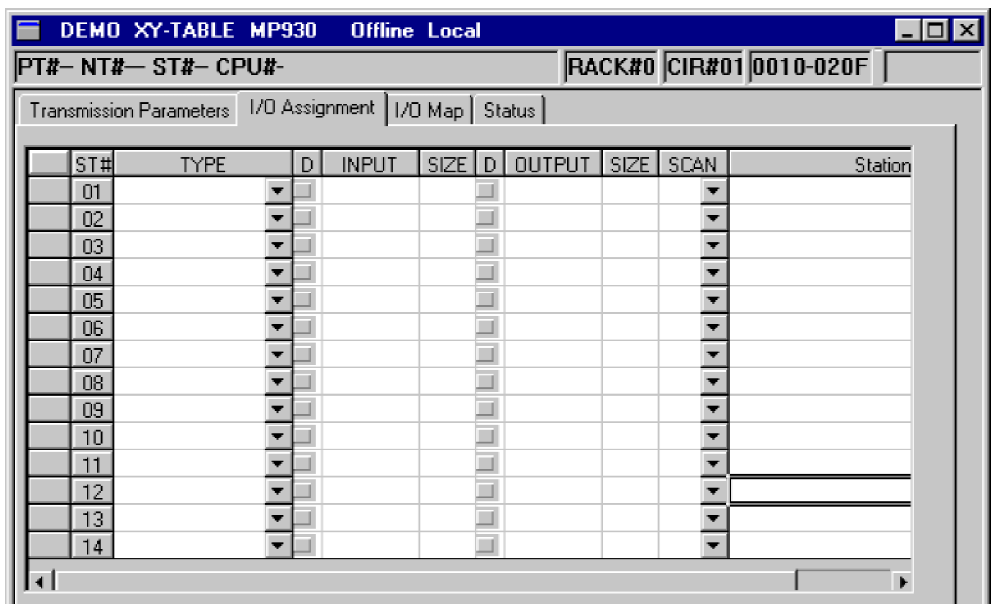
6. Double-click the ***I/O Assignment*** tab on the MECHATROLINK Definition Screen.



7. The CP-216 Transmission Definition message box will be displayed. Click the **Yes** button.

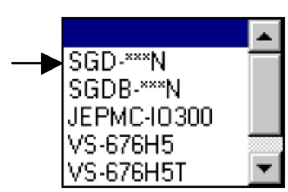


8. In the I/O Assignment tab, set the servo amp and I/O Unit connected to MECHATRO-LINK.

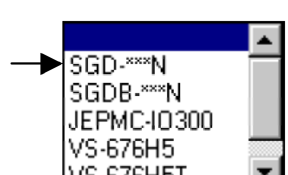


Use the following procedure to allocate to the I/O Unit:

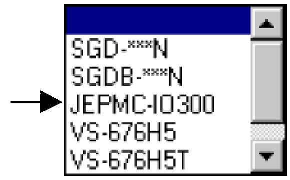
a) Allocate SGD-xxxN to ST#01.
Press the **down arrow** to the right of the ST#01 TYPE, and click ***SGD-xxxN***.



b) Allocate SGD-xxxN to ST#02.
Press the **down arrow** to the right of the ST#02 TYPE, and click ***SGD-xxxN***.

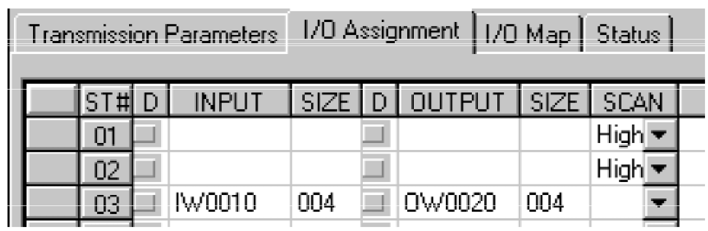


c) Allocate JEPMC-IO300 to ST#03.
Press the **down arrow** to the right of the ST#03 TYPE, and click ***JEPMC-IO300***.

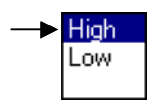


d) Set the I/O address and scan.
Set the JEPMC-IO300 input address, output address, and scan.

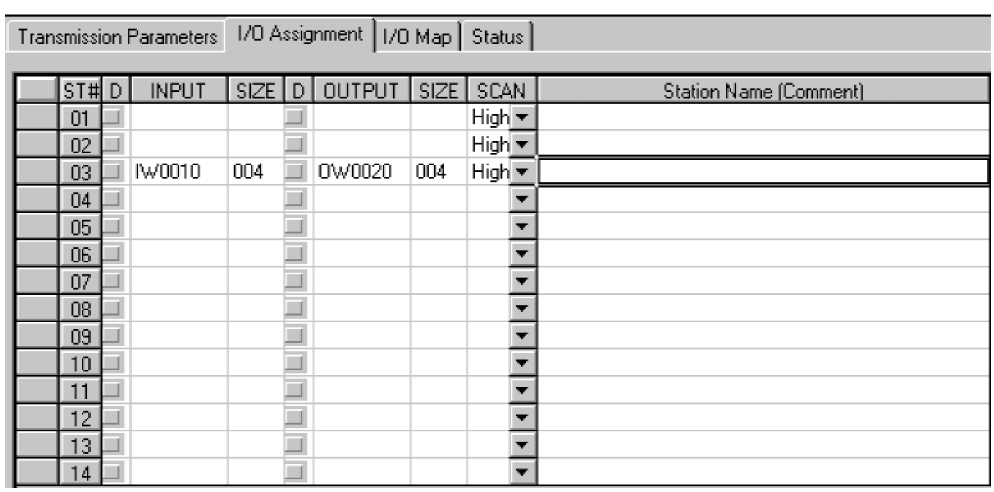
- Double-click the ***INPUT*** area, and then set **IW** and **10**.
- Double-click the ***OUTPUT*** area, and then set **IW** and **20.**



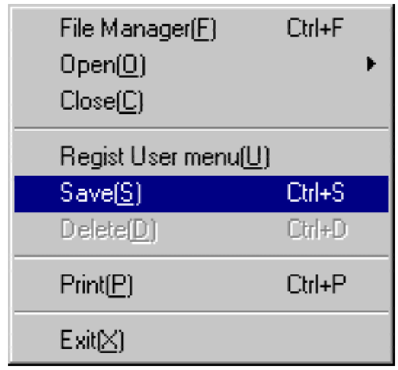
- Click the **down arrow** to the right of SCAN, and click ***High.***



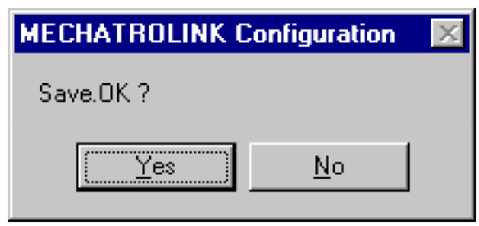
This completes the settings.



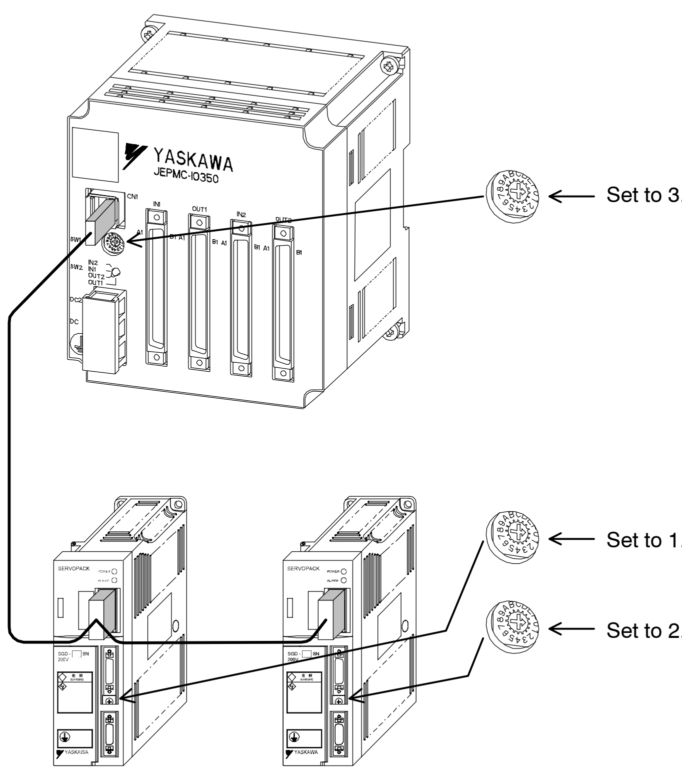
9. Click ***File (F)*** and then ***Save (S)*** from the File menu.



10. Click the **Yes** button. The definition data will be saved.



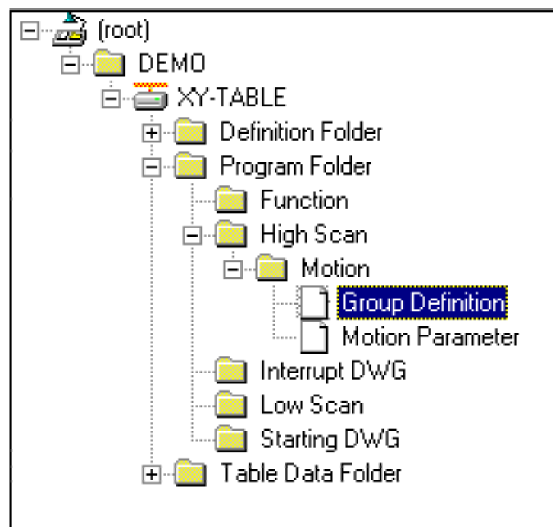
11. Set the station number of the Servopack and I/O Unit according to the settings on the ME-CHATROLINK Definition Screen.



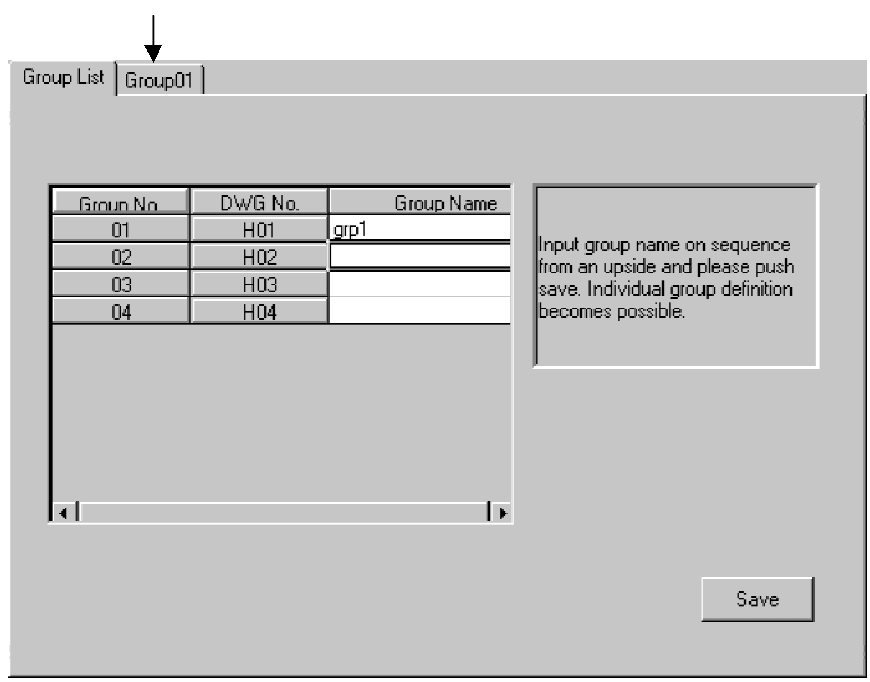
■ Group Definitions

Set the number of axes and tasks, and the axis names required for MP930 motion control.

1. On the File Manager Screen, scroll down in order of ***XY-TABLE*** → ***Program Folder*** → ***High-speed Drawing*** → ***Motion*** → ***Group Definition***, place the cursor on ***Group Definition***, and double-click.



2. Double-click the ***Group Name*** field under Group No. 01, and set **grpl**. Then click the **Save** button to register the group name. The registered Group Number tab will be added to the window.

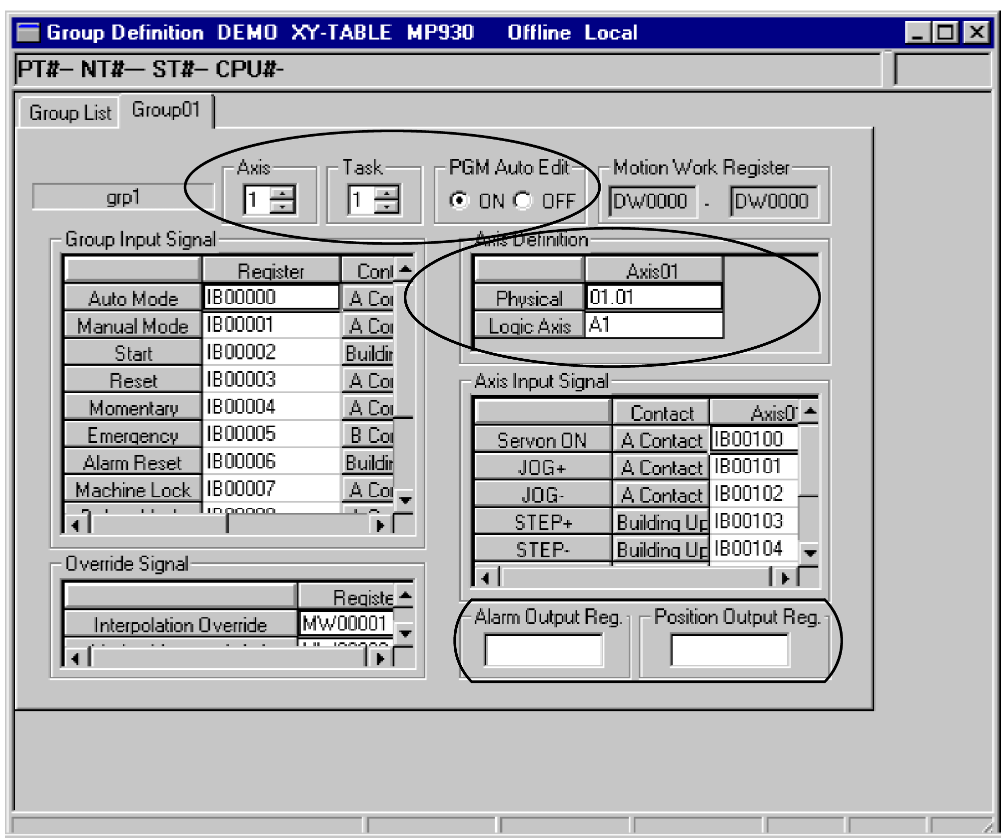


3. The Save Completed message box will be displayed. click the **OK** button.

4. Click the ***Group 01*** tab in the Group Definition window.

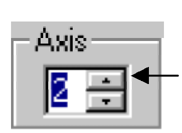
The Group 01 Group Definition Screen will be displayed.

5. Use the following procedure to change the circled parts of the Group 01 Group Definition Screen.



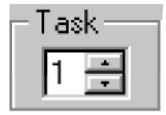
a) Number of Controlled Axes

Click the **up arrow** button and set the number of axes to 2.



b) Number of Tasks

Confirm that the number of tasks is 1.

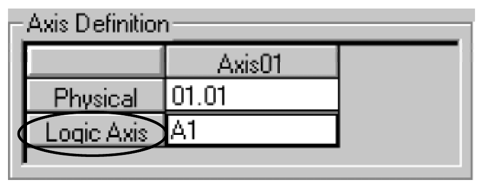


c) PMG Automatic Generation

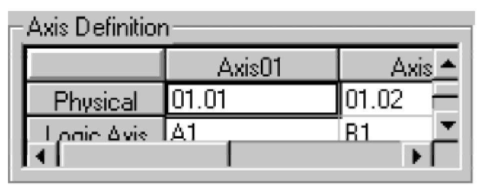
Confirm that PMG automatic generation is set to **Yes**. (The default setting is Yes.)

d) Axis Definition

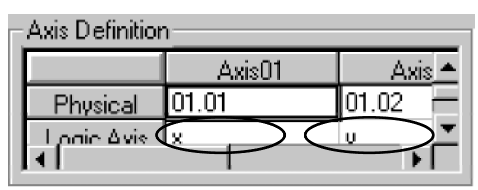
Click the ***Logical Axis*** name.



The two axes set in step a) will be displayed.

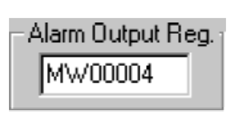


e) Click ***A1*** for Logical Axis Name axis 01, and input **X**.
Click ***B1*** for Logical Axis Name axis 02, and input **Y**.



f) Alarm Output Register

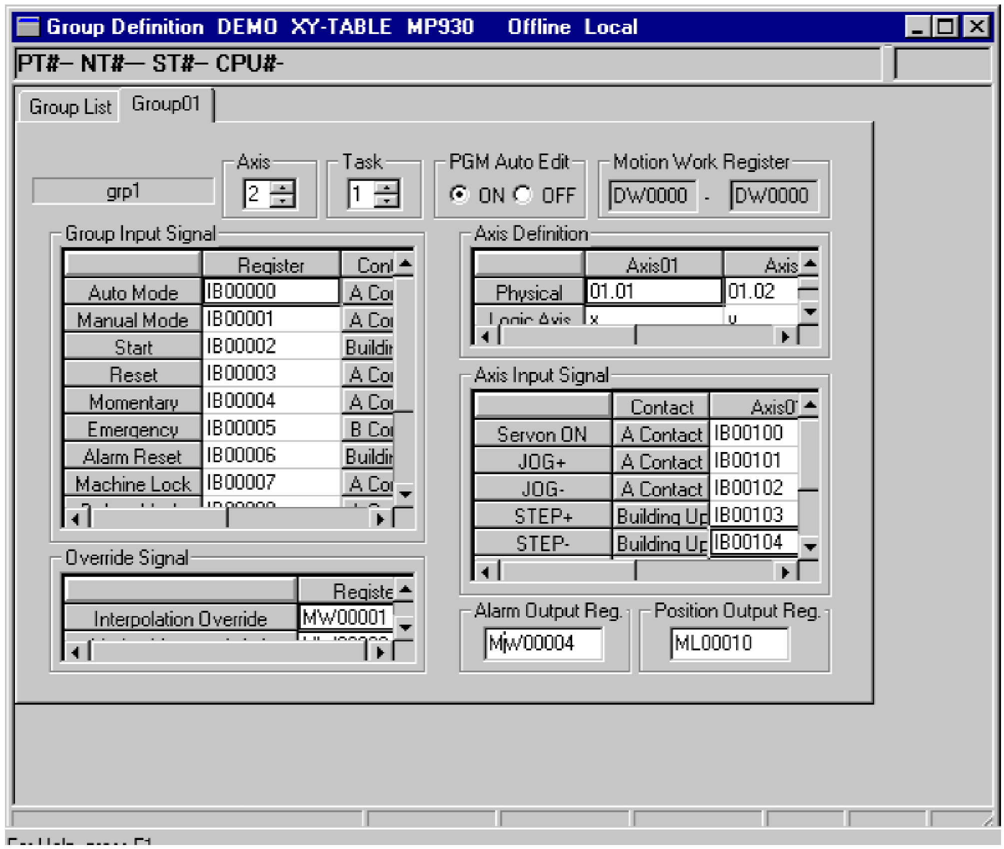
Click the ***Alarm Output Reg.*** field, and input **MW00004**.



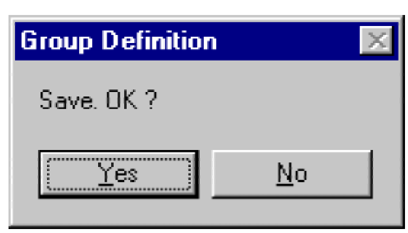
g) Position Output Register

Click the **Position Output Reg.** field, and input **ML00010**.

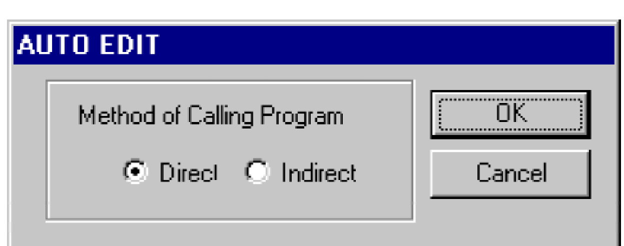
The Group Definition Screen will be as shown in the following illustration after the settings have been completed.



6. Click ***Save (S)*** from ***File (F)*** on the Group Definition menu. Click the **Yes** button in the Group Definition message box.



7. Click **OK** in the Auto Edit message box. For the methods of calling programs, Direct and Indirect, refer to *3.2.3 MOTION PROGRAM CALL Instruction (MSEE) of MP9□□ Machine Controller Ladder Programming User's Manual* (SIEZ-C887-1.2).

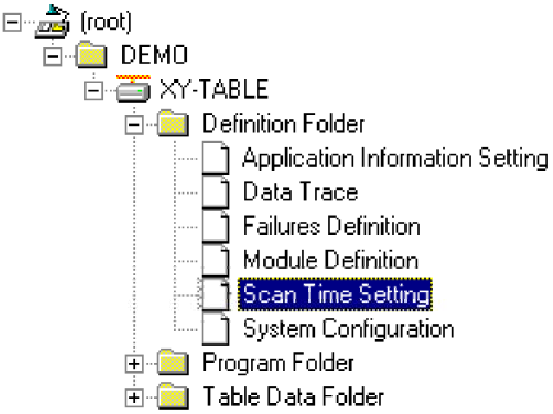


8. The Save Completed message box will be displayed, and the group definition settings have been completed. Click OK.

Scan Time Setting

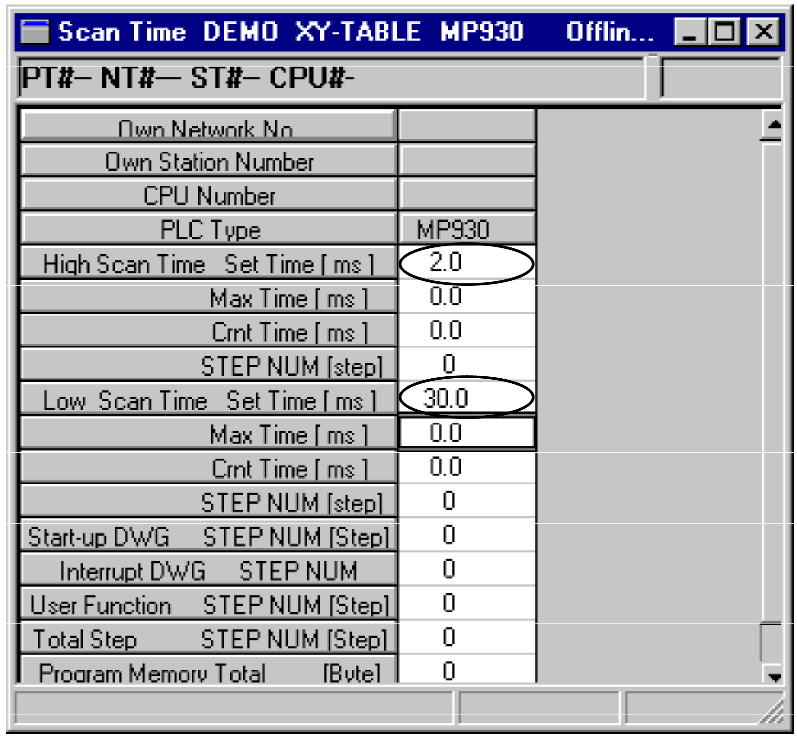
The MP930 PLC sets the cycle for executing user programs (high-speed drawings and low-speed drawings).

1. Click ***Definition Folder*** below ***XY-TABLE*** on the File Manager Screen, place the cursor on ***Scan Time Setting***, and double-click.

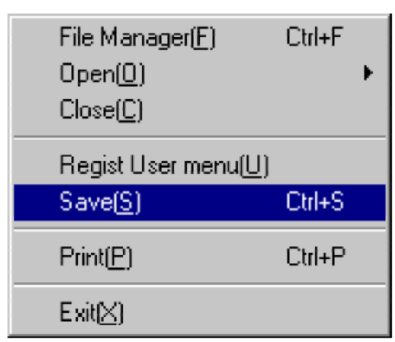


2. The Scan Time message box will be displayed. Click **OK**.

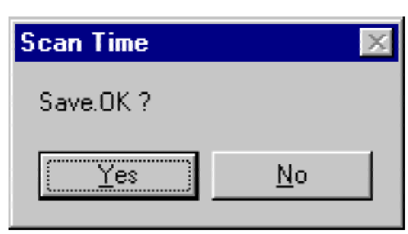
3. Set the **High Scan Time** to **2.0** ms and the **Low Scan Time** to **30** ms.



4. Select ***File (F)*** from the Scan Time menu, then click ***Save (S)***.

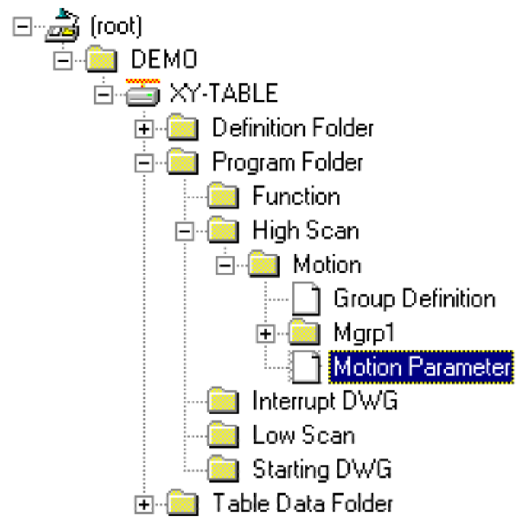


5. Click the **Yes** button in the Scan Time message box.



4.3.7 Motion Parameter Settings

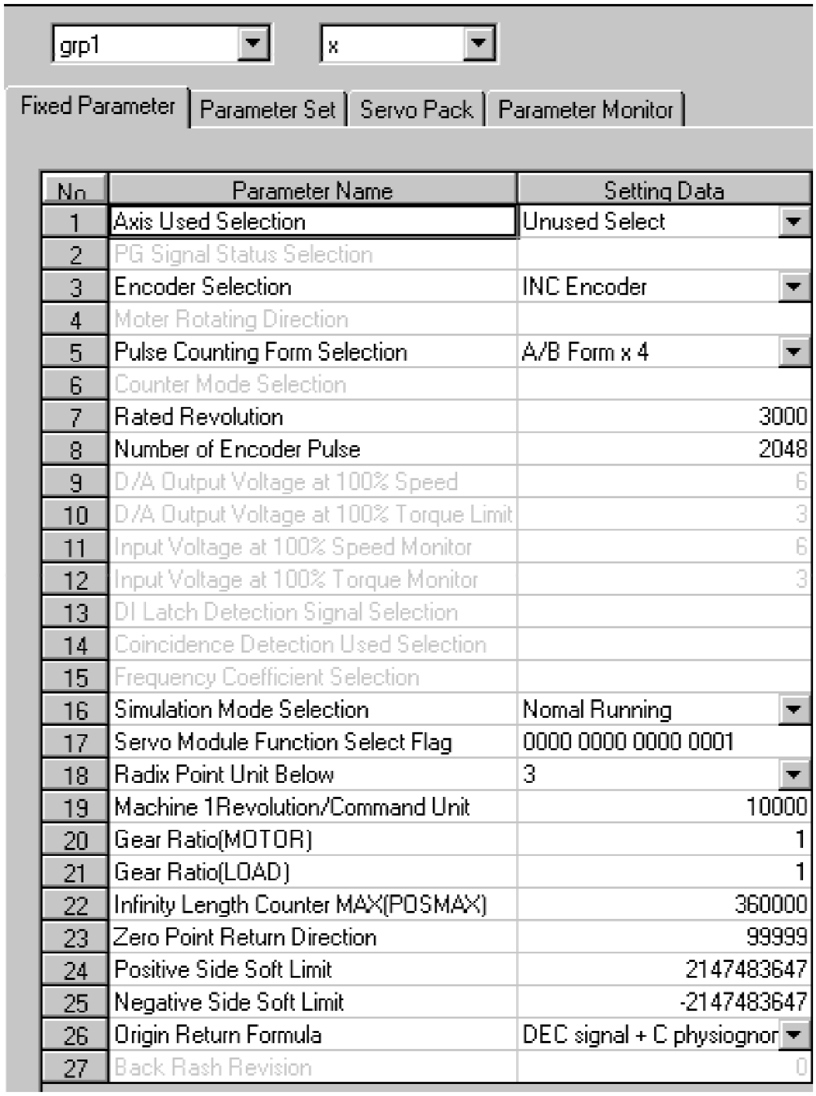
1. On the File Manager Screen, scroll down in order of ***XY-TABLE*** → ***Program Folder*** → ***High-speed Drawing*** → ***Motion*** → ***Motion Parameter***, place the cursor on ***Motion Parameter***, and double-click.



2. Fixed Parameter Settings

Set the following parameter relating to the X axis:

- No.26 Origin Return Method ("Formula"): C phase

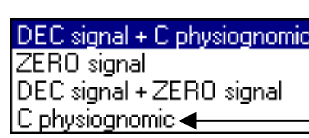


grp1 | x

Fixed Parameter | Parameter Set | Servo Pack | Parameter Monitor

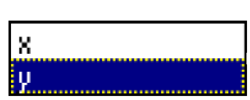
No	Parameter Name	Setting Data
1	Axis Used Selection	Unused Select
2	PG Signal Status Selection	
3	Encoder Selection	INC Encoder
4	Moter Rotating Direction	
5	Pulse Counting Form Selection	A/B Form x 4
6	Counter Mode Selection	
7	Rated Revolution	3000
8	Number of Encoder Pulse	2048
9	D/A Output Voltage at 100% Speed	6
10	D/A Output Voltage at 100% Torque Limit	3
11	Input Voltage at 100% Speed Monitor	6
12	Input Voltage at 100% Torque Monitor	3
13	DI Latch Detection Signal Selection	
14	Coincidence Detection Used Selection	
15	Frequency Coefficient Selection	
16	Simulation Mode Selection	Nomal Running
17	Servo Module Function Select Flag	0000 0000 0000 0001
18	Radix Point Unit Below	3
19	Machine 1Revolution/Command Unit	10000
20	Gear Ratio(MOTOR)	1
21	Gear Ratio(LOAD)	1
22	Infinity Length Counter MAX(POSMAX)	360000
23	Zero Point Return Direction	99999
24	Positive Side Soft Limit	2147483647
25	Negative Side Soft Limit	-2147483647
26	Origin Return Formula	DEC signal + C physiognor
27	Back Rash Revision	0

Click the **down arrow** to the right of ***Origin Return Formula***, and click ***C physiognomic***.



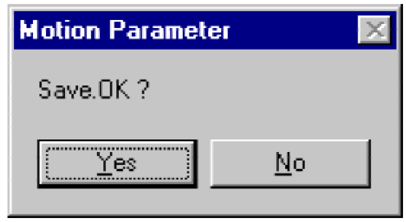
3. Setting Y Axis Fixed Parameters

Click the **down arrow** in the axis name field, and click ***Y***.



As with procedure 2, set No.26 ***Origin Return Formula*** to ***C physiognomic***.

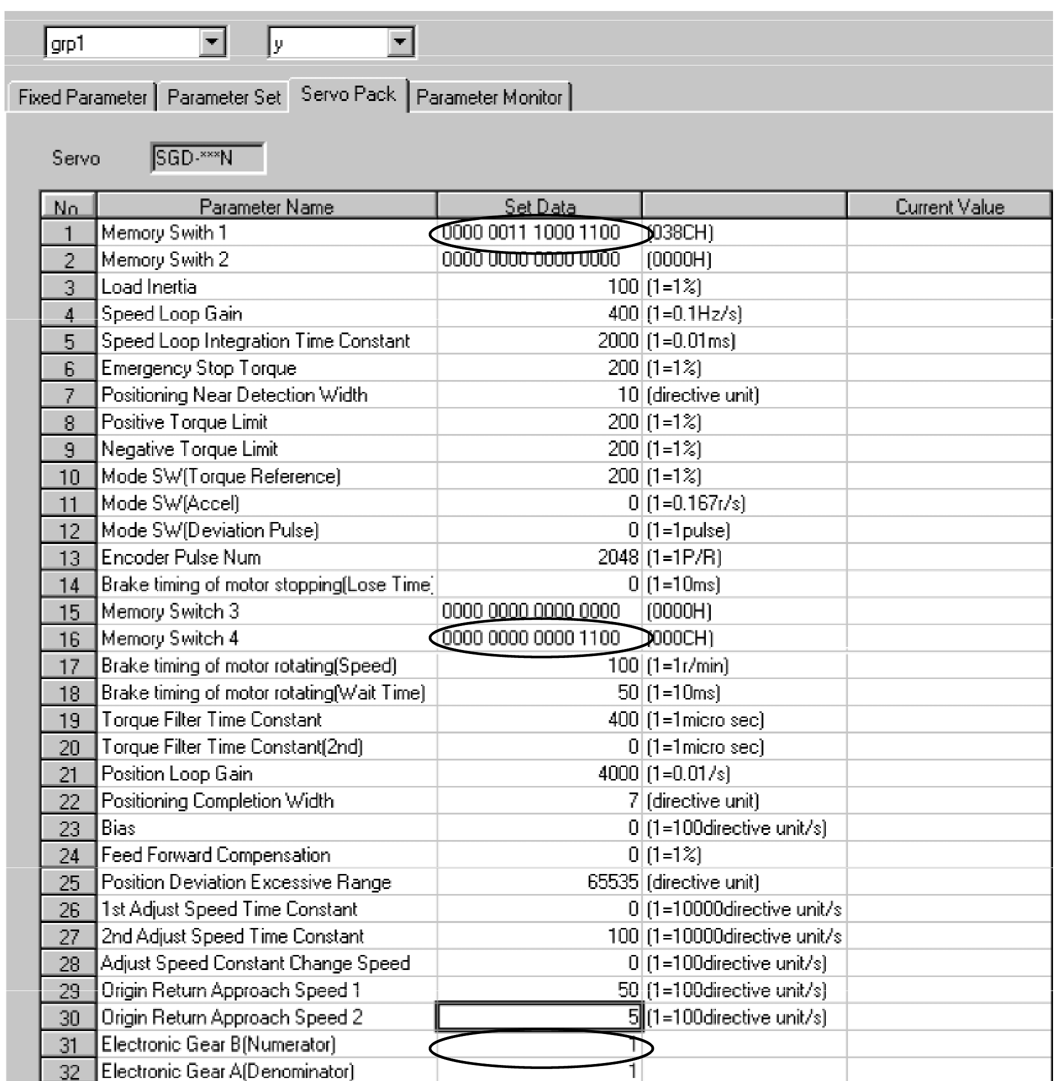
4. Click the Servopack tab and then click the **Yes** button for “Save OK?” in the message box.



5. Servopack Parameter Settings

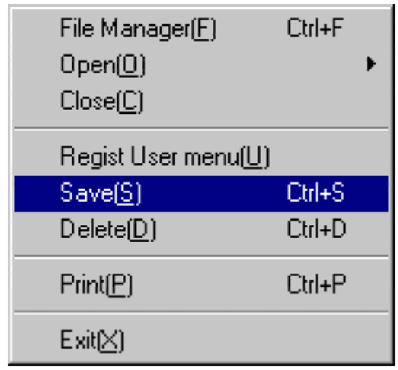
Change the following parameter settings. Set 2 as the axis number.

No.1:	Memory switches 1	038CH	P-OT, N-OT mask added
No.16:	Memory switches 4	000CH	P-SOT, N-SOT mask added
No.31:	Electronic gear B (numerator)	1	

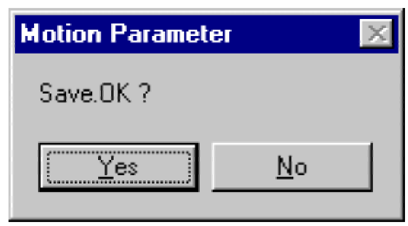


No	Parameter Name	Set Data		Current Value
1	Memory Swith 1	0000 0011 1000 1100	(038CH)	
2	Memory Swith 2	0000 0000 0000 0000	(0000H)	
3	Load Inertia	100	(1=1%)	
4	Speed Loop Gain	400	(1=0.1Hz/s)	
5	Speed Loop Integration Time Constant	2000	(1=0.01ms)	
6	Emergency Stop Torque	200	(1=1%)	
7	Positioning Near Detection Width	10	(directive unit)	
8	Positive Torque Limit	200	(1=1%)	
9	Negative Torque Limit	200	(1=1%)	
10	Mode SW(Torque Reference)	200	(1=1%)	
11	Mode SW(Accel)	0	(1=0.167r/s)	
12	Mode SW(Deviation Pulse)	0	(1=1pulse)	
13	Encoder Pulse Num	2048	(1=1P/R)	
14	Brake timing of motor stopping(Lose Time)	0	(1=10ms)	
15	Memory Switch 3	0000 0000 0000 0000	(0000H)	
16	Memory Switch 4	0000 0000 0000 1100	(000CH)	
17	Brake timing of motor rotating(Speed)	100	(1=1r/min)	
18	Brake timing of motor rotating(Wait Time)	50	(1=10ms)	
19	Torque Filter Time Constant	400	(1=1micro sec)	
20	Torque Filter Time Constant(2nd)	0	(1=1micro sec)	
21	Position Loop Gain	4000	(1=0.01/s)	
22	Positioning Completion Width	7	(directive unit)	
23	Bias	0	(1=100directive unit/s)	
24	Feed Forward Compensation	0	(1=1%)	
25	Position Deviation Excessive Range	65535	(directive unit)	
26	1st Adjust Speed Time Constant	0	(1=10000directive unit/s	
27	2nd Adjust Speed Time Constant	100	(1=10000directive unit/s	
28	Adjust Speed Constant Change Speed	0	(1=100directive unit/s)	
29	Origin Return Approach Speed 1	50	(1=100directive unit/s)	
30	Origin Return Approach Speed 2	5	(1=100directive unit/s)	
31	Electronic Gear B(Numerator)	1		
32	Electronic Gear A(Denominator)	1		

6. After completing the settings, click ***File (F)*** → ***Save (S)*** from the File menu.



7. Click the **Yes** button. The Servopack parameters will be saved.

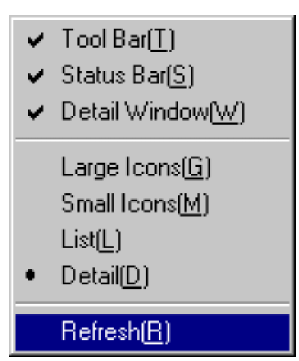


4.3.8 Creating and Saving Motion Programs

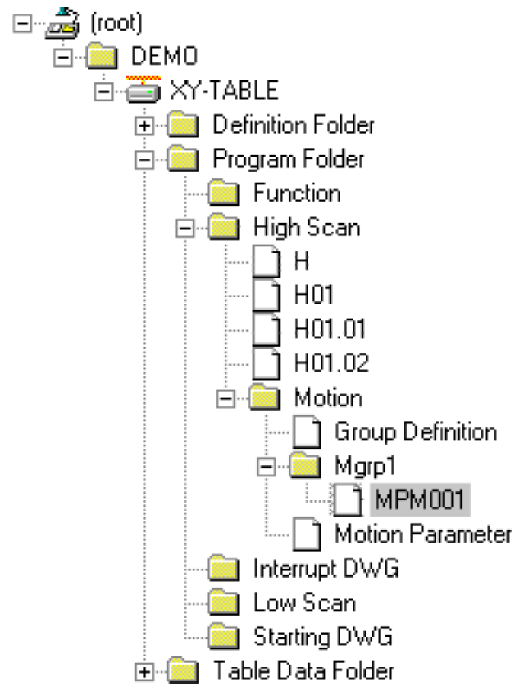
Creating Motion Programs

Use the following procedure to create a motion program.

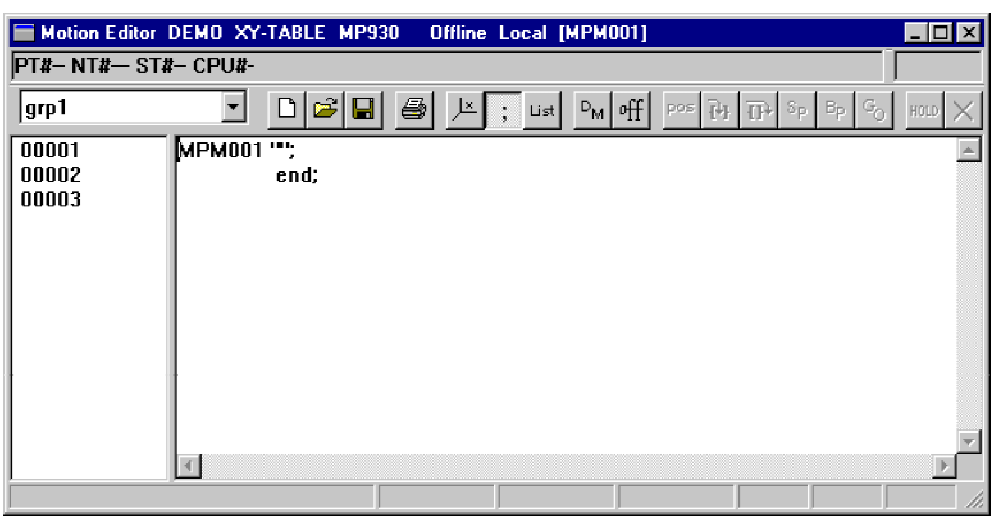
1. Click ***Refresh (R)*** from ***View (V)*** on the File Manager menu. This refreshes the display information in the XY-TABLE folder.



2. On the XY-TABLE Folder of the File Manager Screen, scroll down in order of ***Program Folder → High-speed Drawing → Motion → Mgrpl → MPM001***, place the cursor on ***MPM001***, and double-click.



The motion program MPM001 file will be displayed.



4

3. Input the following program in the part marked “Input here.”

```
MPM001 "";
        fmx t8000000;                     "setup interpolation max. feed speed
        mw0001=10000;                     "interpolation speed overrid=100.0%
        iac t300;                         "time constant for acceleration for interpolation= 300ms
        idc t500;                         "time constant for deceleration for interpolation= 500ms
        vel [x]6000 [y]1000;              "setup axis x,y feed speed
        zrn [x]0 [y]0;                    "home position return
        mov [x]100.0 [y]100.0;            "positioning (100,100)
        inc mvs [x]200.0 f5000000;        "linear interpolation axis x +200.0 (inc. mode)
        mvs [y]200.0;                     "axis-y +200.0 (inc. mode)
        mvs [x]-200.0;                    " axis-x -200.0  (inc. mode)
        mvs [y]-200.0;                    " axis-y -200.0  (inc. mode)
        abs mvs [x]200.0 [y]200.0;        " axis-x,y (200,200)  (abs. mode)
        pln [x] [y];                      "circular interpolation plane
        mcc [x]200.0 [y]200.0 u200.0 v300.0 f5000000;  "circular interpolation
        mvs [x]100.0 [y]100.0;
        end;
```

Saving Motion Programs

Use the following procedure to save a motion program that has been created.

1. Click ***Save*** on the Motion Editor tool bar.

2. Click the **Yes** button for the Overwrite OK? message box. Compilation will start, and the motion program will be saved.

MPM001 Operation

1. Start.
2. X, Y axis origin return.
3. Moves by rapid traverse speed to reference point (100.0, 100.0).
4. Draws a square in order of +X → +Y → –X → –Y by linear interpolation.
5. Moves to (200.0, 200.0) by linear interpolation.
6. One-circle operation at the center point (200.0, 300.0).
7. Moves by linear interpolation to reference point (100.0, 100.0).
8. End.

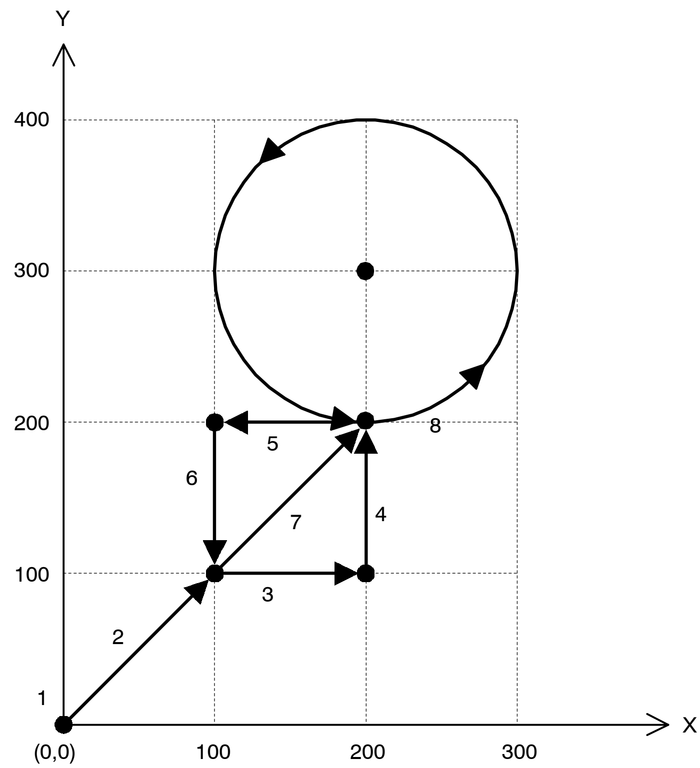


Figure 4.2 Move Operation Chart According to Program

4

4

4.3.9 Ladder Logic Programs

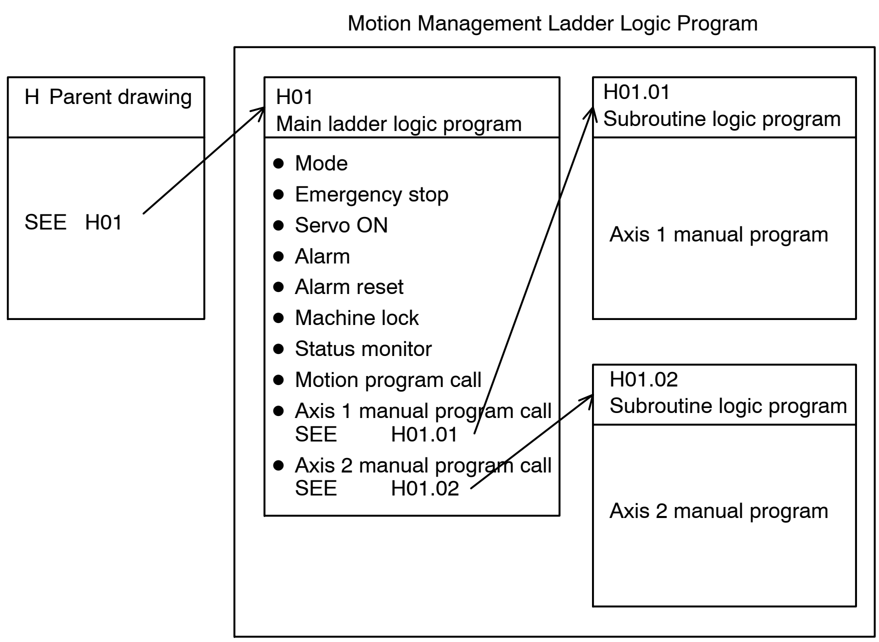
■ Overview

Ladder logic programs are automatically generated on the CP-717 by selecting Yes for PGM Automatic Generation on the Group Definition Screen and saving.

These ladder logic programs, called motion management ladder logic programs, are used to execute JOG, STEP, and HOME RETURN in manual mode, and to execute motion programs in automatic mode.

■ Structure of Ladder Logic Programs

The following illustration shows the structure of an automatically generated motion management ladder logic program (MM ladder logic program).



■ Control Specifications

Motion management ladder logic programs with the following specifications are automatically generated for the Test Unit.

Number of controlled axes	2 axes
Number of tasks	1
Number of groups	1

■ External Signal Allocation

The external signals used by motion management ladder logic programs are allocated as shown in the following table.

Group Input Signals		Axis 1 Input Signals		Axis 2 Input Signals	
IB00000	Automatic mode	**IB00100**	Servo ON	**IB00110**	Servo ON
IB00001	Manual mode	**IB00101**	JOG+	**IB00111**	JOG+
IB00002	Start	**IB00102**	JOG−	**IB00112**	JOG−
IB00003	Reset	**IB00103**	STEP+	**IB00113**	STEP+
IB00004	Pause	**IB00104**	STEP−	**IB00114**	STEP−
IB00005	Emergency stop	**IB00105**	ZRN	**IB00115**	ZRN
IB00006	Alarm reset	**IB00106**	Zero point setting	**IB00116**	Home position setting
IB00007	Machine lock setting	**IB00107**	Stop	**IB00117**	Stop
IB00008	Block operation mode				
IB00009	Block operation				
IB0000A	Skip 1 operation				
IB0000B	Skip 2 operation				

4

Registers Used by Motion Management Ladder Logic Programs

Data Transfer between Main and Sub-ladder Logic Programs

MW00002 (1 word) is used as the register for data transfer between main and sub-ladder logic programs.

Registers in the Main Ladder Logic Program (H01)

The following illustration shows the configuration of the group work registers used by the main ladder logic program.

Register Number	Size	Description
DW00100	1 word	Group status
DW00101	1 word	Group control signal
DW00102	1 word	Automatic, common
DW00103	1 word	Manual, common
DW00104	1 word	Axis 1 manual status
DW00105	1 word	Axis 2 manual status
:		:
DW00151	1 word	Axis 48 manual status

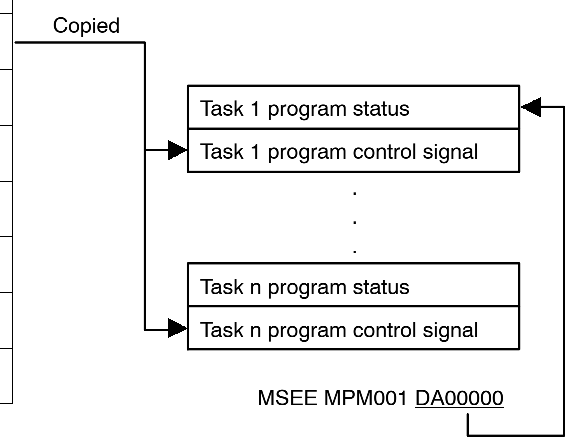


Figure 4.3 Group Work Register Configuration

A detailed description of the registers is given in the following table.

Program Status (DW00100)		Program Control Signal (DW00101)	
b0	Program running	**b0**	Program start request
b1	Program paused	**b1**	Program pause request
b2	(Used by the system.)	**b2**	Program forced stop request
b3	(Used by the system.)	**b3**	Program debugging mode selection
b4	Program being debugged	**b4**	Program debugging start request
b8	Program alarm generated	**b5**	Alarm reset request
bB	Debugging mode (EWS debugging)	**b8**	Skip 1 information
bE	Main program duplication error	**b9**	Skip 2 information
bF	Main program number exceeded error	–	–

Automatic, Common (DW00102)		Manual, Common (DW00103)	
b0	Stopped for emergency	**b0**	Operating manually
b1	Status history	–	–
b2	Debugging start history	–	–
b3	Automatic mode status OFF request	–	–
b4	Manual mode status OFF request	–	–
b5	Program start request	–	–

Manual Status (DW00104)	
b0	Axis alarm generated
b1	Command duplication command alarm
b2	Operating manually

Registers in Sub-Ladder Logic Programs (H01.01, H01.02)

The following table shows the configuration of the group work registers used by the sub-ladder logic programs.

Register Number	Size	Description
DW00100	1 word	Manual status
DW00101	1 word	Command/Response
DW00102	1 word	FEED status
DW00103	1 word	STEP status
DW00104	1 word	ZRET status
DW00105	1 word	ZSET status

A detailed description of the registers is given in the following table.

Manual status (DW00100)		Axis Command/Response (DW00101)	
b0	Axis alarm generated	**b0**	Command = No command
b1	Command duplication command alarm	**b1**	Command = Interpolation end segment
b2	Operating manually	**b2**	Command = FEED
b3	Manual system command completed	**b6**	Response = No command
b4	Command interrupted	**b7**	Response = POSITIONING
–	–	**b8**	Response = EXTERNAL POSITIONING
–	–	**b9**	Response = ZRET
–	–	**bA**	Response = Interpolation
–	–	**bB**	Response = Interpolation end segment
–	–	**bC**	Response = FEED
–	–	**bD**	Response = STEP
–	–	**bE**	Response = ZSET

FEED status (DW00102)		STEP status (DW00103)	
b0	FEED start request	**b0**	STEP start request
b1	FEED command completion check	**b1**	STEP start history
b2	FEED execution	**b2**	STEP execution
b3	FEED command operating	**b3**	STEP command operating
b4	FEED command completed	**b4**	STEP command completed

ZRET status (DW00104)		ZSET status (DW00105)	
b0	ZRET start request	**b0**	ZSET start request
b1	ZRET start history	**b1**	ZSET start history
b2	ZRET execution	**b2**	ZSET execution
b3	ZRET command operating	**b3**	ZSET command operating
b4	ZRET command completed	**b4**	ZSET command completed

Motion Management Ladder Logic Programs

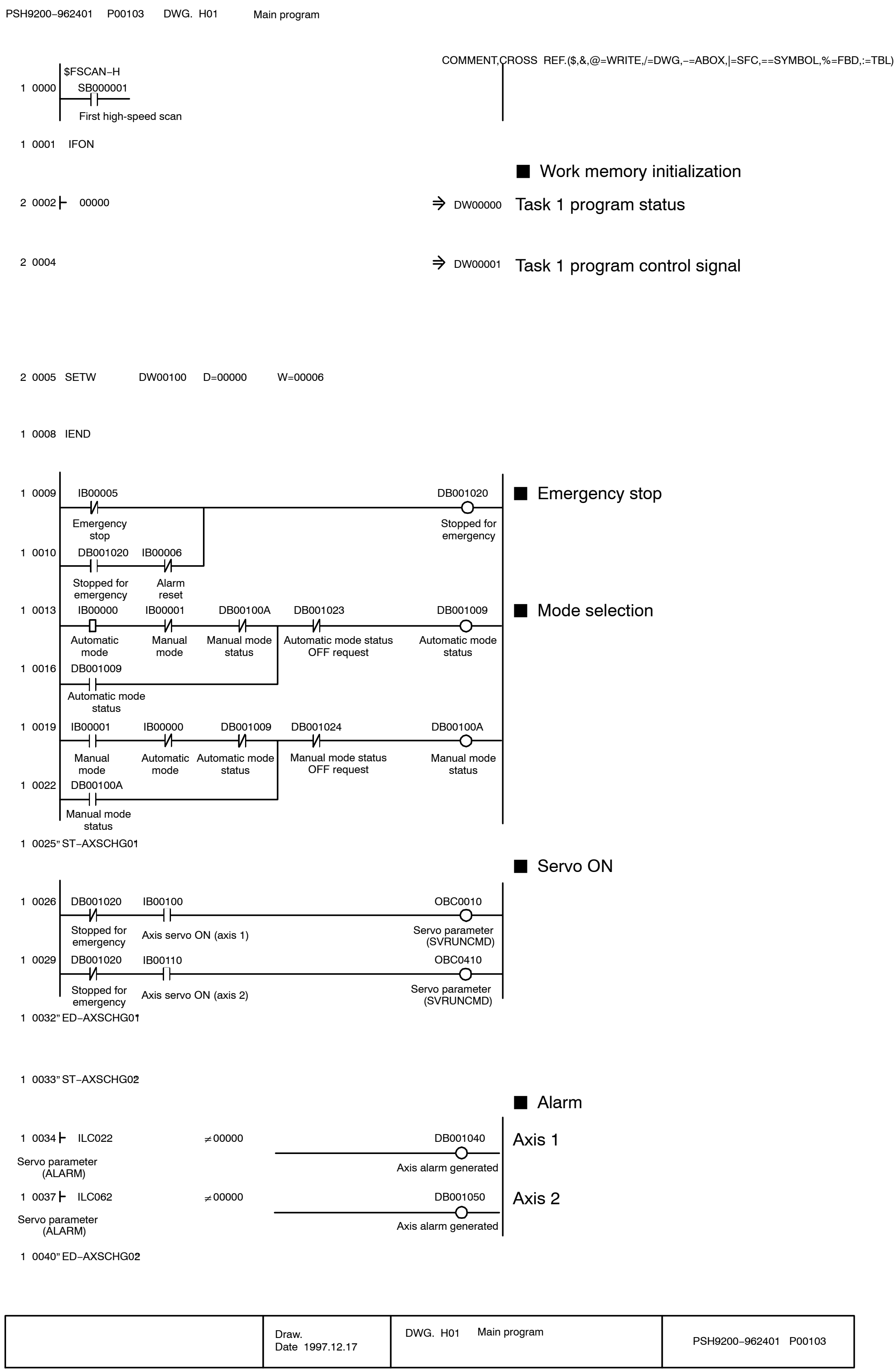
The programs that are automatically generated on the Group Definition Screen are shown in the following illustrations.

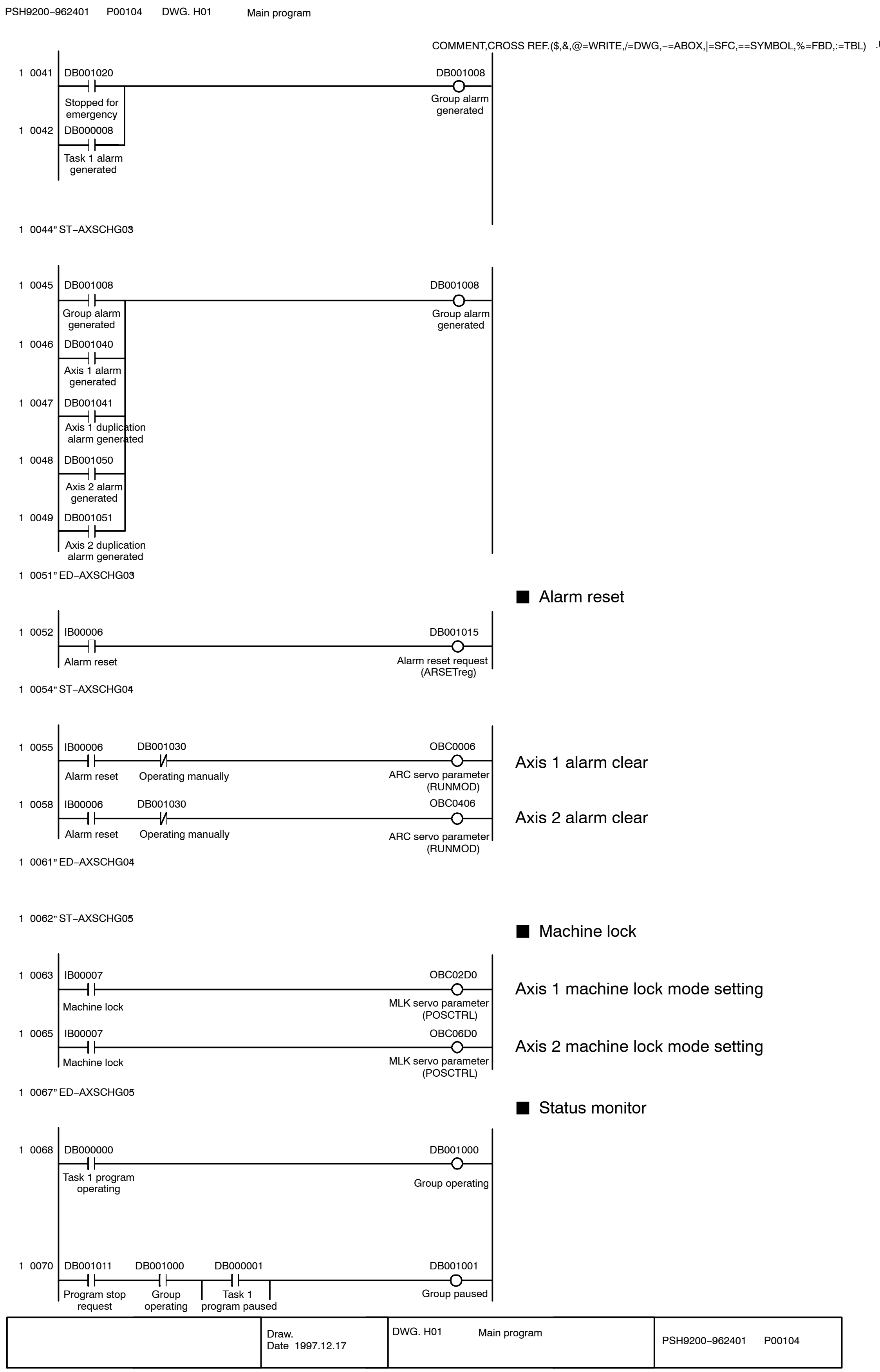
H Drawing Main Program

```
PSH9200-962401    P00101    DWG. H        Main program

                     COMMENT,CROSS  REF.($,&,@=WRITE,/=DWG,-=ABOX,|=SFC,==SYMBOL,%=FBD,:=TBL)

  1  0000"SEESTT"

  1  0001  SEE          H01   (Motion management ladder logic program call)

  1  0002"SEE01"

  0  0003  DEXD
```

	Draw. Date 1997.12.17	DWG. H Main program	PSH9200-962401 P00101

Main Motion Management Ladder Logic Program





■ Alarm reset

Axis 1 alarm clear

Axis 2 alarm clear

■ Machine lock

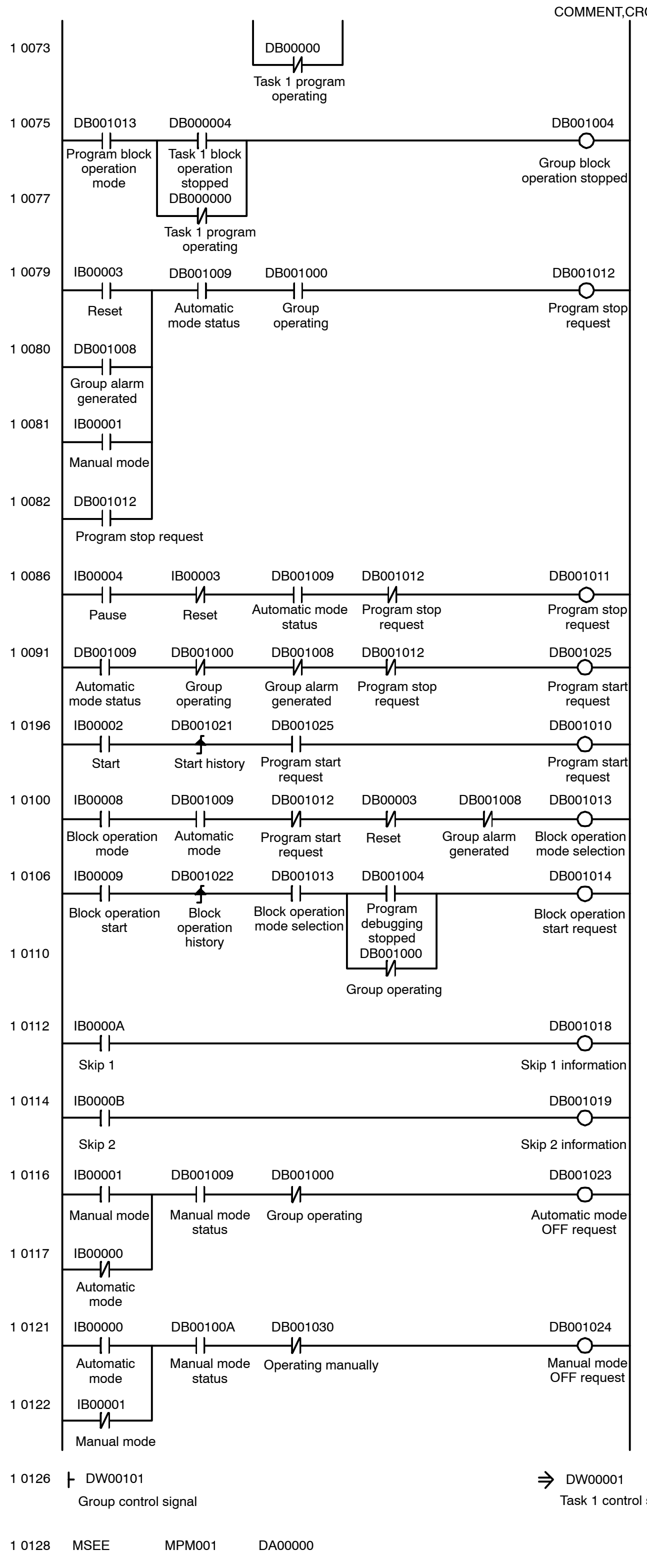
Axis 1 machine lock mode setting

Axis 2 machine lock mode setting

■ Status monitor

4.3.9 Ladder Logic Programs

PSH9200–962401 P00105 DWG. H01 Main program



■ Stop

■ Pause

■ Start

■ Block operation

■ Skip operation

■ Mode resetting

■ Control signal distribution

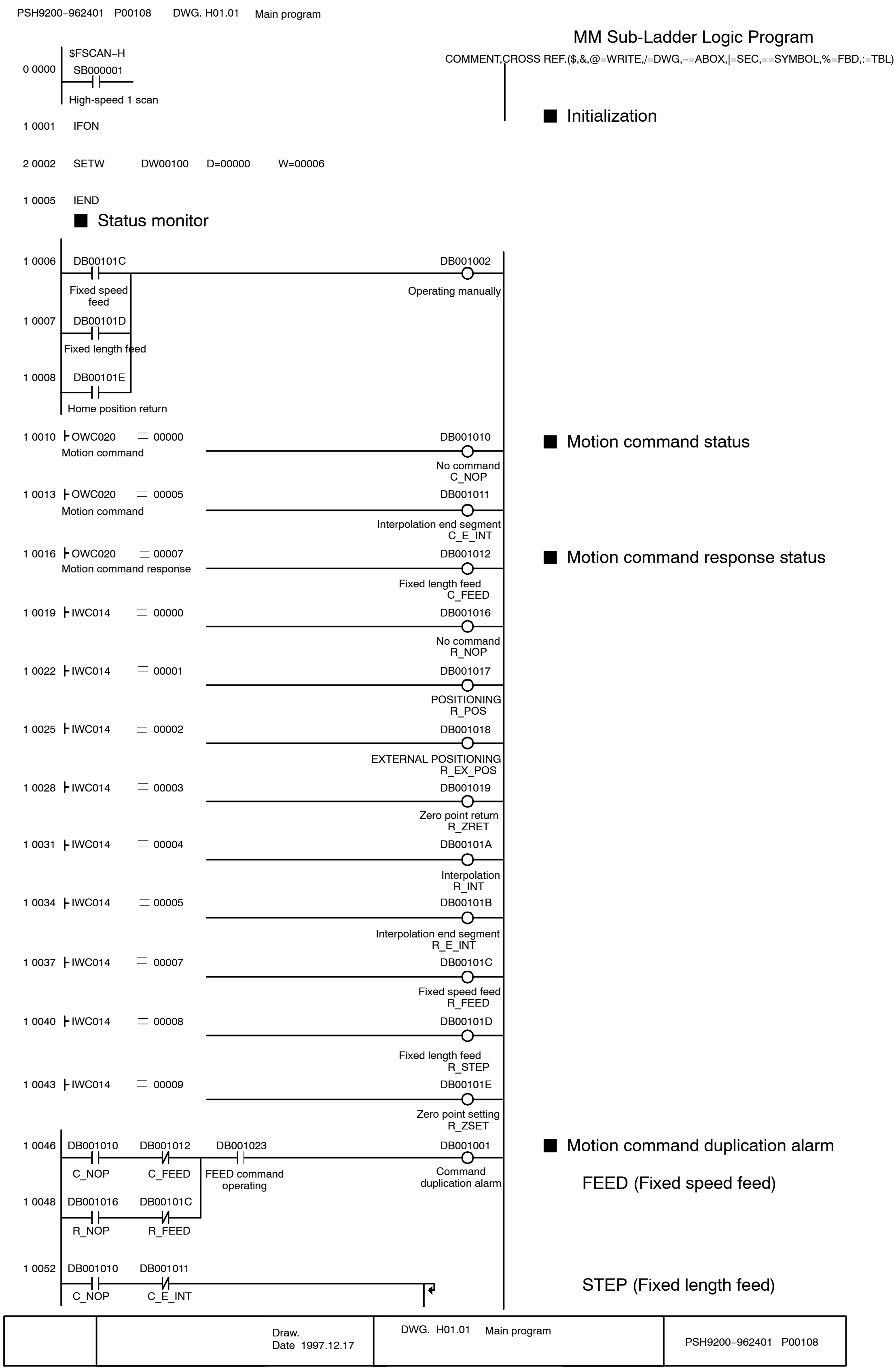
■ Motion program call (Task 1)

	Draw. Date 1997.12.17	DWG. H01 Main program	PSH9200–962401 P00105

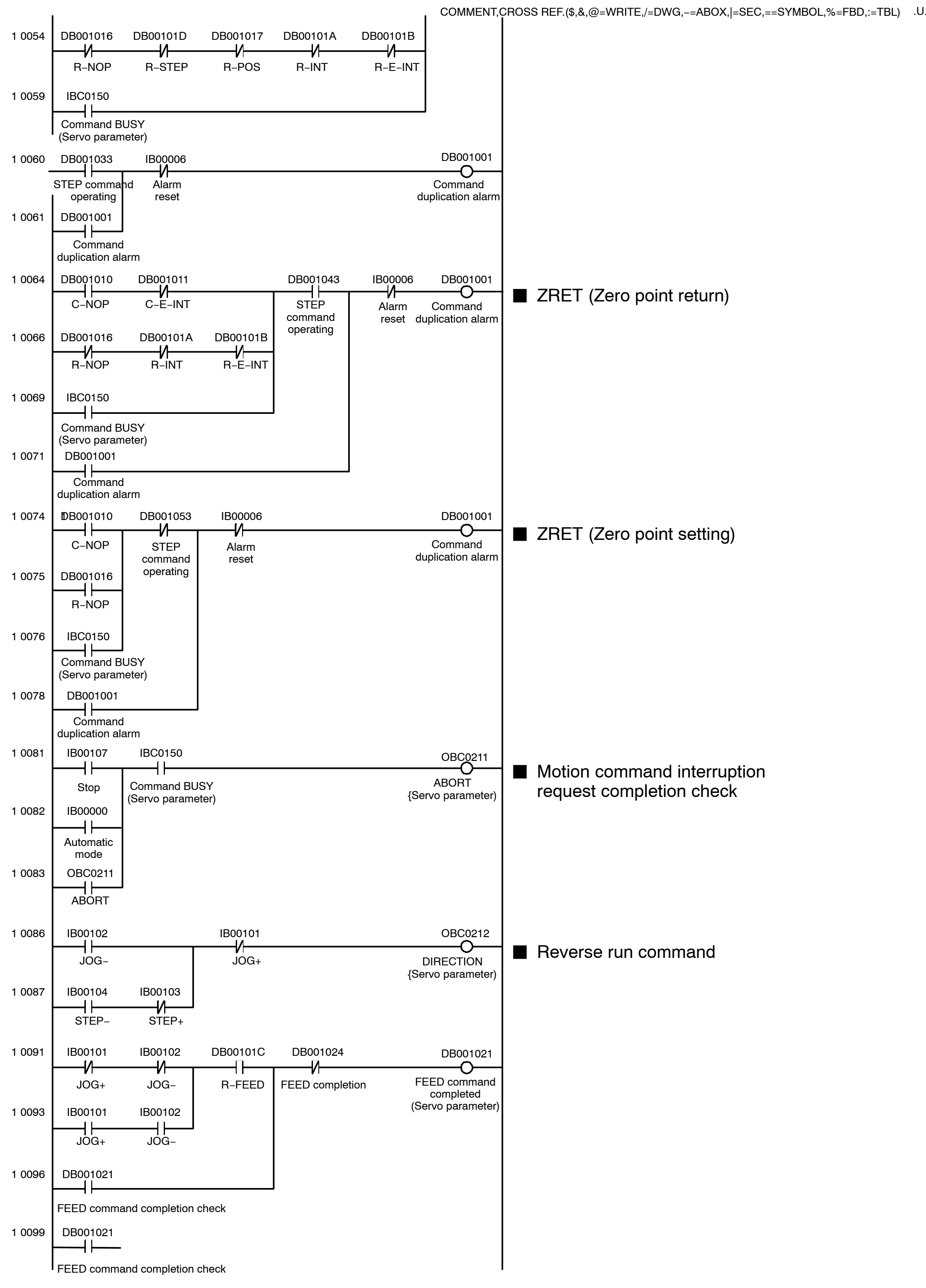
	Draw. Date 1997.12.17	DWG. H01.01 Main program	PSH9200−962401 P00106

4

Axis 1 Manual Program



PSH9200–962401 P00109 DWG. H01.01 Main program



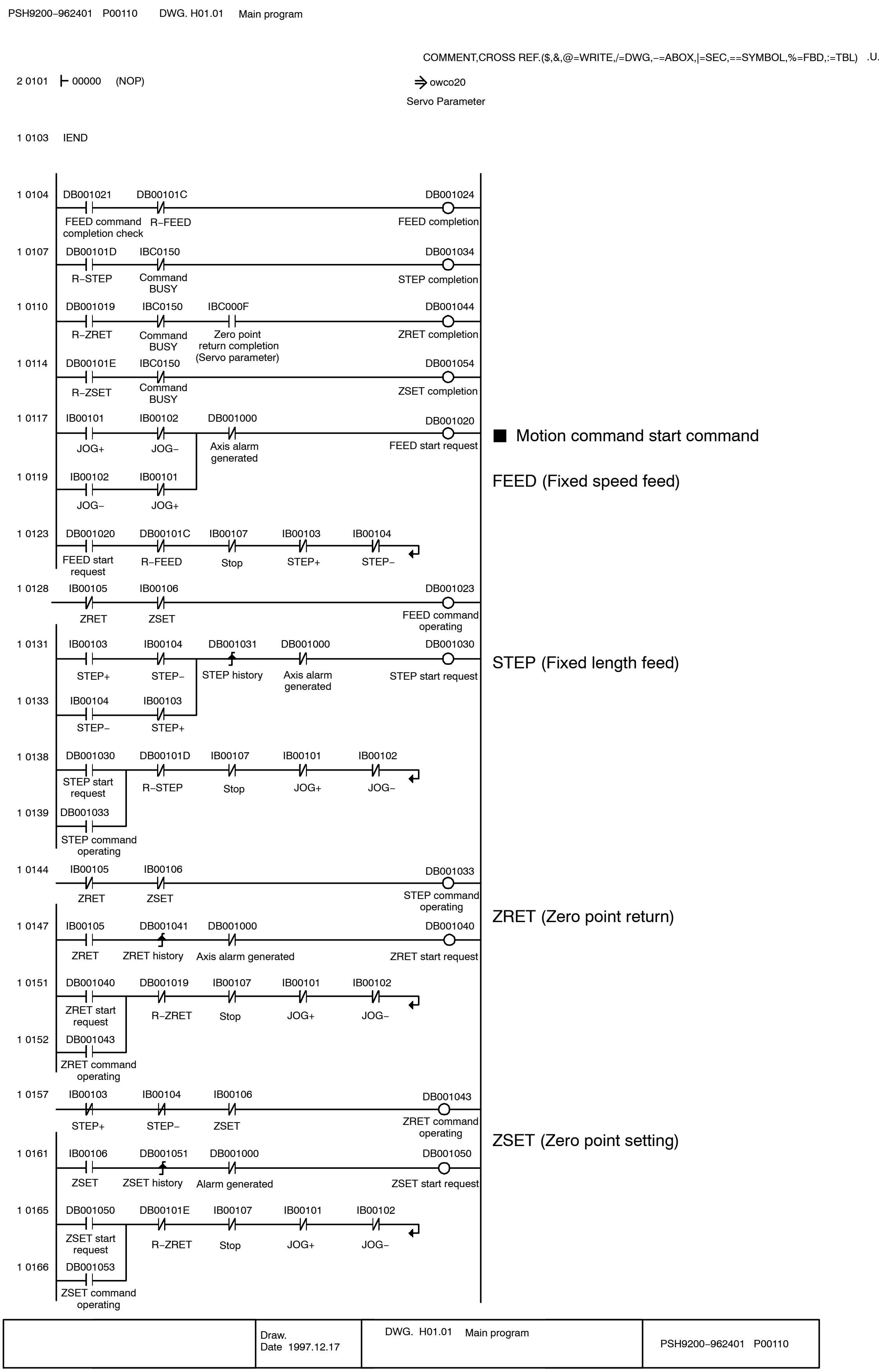
■ ZRET (Zero point return)

■ ZRET (Zero point setting)

■ Motion command interruption request completion check

■ Reverse run command

	Draw. Date 1997.12.17	DWG. H01.01 Main program	PSH9200–962401 P00109



■ Motion command start command

FEED (Fixed speed feed)

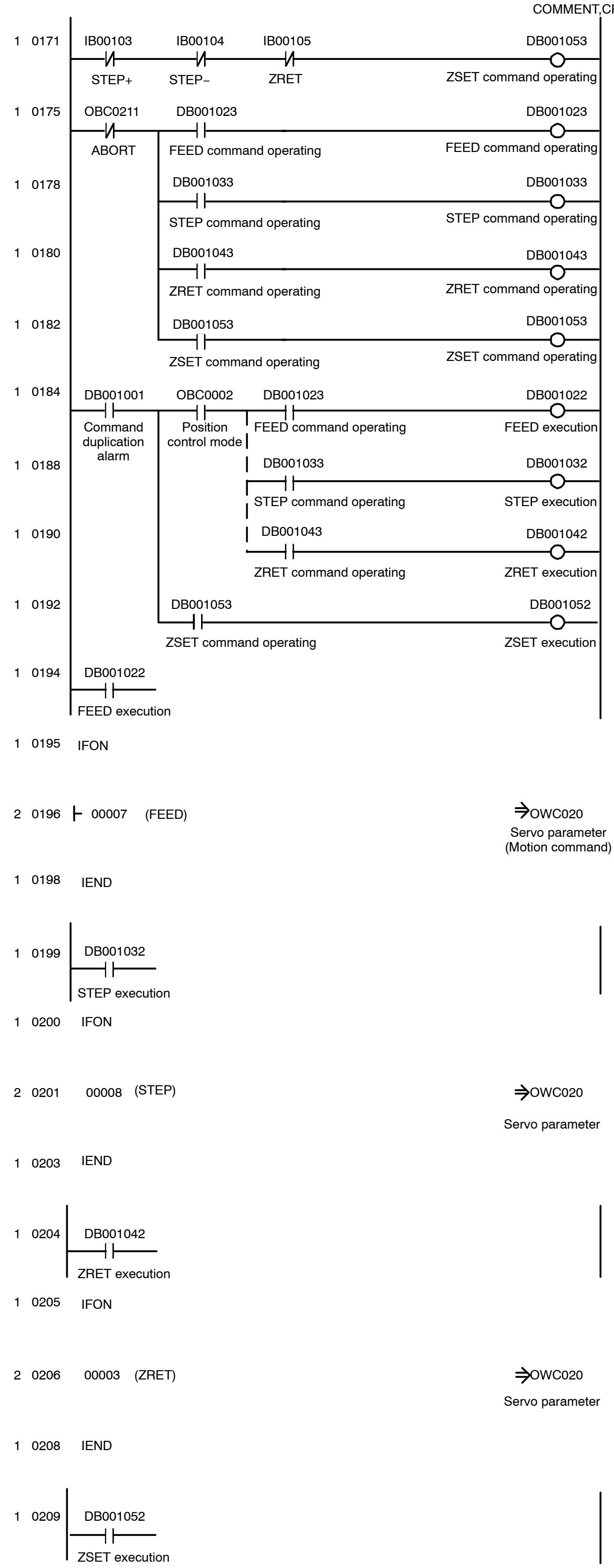
STEP (Fixed length feed)

ZRET (Zero point return)

ZSET (Zero point setting)

PSH9200−962401 P00111 DWG. H01.01 Main program

COMMENT,CROSS REF.($,&,@=WRITE,/=DWG,−=ABOX,|=SFC,==SYMBOL,%=FBD,:=TBL) .U.



■ Command execution

	Draw. Date 1997.12.17	DWG. H01.01 Main program	PSH9200−962401 P00111

4.3.9 Ladder Logic Programs

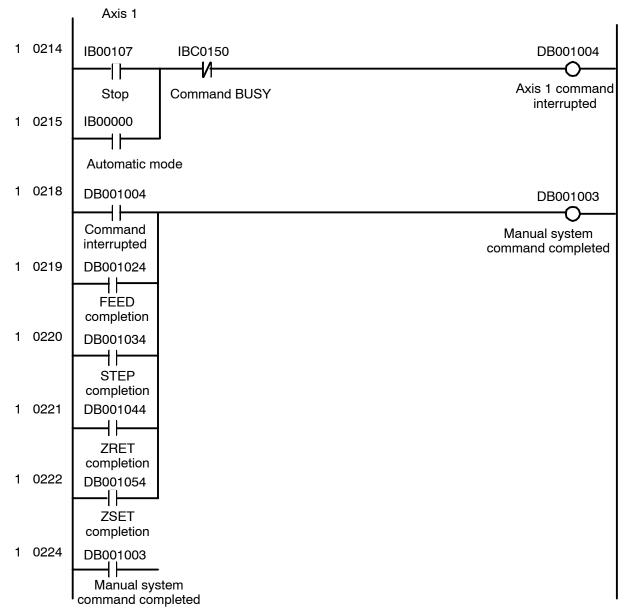
PSH9200-962401 P00112 DWG. H01.01 Main program

COMMENT,CROSS REF.($,&,@=WRITE,/=DWG,-=ABOX,|=SFC,==SYMBOL,%=FBD,:=TBL) .U.

1 0210 IFON

2 0211 ⊦ 00009 (ZSET) ⇒OWC020
Servo Parameter

1 0213 IEND



■ Stop command

■ Motion command end command

2 0225 IFON

2 0226 ⊦ 00000 (NOP) ⇒OWC020
Motion command

2 0228 ⊦OWC021 Motion command control flag ∧ HFFFD (ABORT=OFF) ⇒OWC021 Motion command control flag

1 0231 IEND

1 0232 ⊦DW00100 Manual status ⇒MW00002 Manual status

1 0234 DEND
Subladder end

	Draw. Date 1997.12.17	DWG. H01.01 Main program	PSH9200-962401 P00112

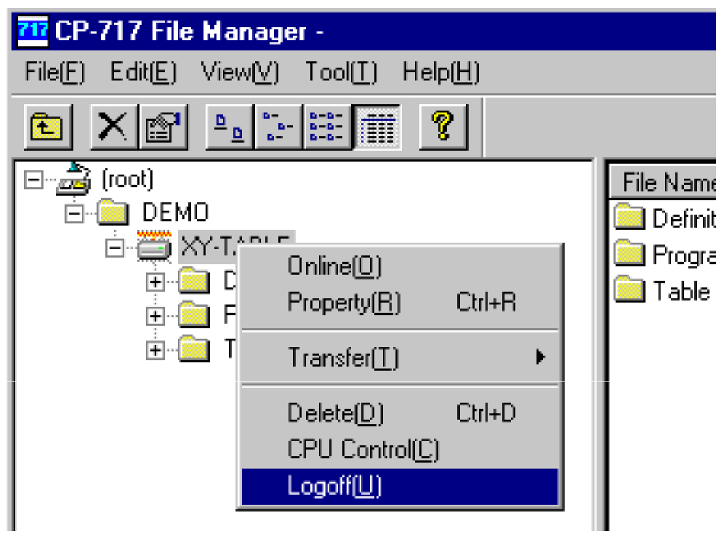
4.3.10 Transferring Definitions, Parameters, and Programs

Preparations for Transfer

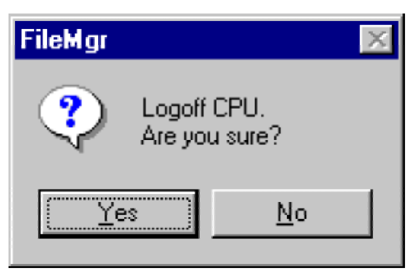
Use the following procedure to set the CPU to STOP status after switching from offline to online.

1. Logoff from Offline

On the File Manager Screen, right click the ***XY-TABLE*** PLC folder and click ***Logoff (U)***.

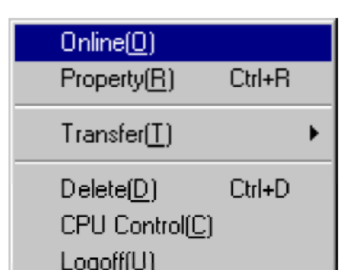


2. Click the **Yes** button for the Logoff CPU Are you sure? message box.



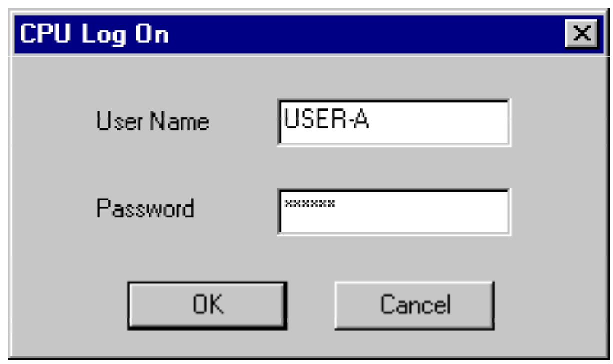
3. Online Logon

a) On the File Manager Screen, right click the ***XY-TABLE*** PLC folder and click ***Online***.

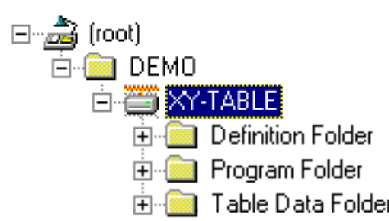


b) Double-click the ***XY-TABLE*** PLC folder.

c) Input the user name **USER-A** and password **USER-A**, and click the **OK** button.

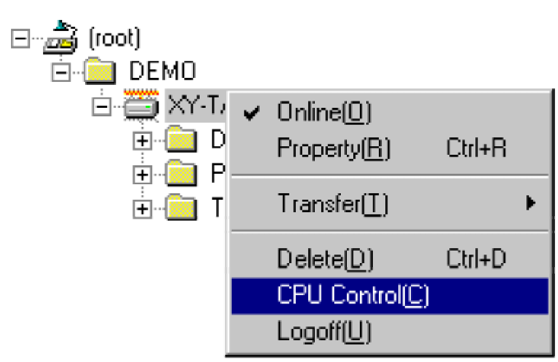


The program, definition, and table data folders will be displayed in the XY-TABLE PLC folder, and logon has been completed.



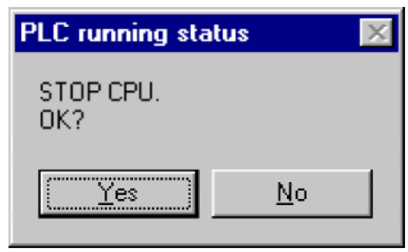
4. CPU STOP Operation

a) Right click the ***XY-TABLE*** PLC folder and click ***CPU Control (C)***.

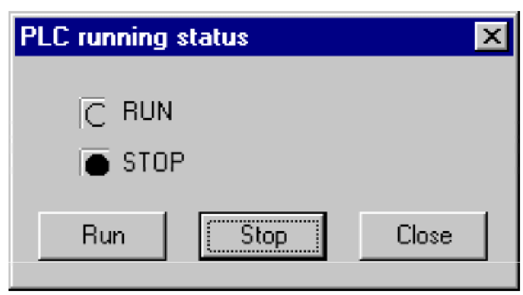


b) Click the **Stop** button in the PLC Running Status message box.

c) Click the **Yes** button for the STOP CPU OK?” message box.



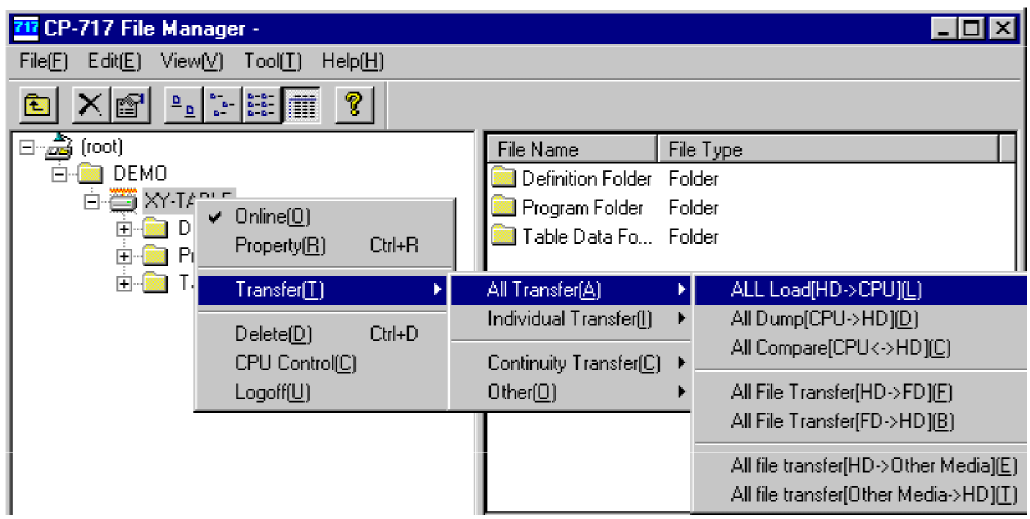
d) Confirm that the message box has entered STOP status, and click the **Close** button.



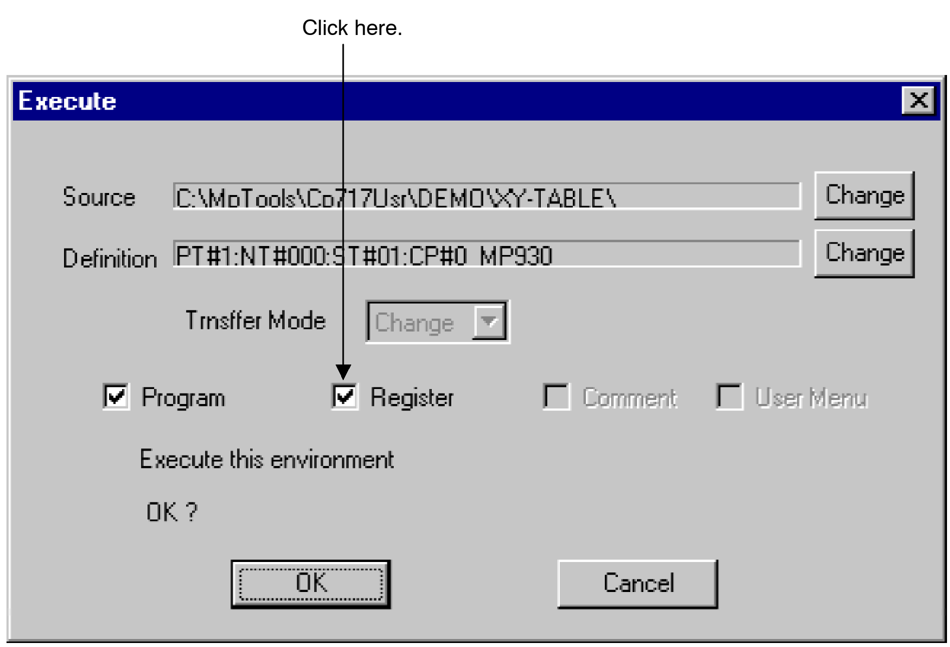
■ Transfer Procedure

The following screens show the transfer procedure for definitions, parameters, and programs.

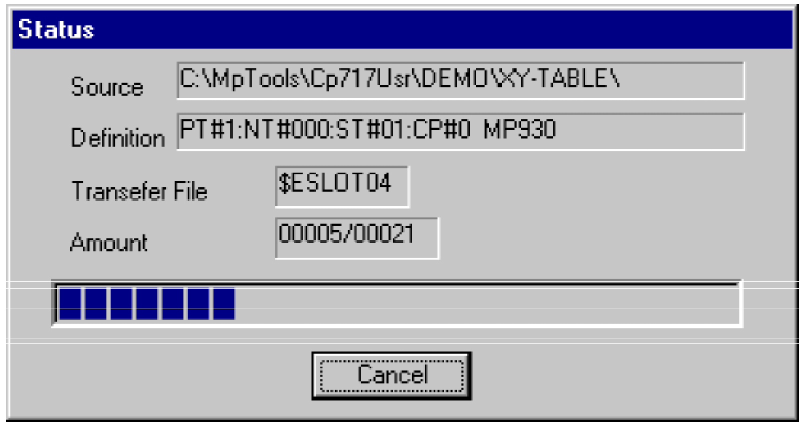
1. Right click the ***XY-TABLE*** PLC folder, scroll down in order of ***Transfer (T)*** → ***All Transfer (A)*** → ***All Load (HD → CPU) (L)***, place the cursor on ***All Load (HD → CPU) (L)***, and click.



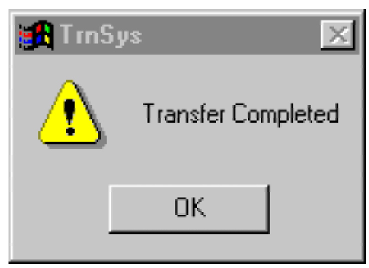
2. Registers do not need to be transferred, so turn OFF the register selection, and click **OK** to start the transfer.



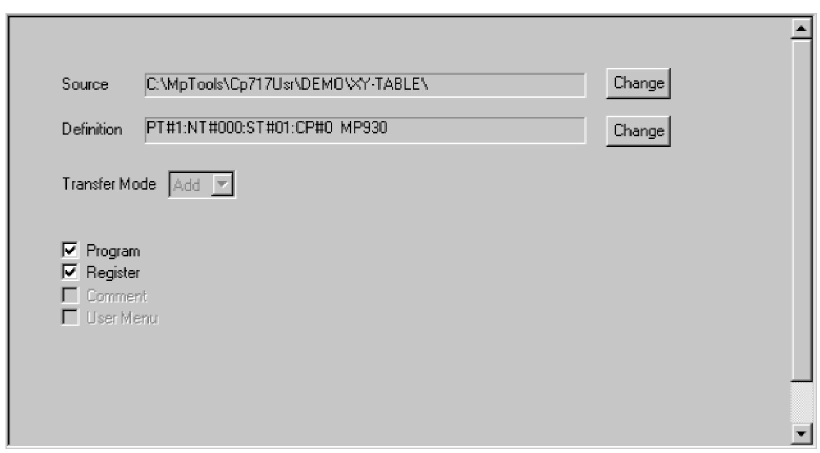
3. The Status Screen will be displayed during transfer.



4. When transfer has been completed, the next message box will be displayed. Click the **OK** button in the Transfer Completed message box.



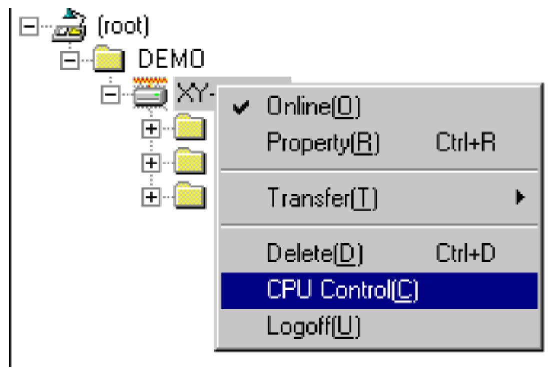
5. The Completion Confirmation Screen will be displayed.



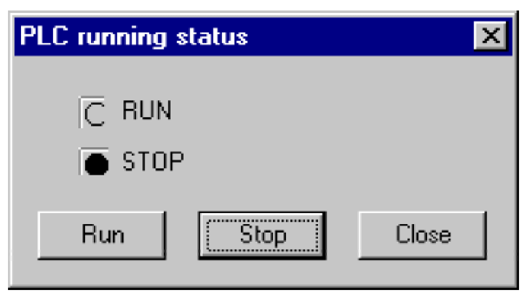
6. CPU RUN Operation

Once the transfer has been completed, perform a CPU RUN operation, and execute the user program according to the definitions, parameters, and programs that have been sent.

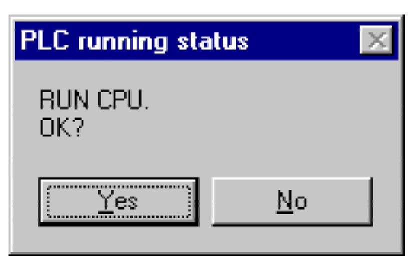
a) Right click the ***XY-TABLE*** PLC folder and click ***CPU Control (C)***.



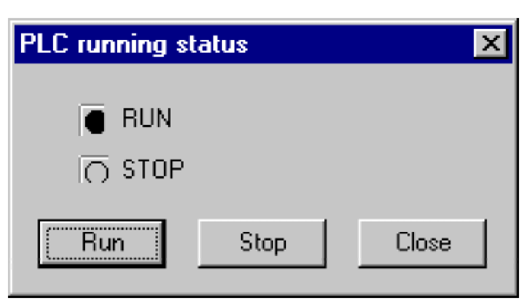
b) Click the **RUN** button in the PLC running Status message box.



c) Click the **Yes** button in the RUN CPU OK? message box.



d) Confirm that the message box has entered RUN status, and click the **Close** button.



User programs are executed by this operation. The MC Unit RUN indicator will light, and the Servopack ALARM indicator will go out.

7. Servopack Parameter Transfer

Open the Motion Parameter Screen and open the Servopack Screen, confirm that the servo parameters saved by the procedure discussed in *4.3.7 Motion Parameter Settings* have been loaded, and perform ***File (F) → Save (S)***. This process is required for each axis.

4.3.11 Checking Operations

After wiring has been completed, and after the definitions, parameters, motion programs, and ladder logic programs have been created and transferred, use the following procedure to check operations.

■ Operation Check Procedure

1. Set the Automatic and Manual switches to Manual.
2. Set the emergency stop signal to ON.
3. Set the Servo ON/OFF switch to ON.
4. Press the JOG buttons.
5. Set the Automatic and Manual switches to Automatic.
6. Press the Start button and start the motion program.

■ Selecting Manual Mode

In the switch box, set the Automatic switch to OFF and the Manual switch to ON to select Manual mode.

■ Setting the Emergency Stop Signal to ON

Switch the emergency stop signal switch from OFF to ON. Because this signal is an N.C. contact, the emergency stop will be reset to ON status.

■ Setting the Servo to ON

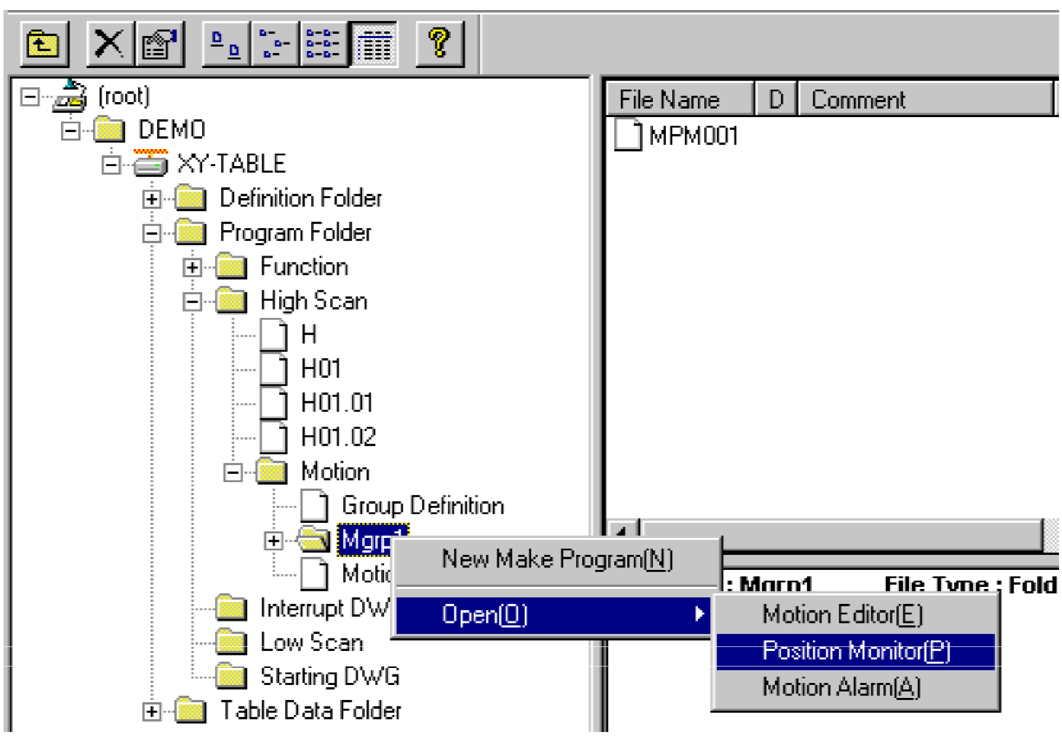
Switch the SERVO ON/OFF switch in the switch box from OFF to ON. The Servopack power will be clamped, and will enter servo clamp status.

JOG Operation Check

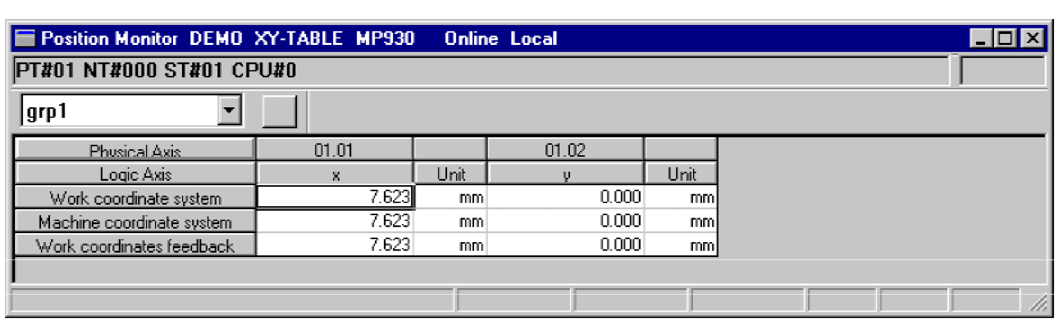
Check the JOG operation of each axis.
When the X+ button is pressed, the X axis will move in the positive direction while the button is being pressed. The current position of the X axis on the Position Monitor Screen will increase.
When the X– button is pressed, the X axis will move in the negative direction while the button is being pressed. The current value of the X axis on the Position Monitor Screen decreases.
Perform the same operation for the Y axis.

The procedure for displaying and checking the current position is as follows:

1. On the File Manager Screen, scroll down in order of ***Program Folder → High-speed Drawing → Motion → Mgrp1***, right click ***Mgrp1***, and click ***Open (O) → Position Monitor (P)***.



2. The Position Monitor Screen will be displayed. The current position of each axis can be monitored.



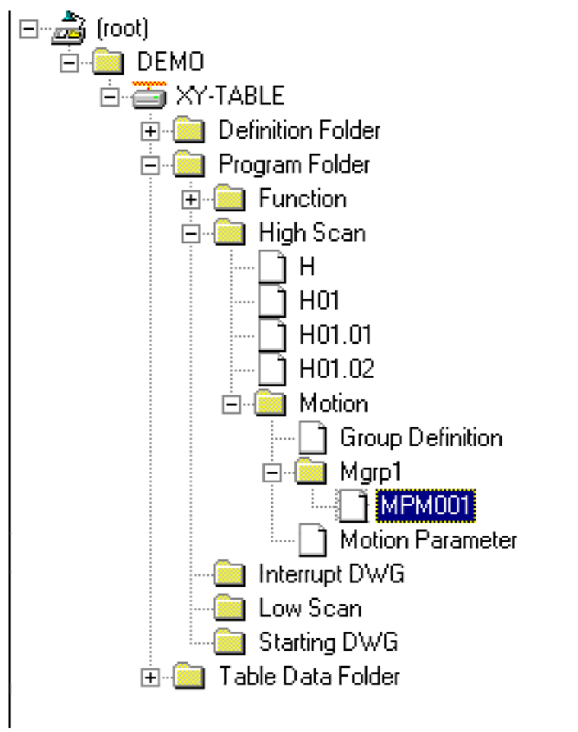
Changes in the current position can be checked by pressing the X+ and X– buttons.

Motion Program Operation Check

Set the Automatic switch to ON and the Manual switch to OFF to select Automatic mode.
Next, press the Start button to execute the motion program.
To monitor the motion program, display and check the Motion Editor Screen.

The procedure referred to above is shown in more detail in the following illustrations.

1. Set the Automatic switch to ON and the Manual switch to OFF to switch to Automatic mode.
2. On the File Manager Screen, scroll down in order of ***Program Folder*** → ***High-speed Drawing*** → ***Motion*** → ***Mgrp1*** → ***MPM001***, and double-click ***MPM001***.



3. Press the switch box Start button. The block being executed will be displayed in reverse video.

```
MPM001 "";
        fmx t8000000;                "setup interpolation max. feed speed
        mw0001=10000;                "interpolation speed overrid=100.0%
        iac t300;                    "time constant for acceleration for interpolation= 300ms
        idc t500;                    "time constant for deceleration for interpolation= 500ms
        vel [x]6000 [y]1000;         "setup axis x,y feed speed
        zrn [x]0 [y]0;               "home position return
        mov [x]100.0 [y]100.0;       "positioning (100,100)
        inc mvs [x]200.0 f5000000;   "linear interpolation axis x +200.0 (inc. mode)
        mvs [y]200.0;                "axis-y +200.0 (inc. mode)
        mvs [x]-200.0;               " axis-x -200.0  (inc. mode)
        mvs [y]-200.0;               " axis-y -200.0  (inc. mode)
        abs mvs [x]200.0 [y]200.0;   " axis-x,y (200,200)  (abs. mode)
        pln [x] [y];                 "circular interpolation plane
        mcc [x]200.0 [y]200.0 u200.0 v300.0 f5000000;  "circular interpolation
        mvs [x]100.0 [y]100.0;
        end;
```

5

Parameters

This chapter describes the procedure for setting parameters needed to run the MP930 system.

5.1 Description of Parameters

This section describes parameters critical to motion functions in the MP930.

5.1.1 Parameter Classifications

Parameters are specific constants needed for MP930 module motion functions. Set these parameters to values appropriate for machine specifications as well as for applicable Servodrive (Servomotor + Servopack) performance.

Use a CP-717 Programming Device to create and edit parameters.

Refer to *5.1.2 List of Parameters* and *Appendix C Parameter Lists* at the end of this manual for lists of parameters.

Parameters are classified into the following four types.

Classification	Register No.	Description
Fixed Parameters	No registers	These parameters set machine, Servomotor, encoder, and other mechanical conditions. They are not normally changed once they are set, and they cannot be changed while the system is running.
Setting Parameters	OWxx00 to OWxx3F	These parameters are used to provide commands to the servo control section. They can be set from a motion programs or ladder logic programs while the system is running.
Monitor Parameters	IWxx00 to IWxx3F	These parameters are servo monitor data reported by the servo control section. They can serve as reference for a motion programs or ladder logic programs.
Servo Parameters	Cn-0001 to Cn-003F	This are the Servopack user constants that are set on the Motion Parameter Screen.

The following table describes the procedures used to create, edit or change parameters.

Applicable Peripheral Device	Procedure
Personal Computer Programmer	Edits parameters in the Definitions Folder from the Setting Screen.
Motion Program	Uses motion programs to set setting parameters (output registers Owxx00 to Owxx3F) with substitution statements.
Ladder Logic Program	Stores parameters set directly from a ladder logic program.

5.1.2 Parameter Lists

■ Fixed Parameters (Cannot be Changed from a Program)

No.	Name	Size	Setting Range	Meaning	Default
1	Axis selection	1 word	–	This parameter is not used.	0
2	Not used.		–	–	–
3	Encoder selection	1 word	0, 1	0: Incremental encoder 1: Absolute encoder	0
4	Not used.		–	–	–
5	Pulse counting method	1 word	4 to 6	4: A/B pulse × 1 5: A/B pulse × 2 6: A/B pulse × 4	6
6	Not used.	–	–	–	–
7	Rated motor speed	1 word	–	Not used.	3000
8	Number of FB pulses per revolution	1 word	Multiples of 4 in a range from 4 to 65535	1 = 1 pulse	2048
9 to 15	Not used.	–	–	–	–
16	Simulation mode selection	1 word	–	Not used.	0
17	Servo controller function selection flags	1 word	–	–	1
18	No. of digits left of radix point	1 word	0, 1, 2, 3, 4, 5	Minimum reference unit is determined by this parameter and reference unit selection.	3
19	Machine rotation/reference unit	2 words	1 to 2^{31}–1	1 = 1 reference unit	10000
20	Gear ratio, servomotor end	1 word	1 to 65535		1
21	Gear ratio: load end	1 word	1 to 65535		1
22	Maximum infinite length counter value (POSMAX)	2 words	1 to 2^{31}–1	1 = 1 reference unit	360000
23	Maximum absolute encoder rotations	2 words	1 to 2^{31}–1	1 = 1 rotation Set according to the encoder specifications.	99999
24	Positive stored stroke limit	2 words	–2147483648 to 2147483647	1 = 1 reference unit	2^{31}–1
25	Negative stored stroke limit	2 words	–2147483648 to 2147483647	1 = 1 reference unit	-2^{31}
26	Zero point return method	1 word	0 to 3	0: DEC signal + C phase 1: ZERO signal 2: DEC signal + ZERO signal 3: C phase	0
27	Not used.	–	–	–	–

■ Setting Parameters

No.	Name	Register No.	Setting Range	Meaning	Default
1	Operating mode settings	OWxx00	Set by bit	–	–
2	Drive run command settings	OWxx01	Set by bit	–	–
3 to 6	–	–	–	Not used.	–
7	Zero point position offset	OLxx06	0 to $\pm 2^{31}-1$	1 = 1 reference unit	0
8 to 10	–	–	–	Not used.	–
11	Linear acceleration time	OWxx0C	0 to 32767	1 = 1 ms	0
12 to 14	–	–	–	Not used.	–
15	Position loop gain	OWxx10	1 to 32767	1 = 0.1/S	400
16	Feed forward gain	OWxx11	0 to 200	1 = 1%	0
17	Position reference pulse	OLxx12	-2^{31} to $2^{31}-1$	1 = 1 reference unit	0
18	Averaged number of revolutions	OWxx14	0 to 65535	1 = 100 µs	0
19 to 25	–	–	–	Not used.	–
26	Speed loop gain	OWxx1D	1 to 32767	1 = 0.1 Hz	400
27	–	–	–	Not used.	–
28	Motion command code	OWxx20	0 or more	Servopack reference command number	
29	Motion command control flags	OWxx21	Set by bit	–	–
30	rapid traverse speed	OLxx22	0 to $2^{31}-1$	1 = 10^n reference units/min (n: No. of digits left radix point)	0
31	External positioning distance traveled	OLxx24	-2^{31} to $2^{31}-1$	1 = 1 reference unit	0
32	Stopping distance	OLxx26	-2^{31} to $2^{31}-1$	1 = 1 reference unit	0
33	STEP distance traveled	OLxx28	0 to $2^{31}-1$	1 = 1 reference unit	1000
34	–	–	–	Not used.	–
35	Override	OWxx2C	0 to 32767	1 = 0.01%	10000
36	Position control control flags	OWxx2D	–	–	–
37	Workpiece coordinate offset	OLxx2E	-2^{31} to $2^{31}-1$	1 = 1 reference unit	0
38	Preset No. of POSMAX turns	OLxx30	-2^{31} to $2^{31}-1$	1 = 1 reference unit	0
39	Second in-position width	OWxx32	0 to 65535	1 = 1 reference unit	0
40	Zero point output width	OWxx33	0 to 65535	1 = 1 reference unit	10

No.	Name	Register No.	Setting Range	Meaning	Default
41	Positioning completed check time	OWxx34	0 to 65535	1 = 1 ms	0
42	MECHATROLINK servo user constant No.	OWxx35	b0 to b11: 1 to 4095	User constant Cn	0
			b12 to b15: 1 to 2	Number of words	0
43	MECHATROLINK servo user constant	OLxx36	-2^{31} to $2^{31}-1$	–	0
44	Encoder position at power OFF (Low)	OLxx38	-2^{63} to $2^{63}-1$	1 = 1 pulse	0
45	Encoder position at power OFF (High)	OLxx3A			
46	Absolute pulse position at power OFF (Low)	OLxx3C	-2^{63} to $2^{63}-1$	1 = 1 pulse	0
47	Absolute pulse position at power OFF (High)	OLxx3E			

IMPORTANT

Register Numbers

The address underlined in OWxx00 for each axis is the register number OWC000 in the Parameter Table with the following offset added.

Offset for each axis = (Axis No. − 1) × 40H (64 words)

Leading address of setting parameters for each axis = Offset for each axis + OWC000

The following table lists the servo setting parameter register numbers for each axis.

Axis No.	OW Address	Axis No.	OW Address
1	OWC000 to OWC03F	8	OWC1C0 to OWC1FF
2	OWC040 to OWC07F	9	OWC200 to OWC23F
3	OWC080 to OWC0BF	10	OWC240 to OWC27F
4	OWC0C0 to OWC0FF	11	OWC280 to OWC2BF
5	OWC100 to OWC13F	12	OWC2C0 to OWC2FF
6	OWC140 to OWC17F	13	OWC300 to OWC33F
7	OWC180 to OWC1BF	14	OWC340 to OWC37F

5

■ Monitor Parameters

No.	Name	Register No.	Setting Range	Meaning	Remarks
1	Operating status	IWxx00	Set by bit	–	–
2	MECHATROLINK servo status	IWxx01	Set by bit	–	–
3	Calculated position monitored in the machine coordinate system	ILxx02	-2^{31} to $2^{31}-1$	1 = 1 reference unit	CPOS
4	–	ILxx04	–	Not used.	–
5	Latch position monitored in the machine coordinate system	ILxx06	-2^{31} to $2^{31}-1$	1 = 1 reference unit	LPOS
6	Feedback position monitored in the machine coordinate system	ILxx08	-2^{31} to $2^{31}-1$	1 = 1 reference unit	APOS
7 to 10	–	IWxx0A to IWxx0E	–	Not used.	–
11	Parameter No. out of range	IWxx0F	1 to 47 101 to 127	–	–
12, 13	–	IWxx10 to IWxx13	–	Not used.	–
14	Motion command response code	IWxx14	0 to 65535	Motion command currently executing.	–
15	Motion command status	IWxx15	Set by bits	–	–
16	No. of digits left of radix point	IWxx16	0 to 5	Same as fixed parameter 18.	–
17	Position control status	IWxx17	Set by bit	–	–
18	Reference position in the machine coordinate system	ILxx18	-2^{31} to $2^{31}-1$	1 = 1 reference unit	MPOS
19	–	ILxx1A	–	Not used.	–
20	POSMAX monitoring	ILxx1C	1 to $2^{31}-1$	1 = 1 reference unit	
21	No. of POSMAX turns	ILxx1E	-2^{31} to $2^{31}-1$	1 = 1 rotation	
22	Monitor data for the servo drive user	ILxx20	-2^{31} to $2^{31}-1$	–	–
23	Alarm	ILxx22	Set by bits	Servo-related alarms	–
24	Servo driver alarm code	IWxx24		Servo Amp alarm codes	–
25	Servo driver I/O monitoring	IWxx25	Set by bits	–	–
26	Speed reference output monitoring	ILxx26	-2^{31} to $2^{31}-1$	1 = 1 reference unit/s, pulse/s	–
27	MECHATROLINK servo user constant monitoring	ILxx28	-2^{31} to $2^{31}-1$	–	–

No.	Name	Register No.	Setting Range	Meaning	Remarks
28	–	IWxx30 to IWxx37	–	Not used.	–
29	Encoder position at power OFF (Low)	ILxx38	-2^{31} to $2^{31}-1$	1 = 1 pulse	For ABS system infinite length position control
30	Encoder position at power OFF (High)	ILxx3A			
31	Absolute pulse position at power OFF (Low)	ILxx3C	-2^{31} to $2^{31}-1$	1 = 1 pulse	For ABS system infinite length position control
32	Absolute pulse position at power OFF (High)	ILxx3E			

IMPORTANT

Register Numbers

The address underlined in IWxx00 for each axis is the register number IWC000 in the Parameter Table with the following offset added.

Offset for each axis = (Axis No. – 1) × 40H (64 words)

The leading address of the monitor parameters for each axis = Offset for each axis + IWC000

The following table lists the servo setting parameter register numbers for each axis.

Axis No.	IW Address	Axis No.	IW Address
1	IWC000 to IWC03F	8	IWC1C0 to IWC1FF
2	IWC040 to IWC07F	9	IWC200 to IWC23F
3	IWC080 to IWC0BF	10	IWC240 to IWC27F
4	IWC0C0 to IWC0FF	11	IWC280 to IWC2BF
5	IWC100 to IWC13F	12	IWC2C0 to IWC2FF
6	IWC140 to IWC17F	13	IWC300 to IWC33F
7	IWC180 to IWC1BF	14	IWC340 to IWC37F

■ User Constants

No.	Name	Size	Units	Range	Default Setting
Cn-0001	Memory switches 1	2	bits		0380H
Cn-0002	Memory switches 2	2	bits		0000H
Cn-0003	Load inertia	2	%	0-65535	100
Cn-0004	Speed loop gain	2	0.1 Hz	1-20000	400
Cn-0005	Speed loop integration time constant	2	0.01 ms	100-65535	2000
Cn-0006	Emergency stop torque	2	%	0-MAX	MAX
Cn-0007	Positioning proximity detection width	2	Reference units	0-10000	10
Cn-0008	Positive torque limit	2	%	0-MAX	MAX
Cn-0009	Negative torque limit	2	%	0-MAX	MAX
Cn-000A	Reserved by system.	2	–	–	2048
Cn-000B	Reserved by system.	2	–	–	0000H
Cn-000C	Mode switch: Torque reference	2	%	0-32767	200
Cn-000D	Mode switch: Speed reference	2	r/min	0-32767	0
Cn-000E	Mode switch: Acceleration	2	0.167 r/s^2	0-3000	0
Cn-000F	Mode switch: Error pulse	2	pulses	0-10000	0
Cn-0010	Reserved by system.	2	–	–	0000H
Cn-0011	No. of encoder pulses	2	P/R	513-32767	2048
Cn-0012	Brake timing for Servomotor stop (delay from reference to SVOFF)	2	10 ms	0-50	0
Cn-0013	Memory switches 3	2	bits	–	0000H
Cn-0014	Memory switches 4	2	bits	–	0000H
Cn-0015	Brake timing with servomotor running: Reference output speed)	2	r/min	0-MAX	100
Cn-0016	Brake timing with servomotor running: Wait time from SVOFF to reference)	2	10 ms	10-100	50
Cn-0017	Torque reference filter time constant	2	μs	0-25000	400
Cn-0018	Secondary torque reference filter time constant	2	μs	0-25000	0
Cn-0019	Reserved by system.	2	–	–	0000H
Cn-001A	Position loop gain	2	0.01/s	1-50000	4000
Cn-001B	Positioning completed width	2	Reference units	0-250	7
Cn-001C	Bias	2	100 reference units/s	0-MAX	0
Cn-001D	Feed forward compensation	2	%	0-100	0
Cn-001E	Position error overflow range	2	Reference units	1-65535	65535

No.	Name	Size	Units	Range	Default Setting
Cn-001F	First level linear acceleration/deceleration time constant	2	1,000 reference units/s^2	0-65535	0
Cn-0020	Second level linear acceleration/deceleration time constant	2	10,000 reference units/s^2	0-65535	100
Cn-0021	Acceleration/deceleration time constant switching speed	2	1,000 reference units/s	0-65535	0
Cn-0022	Zero point return approach speed 1	2	1,000 reference units/s	0-65535	50
Cn-0023	Zero point return approach speed 2	2	1,000 reference units/s	0-65535	5
Cn-0024	Electronic gear ratio, numerator	2		1-32768	4
Cn-0025	Electronic gear ratio, denominator	2		1-32768	1
Cn-0026	Average move time	2	100 μs	0-5100	0
Cn-0027	Feed forward reference filter	2	μs	0-64000	0
Cn-0028	Final distance traveled to zero point return	4	reference units	–2147483648 to 2147483647	100
Cn-002A	Zero point position range	2	reference units	0-65535	10
Cn-002B	Final distance traveled to external positioning	4	reference units	–2147483648 to 2147483647	100
Cn-002D	Exponential acceleration speed bias	2	reference units/s	0-32767	0
Cn-002E	Exponential acceleration time constant	2	100 μs	0-5100	0
Cn-002F	Forward direction software limit	4	reference units	–2147483648 to 2147483647	8192 × 99999
Cn-0031	Reverse direction software limit	4	reference units	–2147483648 to 2147483647	8192 × 99999
Cn-0033	Absolute encoder zero point position offset	4	reference units/s	–2147483648 to 2147483647	0
Cn-0035	–	2	–	–	0000H
Cn-0036	Reserved by system.	2	–	–	0000H
Cn-0037	Motor selection (SGDB only)	2	–	0 to 255	255
Cn-0038	PG power supply voltage change (SGDB only)	2	0.1 mV	52000 to 58000	52500
Cn-0039	Reserved by system.	2	–	–	0000H
Cn-003A	Reserved by system.	2	–	–	0000H
Cn-003B	Reserved by system.	2	–	–	0000H
Cn-003C	Reserved by system.	2	–	–	0000H
Cn-003D	Reserved by system.	2	–	–	0000H
Cn-003E	Reserved by system.	2	–	–	0000H
Cn-003F	Reserved by system.	2	–	–	0000H

IMPORTANT

Do not normally use the electronic gear ratio parameters on the Servopack, but set the Cn-0024 and Cn-0025 to 1 during set up. When the electronic gear ratio on Servopack is 1:1, the reference unit on the Servopack is in pulse.

■ Memory Switch Bit Details

The following describes individual memory switch bits (bit user constants) from the list of Servopack user constants.

Cn-001: Memory Switches 1

Cn-001: The following table describes the bits in memory switches 1.

Bit	Name	Description	Default
0	SV_ON mask	0: SV_ON/SV_OFF enabled 1: Always SV_ON	0
1	SENS_ON mask	0: SENS_ON/SENS_OFF enabled 1: Always SENS_ON	0
2	P-OT mask	0: P-OT enabled 1: P-OT signal mask (Always disabled)	0
3	N-OT mask	0: N-OT enabled 1: N-OT signal mask (Always disabled)	0
4	–		0
5	Power outage mask	0: Servo alarm after recovery from power outage 1: Power outage mask (No servo alarm with power outage recovery)	0
6	Base block power outage prevention method	0: Dynamic brake (DB) stop 1: Free run stop	0
7	Status after dynamic brake stop	0: Cancel dynamic brake 1: Do not cancel dynamic brake	1
8	Operation with OT stop	0: Stop according to bit 6 setting 1: Decelerate to a stop using emergency stop torque	1
9	Operation after decelerating to a stop using OT emergency stop torque	0: Servo OFF after decelerating to a stop 1: Zero clamp after decelerating to a stop	1
A	Position error with servo OFF	0: Clear position error 1: Hold position error	0
B	Mode switch function	0: Mode switch function enabled (according to bits C and D) 1: Mode switch function disabled	0
C	Mode switch selection	00: Mode switch selection: Internal torque reference 01: None (Do not use this setting.) 10: Mode switch selection: Acceleration 11: Mode switch selection: Error pulse	0
D			0

Bit	Name	Description	Default
E	Encoder selection	0: Incremental encoder 1: Absolute encoder	0
F	–		0

IMPORTANT Never change the default setting of bits with a dash (–) in the name column.

Cn-002: Memory Switches 2

Cn-002: The following table describes the bits in memory switches 2.

Bit	Name	Description	Default
0	Reverse rotation mode	0: Sets counterclockwise as the forward direction 1: Sets clockwise as the forward direction,	0
1	Zero point error detection mask	0: Sets zero point error detection (only with an absolute encoder) 1: Zero point detection mask (no detection)	0
2	–	–	0
3	–	–	0
4	–	–	0
5	–	–	0
6	Software limit check by reference target	0: No check 1: Check	0
7	–	–	0
8	Servomotor selection	0: SGM 1: SGMP	0
9	–	–	0
A	–	–	0
B	–	–	0
C	–	–	0
D	–	–	0
E	–	–	0
F	–	–	0

IMPORTANT Never change the default setting of bits with a dash (–) in the name column.

Cn-0013: Memory Switches 3

Cn-0013: The following table describes the bits in memory switches 3.

Bit	Name	Description	Default
0	–	–	0
1	–	–	0
2	–	–	0
3	–	–	0
4	–	–	0
5	–	–	0
6	–	–	0
7	–	–	0
8	–	–	0
9	–	–	0
A	Reserved by system.*	–	0
B	Reserved by system.*	–	0
C	–	–	0
D	–	–	0
E	–	–	0
F	–	–	0

* See *7.3.3* in the Σ-*Series SGM□/SGD-□N User's Manual* (SIE-S800-26.3).

IMPORTANT

Never change the default setting of bits with a dash (–) in the name column.

Cn-0014: Memory Switches 4

Cn-0014: The following table describes the bits in memory switches 4.

Bit	Name	Description	Default
0	–	–	0
1	Zero point return direction	0: Forward 1: Reverse	0
2	P-SOT mask	0: P-SOT enabled 1: P-SOT disabled	0
3	N-SOT mask	0: N-SOT enabled 1: N-SOT disabled	0
4	–	–	0
5	–	–	0
6	–	–	0
7	–	–	0
8	–	–	0
9	Brake operation	0: Operate from the BRK-ON/BRK_OFF command 1: Operation from the Servopack (BRK-ON/BRK_OFF disabled)	0
A	P-OT signal	0: Positive logic 1: Negative logic	0
B	N-OT signal	0: Positive logic 1: Negative logic	0
C	DEC signal	0: Positive logic 1: Negative logic	0
D	–	–	0
E	–	–	0
F	–	–	0

IMPORTANT

1. Never change the default setting of bits with a dash (–) in the name column.
2. Set bits 2 and 3 of Servopack user constant Cn-0014 to 1 to disable P-SOT and N-SOT.
3. Set "1: Operation from the Servopack" for brake operation. The brake signal is output in sequence with Servopack ON/OFF.

■ Relationship of Servopack User Constants to MP930 Parameters

Some MP930 controller parameters and Servo Amp user constants have the same function. Set these parameters carefully.

List of Parameters Requiring Special Attention

The following table shows parameters with the same function.

MP930	Servopack
OWxx10: Position loop gain	Cn-000A: Position loop gain
OWxx11: Feed forward gain	Cn-001D: Feed forward compensation
OWxx1D: Speed loop gain	Cn-0004: Speed loop gain
OWxx0C: Linear acceleration time	Cn-0020: Second level linear acceleration/deceleration time constant
OWxx14: Averaged number of revolutions	Cn-0026: Average move time
Fixed parameter 3: Encoder selection	Cn-0001 bE: Encoder selection
Fixed parameter 8: Number of FB pulses per revolution	Cn-0011: No. of encoder pulses
Fixed parameter 21: Gear ratio, load end	Cn-0024: Electronic gear ratio, numerator
Fixed parameter 20: Gear ratio, servomotor end	Cn-0025: Electronic gear ratio, denominator
Fixed parameter 17 b7: Forward Stored Stroke Limit Function Selection	Cn-0014 b2: 0: P-SOT mask
Fixed parameter 17 b8: Reverse Stored Stroke Limit Function Selection	Cn-0014 b3: N-SOT mask
Fixed parameter 24: Positive stored stroke limit	Cn-002F: Forward direction soft limit
Fixed parameter 25: Negative stored stroke limit	Cn-0031: Reverse direction soft limit
OWxx33: Zero point output width	Cn-002A: Zero point position range

Parameters Motion Programs Can Write

The following Servopack user constants can be written from a motion program. (Servopack user constants are simultaneously written whenever setting parameters are written from a motion program).

Parameter Name	Motion Program Format		MP930	Servopack
Second Level Linear Acceleration/Deceleration Time Constant	ACC[X]6000;	→	OWxx0C	Cn-0020
Average Move Time	SCC[X]6000;	→	OWxx14	Cn-0026

5

Parameters MP930 Motion Commands Can Write

Motion commands can be used for the following parameters to write settings on the Controller to the Servopack.

Parameter Name	Controller	Servopack
Speed Loop Gain	OWxx1D	Cn-0004
Position Loop Gain	OWxx10	Cn-001A
Feed Forward Compensation	OWxx11	Cn-001D

The following procedure must be used to change parameters.

◀EXAMPLE▶

- Example showing the procedure for writing position loop gain from a motion program.

```
WHILE OWxx20<>;        □Check to see if motion command OWxx20 is set to 0 (NOP).
EOX;                   □ONE SCAN WAIT command
WEND;
OWxx10 = 200;          □Position loop gain: Store the value at OWxx10.
OWxx20 = 15;           □Set motion command OWxx20 to 15 (KPS command).
WHILE IWxx14<>15;      □Wait until the command response is 15 (KPS command).
EOX;
WEND;
OWxx20 = 0             □Set motion command OWxx20 to 0 (NOP).
```

arameters that Must Be the Same for MP930 and Servopack

Motion control will not function properly if the following parameters are not the same.

Parameter Name	MP930	Servopack
Encoder Selection	Fixed Parameter 3	Cn-0001 bE
No. of Encoder Pulses	Fixed Parameter 8	Cn-0011

Parameters Set Either on Controller or Servopack

Motion control will not function properly if both the following sets of parameters are used at the same time.

Parameter Name	MP930	Servopack
Electronic Gear Ratio, Numerator	Fixed Parameter 20	Cn-0024
Electronic Gear Ratio, Denominator	Fixed Parameter 21	Cn-0025

IMPORTANT

Do not normally use the electronic gear ratio parameters on the Servopack, but set the Cn-0024 and Cn-0025 to 1 during set up.
When the electronic gear ratio on the Servopack is 1:1, the reference unit on the Servopack is in pulse.

Servopack Parameters That Must Not Be Used

Parameter Name	MP930	Servopack
P-SOT Mask	Fixed Parameter 17 b7	Cn-0014 b2
N-SOT Mask	Fixed Parameter 17 b8	Cn-0014 b3
Forward Direction Software Limit	Fixed Parameter 24	Cn-002F
Reverse Direction Software Limit	Fixed Parameter 25	Cn-0031

IMPORTANT Set P-SOT and N-SOT masks on the Servopack by setting Cn-0014 b2 and b3 to 1 during set up.

Parameters That Look Similar but Are Different

MP930	Servopack
Zero Point Output Width: OWxx33	Zero Point Position Range: Cn-002A

The MP930 parameter is used for zero point position output.

5.2 Parameter Settings

5.2.1 Fixed Parameters

The following table describes the fixed parameter settings used to set machine, Servomotor, encoder, and other mechanical conditions for the controlled axes.

Be sure to look through these because they are basic parameters. The following table describes the settings of these parameters.

Parameter No.	Name	Description	Default
1	Axis selection	Not used.	0
3	Encoder selection	Selects the type of encoder that is used. 0: Incremental encoder 1: Absolute encoder	0
5	Pulse counting method	Selects the multiplier for the number of encoder pulses. 4: A/B pulse × 1 5: A/B pulse × 2 6: A/B pulse × 4 **Note** Set either 4, 5 or 6 above as the pulse counting method; the system will not function properly if 1, 2 or 3 is set.	6
7	Rated motor speed	Not used.	3000
8	Number of FB pulses per revolution	Sets the number of feedback pulses per one Servomotor rotation (no multiplier). Set this parameter based on the specifications of the encoder that is used. • Setting range: 4 to 65532 [pulses] **Note:** Set a multiple of 4.	2048
16	Simulation mode selection	Not used.	0

5

<table>
<tr><th>Parameter No.</th><th>Name</th><th colspan="2">Description</th><th>Default</th></tr>
<tr><td rowspan="9">17</td><td rowspan="9">Servo controller function selection flags</td><td>b0 to b3
0 to 7:</td><td>Reference unit selection range
Sets reference unit for positioning control.
0: pulse (Electronic gear disabled)
1: mm
2: deg
3: inch</td><td>1</td></tr>
<tr><td>b4:</td><td>Electronic gear
Enables or disables the electronic gear in the servo controller. Use in combination with Reference Unit Selection.
0: Disabled
1: Enabled
Note: The electronic gear will be disabled even if this flag is enabled if Reference Unit Selection = pulse.</td><td>0</td></tr>
<tr><td>b5:</td><td>Finite length/infinite length axis selection
Determines whether or not there is a limit on controlled axis movement.
<table>
<tr><th>b5</th><th>Meaning</th><th>Reference Range</th></tr>
<tr><td>0</td><td>• Sets finite length axis.
• The axis will have limited movement.
• The stored stroke limit function is enabled.</td><td>-21^{31} to $2^{31}-1$</td></tr>
<tr><td>1</td><td>• Sets infinite length axis.
• The axis will have unlimited movement.
• The stored stroke limit function is disabled.</td><td>–</td></tr>
</table></td><td>0</td></tr>
<tr><td>b6:</td><td>Not used.</td><td>0</td></tr>
<tr><td>b7:</td><td>Positive stored stroke limit function selection
Enables or disables the stored stroke limit function in the positive direction.
0: Disabled, 1: Enabled</td><td>0</td></tr>
<tr><td>b8:</td><td>Negative stored stroke limit function selection
Enables or disables the stored stroke limit function in the negative direction.
0: Disabled, 1: Enabled</td><td>0</td></tr>
<tr><td>b9:</td><td>Override function selection
Enables or disables the speed override function.
0: Disabled, 1: Enabled
The OWxx2C: Override setting is enabled if this parameter is enabled. Override is fixed at 100% if this parameter is disabled.</td><td>0</td></tr>
<tr><td>b10 to b15:</td><td>Not used.</td><td>0</td></tr>
</table>

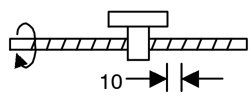
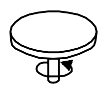
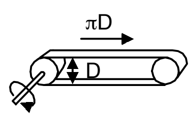
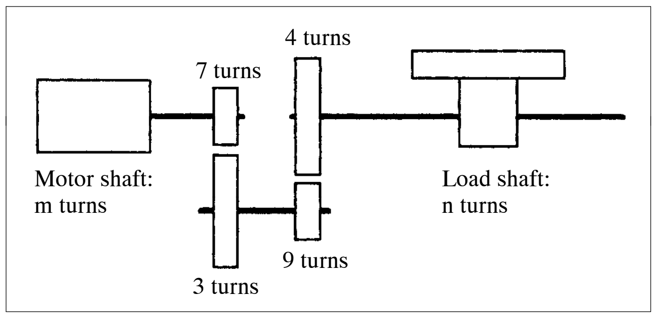
<table>
<tr><th>Parameter No.</th><th>Name</th><th colspan="2">Description</th><th>Default</th></tr>
<tr><td>18</td><td>No. of digits left of radix point</td><td colspan="2">The minimum reference unit is determined by this parameter and Reference Unit Selection in the servo controller function selection flags (b0 to b3).
Table 5.1 shows the parameter settings.</td><td>3</td></tr>
<tr><td rowspan="4">19</td><td rowspan="4">Machine rotation/reference unit</td><td colspan="2">Sets the amount a load moves (reference units) per load axis rotation.
• Setting range: 1 to 2^{31} – 1</td><td rowspan="3">10000</td></tr>
<tr><td>Ball screw

Ball screw pitch = 10 mm</td><td>Ball screw pitch = 10 mm
Reference Unit Selection = mm
Number of digits left of t radix point = 3
↓
Set the amount of movement per machine 1 rotation to 10000.</td></tr>
<tr><td>Rotating table

One rotation = 360°</td><td>One table rotation = 360°
Reference Unit Selection = deg
Number of digits left of radix point = 3
↓
Amount of movement per machine 1 rotation to 360000</td></tr>
<tr><td></td><td>One roller rotation = 360°
Reference Unit Selection = mm
Number of digits left of radix point = 3
↓
Set the amount of movement per machine 1 rotation to πD × 1000.</td><td>10000</td></tr>
</table>

Table 5.1 Parameter for Number of Digits Left of Radix Point

Parameter Setting	Reference Unit (pulse)	Reference Unit (mm)	Reference Unit (deg)	Reference Unit (inch)
No. of digits left of radix point = 0	1 pulse	1 mm	1°	1″
No. of digits left of radix point = 1	1 pulse	0.1 mm	0.1°	0.1″
No. of digits left of radix point = 2	1 pulse	0.01 mm	0.01°	0.01″
No. of digits left of radix point = 3	1 pulse	0.001 mm	0.001°	0.001″
No. of digits left of radix point = 4	1 pulse	0.0001 mm	0.0001°	0.0001″
No. of digits left of radix point = 5	1 pulse	0.00001 mm	0.00001°	0.00001″

5

<table>
<tr><th>Parameter No.</th><th>Name</th><th>Description</th><th>Default</th></tr>
<tr><td>20</td><td>Gear ratio, servomotor end</td><td rowspan="2">The gear ratio between the motor and load is determined by these parameters. The following two values are set for a configuration in which the load shaft will turn n times in response to m turns of the motor shaft.
• Gear ratio, servomotor end: m
• Gear ratio, load end: n
Setting Example


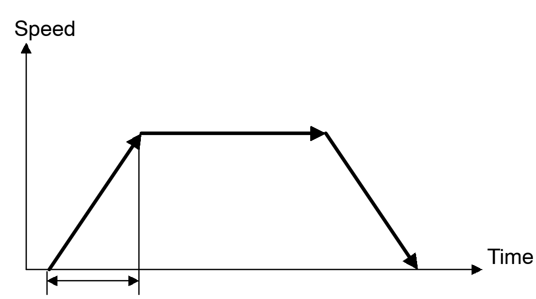
In the above example, the reduction ratio is n/m, or 3/7 x 4/9 = 4/21. The following settings would thus be made.
Gear ratio, servomotor end: 21
Gear ratio, load end: 4</td><td>1</td></tr>
<tr><td>21</td><td>Gear ratio, load end</td><td>1</td></tr>
<tr><td>22</td><td>Maximum infinite length counter value (POSMAX)</td><td>Set the period of an infinite axis in reference units. This parameter is valid when an infinite length axis has been set in the fixed parameters.
• Setting range: 1 to $2^{31} - 1$</td><td>360000</td></tr>
<tr><td>23</td><td>Maximum absolute encoder rotations</td><td>Set the maximum number of rotations handled by the absolute encoder. This parameter is valid when an absolute encoder is being used and an infinite length axis has been set. Set according to the encoder specifications.
• Setting range: 1 to $2^{31} - 1$</td><td>99999</td></tr>
<tr><td>24</td><td>Positive stored stroke limit</td><td rowspan="2">Set the positions at which the stored stroke limit function is to operate on the machine coordinate system.
• Setting range: -2^{31} to $2^{31} - 1$ (1 = 1 reference unit)
Whether or not the stored stroke limit function is used is set in bits b7 and b8 of the Servo Controller Function Selection flags in the fixed parameters.</td><td>$2^{31}-1$</td></tr>
<tr><td>25</td><td>Negative stored stroke limit</td><td>-2^{31}</td></tr>
<tr><td>26</td><td>Zero point return method</td><td>0: DEC signal + C-phase
1: ZERO signal
2: DEC signal + ZERO signal
3: C-phase
• Setting range: 0 to 3</td><td>0</td></tr>
</table>

5.2.2 Setting Parameters

The following table describes parameters that need to be set to execute functions. Be sure to look through these because they are basic parameters.

The following tables describes the settings of these parameters.

<table>
<tr><th>Parameter No.</th><th>Name</th><th>Register No.</th><th colspan="3">Description</th><th>Default</th></tr>
<tr><td rowspan="7">1</td><td rowspan="7">Axis selection</td><td rowspan="7">OWxx00</td><td colspan="4">Sets the Control Mode. The only valid bits are b6 and b9, and all others are invalid even if they are set.</td></tr>
<tr><td colspan="3">b6: ACR: Alarm clear
The following monitor parameters will clear when this bit turns ON.</td><td>0</td></tr>
<tr><td>Name</td><td>Register No.</td><td colspan="2">Meaning</td></tr>
<tr><td>Operating Status</td><td>IWxx00</td><td colspan="2">b1: Servo parameter setting error
b2: Servo fixed parameter setting error</td></tr>
<tr><td>Parameter no. out of range</td><td>IWxx0F</td><td colspan="2">1 to 99: Setting parameter error
101 to 109: Fixed parameter</td></tr>
<tr><td>Alarm</td><td>ILxx22</td><td colspan="2">See the list of monitor parameters for more details.</td></tr>
<tr><td colspan="3">b9: ZRNDIR zero point return direction
Sets the direction for returning to the zero point.
0: Reverse direction
1: Forward direction</td><td>0</td></tr>
<tr><td rowspan="2">2</td><td rowspan="2">Drive run command</td><td rowspan="2">OWxx01</td><td>b0:
(RUN/DO0)</td><td colspan="2">Servo ON signal or general-purpose DO. Turns ON the servo power supply when this parameter goes from OFF to ON. Turns it OFF when this parameter goes from ON to OFF.</td><td>0</td></tr>
<tr><td>b13:
(SPDTYPE)</td><td colspan="2">Rapid traverse reference type
Sets rapid traverse speed parameter selection.
0: Use OLxx22
Unit: 10^{n} reference units/min</td><td>0</td></tr>
<tr><td>3 to 6</td><td>–</td><td>–</td><td colspan="3">Not used.</td><td>–</td></tr>
</table>

<table>
<tr><th>Parameter No.</th><th>Name</th><th>Register No.</th><th colspan="3">Description</th><th>Default</th></tr>
<tr><td rowspan="6">7</td><td rowspan="6">Zero point position offset</td><td rowspan="6">OLxx06</td><td colspan="3">This parameter is used to set the zero point position for the machine coordinate system. The significance of the parameter varies with the type of encoder used and by finite length/infinite length axis selection.</td><td rowspan="6">0</td></tr>
<tr><td rowspan="2">Finite length axis</td><td>INC axis</td><td>ABSOFF always enabled.</td></tr>
<tr><td>ABS axis</td><td>ABSOFF always enabled.</td></tr>
<tr><td rowspan="2">Infinite length axis</td><td>INC axis</td><td>ABSOFF always enabled.</td></tr>
<tr><td>ABS axis</td><td>Enabled only when setting the zero point (Used to define ABS system infinite length position control data)</td></tr>
<tr><td colspan="3">• Setting range: 0 to ±231–1 [Reference units]</td></tr>
<tr><td>8 to 10</td><td>–</td><td>–</td><td colspan="3">Not used.</td><td>–</td></tr>
<tr><td>11</td><td>Linear acceleration time</td><td>OWxx0C</td><td colspan="3">Sets the linear acceleration time for rapid traverse speed. The same value is used for deceleration time and is valid when the ACC motion command is executed.


• Setting range: 0 to 32767 [ms]</td><td>0</td></tr>
<tr><td>12 to 14</td><td>–</td><td>–</td><td colspan="3">Not used.</td><td>–</td></tr>
</table>

Parameter No.	Name	Register No.	Description	Default
15	Position loop gain	OWxx10	Sets position loop gain for the servo system. Position loop gain is needed to set response performance for the servo system and is valid when the KPS motion command is executed. The following are setting guidelines. 40 to 250 Excellent response (Watch for hunting.) Ordinary response Set a value appropriate for machine rigidity and inertia as well as the type of Servomotor. • Setting range: 1 to 32767 [0.1/s]	400
16	Feed forward gain	OWxx11	Reduces positioning time by applying feed forward control. • Setting range: 0 to 200 [%] Position reference and actual position errors decrease with higher settings. The machine may start to vibrate if the setting is too high.	0
17	Position reference pulse	OLxx12	It does not set the reference value (target position) as is, but rather adds a value to XREF to create a new XREF that is used to determine the amount of incremental movement. • Setting range: -2^{31} to 2^{31} (1 = 1 reference unit)	0
18	Averaged number of revolutions	OWxx14	• Setting range: 0 to 65535 (1 = 1 ms)	0
19 to 25	–	OWxx15 to OWxx1C	Not used.	–
26	Speed loop gain	OWxx1D	This parameter is proportional gain for the speed controller. Set the parameter at 400 or less when operating the Servomotor under no-load conditions. • Setting range: 1 to 32767 (1 = 0.1 Hz)	400
27	–	–	Not used.	–

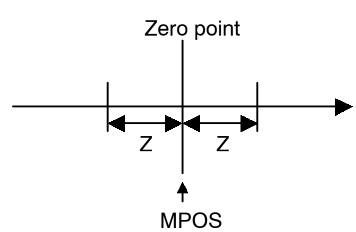
Parameter No.	Name	Register No.	Description		Default
28	Motion command code	OWxx20	Set the following command numbers for this parameter to send commands to the MECHATROLINK servo.		0
			0: NOP	No command	
			1: POSING	Positioning	
			2: EX_POSING	External positioning	
			3: ZRET	Zero point return	
			4: INTERPOLATE	Interpolation	
			5: ENDOF_ INTERPOLATE	Interpolation end segment	
			6: LATCH	Interpolation with position detection functions	
			7: FEED	Fixed speed feed	
			8: STEP	Fixed length feed	
			9: ZSET	Zero point setting	
			10: ACC	Change the acceleration time	
			11	–	
			12: SCC	Change the filter time constant	
			13: CHG_FILTER	Change the filter type	
			14: KVS	Change the speed loop gain (kv)	
			15: KPS	Change the position loop gain (kp)	
			16: KFS	Change the feed forward (kf)	
			17: CN_RD	Read the MECHATROLINK servo user constant	
			18: CN_WR	Write the MECHATROLINK servo user constant	
			19: ALM_MON	Monitor the MECHATROLINK servo alarms	
			20: ALMHIST_MON	Monitor the MECHATROLINK servo alarm history	
			21: ALMHIST_CLR	Clear the MECHATROLINK servo alarm history	

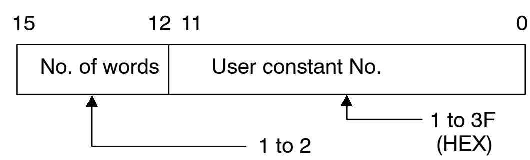
5

Parameter No.	Name	Register No.	Description		Default
29	Motion command control flags	OWxx21	b0: HOLD	Hold the command	0
			b1: ABORT	Abort the command	
			b2: DIRECTION	Direction of movement (0: Forward, 1: Reverse)	
			b3: P_PI	P and PI speed loop control switching (0: PI)	
			b4 to b7: FILTER-TYPE	Filter type selection 0: No filter 1: Exponential filter 2: Average move filter Note: Execute CHG-FILTER command (OWxx20 = 13) after having changed the filter type.	
			b8 to b15:	Not used.	
30	Rapid traverse speed	OLxx22	This speed parameter is used for move commands like JOG/STEP, MVE, and MVT. Setting unit: 1 = 10^n reference units/min (n: Fixed parameter: No. of digits left of radix point) 1 = 10^3 reference units/min when electronic gear is disabled • Setting range: 0 to $2^{31} - 1$ Related Parameters • Fixed parameter 18: No. of digits left of radix point • Setting parameter OWxx2C: Override		0
31	External positioning distance traveled	OLxx24	Sets the amount of movement after an external positioning command signal is input. • Setting range: -2^{31} to $2^{31} - 1$ (1 = 1 reference unit)		0
32	Stopping distance	OLxx26	This is the distance used to decelerate to a stop to provide a software limit function when an interpolation-related command is executed. Since the parameter is set automatically when an interpolation-related command is executed, users do not need to set this parameter directly. • Setting range: -2^{31} to $2^{31} - 1$ (1 = 1 reference unit)		0
33	STEP distance traveled	OLxx28	Sets the amount of movement for the STEP command. • Setting range: 0 to $2^{31} - 1$ • Unit: Reference unit		1000
34	–	–	Not used.		–

Parameter No.	Name	Register No.	Description		Default
35	Override	OWxx2C	Sets an override percentage with respect to setting parameter OWxx22: Rapid traverse speed in 0.01% units. • Setting range: 0 to 32,767 (1 = 1 0.01%) Related Parameters • Setting parameter OLxx22: Rapid traverse speed		10000
36	Position control control flags	OWxx2D	b0: MLK	Sets Machine Lock Mode. • Turns the machine lock function for position control in the controller ON and OFF. • The axis does not actually move, but rather the current calculated value is updated instead in Machine Lock Mode.	0
			b1: TPRSREQ	Request for preset number of POSMAX turns. • With an infinite length axis, a turn is counted every time the position value exceeds POSMAX and the count is stored at monitor parameter ILxx1E: No. of POSMAX turns. • The number of turns can be preset at setting parameter OLxx30: Preset data for the number of POSMAX turns by turning ON the preset number of POSMAX turns request flag. Related Parameters • Fixed parameter 22: Maximum value for an infinite length counter • Setting parameter OLxx30: Preset data for the number of POSMAX turns • Monitor parameter ILxx1E: Number of POSMAX turns.	0
			b2: ABSLDREQ	ABS system infinite length position control data LOAD request If this flag is ON during infinite length axis selection in an ABS system, then position data in the controller will be initialized to position data that was used prior to turning power OFF. Related Parameters • Setting parameter OLxx38 to OLxx3E: ABS system infinite length position control data	0
			b3 to b11:	Not used.	0

<table>
<tr><th>Parameter No.</th><th>Name</th><th>Register No.</th><th colspan="4">Description</th><th>Default</th></tr>
<tr><td rowspan="19">36</td><td rowspan="19">Position control control flags (continued)</td><td rowspan="19">OWxx2D</td><td rowspan="19">b12 to b15:
USRMON SEL</td><td colspan="3">Servo user monitor data selection
The following position data in a MECHATROLINK servo can be monitored if specified by this parameter. Data is monitored in ILxx20.</td><td rowspan="19">0</td></tr>
<tr><td>Code</td><td>Description</td><td>Unit</td></tr>
<tr><td>0</td><td>Reference position in the reference coordinate system</td><td>pulse</td></tr>
<tr><td>1</td><td>Machine reference position in the machine coordinate system</td><td>pulse</td></tr>
<tr><td>2</td><td>Position error</td><td>pulse</td></tr>
<tr><td>3</td><td>Feedback position for the machine coordinate system</td><td>pulse</td></tr>
<tr><td>4</td><td>Counter latch position for the machine coordinate system</td><td>pulse</td></tr>
<tr><td>5</td><td>Internal reference position in the reference coordinate system</td><td>pulse</td></tr>
<tr><td>6</td><td>Final target positioning in the reference coordinate system</td><td>pulse</td></tr>
<tr><td>7</td><td>–</td><td></td></tr>
<tr><td>8</td><td>Feedback speed</td><td>pulse/s</td></tr>
<tr><td>9</td><td>Reference speed</td><td>pulse/s</td></tr>
<tr><td>A</td><td>Final target reference speed</td><td>%</td></tr>
<tr><td>B</td><td>Torque reference</td><td>%</td></tr>
<tr><td>C</td><td>–</td><td></td></tr>
<tr><td>D</td><td>–</td><td></td></tr>
<tr><td>E</td><td>Option monitor 1</td><td></td></tr>
<tr><td>F</td><td>Option monitor 2</td><td></td></tr>
<tr><td colspan="3">Note: The units are when the electronic gear ratio is 1:1.</td></tr>
</table>

Parameter No.	Name	Register No.	Description	Default
37	Workpiece coordinate offset	OLxx2E	This parameter stores offset when a POS motion command is specified. • Setting range: -2^{31} to $2^{31} - 1$ (1 = 1 reference unit)	0
38	Preset No. of POSMAX turns	OLxx30	With an infinite length axis, a turn is counted every time the position value exceeds POSMAX and the count is stored at monitor parameter ILxx1E: No. of POSMAX turns. The number of turns can be preset by turning ON setting parameter OWxx32D b1: Preset request for the number of POSMAX. • Setting range: -2^{31} to $2^{31} - 1$ (1 = 1 reference unit) Related Parameters • Fixed parameter 22: Maximum value for an infinite length counter • Setting parameter OWxx2D b1: Preset data for the number of POSMAX turns • Monitor parameter ILxx1E: Number of POSMAX turns.	0
39	Second in-position width	OWxx32	Turns ON monitor parameter b2: Second INP completed if the difference between the reference position and the feedback position after distribution has been completed is within the range set for this parameter. The parameter can be set using an INP motion command. • Setting range: 0 to 65535 (1 = 1 reference unit)	0
40	Zero point output width	OWxx33	Turns to position control status IWxx17 b1, ZERO = ON if the absolute value for the machine coordinate system reference position (MPOS) is within the zero point position output width OWxx33 after the machine returns to the zero point (motion command status IWxx15 b6, ZRNC = ON).  • Setting range: 0 to 65535 (1 = 1 reference unit) Related Parameters • Monitor parameter ILxx17 b1: Zero point position	0

Parameter No.	Name	Register No.	Description	Default
41	Positioning completion check time	OWxx34	Sets the time from the completion of distrubution to the completion of positioning. • An alarm will be generated if positioning is not completed within the time set here. → Positioning time exceeded • The completion of positioning will not be checked if this parameter is set to 0. • Setting range: 0 to 65535 (1 = 1 ms)	0
42	MECHATROLINK servo user constant No.	OWxx35	Sets the user constant number and the number of words when reading or writing Servopack parameters (See OWxx20.). 	0
43	MECHATROLINK servo user constant	OLxx36	Stores setting data when writing Servopack parameters. • Setting range: -2^{31} to $2^{31}-1$	0
44	Encoder position at power OFF, low	OLxx38	These parameters are used for ABS system infinite length position control. Encoder position at power OFF. low and high are paired data that together are called the ABS system infinite length position control information. • Setting range: -2^{63} to $2^{63}-1$ (1 = 1 pulse) Related Parameters • Fixed parameter 3: Encoder selection • Fixed parameter 17: Servo controller function selection flag b5: Finite length and infinite length.	0
45	Encoder position at power OFF, high	OLxx3A		
46	Absolute pulse position at power OFF, low	OLxx3C		
47	Absolute pulse position at power OFF, high	OLxx3E		

5.2.3 Monitor Parameters

The following table describes the servo monitor data reported by the servo controller. Be sure to look through these because they are basic parameters.

These monitor parameters can be used as reference from ladder logic and motion programs.

The following table describes the monitor parameters.

No.	Name	Register No.	Description		Remarks
1	Operating status	IWxx00	Monitors servo controller operating status.		–
			b1: PRMERR	Setting parameter setting error	–
			b2: FPRMERR	Fixed parameter setting error	–
			b3 to b6	Not used.	–
			b7: SVCRDY	Servo controller operation preparations complete	–
			b8: SVCRUN	Servo controller is running	–
			b9	Not used.	–
			b10	Not used.	–
			b11	Not used.	–
			b12	Not used.	–
			b13: POSCOMP	Positioning complete signal	–
			b14	Not used.	–
			b15	Not used.	–
2	MECHATROLINK servo status	IWxx01	b0:ALARM	Turns ON when an alarm is detected on the Servopack. Reset the alarm by executing the alarm reset (OWxx00 b6).	–
			b1: WARNG	Turns ON when a warning is detected on the Servopack. Reset the alarm by executing the alarm reset (OWxx00 b6).	–
			b2: CMDRDY	Turns ON when the MECHATROLINK is ready to receive the command.	–
			b3: SVON	Turns ON when the servo turns ON.	–
			b4: PON	Turns ON when the main power supply is ON.	–
			b5: MLOCK	Turns ON in machine lock status. Set the machine lock by setting the OWxx2D b0.	–

No.	Name	Register No.	Description		Remarks
2	MECHATROLINK servo status	IWxx01	b6: ZPOINT	Turns ON when the feedback position (OLxx08) on the machine coordinate system is in the set zero point range. Use the user constant Cn-002A to set the zero point range.	–
			b7: PSET	Turns ON when the distribution (IWxx01 b8) has been completed and the feedback position on the machine coordinate system (ILxx08) is in the positioning completion range of the target position. Use the user constant Cn-001B to set the positioning completion range.	–
			b8: DEN	Turns ON when the reference position is on the final target positioning location. Completes the distribution of position reference.	–
			b9: T_LIM	Turns ON when the torque limit value limits the torque reference. Use the user constants Cn-008 (positive torque limit) and Cn-009 (negative torque limit) to set the torque limit.	–
			b10: L_CMP	Enters latch mode during execution of a latch system command. Turns ON when latch position data is prepared after having received a latch signal. Turns OFF when receiving a signal other than a latch system command and cancels the latch mode. The latch system command indicates the motion commands, EXPOSING (OWxx20 = 2) and ZRET (OWxx20 = 3).	–
			b11: NEAR	Turns ON when the feedback position (ILxx08) for the machine coordinate system is within the range of the positioning proximity detection width of the target position. Use the user constant Cn-0007 to set the positioning proximity detection width. The completion of distribution DEN (IWxx01 b8) does not affect NEAR.	–
			b12: P-SOT (Not used)	Not used since the P-SOT mask on the Servopack is set. Note: Be sure to set the P-SOT mask (user constant Cn-0004 b3) on the Servopack to 1.	–

5

<table>
<tr><th>No.</th><th>Name</th><th>Register No.</th><th colspan="2">Description</th><th>Remarks</th></tr>
<tr><td rowspan="2">2</td><td rowspan="2">MECHATROLINK servo status</td><td rowspan="2">IWxx01</td><td>b13: N-SOT (Not used)</td><td>Not used since the N-SOT mask on the Servopack is set.
Note: Be sure to set the N-SOT mask (user constant Cn-0004 b3) on the Servopack to 1.</td><td>–</td></tr>
<tr><td>b14 and b15: RESERVED</td><td>For system reserve</td><td>–</td></tr>
<tr><td>3</td><td>Calculated position monitored in the machine coordinate system</td><td>ILxx02</td><td colspan="2">• This is the calculated position (CPOS) for the machine coordinate system, and it is updated when machine lock is ON.
• If CPOS is used as the current reference position for movement reference, then the confirmed CPOS position status (DEN = ON or PSET = ON status) must be used as the reference.</td><td>–</td></tr>
<tr><td>5</td><td>Latch position monitored in the machine coordinate system</td><td>ILxx06</td><td colspan="2">This parameter is the latch position (LPOS) for the machine coordinate system. It is updated when an external positioning command is executed and latching is completed</td><td>–</td></tr>
<tr><td>6</td><td>Feedback position monitored in the machine coordinate system</td><td>ILxx08</td><td colspan="2">This parameter is the feedback position (APOS) for the machine coordinate system.</td><td>–</td></tr>
<tr><td>11</td><td>Parameter No. out of range</td><td>ILxx0F</td><td colspan="2">• When a fixed or setting parameter is set outside the acceptable range, then the most recent parameter number that caused the setting range error is stored for this parameter.
• Setting parameters: 1 to 47
• Fixed parameters: 101 to 127</td><td>–</td></tr>
<tr><td>14</td><td>Motion command response code</td><td>IWxx14</td><td colspan="2">• This is the response code for setting parameter OWxx20: Motion command and it is used to store the executing motion command.
• The codes are all the same as that for OWxx20.</td><td>–</td></tr>
</table>

No.	Name	Register No.	Description		Remarks
15	Motion command status	IWxx15	These flags indicate the executing status of motion commands.		–
			b0: BUSY	Command executing • Indicates that a motion command is being executed and that a new motion command cannot be issued.	–
			b1: HOLDL	Command hold completed • Turns ON when a request has been issued to put a motion command on hold in setting parameter OWxx21 b0: Hold command and the hold has been completed.	–
			b2: DEN	Distribution complete • Stores distribution completed (DEN) status for the MECHATROLINK servo.	–
			b3: ZSET	Zero point setting complete • Turns ON when the ZSET motion command has finished executing.	–
			b4: EX_LATCH	External positioning signal latch complete • Turns ON when latching by an external signal is completed when the EX_POJING/LATCH motion command is executed.	–
			b5: FAIL	Command error end status • Turns ON when an error occurs in motion command processing.	–
			b6: ZRNC	Zero point return complete status • Turns ON when zero point return has been completed.	–
			b7 to b15	Not used.	–
16	Number of digits left of radix point	IWxx16	Monitors fixed parameter 18: Number of digits left of radix point.		–

5

No.	Name	Register No.	Description		Remarks
17	Position control status	IWxx17	These flags indicate position control status.		–
			b0: MLKL	Machine lock ON	–
			b1: ZERO	Zero point position • Turns ON when MPOS (machine coordinate system reference position) moves from the machine coordinate system zero point to within the preset range set at setting parameter OWxx33 (zero point position output width) at completion of zero point return.	–
			b2: PSET2	Second INP complete • Turns ON when the difference between the reference position and the feedback position after completing distribution is within the range set at setting parameter OWxx32: Second in-position width.	–
			b3: ABSLDE	ABS system infinite position control data LOAD complete	–
			b4: TPRSE	Preset request for the no. of POSMAX turns completed	–
			b5: GEARM	Copy enabled electronic gear selection	–
			b6: MODSELM	Copy axis selection	–
			b12 to b15: USR-MONSERL	MECHATROLINK servo user monitor data selection response • Stores the type of monitor data stored at monitor parameter ILxx20: MECHATROLINK servo user monitor data.	–
18	Reference position in the machine coordinate system	ILxx18	This is the reference position for the machine coordinate system and it is essentially the same as ILxx02 (CPOS). MPOS does not equal CPOS because the position is not updated when the machine is locked.		–
20	POSMAX monitoring	ILxx1C	Stores the POSMAX value set at fixed parameter 22 and is used when motion and other functions use POSMAX for reference.		–
21	No. of POSMAX turns	ILxx1E	The count goes up and down every time POSMAX is exceeded when using an infinite length rotating axis.		–
22	Monitor data for the servo drive user	ILxx20	Stores monitor data for the MECHATROLINK servo selected at setting parameter OWxx2D b12 to b15:MECHATROLINK servo monitor data selection. See the Σ-*Series SGM□/SGD-□N User's Manual* (SIE-S800-26.3) for more details.		–

No.	Name	Register No.	Description		Remarks
23	Alarms	ILxx22	Monitors servo control section alarms based on bit data.		–
			b0: SVERR	Servo alarm error • Detected Servopack alarms. See IWxx24 for alarm details.	Servo OFF
			b1: OTF	Positive overtravel • Detected Servo Amp overtravel in the positive direction (P-OT signal ON).	–
			b2: OTR	Negative overtravel • Detected Servo Amp overtravel in the negative direction (N-OT signal ON).	–
			b3: SOTF	Positive soft limit • Detected movement toward the positive software limit.	–
			b4: SOTR	Negative soft limit • Detects movement toward the negative software limit.	–
			b5: SVOFF	Servo power supply open • Movement reference was executed with servo power OFF.	Servo OFF
			b6: TIMEOVER	Positioning time exceeded • Positioning was not completed within the time set at OWxx34: Positioning completion check time after distribution.	–
			b7: DISTOVER	Positioning distance traveled exceeded • The move reference that was executed exceeded the positioning distance trav-eled.	–
			b8: FILTYPERR	Filter type change error • The type of filter was changed before distribution was completed.	–

No.	Name	Register No.	Description		Remarks
23	Alarms (continued)	ILxx22	b9: FILTIMERR	Filter time constant change error • The filter time constant was changed before distribution was completed.	–
			b10: MODERR	Control mode error • A Position Control Mode motion command was used in a mode other position control.	–
			b11: ZSET_NRDY	Zero point not set • The zero point is not set and a move reference was executed without setting the zero point.	–
			b12:	Not used.	–
			b13:	Not used.	–
			b14: WDT_ERR	MECHATROLINK servo synchronized communications error • The MP930 detected a synchronized communications error with the MECHATROLINK servo.	Servo OFF
			b15: COM_ERR	MECHATROLINK servo communications error • The MP930 detected a communications error two consecutive times with the MECHATROLINK servo.	Servo OFF
			b16: SVTIMOUT	MECHATROLINK servo command time-out error • A MECHATROLINK servo command was not completed within the specified time.	Servo OFF
			b17: ABSOVER	ABS encoder rotation count exceeded • The number of ABS encoder rotations exceeded the range of the MP930.	–
			b18 to b31	Not used.	–
24	MECHATROLINK servo alarm code	IWxx24	This parameter is used to monitor alarm codes generated by the MECHATROLINK servo. See the Σ-*Series SGM□/SGD-□N User's Manual* (SIE-S800-26.3) for more details. Normally 99H is monitored.		–

No.	Name	Register No.	Description		Remarks
25	MECHATROLINK servo I/O monitoring	IWxx25	This parameter is use to monitor I/O monitor data for the MECHATROLINK servo.		–
			b0: P-OT	Forward direction OT input	–
			b1: N-OT	Reverse direction OT input	–
			b2: DEC	Speed LS input	–
			b3: PA	Encoder A phase input	–
			b4: PB	Encoder B phase input	–
			b5: PC	Encoder C phase input	–
			b6	Not used.	–
			b7	Not used.	–
			b8	Not used.	–
			b9: BRK	Brake status output	–
			b10 to b15	Not used.	–
26	Rapid traverse speed reference output monitoring	ILxx26	This parameter is used to debug the system.		–
27	MECHATROLINK servo user constant monitoring	ILxx28	This parameter allows the MECHATROLINK servo user constant to be read using the CN_RD motion command. User constant data that is read is stored at this parameter.		–
28	–	IWxx30 to IWxx37	Not used.		–
29	Rightmost 2 encoder position words at power OFF	ILxx38	These parameters are used for ABS system infinite length position control.		–
30	Rightmost 2 encoder position words at power OFF	ILxx3A	Encoder position at power OFF and pulse unit position at power OFF are paired data that together are called ABS system infinite length position control information.		–
31	Rightmost 2 absolute pulse position words at power OFF	ILxx3C	The ABS system infinite length position control information must be saved periodically in a low-speed drawing (DWG.L) to M registers.		–
32	Rightmost 2 absolute pulse position words at power OFF	ILxx3E			–

5

6

Controlled Axis Support Functions

This chapter describes controlled axis support functions for positioning control in systems that use the MP930.

6.1 Support Functions for Controlled Axes

6.1.1 Reference Unit

A reference unit is the unit of measure used for positioning. In the MP930 the reference unit can be millimeters, degrees, inches, or pulses. It is also referred to as a minimum reference unit when it expresses the minimum measurement unit for a position. The minimum reference unit is determined by two fixed parameters: Reference Unit Selection and Number of Digits Left of Radix Point.

Table 6.1 Reference Unit

Reference Unit Selection		Reference Units (Electronic Gear Enabled)			Pulse (Electronic Gear Disabled)
		mm	deg	inch	
Number of Digits Left of Radix Point	**0**	1 [mm]	1 [deg]	1 [inch]	1 [pulse]
	1	0.1 [mm]	0.1 [deg]	0.1 [inch]	
	2	0.01 [mm]	0.01 [deg]	0.01 [inch]	
	3	0.001 [mm]	0.001 [deg]	0.001 [inch]	
	4	0.0001 [mm]	0.0001 [deg]	0.0001 [inch]	
	5	0.00001 [mm]	0.00001 [deg]	0.00001 [inch]	

IMPORTANT

The fixed parameter: Number of Digits Left of Radix Point is disabled when pulse units are selected.

6.1.2 Electronic Gear

An electronic gear converts position or speed units into user units (reference units) and internal controller units (pulse units), and it converts pulse units into reference units. The electronic gear function is not used to select a detector (encoder) suitable for the machine system but rather for positioning control.

Forward Direction Electronic Gear Conversion

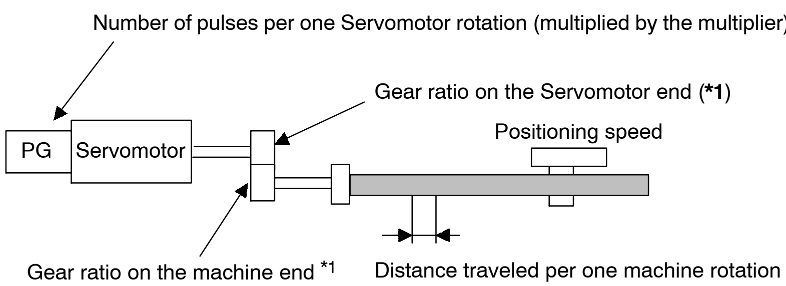
$$\text{Controller units [pulses]} = \frac{\text{Electronic gear numerator}}{\text{Electronic gear denominator}} \times \text{Reference units position [reference units]}$$

Reverse Direction Electronic Gear Conversion

$$\underset{\text{[reference units]}}{\text{User reference units position}} = \frac{\text{Electronic gear numerator}}{\text{Electronic gear denominator}} \times \underset{\text{[pulses]}}{\text{Controller internal unit position}}$$

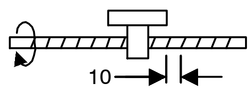
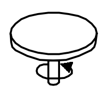
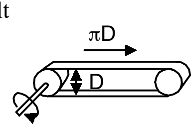
- Electronic gear numerator = Gear ratio on the Servomotor end × Number of pulses per Servomotor 1 rotation (multiplied by the multiplier)
- Electronic gear numerator = Gear ratio on the machine end × Distance traveled per one machine rotation

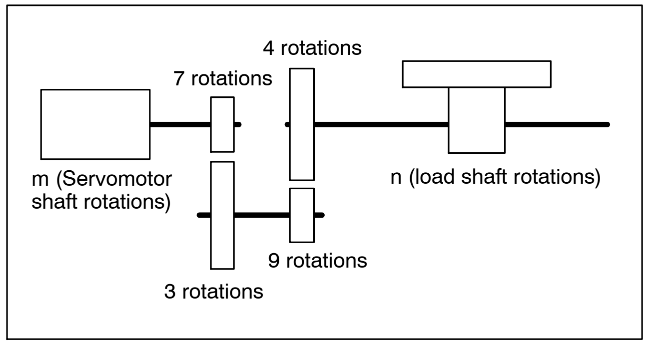
Electronic Gear Concept



* When the electronic gear system is structured so that the shaft on the machine end will turn n times when the shaft on the Servomotor end turns m times, m is the Servomotor-end gear ratio and n is the machine-end gear ratio.

The following parameters are related to the electronic gear.

<table>
<tr><th>Parameter No.</th><th>Name</th><th colspan="2">Description</th><th>Default</th></tr>
<tr><td>18</td><td>No. of digits left of radix point</td><td colspan="2">Minimum reference units are determined by this parameter and Reference Unit Selection for the servo controller function selection flags (b0 to b3).
Parameter set values are described below.</td><td>3</td></tr>
<tr><td rowspan="4">19</td><td rowspan="4">Machine rotation/reference unit</td><td colspan="2">Sets the amount a load moves (reference units) per load axis rotation.
Setting range: 1 to $2^{31}-1$</td><td rowspan="4">10000</td></tr>
<tr><td>Ball screw

Ball screw pitch = 10 mm</td><td>Ball screw pitch = 10 mm,
Reference Unit Selection = mm,
number of digits left of radix point = 3
↓
Set the amount of movement per one machine rotation to 10,000.</td></tr>
<tr><td>Rotating table

One rotation = 360°</td><td>One table rotation = 360°,
Reference Unit Selection = deg, number of digits left of radix point = 3
↓
Set the amount of movement per one machine rotation to 360,000.</td></tr>
<tr><td>Belt
</td><td>Roller 1 rotation = 360°,
Reference unit selection = mm, number of digits left of radix point = 3
↓
Set the amount of movement per one machine rotation to $\pi D \times 1{,}000$.</td></tr>
<tr><td>20</td><td>Gear ratio, Servomotor end</td><td colspan="2">These parameters are used to set the gear ratio for the Servomotor and load. Set the following at values that will allow the load shaft to rotate n times when the Servomotor shaft rotates m times.
• Gear ratio (Servomotor end) = m
• Gear Ratio (Load end) = n</td><td>1</td></tr>
</table>

Parameter No.	Name	Description	Default
21	Gear ratio, load end	Setting example  In the preceding diagram: Deceleration ratio: n/m = 3/7 × 4/9 = 4/21 Consequently, set the gear ratio (Servomotor end) at 21 and the gear ratio (load end) at 4.	1

6.1.3 Override Function

When an axis is moving via rapid traverse or interpolation feed for example, the speed of movement can be changed by setting a percentage (override percentage) with respect to the specified speed.

- The procedure used to set override is different in rapid traverse and interpolation systems.

Rapid traverse	JOG STEP Rapid traverse (MOV)	By axis: Override enabled/disabled (fixed parameter 17 b9) Override (setting parameter OWxx2C)
Interpolation	Linear interpolation (MVS) Circular interpolation (MCW/MCC) Skip (SKP)	Set in a group definition by each group. The default setting is MW00001. This override is always enabled. MW00001 (100 % = 10000)

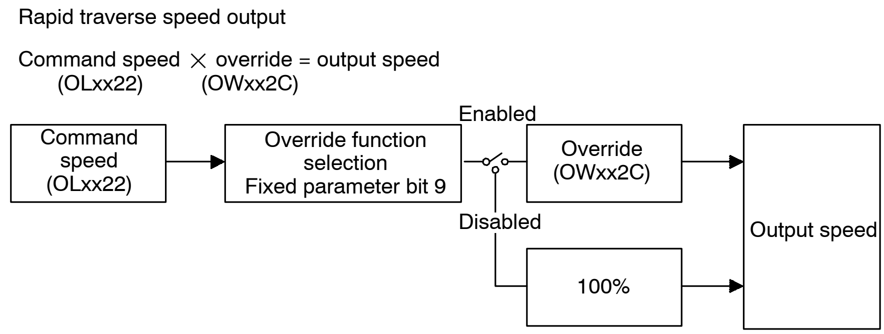
- An override in the range of 0% to 327.67% can be selected for the Rapid Traverse Speed. Set Override (OWxx2C) in the setting parameter for each axis.

TERMS

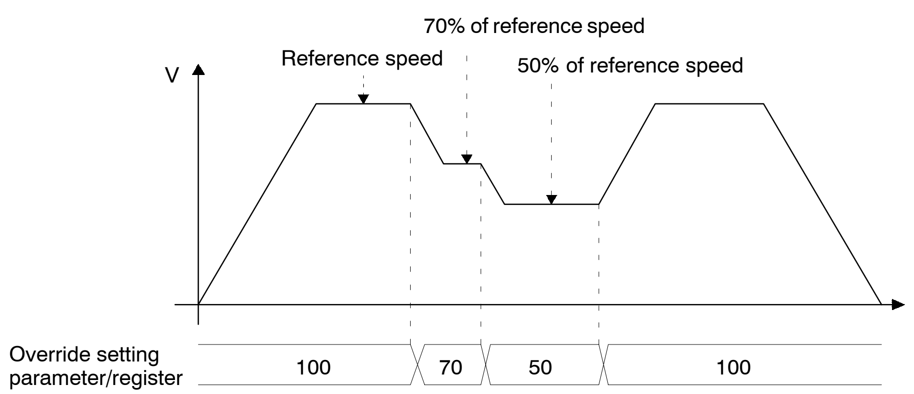
◆ Override

Override often means “to invalidate.” In this manual, however, it should be taken to mean “change” the set value.

- There are three override setting methods: Motion program, ladder logic program, or the Parameter Setting Screen.

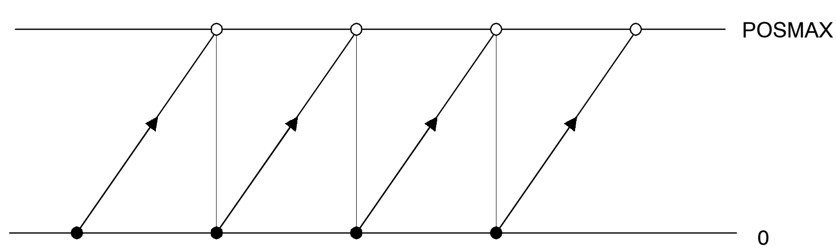


- Override is always enabled during operation, but this can be changed from a ladder program, motion program or parameter settings while an axis is moving.
- A parameter setting error will be generated if the override data setting causes the output speed to fall outside the acceptable range.
- The following illustration shows speed change timing for changes to the override.



6.1.4 Infinite Length Positioning

Infinite Length Positioning is a function that automatically updates machine position, program position and current values at regular intervals according to fixed parameter settings. The function can be used for repeated positioning in one direction.



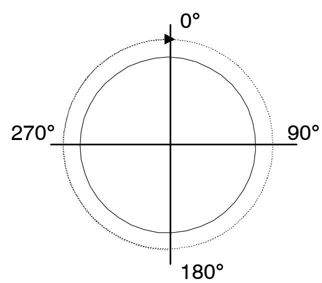
The following shows parameters related to Infinite Length Positioning.

Name	Parameter No.	Setting Range	Remarks
Servo Module Function Selection Flag	Fixed parameter 17 b5	0, 1	0: Finite Length Mode 1: Infinite Length Mode
POSMAX	Fixed parameter 22	1 to $2^{31}-1$	Units: 1 = 1 reference unit

- Procedure for Specifying Incremental Mode in Infinite Length Mode Axis

  A relative reference can be set in Infinite Length Mode Axis at the same range as that for Finite Length Mode Axis. (Range: –214783648 to 214783647.)

◀**EXAMPLE**▶



ZRN [X1] 0;
INC MOV [X1] 180.0;
INC MOV [X1] 2700.0; Move clockwise 7.5 rotations to 0°.

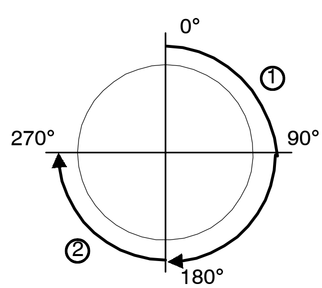
Figure 6.1 Specifying Incremental Mode in Infinite Length Mode Axis

- Procedure for Specifying Absolute Mode in Infinite Length Mode Axis

  Reference codes signify the direction of rotation and reference angles signify absolute position as shown in the figure below when absolute references are set in Infinite Length Mode Axis.

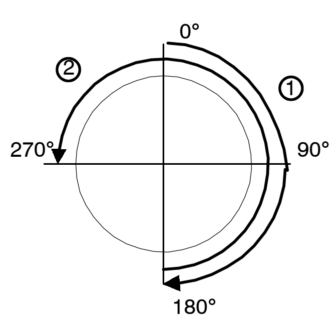
◀**EXAMPLE**▶

Refer to Figures 6.2 and 6.3 where the current position is specified at position 180° .



ZRN [X1] 0;
1 INC MOV [X1] 180.0;
2 ABS MOV [X1] 270.0; Turn clockwise (forward) to 270°.

Figure 6.2 Specifying Absolute Mode in Infinite Length Mode Axis (Example 1)



ZRN [X1] 0;
1 INC MOV [X1] 180.0;
2 ABS MOV [X1] –270.0; Turn counterclockwise (reversed) to 270°.

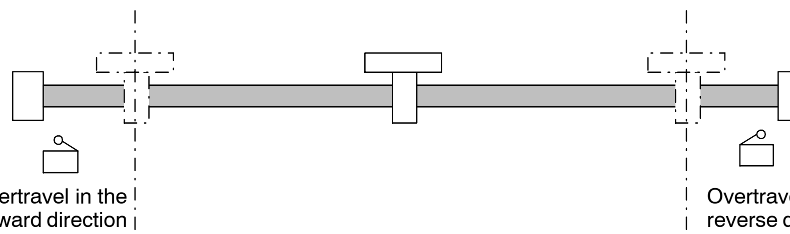
Figure 6.3 Specifying Absolute Mode in Infinite Length Mode Axis (Example 2)

INFO

1. The stroke limit function is disabled when Infinite Length Positioning is selected.
2. When moving to 0° by specifying Absolute Mode in Infinite Length Mode Axis, the axis moves counter-clockwise even if +0.0 is specified. Specify +360.0 to move the axis clockwise.

6.1.5 Stroke Limit Function

The stroke limit function is used to set upper and lower limits at fixed parameters for machine system range of movement so the controller can constantly monitor the operating range of the machine. The function prevents machine runaway or damage due to incorrect operation as well as incorrect references in a motion program.



- The following fixed parameters must be set in order to use the stroke limit function.

Parameter No.	Name	Units	Remarks
17	Servo Module Function Selection Flags b7: Positive stored stroke limit b8: Negative stored stroke limit		 0: Disabled, 1: Enabled 0: Disabled, 1: Enabled
25	Positive stored stroke limit	Units: 1 = 1 reference units	–2147483648 to 2147483647
26	Negative stored stroke limit	Units: 1 = 1 reference units	–2147483648 to 2147483647

- Set the positive and negative stroke limits for a machine coordinate system.
- The machine coordinate system is determined by returning to the zero point.
- The stroke limit function is implemented after the machine returns to the zero point.
- Be sure to return to the zero point after power is turned ON.

Type of Axis Movement	Check	Remarks
Program Operation • **Positioning** • **Interpolation**	Yes	• If a positioning reference is shifted to a position beyond the stroke limit, the axis will be positioned on the stroke limit and an alarm will be generated. • The stroke limit range is constantly checked during an interpolation move, and the axis will decelerate to a stop at the stroke limit position.
JOG Operation	Yes	• If the stroke limit function is enabled, a move is executed to the stroke limit position. After an error is cleared, the axis can be moved to within the stroke range.
STEP Operation	Yes	• If a positioning reference is given for a position beyond the stroke limit, the axis will be positioned on the stroke limit and an alarm will be generated.

7

Absolute Position Detection

This chapter describes an absolute detection system that uses an absolute encoder. Be sure to read this chapter carefully when using a Servomotor equipped with an absolute encoder.

7.1 Structure of the Absolute Position Detection Function

This section describes the Absolute Position Detection Function in the MP930.

7.1.1 Description of the Function

The Absolute Position Detection Function detects the position of the machine even if power is turned OFF. This allows it to set the machine coordinate system automatically and to begin operating automatically without having to return to the zero point after power is turned ON.

The following are features of the absolute position detection system.

- Eliminates the need to return to the zero point after power is turned ON.
- Enables the Stored Stroke Limit Function immediately after power is turned ON.
- Eliminates the need for a zero point dog and overtravel limit switch.

One of the following operating systems can be selected from parameter settings with this function.

- Incremental detection system using an incremental encoder.
- Absolute position detection system using an absolute encoder.
- Incremental position detection system using an absolute encoder.

7.1.2 Structure of Absolute Position Detection

■ Basic Terminology

The following explanation for basic terminology used in this chapter is provided to ensure basic understanding.

Absolute Encoder

Absolute position detection is generally performed in a semi-closed loop using an absolute encoder built into a Servomotor. The detector is comprised of an encoder that is used to detect absolute position within one rotation and a counter that is used to count the number of rotations.

Absolute Data

Absolute data that is stored in an absolute encoder is comprised of the number of rotations (P) from the absolute reference position and position (PO) in a one Servomotor rotation. This absolute data is read as serial data when the machine is turned ON.

All other operations are the same as that for ordinary incremental encoders.

In other words, we can determine the absolute position PO from the equation for absolute value (P) which is absolute value (P) = N $\times$ RP + PO if we use the following:

- Number of rotations from the absolute reference position: N
- Number of pulses per one Servomotor rotation: RP
- Position in one Servomotor rotation: PO

Holding Absolute Data

An absolute encoder uses a battery to maintain absolute data at all times even though power is turned OFF. It also updates data if there is a change.

The battery is connected to the battery terminal of the Servopack.

Yaskawa supplies the battery module (JRMSP-120XC9600) on which the following battery is mounted.

- Type of battery: Lithium
- Battery configuration: ER6VC3, 3.6 V $\times$ 1
- Non-conducting service life: About 1 year

Reading Absolute Data

When power is turned ON, absolute data is read to the Servopack as well as to the MP930 where it is used to automatically calculate the absolute position and set the machine coordinate system. This way the absolute machine position can be detected and automatic operation can begin immediately after power is turned ON.

■ Changes in Status in an Absolute Position Detection System

The following shows changes in status in an absolute position detection system.

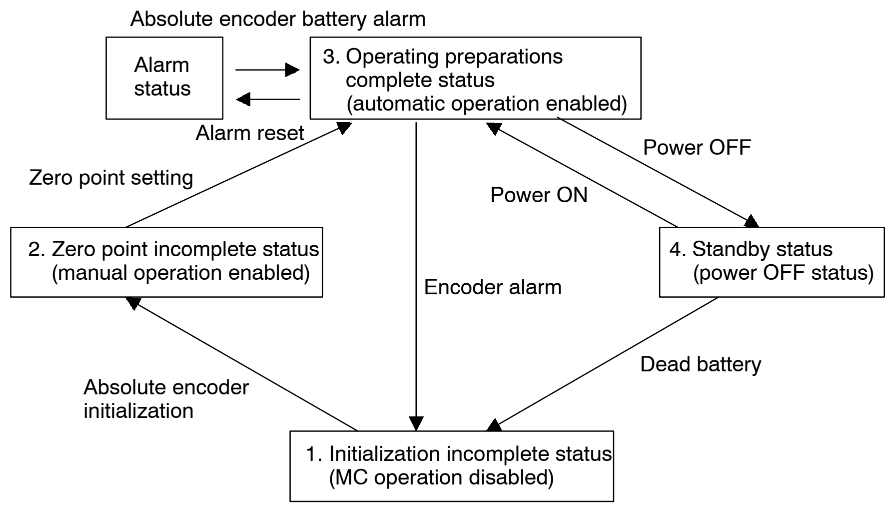


Figure 7.1 Changes in Status in the Absolute Position Detection System

The following describes the status above in more detail.

1. Initialization Incomplete Status

Absolute encoder operation cannot be guaranteed if the encoder has this status. This alarm will occur if this is the first time that the absolute encoder has been used or if all backup power supplies for the absolute encoder have discharged. Be sure to initialized the absolute encoder in these cases. The zero point cannot be set if the encoder has this status.

2. Zero Point Incomplete Status

This status indicates that the zero point setting that is needed to determine the zero point of the machine coordinate system has not been input. In this case an alarm will be generated when the machine is turned ON so reset the alarm and then enter the zero point setting. In zero point incomplete status, the only way to move the axis is through manual JOG and STEP operations.

3. Operation Preparations Complete

This indicates that zero point settings were input, the Absolute Position Detection Function is enabled and the machine is ready for normal operation.

4. Standby Status

This status indicates that machine movement is being detected even though power is OFF and data changes are being updated with absolute encoder rotation. At this time, the absolute encoder is running on battery power.

7

7.2 Starting the Absolute Position Detection Function

This section describes the procedure that is used to start the Absolute Position Detection Function.

7.2.1 System Startup Procedure

The Servopack, Servomotor, and other peripheral devices must be checked before starting up the absolute position detection system.

The system must be started up using the following procedure.

1. Check Devices

   Check to see if the Servopack, Servomotor, and cables are the right products and models for the absolute encoder.

2. Check Parameters Related to the Servopack

   Check to see if the Servopack parameters (user constants) are for an absolute encoder. Also check to see if the number of encoder pulses is set properly.

3. Set Parameters Related to the MP930

   Set all parameters related to the Absolute Position Detection Function.

4. Initialize the Absolute Encoder

   Follow the setup procedure to set the absolute encoder to default values.

5. Zero Point Setting

   Set the zero point as well as the absolute zero point, that is, the machine coordinate zero point.

The status of the encoder will change to operation preparation complete if steps 1 to 5 are successfully completed, and the absolute position detection system will be ready for operation.

Perform the absolute position detection system startup procedure in the following situations.

- When starting up the absolute position detection system for the first time
- When the Servomotor is changed
- When an absolute encoder-related alarm occurs

7.2.2 Setting Related Parameters

This section describes absolute position detection parameters in the MP930 parameters.

Set the following parameters prior to starting up the absolute position detection system.

Table 7.1 MP930 Unit Parameters

Parameter No.	Name	Setting Range	Units	Initial Value
Fixed Parameter 3	Encoder selection	0: Incremental encoder 1: Absolute encoder	–	0
Fixed Parameter 8	Number of FB pulses per Servomotor 1 rotation	Multiples of 4 in a range from 4 to 65535	1 = 1 pulse	2048
Fixed Parameter 17 b5	Finite Length Mode/Infinite Length Mode Axis Selection	0: Finite Length Mode Axis Setting 1: Infinite Length Mode Axis Setting	–	0
Fixed Parameter 22	Maximum Infinite Length Counter value	1 to $2^{31}-1$	1 = 1 reference unit	360000
Fixed Parameter 23	Maximum Absolute Encoder Rotations	1 to $\pm 2^{31}-1$	1 = 1 rotation	99999
Setting Parameter 7 (OLxx06)	Zero Point Position Offset Setting for a Machine Coordinate System	0 to $\pm 2^{31}-1$	reference units	0

Table 7.2 Servopack User Constants

User Constant	Name	Setting Range	Units	Initial Value
Cn-0001bE	Encoder selection	0: Incremental encoder 1: Absolute encoder	–	0
Cn-0032	Absolute Encoder Zero Point Position Offset	0 to $\pm 2^{31}-1$	reference units	0

Encoder Selection Setting

MP930 fixed parameter 3 and Servopack user constant Cn-0001bE are used to set absolute encoder for the axis whose absolute position will be detected.

Absolute position detection system can be set for each axis and any combination of incremental and absolute position detection is possible in this system.

7

Setting the Number of Encoder Pulses

This parameter is used to set an applicable number of absolute encoder pulses at fixed parameter 8 for the MP930 and user constant Cn-0011 for the Servopack.

- MP930 fixed parameter 8
- Servopack user constant Cn-0011

Note: Be sure to set both parameters.

Infinite Length Mode Axis/Finite Length Mode Axis Selection

This setting is used to set limits for controlled axis movement.

Maximum Number of Absolute Encoder Rotations (Fixed Parameter No. 23)

The difference between machine coordinate values stored when power is turned OFF and machine coordinate values the next time power is turned ON is converted to pulses, but an ABS Encoder Rotation Exceeded error will be generated if the converted number of pulses is greater than number of pulses for half the maximum number of absolute encoder rotations.

Maximum Infinite Length Counter Value

The Maximum Infinite Length Counter value is used to set the number of Infinite Length Mode Axis rotations in reference units. This parameter is enabled when an absolute encoder is used as an Infinite Length Mode Axis.

Zero Point Position Offset Setting for a Machine Coordinate System

This parameter is used to determine the zero point position of a machine coordinate system. Its meaning will depend on the type of encoder that is used and whether Finite Length Mode Axis or Infinite Length Mode Axis is selected.

Finite Length Mode Axis	INC axis	Parameter (OLxx06): ABS OFF is always enabled.
	ABS axis	Parameter (OLxx06): ABS OFF is always enabled.
Infinite Length Mode Axis	INC axis	Parameter (OLxx06): ABS OFF is always enabled.
	ABS axis	Only enabled when a zero point is set. (Used to define ABS System Infinite Length Mode Axis Control Data.)

- ABS Finite Length Mode Axis

  Setting parameter OLxx06: Zero Point Position Offset Setting is always valid. The zero point for the machine coordinate system can be changed simply by changing this parameter. This is why the zero point does not have to be set in ABS Finite Length Mode.

- ABS Infinite Length Mode Axis

  Setting parameter OLxx06: Zero Point Position Offset Setting is always valid when a zero point is set. If the zero point position offset is set when a zero point is set, the positive electronic gear conversion value is used as the current machine coordinate system position.

  Set the desired position at setting parameter OLxx06: Zero Point Position Offset Setting.

7.2.3 Initializing the Absolute Encoder

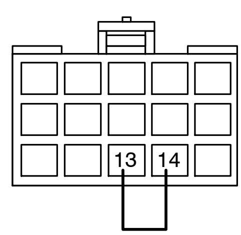
Initialize the absolute encoder in the following situations.

- When the absolute position detection system is started up for the first time
- When the number of rotations from the absolute reference position of the absolute encoder needs to be initialized to 0
- When a Servomotor has been left disconnected for more than four days with no battery connected to the absolute encoder
- When an alarm occurs

■ Initializing a 12-bit Absolute Encoder

Follow the procedure below to initialize a 12-bit absolute encoder.

1. Properly connect the Servopack, Servomotor, and MP930.
2. Reset Absolute Position Data in the encoder.
   a) Disconnect the connector on the encoder end.
   b) Use a short piece to short together connector pins 13 and 14 on the encoder end for 1 to 2 seconds.
   c) Remove the short piece and insert the connector securely in its original position.



3. Rewire the cables using normal wiring and make sure the encoder battery is connected.
4. Turn ON the system.

   Repeat the procedure starting from step 1 if an Absolute Encoder Alarm occurs, otherwise the system has been successfully initialized.

■ Initializing a 15-bit Absolute Encoder

Follow the procedure below to initialize a 15 bit-type absolute encoder.

1. Turn OFF the Servopack and MP930.
2. Discharge the large-capacity capacitor in the encoder using one of the following methods.

- At the encoder end connector
  a) Disconnect the connector on the Servopack end.
  b) Use a short piece to short together connector pins 10 and 13 on the encoder end.
  c) Leave the pins shorted for at least 2 minutes.
  d) Remove the short piece and insert the connector securely in its original position.

- At the Servopack end connector

  a) Disconnect the connector on the encoder end.

  b) Use a short piece to short together connector pins P and S on the encoder end.

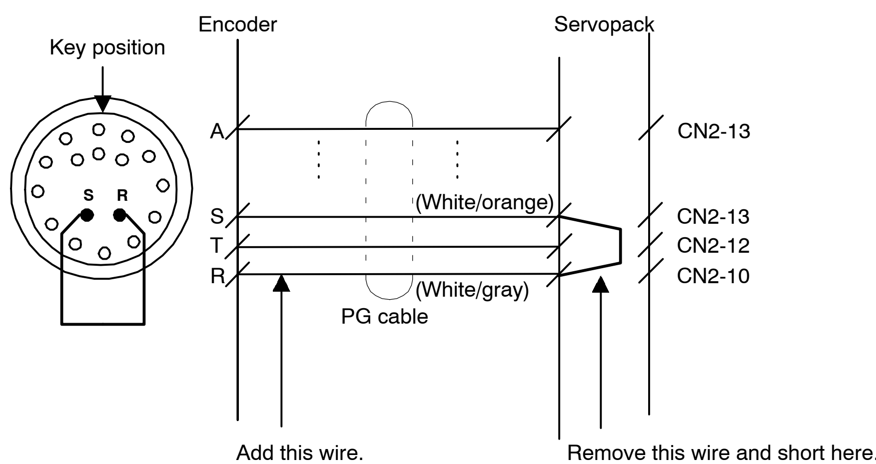
  c) Leave the pins shorted for at least 2 minutes.

  d) Remove the short piece and insert the connector securely in its original position.

3. Rewire the cables using normal wiring and make sure the encoder battery is connected.

4. Turn ON the system.

Repeat the procedure starting from step 1 if an Absolute Encoder Alarm occurs, otherwise the system has been successfully initialized.

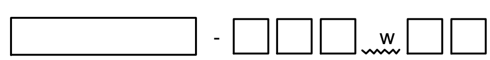


Setup procedure using a PG cable

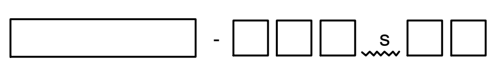
INFO

The following Servomotor models have absolute encoders.

1. 12-bit Encoder



2. 15-bit Encoder



7.3 Using an Absolute Encoder

This section describes precautions regarding use as well as the procedure for setting the zero point when using an absolute encoder.

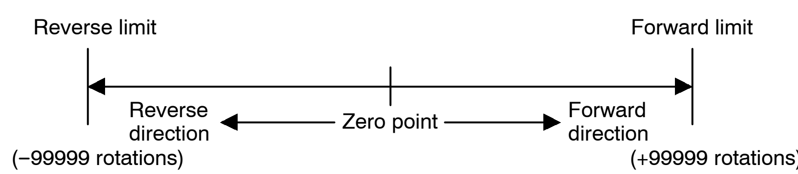
7.3.1 Finite Length Mode Axis

■ Overview

An absolute encoder stores the number of rotations from the encoder zero point in internal memory backed up by battery. This way the zero point of the coordinate system can be determined without returning to the zero point when the system is started up. Once the system is started, the encoder functions just like an incremental encoder.

Unfortunately, the maximum number of rotations from the encoder zero point is ±99999 rotations at which point it is reset to 0. When system power is turned ON therefore, the MP930 position may not be the same before and after power is turned ON.

A Finite Length Mode Axis has only limited movement that falls within a range of ±99999 absolute encoder rotations.



Therefore be sure to note the following precautions when using an absolute encoder as a Finite Length Mode Axis.

- Be sure to initialize the encoder prior to setting the zero point.
- Use an absolute encoder only within the range of ±99999 rotations

Note: The actual machine operating range may vary depending on parameters like the gear ratio.

■ Position Control with a Finite Length Mode Axis

Initialize the axis position as described next when power is turned ON if an absolute encoder is used as a Finite Length Mode Axis.

> Current position for the machine coordinate system =
> Encoder position when servo power is turned ON* + Setting parameter OLxx06: Zero Point Offset

* Multi-turn data × the number of encoder pulses + initial increment

Setting parameter OLxx06: Zero Point Offset is always enabled in Finite Length Mode. This means that the current position of the machine coordinate system (zero point setting) can be changed at any time.

The meaning of setting parameter OLxx06 will depend on whether Finite Length Mode or Infinite Length Mode is set.

Finite Length Mode

Set –(ILxx02) + OLxx06 at OLxx06 in order to make the current position of the machine coordinate system the zero position.

◀**EXAMPLE**▶ ILxx02 = 10,000 and OLxx06 = 100
Setting the current position of the machine coordinate system to 0 when the zero point is set.
–(10,000) + 100 = –9,900 Set OLxx06 to –9,900
ILxx02: Monitor the calculated position of the machine coordinate system

Infinite Length Mode

Set the desired position at OLxx06 and that setting will be used for the current position of the machine coordinate system when the zero point is set.

◀**EXAMPLE**▶ Setting the current position of the machine coordinate system to 0 when the zero point is set
Set OLxx06 to 0.

⚠ Caution

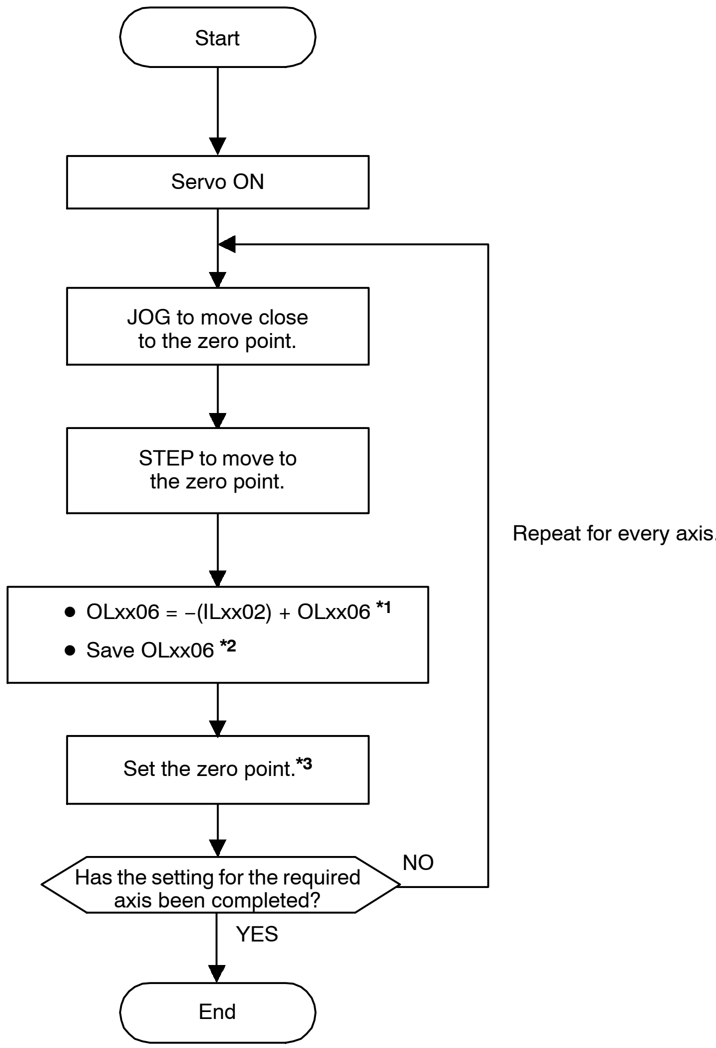
- **Do not change the Zero Point Position Offset (OLxx06) while operating in Finite Length Mode.**

  Otherwise this may cause machine damage or an accident.

■ Setting the Zero Point for a Finite Length Mode Axis

Set the zero point as described here after initializing the absolute encoder in order to set the zero point of the machine coordinate system and to create the machine coordinate system.

The following illustration shows the procedure for setting the zero point for a Finite Length Mode Axis.



*** 1.** The OLxx06 value must be saved when it is set.

*** 2.** See the information on the next page for more details on saving the OLxx06 value.

*** 3.** Execute with the ZSET command.

INFO

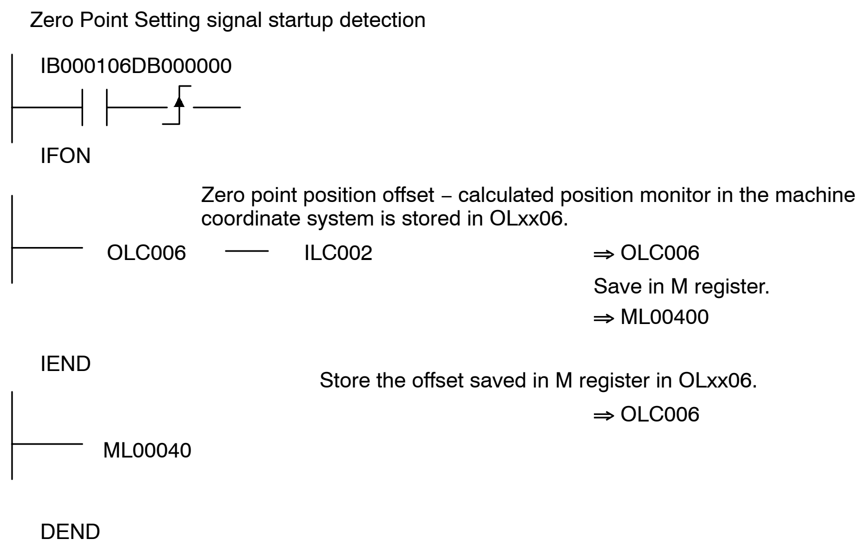
The following methods are used to save the Zero Point Position Offset (OLxx06).

1. Saving in a Ladder Logic Program M Register

Calculate (–(calculated position monitored in the machine coordinate system) + the Zero Point Position Offset setting) and save the results in the M register when they are stored at OLxx06.

Always store the contents saved in M register at setting parameter OLxx06: Zero Point Offset.

Ladder Logic Program Required for a Finite Length Mode Axis: ABS System Finite Length Mode Axis (Axis No. 1)



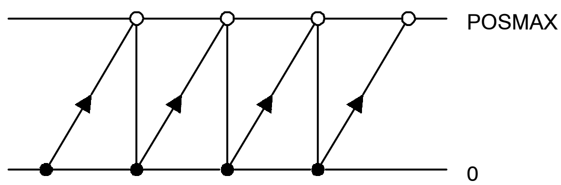
2. Saving the OLxx06: Zero Point Offset from the CP-717 Parameter Screen

After the zero point and the Zero Point Offset (OLxx06) value (current value) are set, use SAVE to save the settings in the Controller. When power is turned back ON, the value that was saved for Zero Point Offset (OLxx06) will be stored automatically.

7.3.2 Infinite Length Mode Axis

■ Description

Infinite Length Positioning is a function that automatically updates machine position, program position (absolute values in a program coordinate system), and current values at regular intervals according to fixed parameter settings. The function can be used for repeated positioning in one direction.



Unfortunately, the maximum number of rotations from the encoder zero point is ±99999 rotations at which point it is reset to 0. When system power is turned ON therefore, the MP930 position may not be the same before and after power is turned ON.

This problem can be resolved using the following method.

■ Position Control for an Infinite Length Mode Axis

Determine the MP930 position when power is turned ON and use the following equation to determine the pulse unit position from the relative encoder position.

Always save the pulse position and encoder position as paired data in memory backed up by battery power. This data is used as pulse position and encoder position at power ON settings the next time power is turned ON in order to determine number of pulses from the relative encoder position using the following equation.

$$\text{Pulse position} = \text{pulse position at power OFF} + (\underbrace{\text{encoder position} - \text{encoder position at power OFF}}_{*})$$

* Indicates the distance traveled while power is OFF (relative encoder position).

INFO

The following explains the terminology used here.

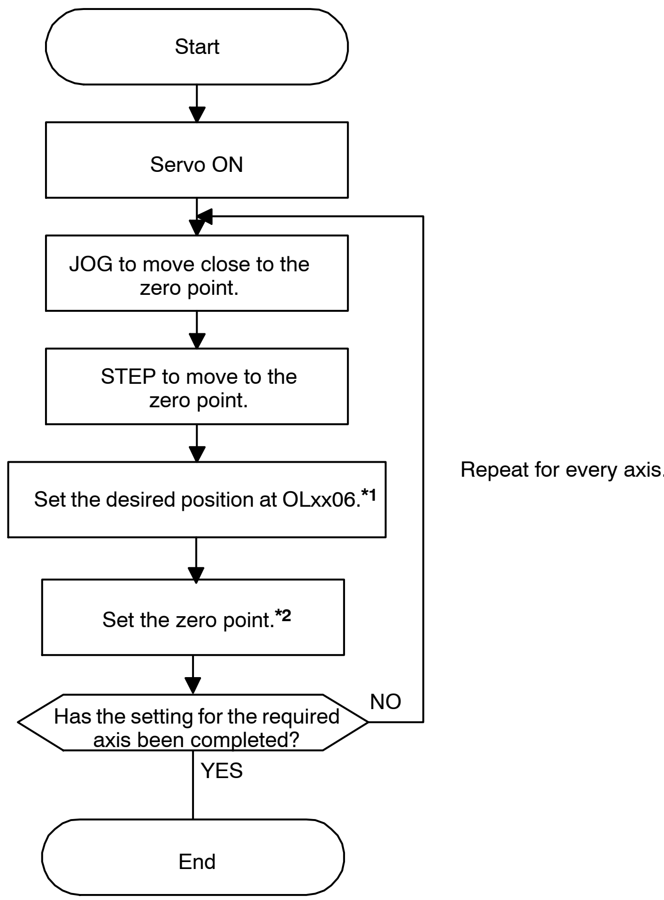
- Encoder position: Position data for an absolute encoder
  (Multi-turn data × the number of encoder pulses + initial increment)
- Pulse position: MP930 position data converted to pulses.

Setting the Zero Point for an Infinite Length Mode Axis

Execute the ZSET motion command (zero point setting).

The system will check pulse position at power OFF, encoder position at power OFF, and all position data when the zero point is set.

The following illustration shows the procedure for setting the zero point for an Infinite Length Mode Axis.



*** 1.** With an Infinite Length Mode Axis, the setting parameter OLxx06: Zero Point Offset is only enabled when the ZSET command is executed. Therefore, the OLxx06 value must be set at the M register. Set the desired coordinates at Zero Point Offset (OLxx06) when using an Infinite Length Mode Axis
Example:
When setting the current stop position to 0 (zero point position for the machine coordinate system)
0 → OLxx06

*** 2.** Execute with the ZSET command.

7

■ Ladder Logic Program for Infinite Length Mode Axis Position Control

Special ladder logic programs for normal operation and for restarting the system are needed for absolute Infinite Length Mode Axis position control when an absolute encoder is used as an Infinite Length Mode Axis.

Normal Operation

1. Check the Zero Point Setting Complete status.

Check to see if monitor parameter Zero Point Setting Complete (IWxx15 bit 3) is ON. If it is, go to step 2.

If it is not, it means that the pulse position at power OFF, encoder position at power OFF and all position data was not checked. In that case, restart the system and set up the position data again or execute the ZSET motion command (zero point setting) to check the position data all over from the start.

2. Save the pulse position at power OFF and encoder position at power OFF.

Use the customer's ladder logic program to save the following monitor parameters with high-speed scan timing at an M register backed up by battery.

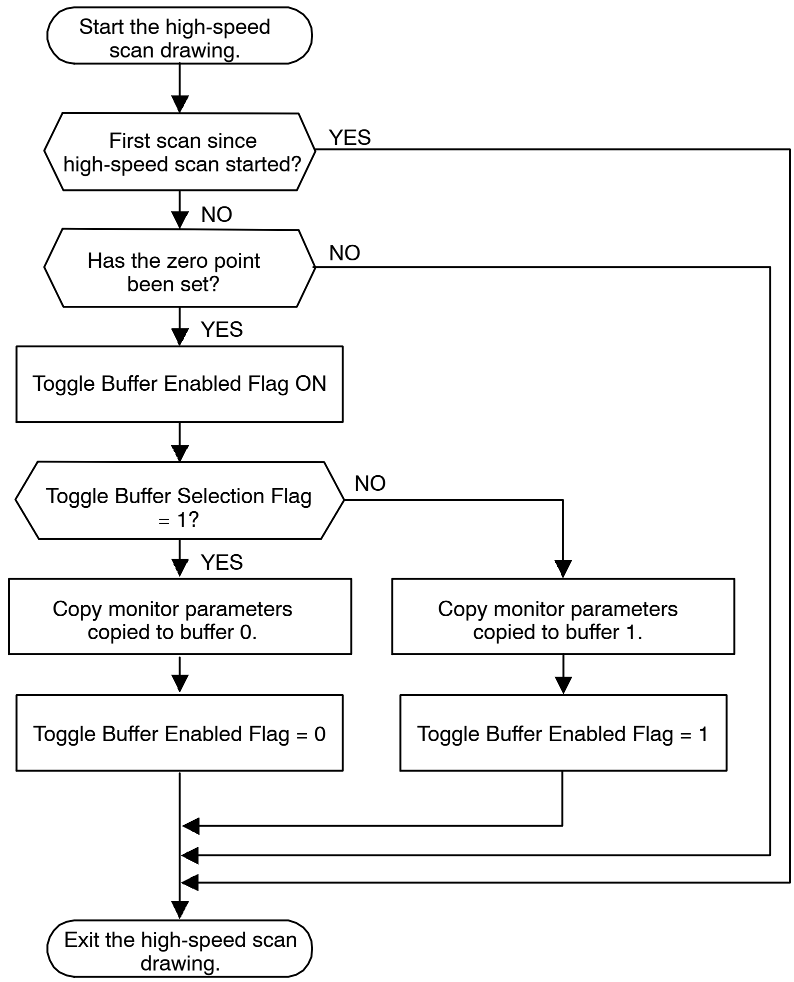
Monitor parameter: Encoder Position at Power OFF (All four words at ILxx38/ILxx3A)
Monitor parameter: Pulse Position at Power OFF (All four words at ILxx3C/ILxx3E)

The M register that is used to save the above monitor parameters is structured as shown below.

<table>
<tr><td rowspan="3">MWxxxxx</td><td>Bit 0</td><td colspan="2">Toggle Buffer Enabled Flag (0: Disabled/1: Enabled)</td></tr>
<tr><td>Bit 1</td><td colspan="2">Toggle Buffer Selection Flag (0: Buffer 0/1: Buffer 1)</td></tr>
<tr><td>Bit 2</td><td colspan="2">Position Data Re-setup Request Flag (0: Complete/1: Request)</td></tr>
<tr><td>MLxxxxx + 1</td><td colspan="3">Not used</td></tr>
<tr><td rowspan="2">MLxxxxx + 2
MLxxxxx + 4</td><td rowspan="4">Buffer 0</td><td rowspan="2">Monitor parameter:
Encoder Position at Power OFF</td><td>Rightmost two words (ILxx38)</td></tr>
<tr><td>Leftmost two words (ILxx3A)</td></tr>
<tr><td rowspan="2">MLxxxxx + 6
MLxxxxx + 8</td><td rowspan="2">Monitor parameter:
Pulse Position at Power OFF</td><td>Rightmost two words (ILxx3C)</td></tr>
<tr><td>Leftmost two words (ILxx3E)</td></tr>
<tr><td rowspan="2">MLxxxxx + 10
MLxxxxx + 12</td><td rowspan="4">Buffer 1</td><td rowspan="2">Monitor parameter:
Encoder Position at Power OFF</td><td>Rightmost two words (ILxx38)</td></tr>
<tr><td>Leftmost two words (ILxx3A)</td></tr>
<tr><td rowspan="2">MLxxxxx + 14
MLxxxxx + 16</td><td rowspan="2">Monitor parameter:
Pulse Position at Power OFF</td><td>Rightmost two words (ILxx3C)</td></tr>
<tr><td>Leftmost two words (ILxx3E)</td></tr>
</table>

Note Two buffers are needed to save the encoder position and the pulse position at power OFF because the program may be exited without checking position data at all four words if power is turned OFF during the high-speed scan.

Use the following flowchart to store values in buffers.



7

The following programming example (ladder logic program) is for the flowchart shown above. The axis used here is axis No. 1 of module No. 1. Change the motion parameter register number if the module and axis numbers are different.

```
H10
                ABS System Infinite Length Mode Axis: Axis No. 1
                First address in the toggle buffer: MW30000

$FSCAN-H (ON for just one scan after high-speed scanning starts)
  SB000001
IFON
  Motion fixed parameter setting error (*1)
  IBC0002
IFON
  Zero Point Setting Completed Flag
  IBC0153
IFON
$ONCOIL                                   Toggle Buffer Enabled Flag ON
  SB000004                                MB300000
  Toggle Buffer Selection Flag
  MB300001
IFON
  Monitor parameters saved in buffer 0
  ILC038                                  ⇒ ML30002
  ILC03A                                  ⇒ ML30004
  ILC03C                                  ⇒ ML30006
  ILC03E                                  ⇒ ML30008
ELSE
  Monitor parameters saved in buffer 1
  ILC038                                  ⇒ ML30010
  ILC03A                                  ⇒ ML30012
  ILC03C                                  ⇒ ML30014
  ILC03E                                  ⇒ ML30016
IEND
                                          Toggle Buffer Selection Flag inverted
  MB300001                                MB300001
ELSE
IEND
IEND
IEND
DEND
```

Turning the System Back ON (Turning the Servo Back ON)

Set up position data again from the customer's ladder logic program using high-speed scan timing as shown below. This is done when system power or servo power is turned back ON.

1. Store Pulse Position at Power OFF and Encoder Position at Power OFF at setting parameters.

Store the Pulse Position at Power OFF and Encoder Position at Power OFF values saved in M register at the following setting parameters.

Monitor parameter: Encoder Position at Power OFF (All four words at OLxx38/OLxx3A)
Monitor parameter: Pulse Position at Power OFF (All four words at OLxx3C/OLxx3E)

Store the contents of the buffer selected by the previously output Toggle Buffer Selection Flag at this time.

2. ABS System Infinite Length Position Control Data LOAD Request

Turn the setting parameter: ABS System Infinite Length Position Control Data LOAD Request (OWxx2D bit 2) OFF, ON and OFF again. This will allow all position data to be checked. The following monitor parameters will then be enabled when monitor parameter: Zero Point Setting Completed (IWxx15 bit 3) is ON.

Monitor parameter: Encoder Position at Power OFF (All four words at ILxx38/ILxx3A)
Monitor parameter: Pulse Position at Power OFF (All four words at ILxx3C/ILxx3E)

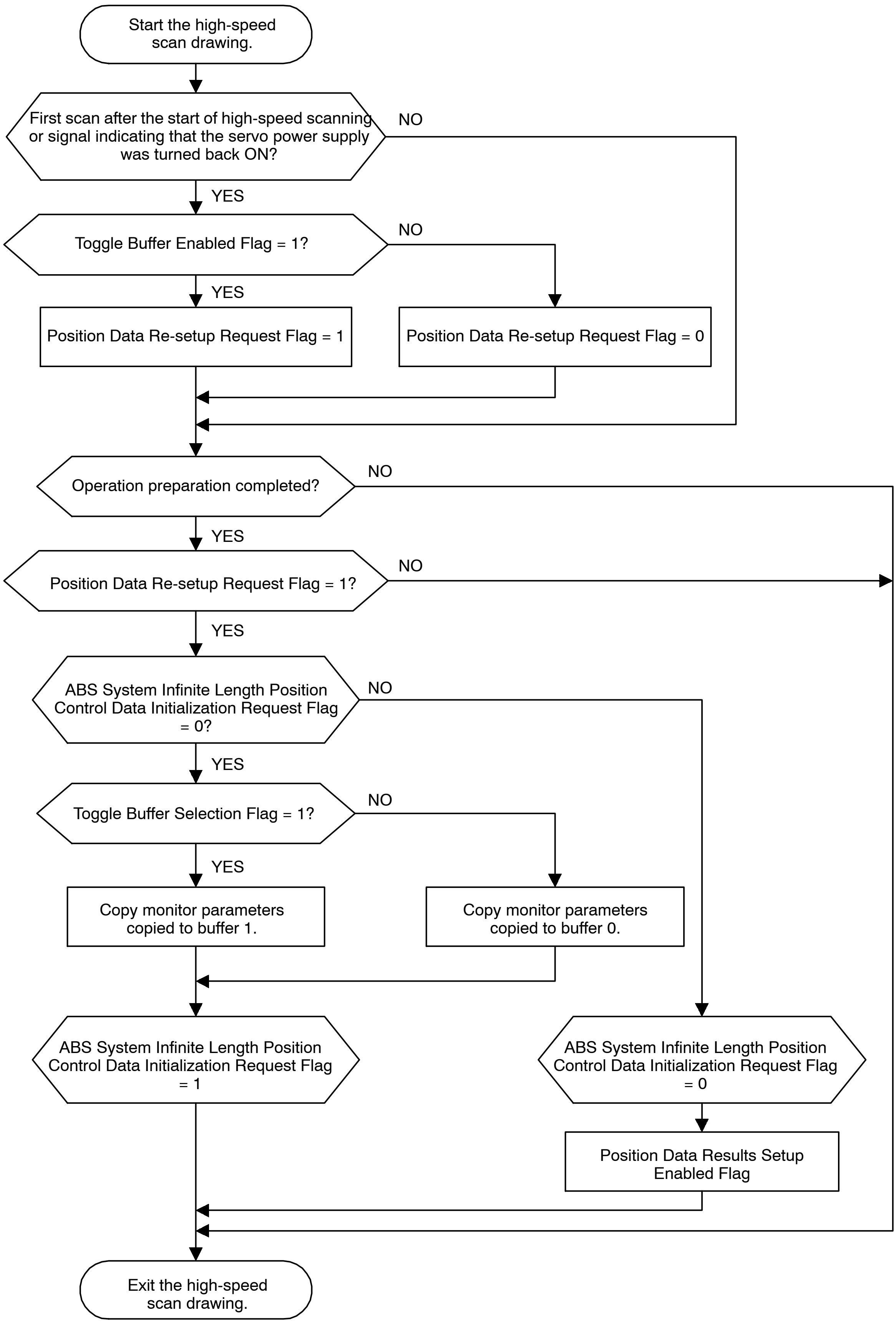
The system will create position data using the following equation when ABS System Infinite Length Position Control Data LOAD is requested.

$$\text{Pulse position} = \text{pulse position at power OFF} + (\underbrace{\text{encoder position} - \text{encoder position at power OFF}}_{*})$$

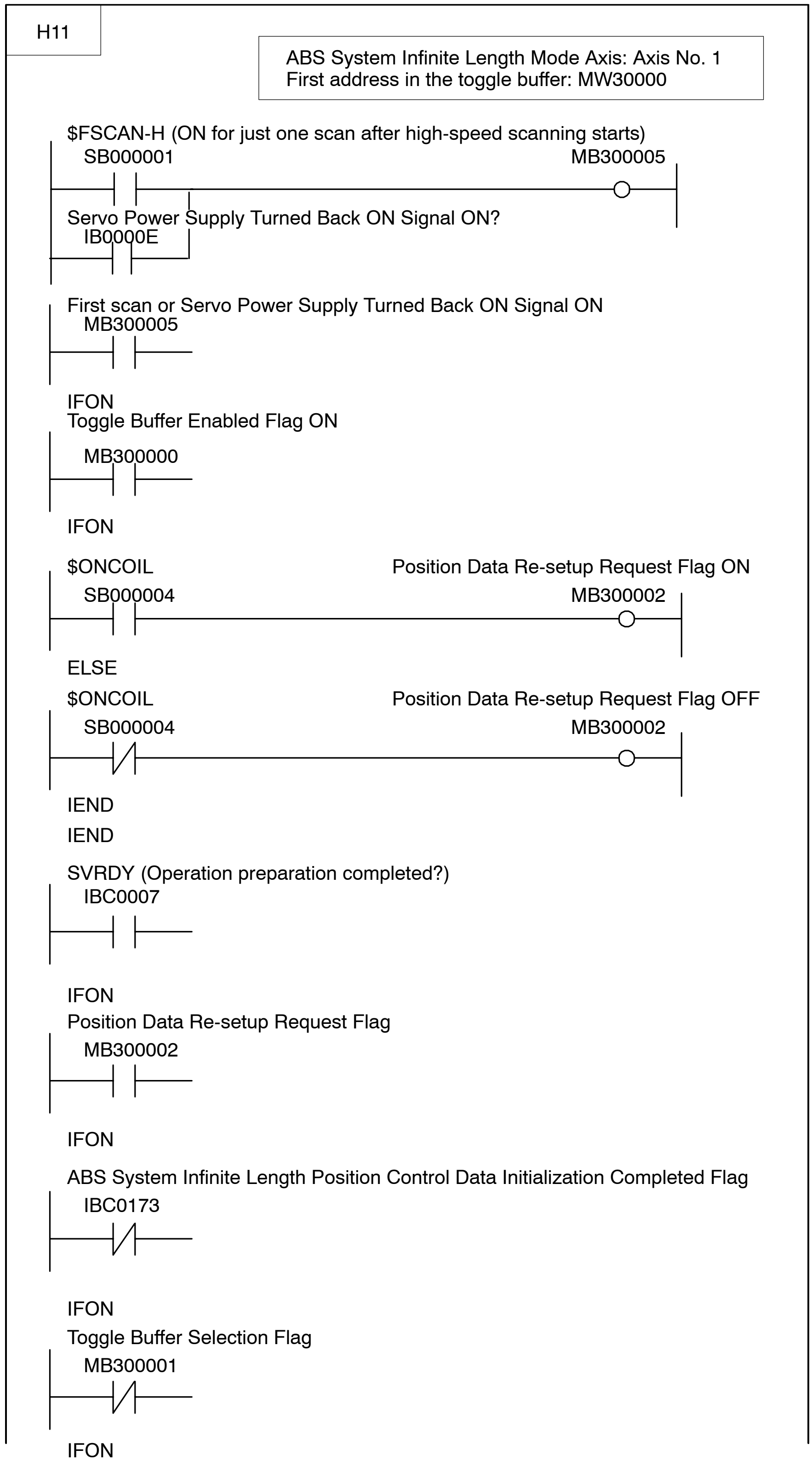
* Indicates the distance traveled while power is OFF.

Execute the following flowchart when Position Data Re-Setup Request is ON.

Follow the procedure below to set up position data again.


Start the high-speed scan drawing.
First scan after the start of high-speed scanning or signal indicating that the servo power supply was turned back ON?
NO
YES
Toggle Buffer Enabled Flag = 1?
NO
YES
Position Data Re-setup Request Flag = 1
Position Data Re-setup Request Flag = 0
Operation preparation completed?
NO
YES
Position Data Re-setup Request Flag = 1?
NO
YES
ABS System Infinite Length Position Control Data Initialization Request Flag = 0?
NO
YES
Toggle Buffer Selection Flag = 1?
NO
YES
Copy monitor parameters copied to buffer 1.
Copy monitor parameters copied to buffer 0.
ABS System Infinite Length Position Control Data Initialization Request Flag = 1
ABS System Infinite Length Position Control Data Initialization Request Flag = 0
Position Data Results Setup Enabled Flag
Exit the high-speed scan drawing.

The following programming example (ladder logic program) is for the flowchart shown above. The axis used here is axis No. 1 of module No. 1. Change the motion parameter register number if the module and axis numbers are different.



```
Setting parameters saved in buffer 0
├── ML30002                                  ⇒ OLC038
├── ML30004                                  ⇒ OLC03A
├── ML30006                                  ⇒ OLC03C
├── ML30008                                  ⇒ OLC03E
    ELSE
Setting parameters saved in buffer 1
├── ML30010                                  ⇒ OLC038
├── ML30012                                  ⇒ OLC03A
├── ML30014                                  ⇒ OLC03C
├── ML30016                                  ⇒ OLC03E
    IEND

  $ONCOIL                       ABS System Infinite Length Position Control Data
    SB000004                    Initialization Request Flag ON   OBC02D2
├────┤ ├─────────────────────────────────────────────────────────( )─┤

    ELSE

  $ONCOIL                       ABS System Infinite Length Position Control Data
    SB000004                    Initialization Request Flag OFF  OBC02D2
├────┤/├─────────────────────────────────────────────────────────( )─┤

  $ONCOIL                              Position Data Re-setup Request Flag OFF
    SB000004                                                     MB300002
├────┤/├─────────────────────────────────────────────────────────( )─┤

    IEND
    IEND
    IEND
    DEND
```

INFO

There are no restrictions on the executing order for ladder logic programs H10 and H11 when an absolute encoder is used as an Infinite Length Mode Axis.

8

Maintenance and Inspection

This chapter describes daily and regular inspection items to ensure that the MP930 can always be used at its best conditions.

8.1 Inspection Items

This section summarizes daily and regular inspection items that must be performed by the customer.

8.1.1 Daily Inspections

The following table lists the daily inspection items.

Table 8.1 Daily Inspection Items

No.	Inspection Item	Inspection Details	Criteria	Action
1	Installation conditions of Module, etc.	Check the mounting screws for looseness. Check whether the covers are all in place.	The screws and covers must be secured correctly.	Retighten the screws.
2	Connection conditions	Check the terminal screws for looseness.	The screws must be tight.	Retighten the screws.
		Check the connectors for looseness.	The connectors must be tight.	Retighten the connector set screws.
		Check the gap between crimp terminals.	There must be an appropriate gap between the terminals.	Correct as necessary.

8

No.	Inspection Item		Inspection Details	Criteria	Action
3	Indicators	POWER indicator	Check whether the indicator is lit.	The indicator must be lit. (It is abnormal if the indicator is unlit.)	
		READY indicator	Check whether the indicator is lit.	The indicator must be lit. (It is abnormal if the indicator is unlit.)	See *Chapter 9 Troubleshooting*.
		RUN indicator	Check whether the indicator is lit while the system is in RUN state.	The indicator must be lit. (It is abnormal if the indicator is unlit.)	See *Chapter 9 Troubleshooting*.
		ERR indicator	Check whether the indicator is unlit.	The indicator must be unlit. (It is abnormal if the indicator is lit.)	See *Chapter 9 Troubleshooting*.
		ALM indicator	Check whether the indicator is unlit.	The indicator must be unlit. (It is abnormal if the indicator is lit.)	See *Chapter 9 Troubleshooting*.
		BAT indicator	Check whether the indicator is unlit.	The indicator must be unlit. (The battery voltage is too low if the indicator is lit.)	Replace the battery.
		I/O indicator	Check whether the indicator comes on and goes off correctly.	The indicator must come on when I/O is ON, and go off when I/O is OFF. It is abnormal if the indicator does not come on or go off as above.	

8.1.2 Regular Inspections

This section describes inspection items that must be performed once or twice every six months to one year. Inspections must also be performed when the equipment is relocated or modified or when the wiring is changed.

Prohibited

- Do not replace the built-in fuse.

  If the customer replaces the built-in fuse, the MP930 may malfunction or break down.

  Contact your Yaskawa representative.

Table 8.2 Regular Inspection Items

No.	Inspection Item		Inspection Details	Criteria	Action
1	Operating environment	Ambient temperature	Check the temperature and humidity with a thermometer and hygrometer, respectively.	0 to 55 °C	If the MP930 is used inside a panel, treat the temperature inside the panel as the ambient temperature.
		Ambient humidity		33% to 95% RH	
		Atmosphere	Check for corrosive gases.	There must be no corrosive gases.	
2	Power supply voltage check	MC Unit	Measure the voltage between 24-VDC terminals.	20.4 to 28.8 VDC	Change the power supply as necessary.
		I/O Unit	Measure the voltage between 24-VDC terminals.	20.4 to 28.8 VDC	
3	Installation conditions	Looseness and excess play	Attempt to move the Unit.	The Module must be secured properly.	Retighten the screws.
		Dust and other foreign matter	Visually check.	The Module must be free from dust and other foreign matter.	Clean.
4	Connection conditions	Check the terminal screws for looseness.	Check by retightening the screws.	The screws must be tight.	Retighten.
		Gap between crimp terminals	Visually check.	There must be an appropriate gap between the terminals.	Correct.
		Looseness of connectors	Visually check.	The screws must be tight.	Retighten the connector set screws.
5	Battery	Battery	Check the “BAT” indicator on the front panel of the MC Unit.	The “BAT” indicator must be unlit.	If the “BAT” indicator is lit, replace the battery.

8

8.2 Battery

The MC Unit has one replaceable built-in battery, which is used to prevent the programs and data stored in the memory from being lost when a power failure occurs (i.e., when the power supply to the MC Unit is interrupted).

8.2.1 Battery Life

The built-in battery can retain the contents of the memory until the total time of power interruptions reaches one year. The warrantee period of the battery is five years from the date of purchase. However, these values differ according to the operating conditions, including the ambient temperature.

If the "BAT" indicator on the MC Unit lights, replace the battery with a replacement battery (DE9403582-1) within two weeks. Any delay in battery replacement will result in the programs and data stored in the memory being lost.

8.2.2 Battery Replacement

This section describes how to replace the battery.

■ Preparations

1. Saving the Memory Contents

   Before replacing the battery, save the programs and data from the memory of the MC Module to floppy disks or a hard disk. The saved programs and data will be used if the programs and data are accidentally deleted during battery replacement.

2. Obtain a Replacement Battery

   Obtain a replacement battery (DE9403582-1). This battery is not commercially available, and must be order from your nearest Yaskawa sales representative. The appearance of the battery is illustrated below.

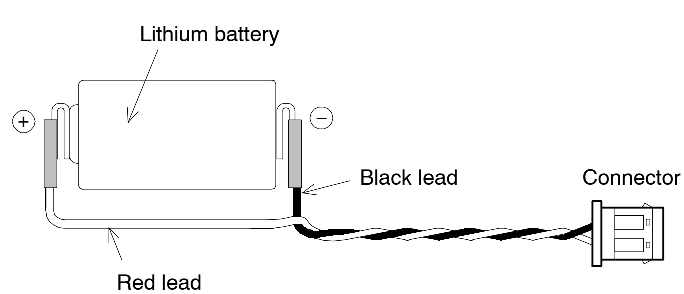


Figure 8.1 DE9403582-1 (Battery with Cable)

■ Replacing the Battery

Replace the battery according to the following procedure.

1. Make sure that the POWER indicator on the MC Unit is lit.
2. Open the battery cover on the lower part of the MC Unit.
3. Disconnect the connector on the end of the built-in battery lead from the connector on the MC Unit, then remove the built-in battery from the battery holder.
4. Firmly connect the connector on the end of the replacement battery lead to the connector on the MC Unit. Then, place the replacement battery into the battery holder. If the replacement battery is placed into the battery holder before connecting these connectors, the connector portion will be too narrow for your fingers to get in.
5. Make sure that the BAT indicator on the MC Unit is unlit.
6. Close the cover.

This completes the battery replacement procedure.

IMPORTANT

Be sure to replace the battery with the power supply to the MC Unit turned ON.

Replacing the battery with the power supply to the MC Unit turned OFF will result in the programs and data stored in the memory being lost.

9

Troubleshooting

This chapter describes the details, causes, and remedies for errors that can occur when using the system.

9

Reading this Chapter (Troubleshooting Procedures)

This section describes three checks available for checking the system when an errors occurs. They are checks by symptoms, error codes, and monitor functions of peripheral equipment. Checking procedures are categorized by status conditions to help determine the cause quickly.

■ Checking by Symptoms

Here, factors like indicators on the front of the Unit and the control status of all devices are visually checked to determine a cause and implement corrections.

■ Checking by Error Codes

Here, error codes generated when errors occur are monitored to determine a cause and implement corrections. Errors are classified into two groups.

Classification	Type of Error Code
Sequence Control Error Code	1. Starting from system register (S register) SW00040
Motion Control Error Code	2. Motion program error code 3. Servo error by axis

■ Checking by Monitor Functions of Peripheral Equipment

Here, the monitor functions of peripheral equipment are used to determine the control status and to find the cause of errors. The status of the following functions can be checked.

- Program monitoring
- Position monitoring
- Error monitoring
- Traces

9.1 Overview of Troubleshooting

This section shows the basic troubleshooting flow and provides a list of errors.

9.1.1 Basic Troubleshooting Flow

When a problem occurs, it is important to determine the cause and treat the problem fast to get the system up and running as quickly as possible. The following table shows the basic troubleshooting flow.

No.	Point	Basic Details Examined
1	**Visual Check**	a) Equipment operation (status while stopped) b) Power ON/OFF c) I/O equipment status d) Wiring status e) Status of indicators (indicators on all Units) f) Status of all switches (DIP switches and other switches) g) Parameters and program content check
2	**Error Check**	Observe whether the following alters the error in any way. a) Stopping the Controller. b) Resetting the alarm. c) Turning power OFF and ON.
3	**Narrowing the Range**	Consider possible failure locations based on the results of 1 and 2 above. a) Is the problem in the Controller or external? b) Is the problem in sequence control or motion control? c) Is the problem software or hardware?

9

9.1.2 Indicator Errors

Error details can be checked by the status of indicators on the front of the MP930 Units.

In the process, we narrow down the repair location in a program by getting an overview of the error from indicators, checking the contents of the system (S) registers, examining the drawing or function number that caused the error and then getting an overview of operation error details.

■ Indicators

The following indicators show operating status and error details for the MP930.

Indicator Section	indicator Name	Indicator Color	Significance When Lit
RDY O O RUN ERR O O ALM PRT1 O O PRT2 BAT O O	**RDY**	Green	System operating normally
	RUN	Green	User program running
	ERR	Red	System error
	ALM	Red	Minor system error
	PRT1	Green/Red	Port 1 sending/receiving
	PRT2	Green/Red	Port 2 sending/receiving.
	BAT	Red	Low battery voltage

Indicator Details

The following describes details and remedies for indicators showing operating status and errors in the MP930.

Classification	Indicator					Indicator Details	Remedy
	RDY	RUN	ALM	ERR	BAT ALM		
Normal	Unlit	Unlit	Unlit	**Lit**	Unlit	Hardware reset status	If this status continues for more than a second, the problem is a user program error or hardware failure. Troubleshoot system errors.
	Unlit	Unlit	Unlit	Unlit	Unlit	Initializing/busy	
	Unlit	**Lit**	Unlit	Unlit	Unlit	Drawing A executing	
	Lit	Unlit	Unlit	Unlit	Unlit	User program stopped (Offline Stop Mode)	This status will occur if a program is stopped from the CP-717 or by turning OFF the RUN switch.
	Lit	**Lit**	Unlit	Unlit	Unlit	User program executing normally	This status will occur during normal operation.
Error	Unlit	**Lit**	Unlit	**Lit**	Unlit	Serious failure has occurred.	See *Processing Flow When a User Program Error Occurs*.
	Lit	Unlit	Unlit	**Lit**	Unlit	1. Program memory initialization incomplete 2. Improper scan time setting	• Clear program memory from the System Definition Screen on the CP717. • If this does not restore the system, then hardware has probably failed.
	Unlit	Unlit	Unlit	Flashing	Unlit	Hardware errors No. of flashes 2: RAM diagnosis error 3: ROM diagnosis error 4: CPU function diagnosis error 5: FPU function diagnosis error	Troubleshoot system errors.

Classification	Indicator					Indicator Details	Remedy
	RDY	RUN	ALM	ERR	BAT ALM		
Warning	–	–	–	–	**Lit**	Battery alarm	Replace the battery.
	Lit	**Lit**	**Lit**	Unlit	Unlit	1. Operation error	See *Operation Error Remedies*.
						2. I/O error	See *I/O Error Remedies*.
						3. Wrong interrupt occurs	–
	System register warning (no indicator display)					1. CP-717 connection data	See *System Status*.
						2. Hardware status (power interruption, RUN/STOP, Test Mode, etc.)	See *System Status*.
Test Mode	Unlit	Unlit	Unlit	**Lit**	Unlit	Hardware errors No. of flashes 2: RAM diagnosis error 3: ROM diagnosis error 4: CPU function diagnosis error 5: FPU function diagnosis error 6: RTC interrupt time diagnosis error 7: WDT over time diagnosis error	Troubleshoot system errors.

9.2 System Errors

This section describes system error details and remedies.

9.2.1 Overview of System Errors

Indicators on the front panel of the MC Unit indicate the operating and error status of the MP930. Used the system (S) registers to get for more details on errors. Carefully check system register details to figure out the failure location and implement corrections.
The following sections describes the system register in more detail.

■ System Register Allocation

The following illustration shows the configuration of the system registers.

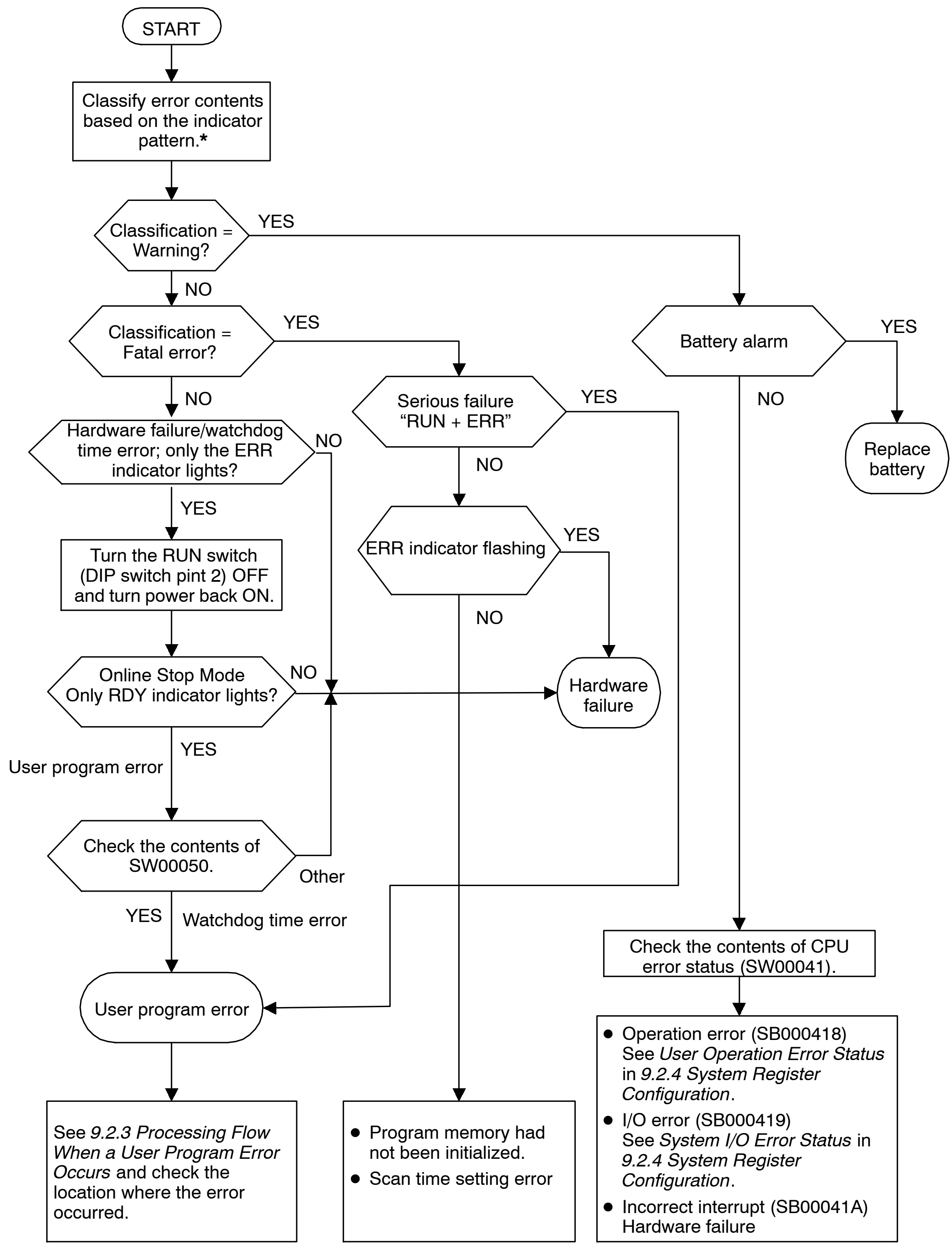
SW00000	System - Service Register
SW00030	System Status*
SW00050	System Error Status*
SW00080	User Operation Error Status*
SW00090	System Service Execution Status
SW00100	Interrupt Input Error Status
SW00110	User Operation Error Status, Details*
SW00200	System I/O Error Status
SW00424	Reserved for the System
SW00500	System Analysis Status
SW00530	Reserved for the System
SW00600	System Operation Error Status
SW00620	Reserved for the System
SW00800 **SW01023**	Reserved for optional modules

* See *9.2.4 System Register Configuration* for more details.

9

9.2.2 Processing Flow When a System Error Occurs

The following illustration shows the processing flow when a system error occurs.



* See *Indicator Details* in *9.1.2 Indicator Errors* for more details on indicator patterns.

9

9.2.3 Processing Flow When a User Program Error Occurs

A serious failure has probably occurred if the RUN and ERR indicators are lit on the front panel of the MP930. Use the following procedures to check the error program.

1. Check by Type of Serious Failure Error

Check the contents of SW00050 (Error Type) to determine if the error is a system error or a user program error.

2. Check by Type of Error Program

Check the contents of SW00055 (Program Type) to determine if the error occurred in a drawing or in a function.

3. Check the Error Drawing

Check the contents of SW00054 (Error Task) and SW00056 (DWG No.) to find the error drawing.

4. Check the Error Function

The error occurred in a function if SW00056 (DWG No.) reads 01zzH.
Check the contents of SW00057 (Error Task) and SW00058 (DWG No.) to find the error drawing.
Check SW00059 (Function Referencing DWG Step No.) for the STEP number where an operation error occurred.

5. Check Whether an Operation Error Occurred

Check the error count of all drawings at SW00080 to SW00088. Operation errors are occurring if the count is going up.

6. Check the Details and Location of Operation Errors

a) Check Error Details

Check error codes for drawings where the error count is going up.

DWG.A: SW00111	DWG.H: SW00143
DWG.I: SW00127	DWG.L: SW00175

b) Check the DWG Number

Check the error DWG number for the DWG number where an error occurred.

DWG.A: SW00120	DWG.H: SW00152
DWG.I: SW00136	DWG.L: SW00184

c) Check the Function Referencing DWG Number and Function Referencing STEP Number if an error occurred in a function.

DWG.A: SW001201, 2	DWG.H: SW00153,4
DWG.I: SW00137, 8	DWG.L: SW00185, 6

7. Correct the Program

Correct the program at the point where the error occurred.

9

9.2.4 System Register Configuration

■ System Status

System status indicates the operating status and error details for the system. System status details are used to determine whether hardware or software is the cause of an error.

Table 9.1 System Status List

Name	Register No.	Contents		
Reserved for the System	SW00030 to SW00039	(Not used)		
CPU Status	SW00040	SB000400	READY	1: Normal, 0: Communications/Self-diagnosis error
		SB000401	RUN	1: Running, 0: Stopped
		SB000402	ALARM	1: Alarm, 0: Normal
		SB000403	ERROR	1: Error, 0: Normal
		SB000404	RESUME	1: Continued startup operation 0: New startup operation
		SB000405	START STATUS	1: Restoration from power interruption 0: Ordinary restoration
		SB000406	Reserved by system.	(Not used)
		SB000407	WEN	1: WRITE enabled, 0: WRITE disabled
		SB000408	Reserved by system.	(Not used)
		SB000409		
		SB00040A		
		SB00040B		
		SB00040C		
		SB00040D		
		SB00040E	OPERATION STOP REQUEST	1: STOP selection, 0: RUN selection
		SB00040F	Reserved by system.	(Not used)

9

Name	Register No.	Contents		
CPU Error Status	SW00041	SB000410	Serious failure	1: WDGE, undefined command See SW00050 for more details.
		SB000411	Program memory error	1: Program memory error
		SB000412	Calendar IC error	1: Calendar IC error
		SB000413	Reserved by system.	(Not used)
		SB000414		
		SB000415		
		SB000416		
		SB000417		
		SB000418	User operation error	1: User operation error
		SB000419	I/O error	1: I/O error
		SB00041A	Reserved by system.	(Not used)
		SB00041B		
		SB00041C to SB00041F		
Software Switch Selection Status	SW00047	SB000470	Startup mode in case of power interruption	0: New startup, 1: Continued startup
		SB000471 SB000472	Reserved by system.	(Not used)
		SB000473	Program WRITE selection	0: WRITE enabled, 1: WRITE disabled
		SB000474	Startup mode in case of ordinary power interruption	0: New startup, 1: Continued startup
		SB000475	Hot swapping program interlock	0: Invalid, 1: Valid
		SB000476 to SB00047F	Reserved by system.	(Not used)

Name	Register No.	Contents		
Hardware Status Configuration	SW00048	SB000480	FLASH	DIP switch report 0: ON 1: OFF
		SB000481	RUN	
		SB000482	INIT	
		SB000483	TEST	
		SB000484	–	
		SB000485	S.TEST	
		SB000486	–	
		SB000487	Battery alarm	1: Battery alarm
Reserved by system.	SW00049	SB000490 to SB00049F	Reserved by system.	(Not used)

System Error Status

The following table lists data when a system error status list is generated.

Table 9.2 System Error Status List

Name	Register No.	Contents	
Error Type	SW00050	0001H	Watchdog time error
		0002H	Bus time over
		0005H	Execution of a single-step interrupt
		0006H	Execution of a breakpoint interrupt
		0007H	Bound error
		0008H	Execution of an undefined command
		0009H	Coprocessor error
		000CH	Double fault
		000DH	Illogical TTS
		000EH	Segment does not exist
		000FH	Stack error
		0010H	General protection error
		0011H	Page fault
		0012H	Segment boundary check
		0041H	ROM diagnosis error
		0042H	RAM diagnosis error
		0043H	CPU diagnosis error
		0044H	FPU diagnosis error
		0081H	Overflow, underflow*
		0083H	0 division*
		0084H	FPU segment over*
		0085H	FPU operation error*
		0088H	Index error*
		0090H	General protection error*
Error Code	SW00051	For system error analysis	
Error IP	SW00052	For system error analysis	
Error CS	SW00053	For system error analysis	

* These errors occur only in version A11 and A12. In the versions other than A11 and A12, the errors occur as user operation error.

9

Name	Register No.	Contents	
Error Task	SW00054	0001H: System	0004H: DWG.H
		0002H: DWG.A	0005H: DWG.L
		0003H: DWG.I	
Program Type	SW00055	0001H: System	0004H: DWG.H
		0002H: DWG.A	0005H: DWG.L
		0003H: DWG.I	0008H: Function
		0010H: Main motion program 0011H: Motion subprogram	
Error DWG No.	SW00056	Parent drawing: FFFFH, Function: 0100H Child drawing: xx00H (Hxx: Child drawing no.) motion program No. Grandchild drawing: xxyyH (Hyy: Grandchild drawing No.)	
Function Referencing DWG Type	SW00057	Type of DWG that references the function in which an error occurred. 0001H: DWG.A 0002H: DWG.I 0003H: DWG.H 0005H: DWG.L	 0008H: Function 0010H: Main motion program 0011H: Motion subprogram
	SW00058	Type of DWG that references the function in which an error occurred. Parent drawing: FFFFH, Function: 0100H Child drawing: xx00H (Hxx: Child drawing no.) motion program no. Grandchild drawing: xxyyH (Hyy: Grandchild drawing no.)	
Function Referencing DWG Type	SW00059	STEP Number of the DWG that references the function in which an error occurred. 0 when DWG err + motion program	
Error Data	SW00060	SW00060	For system error analysis (ES)
		SW00061	For system error analysis (DS)
		SW00062	For system error analysis (DI)
		SW00063	For system error analysis (SI)
		SW00064	For system error analysis (BP)
		SW00065	For system error analysis (SP)
		SW00066	For system error analysis (BX)
		SW00067	For system error analysis (DX)
		SW00068	For system error analysis (CX)
		SW00069	For system error analysis (AX)
		SW00070 to SW00079	Reserved by system.

User Operation Error Status

The following tables list data when a user operation error occurs.

Table 9.3 User Operation Error Status - 1

Name	Register No.	Contents
DWG.A Error Count	SW00080	Operation error code: See User Operation Error Status - 3. Error code when an index error occurs: User Operation Error Status - 4.
Error Code	SW00081	
DWG.I Error Count	SW00082	
Error Code	SW00083	
DWG.H Error Count	SW00084	
Error Code	SW00085	
Reserved by system.	SW00086	
	SW00087	
DWG.L Error Count	SW00088	
Error Code	SW00089	

Table 9.4 User Operation Error Status - 2

<table>
<tr><th rowspan="2">Name</th><th colspan="4">Register No.</th><th rowspan="2">Remarks</th></tr>
<tr><th>DWG.A</th><th>DWG.I</th><th>DWG.H</th><th>DWG.L</th></tr>
<tr><td>Error Count</td><td>SW00110</td><td>SW00126</td><td>SW00142</td><td>SW00174</td><td rowspan="16">Error DWG number

Parent drawing: FFFFH
Child drawing: xx00H
(Hxx: Child drawing no.)
Grandchild drawing: xxyyH
(Hyy: Grandchild drawing no.)
Function: 0100H

Function Referencing DWG Number

Number of the DWG that references the function in which an error occurred.

Function Referencing DWG STEP Number

STEP number of the DWG that references the function in which an error occurred. This will be “□0□” if the error occurred inside the DWG.</td></tr>
<tr><td>Error Code</td><td>SW00111</td><td>SW00127</td><td>SW00143</td><td>SW00175</td></tr>
<tr><td rowspan="2">Error A Register</td><td>SW00112</td><td>SW00128</td><td>SW00144</td><td>SW00176</td></tr>
<tr><td>SW00113</td><td>SW00129</td><td>SW00145</td><td>SW00177</td></tr>
<tr><td rowspan="2">Modification A Register</td><td>SW00114</td><td>SW00130</td><td>SW00146</td><td>SW00178</td></tr>
<tr><td>SW00115</td><td>SW00131</td><td>SW00147</td><td>SW00179</td></tr>
<tr><td rowspan="2">Error F Register</td><td>SW00116</td><td>SW00132</td><td>SW00148</td><td>SW00180</td></tr>
<tr><td>SW00117</td><td>SW00133</td><td>SW00149</td><td>SW00181</td></tr>
<tr><td rowspan="2">Modification F Register</td><td>SW00118</td><td>SW00134</td><td>SW00150</td><td>SW00182</td></tr>
<tr><td>SW00119</td><td>SW00135</td><td>SW00151</td><td>SW00183</td></tr>
<tr><td>Error IP</td><td>SW00120</td><td>SW00136</td><td>SW00152</td><td>SW00184</td></tr>
<tr><td>Error CS</td><td>SW00121</td><td>SW00137</td><td>SW00153</td><td>SW00185</td></tr>
<tr><td>Error DWG No.</td><td>SW00122</td><td>SW00138</td><td>SW00154</td><td>SW00186</td></tr>
<tr><td>Function Referencing DWG Type</td><td>SW00123</td><td>SW00139</td><td>SW00155</td><td>SW00187</td></tr>
<tr><td>Function Referencing DWG STEP Number</td><td>SW00124</td><td>SW00140</td><td>SW00156</td><td>SW00188</td></tr>
<tr><td>Reserved by system.</td><td>SW00125</td><td>SW00141</td><td>SW00157</td><td>SW00189</td></tr>
</table>

Table 9.5 User Operation Error Status - 3

Name	Error Code	Error Contents	User	System Default
Integer Operation	0001H	Integer operation - underflow	Yes	–32768 [–32768]
	0002H	Integer operation - overflow	Yes	32768 [32768]
	0003H	Integer operation - division error	Yes	The A register remains the same.
	0009H	Double integer operation - underflow	Yes	–2147483648 [–2147483648]
	000AH	Double integer operation - overflow	Yes	2147483648 [2147483648]
	000BH	Double integer operation - division Error	Yes	The A register remains the same.
	000xH	Integer operation error within operation error processing drawing (x = 1 to B)	No	Default indicated above.
Real Number Operation	0010H	Integer storage - non-numeric error	Yes	Store not executed. [00000]
	0011H	Integer storage - underflow	Yes	Store not executed. [–32768]
	0012H	Integer storage - overflow	Yes	Store not executed. [+32768]
	0021H	Integer storage - underflow	Yes	Store not executed. [–1.0E+38]
	0022H	Integer storage - overflow	Yes	Store not executed. [1.0E+38]
	0023H	Real number operation - division-by-zero error	Yes	Operation not executed. The F register remains the same.
	0030H	Real number operation - invalid operation (non-numeric)	No	Operation not executed.
	0031H	Real number operation - exponent underflow	No	0.0
	0032H	Real number operation - exponent overflow	No	Maximum value
	0033H	Real number operation - division error (non-numeric 0/0)	No	Operation not executed.
	0034H	Real number storage - exponent underflow	No	Stores 0.0.

Name	Error Code	Error Contents		User	System Default
Real Number Operation	0040H to 0059H	Real number operation error within a standard system function		No	Interrupt operation and output = 0.0
		0040H: SQRT	0041H: SIN	0042H: COS	0043H: TAN
		0044H: ASIN	0045H: ACOS	0046H: ATAN	0047H: EXP
		0048H: LN	0049H: LOG	004AH: DZA	004BH: DZB
		004CH: LIM	004DH: PI	004EH: PD	004FH: PID
		0050H: LAG	0051H: LLAG	0053H: FGN	0054H: IFGN
		0054H: LAU	0055H: SLAU	0056H: REM	0057H: RCHK
		0058H: BSRCH	0059H: SQRT		
		1000H or 2000H is added for an index error.			
		Operation error in a motion function The number of the function where the error occurred + 200H will be stored.			
		0200H: MOV	0201H: MVS	0202H: MCC	0203H: MCW
		0204H:	0205H: SKP	0206H:	0207H:
		0208H: POS	0209H:	020AH: ACC	020BH: DCC
		020CH: SCC	020DH: VEL	020EH: INP	020FH: IAC
		0210H: IDC	0211H: IFP	0212H: FMX	0213H:
		0214H: MVT	0215H: EXM		

Table 9.6 User Operation Error Status - 4

Name	Error Code	Error Contents		User	System Default
Integer - Real Number Operations	1000H	Index error within DWG		No	Re-executed with i, j = 0
	2000H	Index error within function		No	Re-executed with i, j = 0
Integer Operation	x060H to x077H (x = 1, 2)	Index error within integer system function		No	The A register remains the same.
		x06DH: PI	x06DH: PD	x06FH: PID	x070H: LAG
		x071H: LLAG	x072H: FGN	x073H: IFGN	x074H: DAU
		x075H: SLAU	x076H: FGN	x077H: IFGN	

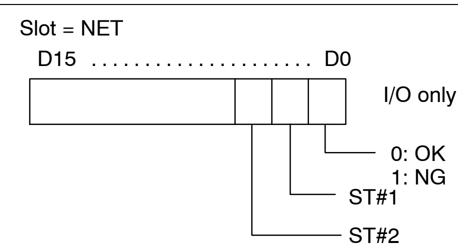
System Service Execution Status

Name	Register No.	Remarks
System Error Count	SW00090	
System Error Code	SW00091	
Failure Occurrence Count	SW00092	
Failure Restoration Count	SW00093	
Reserved by system.	SW00094 to SW00097	(Not used)
Existence Of Data Trace Definition	SW00098	Bit 0 to 3 = Group 1 to 4 Definition exists = 1, No definition = 0
Data Trace Execution Status	SW00099	Bit 0 to 3 = group 1 to 4 Trace executing = 1, Trace stopped = 2

Table 9.7 Latest Data Trace Record Number

Name	Register No.	Remarks
Data Trace Group 1	SW00100	Latest record number
Data Trace Group 2	SW00101	Latest record number
Data Trace Group 3	SW00102	Latest record number
Data Trace Group 4	SW00103	Latest record number

■ System I/O Error Status

Name	Register No.	Remarks
I/O Error Count	SW00200	Number of I/O errors
Number Of Input Errors	SW00201	Number of input errors
Input Error Address	SW00202	Latest input error address (For future use) (Register number of OWxxx)
Output Error Count	SW00203	Number of times an output error has occurred
Output Error Address	SW00204	Latest input error address (For future use) (Register number of OWxxx)
Reserved by system.	SW00205	(Not used)
	SW00206	
	SW00207	
I/O error status	SW00208 to SW00211	(Not used)
	SW00212 to SW00215	(Not used)
	SW00216	

System Operation Error Status

Table 9.8 System Operation Error Code Status - 1

Name	Register No.	Remarks
Error Count	SW00600	Reported when an operation error occurs in the system program.
Error Code	SW00601	
Error A Register	SW00602	
	SW00603	
Modification A Register	SW00604	
	SW00605	
Error F Register	SW00606	
	SW00607	
Modification F Register	SW00608	
	SW00609	
Error IP	SW00610	
Error CS	SW00611	
Error DWG No.	SW00612	
Function Referencing DWG Type	SW00613	
Function Referencing DWG STEP Number	SW00614	
Reserved by system.	SW00615	(Not used)

Table 9.9 System Operation Error Code Status - 2

Name	Error Code	Error Contents	System Default
Integer Operation	0001H	Integer operation - underflow	–32768
	0002H	Integer operation - overflow	+32767
	0003H	Integer operation - division error	0

9

Table 9.10 Optional Module Information

Name	Register No.	Remarks
Optional Module Information	SW00800 to SW00803	Slot 0-mounted module information
	SW00804 to SW00807	(Not used)
	SW00808 to SW00811	(Not used)
	:	
	SW01020 to SW01023	(Not used)

9.3 Motion Errors

This section describes the details and remedies for errors that occur in motion functions.

9.3.1 Description of Motion Errors

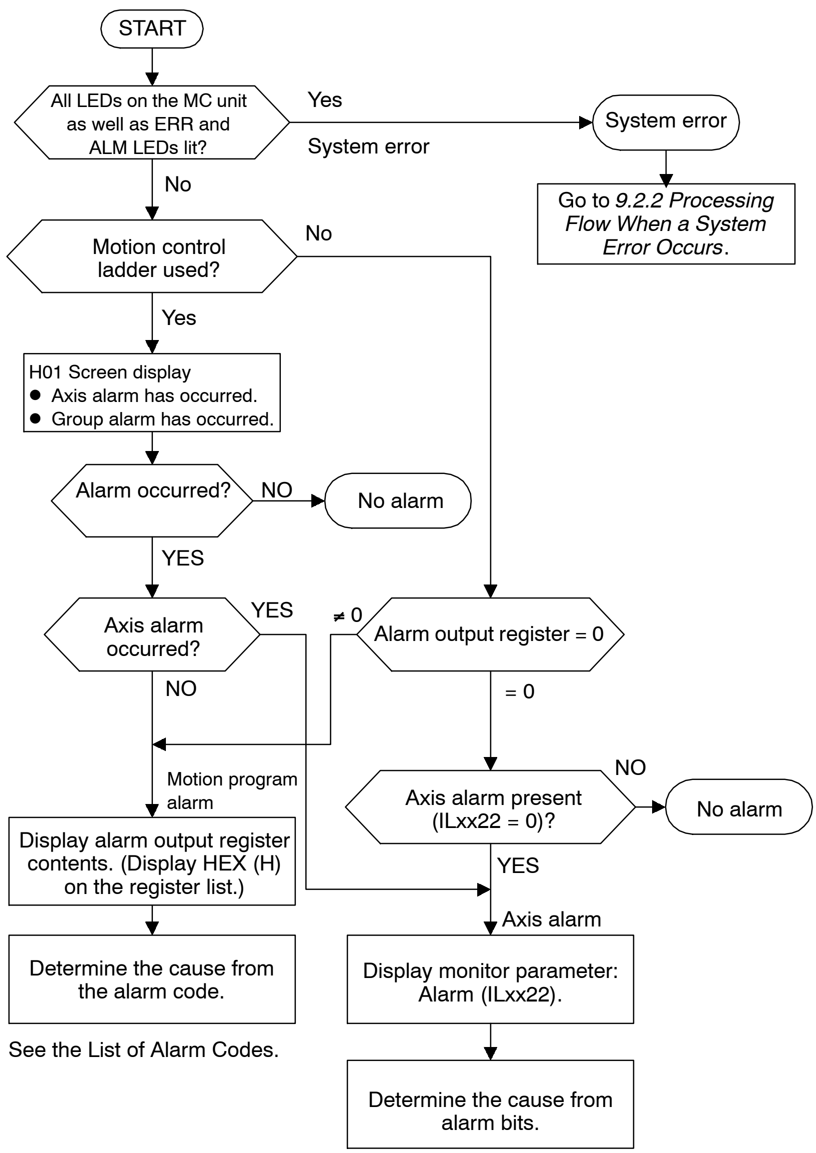
Motion errors in the MP930 are classified as alarms detected in motion programs and axis alarms detected in Servopack units.

The failure location can be determined and appropriate corrections can be taken simply by checking the contents of the alarm output register set from the Define Group Scree) for motion program alarms and the contents of monitor parameter: Alarm (ILxx22) for axis alarms.

9.3.2 Processing Flow When a Motion Error Occurs

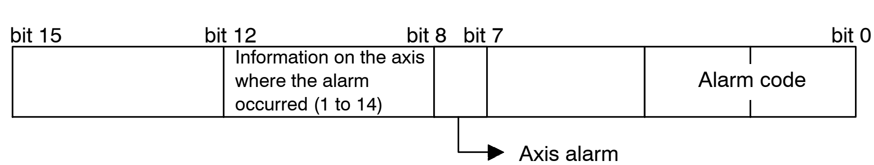
■ Troubleshooting Flow

The following illustration shows the troubleshooting flow when a motion error occurs.



■ Motion Alarm Configuration

The following illustration shows the motion alarm configuration stored in the alarm output register.



List of Motion Program Alarm Codes

The following table lists the List of Motion Program Alarm Codes. Use HEX(H) for the Display Mode when displaying the register list.

Name	Alarm Code	Contents
Program Alarm	**0**	No alarm
	1	–
	2	Division-by-zero error
	3	–
	4	–
	10h	Circumference specified alarm for radius specification
	11h	Exceeded the interpolation feed speed
	12h	No interpolation feed speed specified
	13h	Out of range after changing acceleration and deceleration parameters
	14h	Circular length exceeds LONG_MAX
	15h	No vertical axis specified for circular plane specification
	16h	No horizontal axis specified for circular plane specification
	17h	Exceeded the specified number of axes
	18h	Exceeded the specified number of turns
	19h	Radius exceeds LONG_MAX
	1Ah	Center point specification error
	1Bh	Emergency stop reference executing
	1Ch	Linear interpolation block distance traveled exceeds LONG_MAX
	1Dh	FMX not defined
	1Eh	Address T outside the range
	1Fh	Address P outside the range
	20h	REG data error

9

Name	Alarm Code	Contents
Axis Alarm*	**80h**	Logic-controlled axis use prohibited
	81h	Value exceeding POSMAX specified at Infinite Length Mode Axis specification.
	82h	Distance the axis traveled exceeds LONG_MAX
	83h	Illegal control mode
	84h	Duplicate motion commands
	85h	Duplicate motion command response
	86h	Illegal motion command mode
	87h	Outside the VEL data range setting
	88h	Outside the INP data range setting
	89h	Outside the ACC/SCC/SCC data range setting

* Axis numbers are stored in bits 8 to 11 when an axis alarm occurs.

■ Motion Parameter: Alarm ILxx22 Details

The following tables lists the axis alarm flags (ILxx22).

Parameter No.	Contents	Remarks
b0: SVERR	Servo Amp error • Servopack alarm detected: See IWxx24 for alarm details.	Servo OFF
b1: OTF	Positive direction overtravel • Servo Amp overtravel in the positive direction detected (P_OT signal ON)	
b2: OTR	Negative direction overtravel • Servo Amp overtravel in the negative direction detected (N_OT signal ON)	
b3: SOTF	Positive direction software limit • Machine movement toward the positive software limit range detected	
b4: SOTR	Negative direction software limit • Machine movement toward the negative software limit range detected	
b5: SVOFF	Servo power supply incomplete • A move was executed with the servo OFF.	Servo OFF

Parameter No.	Contents	Remarks
b6: TIOMEOVER	Positioning time exceeded • Positioning was not completed in the time set at OWxx34: Positioning Complete Check Time after distribution.	
b7: DISTOVER	Positioning distance traveled exceeded • A move was executed that exceeded the positioning distance travel limit.	
b8: FILTYPERR	Change filter type error • The type of filter was changed before distribution was completed.	
b9: FILTIMERR	Change filter time constant error • The filter time constant was changed before distribution was completed.	
b10: MODERR	Control mode error • A Position Control Mode motion command is used in a mode other than position control.	
b11: ZSET_NRDY	Zero point not set • The zero point is not set and a move was executed without a set zero point.	
b12: ZSET_MOV	Zero point set during movement • A ZSET motion command was specified while an axis was moving.	
b13: CN_ERR	User constant setting error • Illegal setting when a CN_RD/CN_RD motion command was specified.	
b14: WDT_ERR	MECHATROLINK servo synchronized communications error • The MP930 detected a synchronized communications error with the MECHATROLINK servo.	Servo OFF
b15: COM_ERR	MECHATROLINK servo communications error • The MP930 detected a communications error with the MECHATROLINK servo two consecutive times.	Servo OFF
b16: SVTIMOUT	MECHATROLINK servo command timeout error • A MECHATROLINK servo command was not completed within the specified time.	Servo OFF
b17: ABSOVER	ABS encoder rotation count over • The number of ABS encoder rotations exceeded the range of the MP930.	
b18 to b31:	Not used	

A

External Dimensions

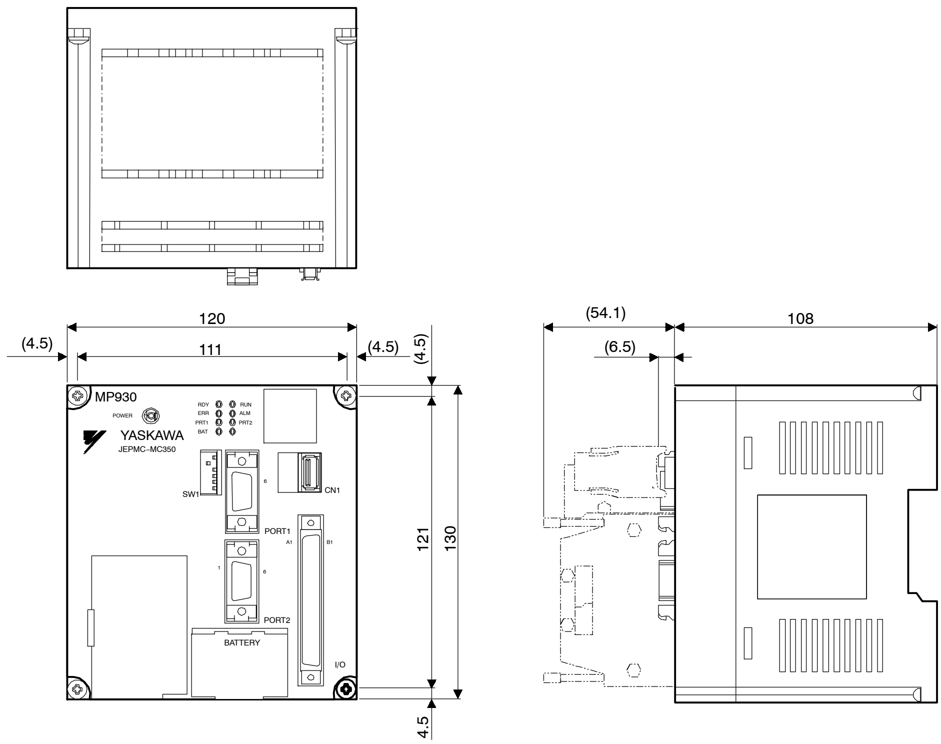
The external dimensions of each unit for the machine controller MP930 are shown.

A

A.1 External Dimensions of the MP930 Units

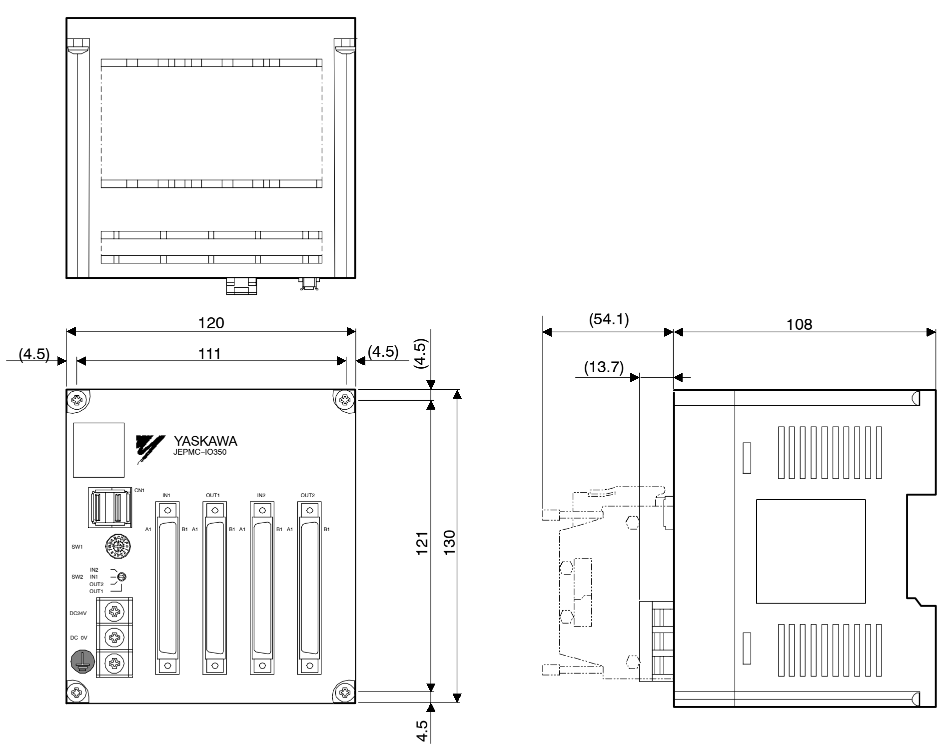
■ MC Unit (Dimensions in mm)

Type: JEPMC-MC350



■ I/O Unit (Dimensions in mm)

Type: JEPMC-IO350



Classification	Command	Name	Programming Format	Function/Meaning
Basic Control Commands	**ABS**	ABSOLUTE MODE	ABS;	Treats all subsequent coordinate words as absolute values.
	INC	INCREMENTAL MODE	INC;	Treats all subsequent coordinate words as incremental values.
	POS	CURRENT POSITION SET	POS [*axis1*] – [*axis2*] – ⋯;	Changes the current values to the desired coordinate values for up to 14 axes simultaneously. Subsequent move commands use this new coordinate system.
	PLN	COORDINATE PLANE SETTING	PLN [*axis1*] [*axis2*]	Designates the coordinate plane to be used for a command requiring a plane designation command.
	MVM	MOVE ON MACHINE COORDINATE	MVM MOV [*axis1*]– [*axis2*]–; or MVM MVS [*axis1*]– [*axis2*]–;	Goes to the target position on the machine coordinate system. The coordinate system set automatically on completion of the zero point return is called a machine coordinate system. This coordinate system is not affected by the POS command.
	PLD	PROGRAM CURRENT POSITION UPDATE	PLD [*axis1*] – [*axis2*] – ⋯;	Updates the program current position for axes shifted by manual intervention. Up to 14 axes can be designated.
Speed and Acceleration/ Deceleration Commands	**ACC**	ACCELERATION TIME CHANGE	ACC [*axis1*] – [*axis2*] – ⋯;	Sets the acceleration time for linear acceleration/deceleration for up to 14 axes simultaneously.
	SCC	S-CURVE TIME CONSTANT CHANGE	SCC [*axis1*] – [*axis2*] – ⋯;	Sets the time constant for moving average acceleration/deceleration for up to 14 axes simultaneously.
	VEL	SET VELOCITY	VEL [*axis1*] – [*axis2*] – ⋯;	Sets the feed speed for up to 14 axes.
	IAC	INTERPOLATION ACCELERATION TIME CHANGE	IAC T–;	Sets the acceleration time for linear acceleration/deceleration for interpolation travel.
	IDC	INTERPOLATION DECELERATION TIME CHANGE	IDC T–;	Sets the deceleration time for linear acceleration/deceleration for interpolation travel.
	IFP	INTERPOLATION FEED SPEED RATIO SETTING	IFP P–;	Designates the maximum feed % for the speed designation during an interpolation feed.
	FMX	MAXIMUM INTERPOLATION FEED SPEED SETTING	FMX T–;	Sets the maximum speed during an interpolation feed. The interpolation acceleration time is the time from “0” until this speed is reached.

A

Classification	Command	Name	Programming Format	Function/Meaning
High-Level Control Commands	**PFN**	IN-POSITION CHECK	MVS [*axis1*] – [*axis2*] – ··· PFN; or PFN [*axis1*] [*axis2*] ;	Proceeds to the next block after the positioning commanded by the interpolation travel command in the same block or a previous block enters the positioning completion range (parameter setting).
	INP	SECOND IN-POSITION CHECK	INP [*axis1*] – [*axis2*] – ···;	Proceeds to the next block after the positioning subsequently commanded by the interpolation travel command with PFN enters the second positioning completion range.
	SNG	IGNORE SINGLE BLOCK SIGNAL	SNG MVS [*axis1*] 100. [*axis2*] 200. F1000;	A block with this command will be executed continuously, even in single-block operation mode. SNG cannot be designated on its own.
	UFC	USER FUNCTION CALL	UFC *Function_name Input_data, Input_address, Output_data*;	Calls a function created by the user.
Sequence Commands	=	SUBSTITUTE	(Result) = (Arithmetic expression)	Substitutes operation results. Performs calculations from left to right (with no order of priority).
	+	ADD	MW– = MW– + MW–; MW– = MW– + 123456; MW– = 123456 + MW–;	Performs integer and real number addition. Calculates combinations of integers and real numbers as real numbers.
	–	SUBTRACT	MW– = MW– – MW–; MW– = MW– – 123456; MW– = 123456 – MW–;	Performs integer and real number subtraction. Calculates combinations of integers and real numbers as real numbers.
	✻	MULTIPLY	MW– = MW– ✻ MW–; MW– = MW– ✻ 123456; MW– = 123456 ✻ MW–;	Performs integer and real number multiplication. Calculates combinations of integers and real numbers as real numbers.
	/	DIVIDE	MW– = MW–/MW–; MW– = MW–/123456; MW– = 123456/MW–;	Performs integer and real number division. Calculates combinations of integers and real numbers as real numbers.
	MOD	REMAINDER	MW– = MW–/MW–; MW– = MOD;	When programmed in the next block after a division, MOD stores the remainder in the designated register.

<table>
<tr><th>Classification</th><th>Command</th><th>Name</th><th>Programming Format</th><th>Function/Meaning</th></tr>
<tr><td>Sequence Commands</td><td>|</td><td>OR (logical OR)</td><td>MB– = MB– | MB–;
MB– = MB– | 1;
MW– = MW– | MW–;
MW– = MW– | H00FF;</td><td>Performs bit/integer logical OR.</td></tr>
<tr><td></td><td>^</td><td>XOR (logical exclusive OR)</td><td>MW– = MW– ^ MW–;
MW– = MW– ^ H00FF;</td><td>Performs integer logical exclusive OR.</td></tr>
<tr><td></td><td>&</td><td>AND (logical AND)</td><td>MB– = MB– & MB–;
MB– = MB– & 1;
MW– = MW– & MW–;
MW– = MW– & H00FF;</td><td>Performs bit/integer logical AND.</td></tr>
<tr><td></td><td>!</td><td>NOT (logical complement)</td><td>MB– = !MB–;
MB– = !1;
MW– = !MW–;
MW– = !H00FF;</td><td>Performs bit/integer logical complement (inverts bits).</td></tr>
<tr><td></td><td>( )</td><td>PARENTHESES</td><td>MW– = MW– & (MW– | MW–);</td><td>The logical arithmetic expression inside parentheses is calculated first.</td></tr>
<tr><td></td><td>S{ }</td><td>SET BIT</td><td>S{MB–} = MB– & MB–;</td><td>If the logical operation result is “true,” the designated bit turns ON. The designated bit does not turn OFF, even if the logical operation result is “false.”</td></tr>
<tr><td></td><td>R{ }</td><td>RESET BIT</td><td>R{MB–} = MB– & MB–;</td><td>If the logical operation result is “true,” the designated bit turns OFF. The designated bit does not turn ON, even if the logical operation result is “false.”</td></tr>
<tr><td></td><td>SIN</td><td>SINE</td><td>SIN (MW–);
SIN (90);</td><td>Obtains the sine of the integer or real number (deg), and returns a real value.</td></tr>
<tr><td></td><td>COS</td><td>COSINE</td><td>COS (MW–);
COS (90);</td><td>Obtains the cosine of the integer or real number (deg), and returns a real value.</td></tr>
<tr><td></td><td>TAN</td><td>TANGENT</td><td>TAN (MF–);
TAN (45.0);</td><td>Obtains the tangent of the real number (deg), and returns a real value.</td></tr>
<tr><td></td><td>ASN</td><td>ARC SINE</td><td>ASN (MF–);
ASN (45.0);</td><td>Obtains the arc sine of the real number (deg), and returns a real value.</td></tr>
<tr><td></td><td>ACS</td><td>ARC COSINE</td><td>ACS (MF–);
ACS (90.0);</td><td>Obtains the arc cosine of the real number (deg), and returns a real value.</td></tr>
</table>

Classification	Command	Name	Programming Format	Function/Meaning
Sequence Commands	**ATN**	ARC TANGENT	ATN (MW–); ATN (45);	Obtains the arc tangent of the integer or real number (deg), and returns a real value.
	SQRT	SQUARE ROOT	SQT (MW–); SQT (100);	Obtains the square root of the integer or real number, and returns a real value.
	BIN	BCD-TO-BINARY	BIN (MW–);	Converts BCD data to binary data.
	BCD	BINARY-TO-BCD	BCD (MW–);	Converts binary data to BCD data.
	= =	MATCH	IF MW– = = MW–; WHILE MW– = = MW–;	Used in an IF or WHILE conditional expression. If the left side and right side match, the condition is “true.”
	< >	MISMATCH	IF MW– < > MW–; WHILE MW– < > MW–;	Used in an IF or WHILE conditional expression. If the left side and right side do not match, the condition is “true.”
	>	GREATER THAN	IF MW– > MW–; WHILE MW– > MW–;	Used in an IF or WHILE conditional expression. If the left side is greater than the right side, the condition is “true.”
	<	LESS THAN	IF MW– < MW–; WHILE MW– < MW–;	Used in an IF or WHILE conditional expression. If the left side is less than the right side, the condition is “true.”
	> =	GREATER THAN OR EQUAL TO	IF MW– >= MW–; WHILE MW– >= MW–;	Used in an IF or WHILE conditional expression. If the left side is greater than or equal to the right side, the condition is “true.”
	< =	LESS THAN OR EQUAL TO	IF MW– <= MW–; WHILE MW– <= MW–;	Used in an IF or WHILE conditional expression. If the left side is less than or equal to the right side, the condition is “true.”
	SFR	RIGHT SHIFT	SFR MB– N– W–;	Shifts only the designated number of word variables to the right.
	SFL	LEFT SHIFT	SFL MB– N– W–;	Shifts only the designated number of word variables to the left.
	BLK	BLOCK MOVE	BLK MW– MW– W–;	Moves the block (constant designation) beginning with the designated bit (word) variable.
	CLR	CLEAR	CLR MB– W–;	Sets the number of constants specified in the variable group beginning with the designated bit (word) variable to OFF (“0”).

Classification	Command	Name	Programming Format	Function/Meaning
Control Commands	**MSEE**	SUBROUTINE CALL	MSEE MPS– ;	Executes the MPS– subroutine.
	TIM	DWELL TIME	TIM T–;	Waits for the period of time specified by T, and then proceeds to the next block.
	IOW	I/O WAIT	IOW MB– = = ***;	Stops execution of the motion program until the conditional expression given in the command is satisfied.
	END	PROGRAM END	END;	Ends the motion program.
	RET	SUBROUTINE RETURN	RET;	Ends the subroutine.
	EOX	ONE SCAN WAIT	EOX;	Separates continuous sequence instructions and forces a wait of one scan before continuing execution.

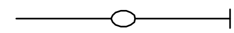
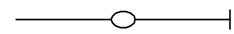
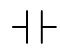
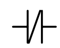
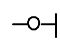
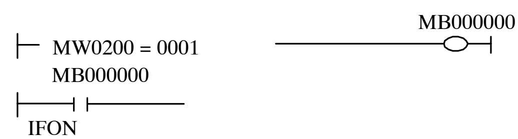
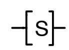
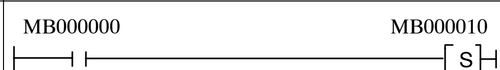
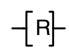
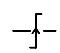
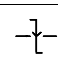
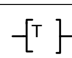
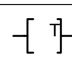
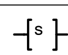
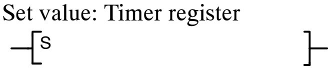
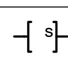
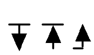
	IF
	ELSE
	IEND
		Branching commands	IF (conditional expression) ;
			(process 1)
			ELSE;
			(process 2)
			IEND;
				Executes process 1 if the conditional expression is satisfied, and executes process 2 if the conditional expression is not satisfied.

	WHILE **WEND**	Repeat commands	WHILE (conditional expression) ; … WEND;	Repeatedly executes WHILE to WEND processing for as long as the conditional expression is satisfied.
	PFORK **JOINTO** **PJOINT**	Parallel execution commands	PFORK label 1, label 2,…; Label 1: Process 1 JOINTO label X Label 2: Process 2 JOINTO label X Label • • Label X: PJOINT;	Executes the blocks designated by the labels in parallel. With a subroutine, a maximum of two labels can be designated. Also, a motion command cannot be used in the block designated by the second label. END and RET cannot be used during parallel execution processing.
	SFORK **JOINTO** **SJOINT**	Selective execution commands	SFORK conditional expression 1? label 1, Conditional expression 2? label 2,…; Label 1: Process 1 JOINTO label X Label 2: Process 2 JOINTO label X Label • • Label X: SJOINT;	Executes process 1 if conditional expression 1 is satisfied, and executes process 2 if the conditional expression 2 is satisfied.

A.2 Ladder Instructions and Standard System Functions

The following table shows a list of the ladder instructions and standard system functions.

Type	Name	Symbol	Abbreviated Instructions	Description
Program Control Instructions	Instructions with [ ]	–	–	–
	CHILD DRAWING CALL	SEE	SEE	Designate the child drawing number or the grandchild drawing number to be called after SEE. SEE H01
	DRAWING END	DEND	END	End of drawing (DWG)
	MOTION PROGRAM CALL	MSEE	MSEE	Designate the motion program number and the MSEE work register address to be called after MSEE. MSEE MPM001 DA00000
	FOR Structure	FOR : : FEND	FOR	Repeats execution statement 1 FOR V = a to b by c V: Can designate any integer register I or J. a, b, c: Can designate an any integer value (b > a > 0, c > 0). FEND: End of FOR instruction.
	WHILE Structure	WHILE : ON/OFF : WEND	WHILE ON OFF	Repeats execution statement 2 WEND: End of WHILE-ON/OFF instruction
	IF Structure –1, –2	IFON/IFOFF : ELSE : IEND	IFON IFOFF ELSE	Conditional execution statement IEND: End of IFON/IFOFF instruction
	FUNCTION CALL	FSTART	FSTART	Calls a function.
	FUNCTION INPUT	FIN	FIN	Function input instruction Stores input data from the designated input register in the function input register.
	FUNCTION OUTPUT	FOUT	FOUT	Function output instruction Stores output data from the function output register in the designated output register.
	COMMENT	“nnnnnnn”	”	A character string enclosed in quotation marks is treated as a comment.
	EXTENSION PROGRAM CALL	XCALL	XCALL	Calls an extension program.

Type	Name	Symbol	Abbreviated Instructions	Description
Direct I/O Instructions	INPUT STRAIGHT	INS	INS	INS MA00100  Executes the input and storage of data with interrupts disabled.
	OUTPUT STRAIGHT	OUTS	OUTS	OUTS MA00100  Executes the setting and output of data with interrupts disabled.
Relay Circuit Instruction	NO CONTACT		][	No limit in a series circuit. Bit designation of any register as a relay number is possible.
	NC CONTACT		]/	No limit in a series circuit. Bit designation of any register as a relay number is possible.
	COIL		@	
	SET COIL		@S	
	RESET COIL		@R	
	RISING PULSE		]P	No limit in a series circuit. Bit designation of any register as a relay number is possible.
	FALLING PULSE		]N	No limit in a series circuit. Bit designation of any register as a relay number is possible.
	10-MS ON-DELAY TIMER		[ON	Set value: Timer register  Set value = any register or constant (setting unit: 10 ms) Timer register = M or D register
	10-MS OFF-DELAY TIMER		[OFF	
	1-S ON-DELAY TIMER		[SON	Set value: Timer register  Set value = any register or constant (setting unit: 10 ms) Timer register = M or D register
	1-S OFF-DELAY TIMER		[SOFF	
	Branching/convergence		,,.	A branching or convergence symbol can be connected to any of the above relay instructions.
Logic Operation Instructions	AND	<	&	Integer designation of any register or constant is possible.
	OR	>	|	Integer designation of any register or constant is possible.
	XOR	⊕	^	Integer designation of any register or constant is possible.

Type	Name	Symbol	Abbreviated Instructions	Description
Numeric Operation Instructions	INTEGER ENTRY	⊢	;	Starts an integer operation. ⊢ MW00280 + 00100 ⇒ MW00220
	REAL NUMBER ENTRY	⊩	;;	Starts a real number operation. ⊩ MW00280 + 00100 ⇒ MW00220
	STORE	⇒	:	Stores the operation result in the designated register.
	ADDITION	+	+	Ordinary numeric addition (with operation error) ⊢ MW00280 +00100 ⇒ MW00220
	SUBTRACTION	–	–	Ordinary numeric subtraction (with operation error) ⊢ MW00280 –00100 ⇒ MW00220
	EXTENDED ADDITION	++	++	Closed numeric addition (without operation error) 0 → 32767 → –32768 → 0
	EXTENDED SUB-TRACTION	– –	– –	Closed numeric subtraction (without operation error) 0 → 32768 → –32767 → 0
	MULTIPLICA-TION	×	*	For integer and long integers, use × and ÷ in combination.
	DIVISION	÷	/	
	MOD	MOD	MOD	Gets the remainder of the division result. ⊢ MW00100 × 0100 ÷ 00121 MOD ⇒ MW00101
	REM	REM	REM	Gets the remainder of the division result. MF00200 REM 1.5 ⇒ MF00202
	INCREMENT	INC	INC	Adds 1 to the designated register. INC MW00100
	DECREMENT	DEC	DEC	Subtracts 1 from the designated register. DEC MW00100
	ADD TIME	TMADD	TMADD	Addition of hours, minutes, and seconds TMADD MW00000, MW00100
	SUBTRACT TIME	TMSUB	TMSUB	Subtraction of hours, minutes, and seconds TMSUB MW00000, MW00100
	SPEND TIME	SPEND	SPEND	Calculates the elapsed time between two times. SPEND MW00000, MW00100

Type	Name	Symbol	Abbreviated Instructions	Description
Numeric Conversion Instructions	SIGN INVERSION	INV	INV	⊦ MW00100 INV If MW00100 = 99, the operation result = –99.
	1'S COMPLEMENT	COM	COM	⊦ MW00100 COM If MW00100 = FFFFH, the operation result = 0000H.
	ABSOLUTE VALUE CONVERSION	ABS	ABS	⊦ MW00100 ABS If MW00100 = –99, the operation result = 99.
	BINARY CONVERSION	BIN	BIN	⊦ MW00100 BIN If MW00100 = 1234H (hexadecimal), the operation result = 1234 (decimal).
	BCD CONVERSION	BCD	BCD	⊦ MW00100 BCD If MW00100 = 1234 (decimal), the operation result = 1234H (hexadecimal).
	PARITY CONVERSION	PARITY	PARITY	Calculates the number of binary bits that are ON. If MW00100 = F0F0H, the operation result = 8.
	ASCII CONVERSION 1	ASCII	ASCII	The designated character string is converted to ASCII code and substituted in the register. MW00200 "ABCDEFG"
	ASCII CONVERSION 2	BINASC	BINASC	Converts 16-bit binary data to 4-digit hexadecimal ASCII code. BINASC MW00100
	ASCII CONVERSION 3	ASCBIN	ASCBIN	Converts the numeric value indicated by a 4-digit hexadecimal ASCII code to 16-bit binary data. ASCBIN MW00100
Numeric Comparison Instructions	<	<	<	⊢ MW00000 < 10000 MB000010 MB000010 IFON
	≦	≦	< =	
	=	=	=	
	≠	≠	< >	
	≧	≧	> =	
	>	>	>	
	RANGE CHECK	RCHK	RCHK	Checks whether or not the value in the A register is in range. ⊦ MW00100 RCHK –1000, 1000

A

Type	Name	Symbol	Abbreviated Instructions	Description
Data Operation Instructions	BIT ROTATION LEFT and BIT ROTATION RIGHT	ROTR ROTL	ROTR ROTL	Example: ROTR Bit-addr Count Width ROTR MB00100A → N = 1 W = 20
	MOVE BITS	MOVB	MOVB	Source Desti. Width MOVB MB00100A → MB00200A W = 20
	MOVE WORD	MOVW	MOVW	Source Desti. Width MOVW MB00100 → MB00200 W = 20
	EXCHANGE	XCHG	XCHG	Source1 Source2 Width XCHG MB00100 → MB00200 W = 20
	SET WORDS	SETW	SETW	Desti. Data Width SETW MW00200 D = 00000 W = 20
	BYTE-TO-WORD EXPANSION	BEXTD	BEXTD	Expands the byte data stored in the word registers into words. BEXTD MW00100 to MW00200 B = 10
	WORD-TO-BYTE COMPRESSION	BPRESS	BPRESS	Collects the lower bytes of the word data stored in the word register area. BPRESS MW00100 to MW00200 B = 10
	BINARY SEARCH	BSRCH	BSRCH	Retrieves the register position that matches the data within the designated register range. BSRCH MW00000 W = 20 D = 100 R = MW00100
	SORT	SORT	SORT	Sorts registers within the designated register range. SORT MW00000 W = 100
	BIT SHIFT LEFT	SHFTL	SHFTL	Shifts the designated bit strings to the left. SHFTL MB00100A N = 1 W = 20
	BIT SHIFT RIGHT	SHFTR	SHFTR	Shifts the designated bit strings to the right. SHFTR MB00100A N = 1 W = 2
	COPY WORD	COPYW	COPYW	Copies the designated register range. COPYW MW00100 → MW00200 W = 20
	BYTE SWAP	BSWAP	BSWAP	The upper and lower bytes of the designated word are swapped. BSWAP MW00100

Type	Name	Symbol	Abbreviated Instructions	Description
Basic Function Instructions	SQUARE ROOT	SQRT	SQRT	Taking the square root of a negative number will result in the square root of the absolute value multiplied by –1. |⊢ MF00100 SQRT
	SINE	SIN	SIN	Input = degrees |⊢MF00100 SIN
	COSINE	COS	COS	Input = degrees |⊢ MF00100 COS
	TANGENT	TAN	TAN	Input = degrees |⊢ MF00100 TAN
	ARC SINE	ASIN	ASIN	|⊢ MF00100 ASIN
	ARC COSINE	ACOS	ACOS	|⊢ MF00100 ACOS
	ARC TANGENT	ATAN	ATAN	|⊢ MF00100 ATAN
	EXPONENT	EXP	EXP	|⊢ MF00100 EXP e MF00100
	NATURAL LOG-ARITHM	LN	LN	|⊢ MF00100 LN $\log_e$ (FM00100)
	COMMON LOG-ARITHM	LOG	LOG	|⊢ MF00100 LOG $\log_{10}$ (FM00100)

A

Type	Name	Symbol	Abbreviated Instructions	Description
DDC Instructions	DEAD ZONE A	DZA	DZA	⊦ MW00100 DZA 00100
	DEAD ZONE B	DZB	DZB	⊦ MW00100 DZB 00100
	UPPER/LOWER LIMIT	LIMIT	LIMIT	⊦ MW00100 LIMIT –00100 00100
	PI CONTROL	PI	PI	⊦ MW00100 PI MA00200
	PD CONTROL	PD	PD	⊦ MW00100 PD MA00200
	PID CONTROL	PID	PID	⊦ MW00100 PID MA00200
	FIRST-ORDER LAG	LAG	LAG	⊦ MW00100 LAG MA00200
	PHASE LEAD/ LAG	LLAG	LLAG	⊦ MW00100 LLAG MA00200
	FUNCTION GENERATOR	FGN	FGN	⊦ MW00100 FGN MA00200
	INVERSE FUNCTION GENERATOR	IFGN	IFGN	⊦ MW00100 IFGN MA00200
	LINEAR ACCELERATOR/ DECELERATOR 1	LAU	LAU	⊦ MW00100 LAU MA00200
	LINEAR ACCELERATOR/ DECELERATOR 2	SLAU	SLAU	⊦ MW00100 SLAU MA00200
	PULSE WIDTH MODULATION	PWM	PWM	⊦ MW00100 PWM MA00200

Type	Name	Symbol	Abbreviated Instructions	Description
Table Data Operation Instructions	TABLE READ	TBLBR	TBLBR	TBLBR TBL1, MA00000, MA00100
	TABLE WRITE	TBLBW	TBLBW	TBLBW TBL1, MA00000, MA00100
	ROW SEARCH	TBLSRL	TBLSRL	TBLSRL TBL1, MA00000, MA00100
	COLUMN SEARCH	TBLSRC	TBLSRC	TBLSRC TBL1, MA00000, MA00100
	TABLE CLEAR	TBLCL	TBLCL	TBLCL TBL1, MA00000
	TABLE BLOCK MOVE	TBLMV	TBLMV	TBLMV TBL1, TBL2, MA00000
	QUEUE TABLE READ	QTBLR	QTBLR	QTBLR TBL1, MA00000, MA00100
	QUEUE TABLE READ AND INCREMENT	QTBLRI	QTBLRI	QTBLRI TBL1, MA00000, MA00100
	QUEUE TABLE WRITE	QTBLW	QTBLW	QTBLW TBL1, MA00000, MA00100
	QUEUE TABLE WRITE AND INCREMENT	QTBLWI	QTBLWI	QTBLWI TBL1, MA00000, MA00100
	QUEUE POINTER CLEAR	QTBLCL	QTBLCL	QTBLCL TBL1

Type	Name	Symbol	Abbreviated Instructions	Description
Standard System Functions	DATA TRACE READ	DTRC-RD	DTRC-RD	Data readout from data trace memory to user memory
	TRACE	TRACE	TRACE	Data trace execution control
	FAILURE TRACE READOUT	FTRC-RD	FTRC-RD	Data readout from failure trace memory to user memory
	SEND MESSAGE	MSG-SND	MSG-SND	Sending a message from a Communications Module
	RECEIVE MESSAGE	MSG-RCV	MSG-RCV	Receiving a message from a Communications Module
	COUNTER	COUNTER	COUNTER	Increments or decrements a counter.
	FIRST-IN FIRST-OUT	FINFOUT	FINFOUT	First-in, first-out
	INVERTER TRACE READ	ITRC-RD	ITRC-RD	Reads inverter trace data to store it in user register.
	INVERTER CONSTANT WRITE	ICNS-WR	ICNS-WR	Writes inverter constants.
	INVERTER CONSTANT READ	ICNS-RD	ICNS-RD	Reads inverter constants to register.

B

Command List

B

The table in Appendix B shows the motion commands and ladder instructions.

B

B.1 Motion Command List

The motion commands are listed in the following table.

Classification	Command	Name	Programming Format	Function/Meaning
Axis Move Commands	**MOV**	POSITIONING	MOV [*axis1*] – [*axis2*] – ···; (Up to 14 axes can be designated.)	Executes positioning at rapid traverse speed for up to 14 axes simultaneously. In programming, replace "–" with the numerical data for each axis.
	MVS	LINEAR INTERPOLATION	MVS [*axis1*] – [*axis2*] – ···F–; (Up to 14 axes can be designated.)	Executes linear travel at interpolation feed speed F for up to 14 axes simultaneously.
	MCW **MCC**	CLOCKWISE CIRCULAR INTERPOLATION COUNTERCLOCKWISE CIRCULAR INTERPOLATION	MCW [*axis1*] – [*axis2*] – R– F–; MCC [*axis1*] – [*axis2*] – U– V– T– F–;	Executes circular interpolation at tangential speed F for two axes simultaneously following radius R (or designated center point coordinates). With the center point coordinate designation, multiple circles can be designated with T–. (T– can also be omitted.)
	MCW **MCC**	CLOCKWISE HELICAL INTERPOLATION COUNTERCLOCKWISE HELICAL INTERPOLATION	MCW [*axis1*] – [*axis2*] –U–V– [*axis3*] –T– F–; MCC [*axis1*] – [*axis2*] –R–[*axis3*] –F–;	Moves three axes simultaneously in a combination of circular interpolation and linear interpolation outside of the circular interpolation plane. Speed F will be the circular interpolation tangential speed. With the center point coordinate designation, the number of turns can be designated with T–. (T– can also be omitted.)
	ZRN	ZERO POINT RETURN	ZRN [*axis1*] – [*axis2*] – ···; (Up to 14 axes can be designated.)	Returns each axis to its zero point.
	SKP	SKIP	SKP [*axis1*]– [*axis2*]– ··· SS–; (Up to 14 axes can be designated.)	If the SKIP signal turns ON during a linear interpolation operation, skips the remaining movement and proceeds to the next block.
	MVT	SET TIME POSITIONING	MVT [*axis1*]– [*axis2*]– ··· T–; (Up to 14 axes can be designated.)	Executes positioning by clamping the feed speed so that travel can be completed at the designated time.
	EXM	EXTERNAL POSITIONING	EXM [*axis1*]– D–;	When an external positioning signal is input while external positioning is being executed, only the travel distance designated by "D–" is positioned with an incremental value, and then the next command is executed.

Classification	Command	Name	Programming Format	Function/Meaning
Basic Control Commands	**ABS**	ABSOLUTE MODE	ABS;	Treats all subsequent coordinate words as absolute values.
	INC	INCREMENTAL MODE	INC;	Treats all subsequent coordinate words as incremental values.
	POS	CURRENT POSITION SET	POS [*axis1*] – [*axis2*] – ⋯;	Changes the current values to the desired coordinate values for up to 14 axes simultaneously. Subsequent move commands use this new coordinate system.
	PLN	COORDINATE PLANE SETTING	PLN [*axis1*] [*axis2*]	Designates the coordinate plane to be used for a command requiring a plane designation command.
	MVM	MOVE ON MACHINE COORDINATE	MVM MOV [*axis1*]– [*axis2*]–; or MVM MVS [*axis1*]– [*axis2*]–;	Goes to the target position on the machine coordinate system. The coordinate system set automatically on completion of the zero point return is called a machine coordinate system. This coordinate system is not affected by the POS command.
	PLD	PROGRAM CURRENT POSITION UPDATE	PLD [*axis1*] – [*axis2*] – ⋯;	Updates the program current position for axes shifted by manual intervention. Up to 14 axes can be designated.
Speed and Acceleration/ Deceleration Commands	**ACC**	ACCELERATION TIME CHANGE	ACC [*axis1*] – [*axis2*] – ⋯;	Sets the acceleration time for linear acceleration/deceleration for up to 14 axes simultaneously.
	SCC	S-CURVE TIME CONSTANT CHANGE	SCC [*axis1*] – [*axis2*] – ⋯;	Sets the time constant for moving average acceleration/deceleration for up to 14 axes simultaneously.
	VEL	SET VELOCITY	VEL [*axis1*] – [*axis2*] – ⋯;	Sets the feed speed for up to 14 axes.
	IAC	INTERPOLATION ACCELERATION TIME CHANGE	IAC T–;	Sets the acceleration time for linear acceleration/deceleration for interpolation travel.
	IDC	INTERPOLATION DECELERATION TIME CHANGE	IDC T–;	Sets the deceleration time for linear acceleration/deceleration for interpolation travel.
	IFP	INTERPOLATION FEED SPEED RATIO SETTING	IFP P–;	Designates the maximum feed % for the speed designation during an interpolation feed.
	FMX	MAXIMUM INTERPOLATION FEED SPEED SETTING	FMX T–;	Sets the maximum speed during an interpolation feed. The interpolation acceleration time is the time from “0” until this speed is reached.

B

Classification	Command	Name	Programming Format	Function/Meaning
High-Level Control Commands	**PFN**	IN-POSITION CHECK	MVS [*axis1*] – [*axis2*] – ··· PFN; or PFN [*axis1*] [*axis2*] ;	Proceeds to the next block after the positioning commanded by the interpolation travel command in the same block or a previous block enters the positioning completion range (parameter setting).
	INP	SECOND IN-POSITION CHECK	INP [*axis1*] – [*axis2*] – ···;	Proceeds to the next block after the positioning subsequently commanded by the interpolation travel command with PFN enters the second positioning completion range.
	SNG	IGNORE SINGLE BLOCK SIGNAL	SNG MVS [*axis1*] 100. [*axis2*] 200. F1000;	A block with this command will be executed continuously, even in single-block operation mode. SNG cannot be designated on its own.
	UFC	USER FUNCTION CALL	UFC *Function_name* *Input_data*, *Input_address*, *Output_data*;	Calls a function created by the user.
Sequence Commands	=	SUBSTITUTE	(Result) = (Arithmetic expression)	Substitutes operation results. Performs calculations from left to right (with no order of priority).
	+	ADD	MW– = MW– + MW–; MW– = MW– + 123456; MW– = 123456 + MW–;	Performs integer and real number addition. Calculates combinations of integers and real numbers as real numbers.
	–	SUBTRACT	MW– = MW– – MW–; MW– = MW– – 123456; MW– = 123456 – MW–;	Performs integer and real number subtraction. Calculates combinations of integers and real numbers as real numbers.
	✻	MULTIPLY	MW– = MW– ✻ MW–; MW– = MW– ✻ 123456; MW– = 123456 ✻ MW–;	Performs integer and real number multiplication. Calculates combinations of integers and real numbers as real numbers.
	/	DIVIDE	MW– = MW–/MW–; MW– = MW–/123456; MW– = 123456/MW–;	Performs integer and real number division. Calculates combinations of integers and real numbers as real numbers.
	MOD	REMAINDER	MW– = MW–/MW–; MW– = MOD;	When programmed in the next block after a division, MOD stores the remainder in the designated register.

<table>
<tr><th>Classification</th><th>Command</th><th>Name</th><th>Programming Format</th><th>Function/Meaning</th></tr>
<tr><td rowspan="13">Sequence Commands</td><td>|</td><td>OR (logical OR)</td><td>MB– = MB– | MB–;
MB– = MB– | 1;
MW– = MW– | MW–;
MW– = MW– | H00FF;</td><td>Performs bit/integer logical OR.</td></tr>
<tr><td>^</td><td>XOR (logical exclusive OR)</td><td>MW– = MW– ^ MW–;
MW– = MW– ^ H00FF;</td><td>Performs integer logical exclusive OR.</td></tr>
<tr><td>&</td><td>AND (logical AND)</td><td>MB– = MB– & MB–;
MB– = MB– & 1;
MW– = MW– & MW–;
MW– = MW– & H00FF;</td><td>Performs bit/integer logical AND.</td></tr>
<tr><td>!</td><td>NOT (logical complement)</td><td>MB– = !MB–;
MB– = !1;
MW– = !MW–;
MW– = !H00FF;</td><td>Performs bit/integer logical complement (inverts bits).</td></tr>
<tr><td>( )</td><td>PARENTHESES</td><td>MW– = MW– & (MW– | MW–);</td><td>The logical arithmetic expression inside parentheses is calculated first.</td></tr>
<tr><td>S{ }</td><td>SET BIT</td><td>S{MB–} = MB– & MB–;</td><td>If the logical operation result is “true,” the designated bit turns ON. The designated bit does not turn OFF, even if the logical operation result is “false.”</td></tr>
<tr><td>R{ }</td><td>RESET BIT</td><td>R{MB–} = MB– & MB–;</td><td>If the logical operation result is “true,” the designated bit turns OFF. The designated bit does not turn ON, even if the logical operation result is “false.”</td></tr>
<tr><td>SIN</td><td>SINE</td><td>SIN (MW–);
SIN (90);</td><td>Obtains the sine of the integer or real number (deg), and returns a real value.</td></tr>
<tr><td>COS</td><td>COSINE</td><td>COS (MW–);
COS (90);</td><td>Obtains the cosine of the integer or real number (deg), and returns a real value.</td></tr>
<tr><td>TAN</td><td>TANGENT</td><td>TAN (MF–);
TAN (45.0);</td><td>Obtains the tangent of the real number (deg), and returns a real value.</td></tr>
<tr><td>ASN</td><td>ARC SINE</td><td>ASN (MF–);
ASN (45.0);</td><td>Obtains the arc sine of the real number (deg), and returns a real value.</td></tr>
<tr><td>ACS</td><td>ARC COSINE</td><td>ACS (MF–);
ACS (90.0);</td><td>Obtains the arc cosine of the real number (deg), and returns a real value.</td></tr>
</table>

Classification	Command	Name	Programming Format	Function/Meaning
Sequence Commands	**ATN**	ARC TANGENT	ATN (MW–); ATN (45);	Obtains the arc tangent of the integer or real number (deg), and returns a real value.
	SQRT	SQUARE ROOT	SQT (MW–); SQT (100);	Obtains the square root of the integer or real number, and returns a real value.
	BIN	BCD-TO-BINARY	BIN (MW–);	Converts BCD data to binary data.
	BCD	BINARY-TO-BCD	BCD (MW–);	Converts binary data to BCD data.
	= =	MATCH	IF MW– = = MW–; WHILE MW– = = MW–;	Used in an IF or WHILE conditional expression. If the left side and right side match, the condition is “true.”
	< >	MISMATCH	IF MW– < > MW–; WHILE MW– < > MW–;	Used in an IF or WHILE conditional expression. If the left side and right side do not match, the condition is “true.”
	>	GREATER THAN	IF MW– > MW–; WHILE MW– > MW–;	Used in an IF or WHILE conditional expression. If the left side is greater than the right side, the condition is “true.”
	<	LESS THAN	IF MW– < MW–; WHILE MW– < MW–;	Used in an IF or WHILE conditional expression. If the left side is less than the right side, the condition is “true.”
	> =	GREATER THAN OR EQUAL TO	IF MW– >= MW–; WHILE MW– >= MW–;	Used in an IF or WHILE conditional expression. If the left side is greater than or equal to the right side, the condition is “true.”
	< =	LESS THAN OR EQUAL TO	IF MW– <= MW–; WHILE MW– <= MW–;	Used in an IF or WHILE conditional expression. If the left side is less than or equal to the right side, the condition is “true.”
	SFR	RIGHT SHIFT	SFR MB– N– W–;	Shifts only the designated number of word variables to the right.
	SFL	LEFT SHIFT	SFL MB– N– W–;	Shifts only the designated number of word variables to the left.
	BLK	BLOCK MOVE	BLK MW– MW– W–;	Moves the block (constant designation) beginning with the designated bit (word) variable.
	CLR	CLEAR	CLR MB– W–;	Sets the number of constants specified in the variable group beginning with the designated bit (word) variable to OFF (“0”).

B

Classification	Command	Name	Programming Format	Function/Meaning
Control Commands	**MSEE**	SUBROUTINE CALL	MSEE MPS– ;	Executes the MPS– subroutine.
	TIM	DWELL TIME	TIM T–;	Waits for the period of time specified by T, and then proceeds to the next block.
	IOW	I/O WAIT	IOW MB– = = ***;	Stops execution of the motion program until the conditional expression given in the command is satisfied.
	END	PROGRAM END	END;	Ends the motion program.
	RET	SUBROUTINE RETURN	RET;	Ends the subroutine.
	EOX	ONE SCAN WAIT	EOX;	Separates continuous sequence instructions and forces a wait of one scan before continuing execution.

	IF
	ELSE
	IEND
		Branching commands	IF (conditional expression) ;
			(process 1)
			ELSE;
			(process 2)
			IEND;
				Executes process 1 if the conditional expression is satisfied, and executes process 2 if the conditional expression is not satisfied.

	WHILE **WEND**	Repeat commands	WHILE (conditional expression) ; … WEND;	Repeatedly executes WHILE to WEND processing for as long as the conditional expression is satisfied.
	PFORK **JOINTO** **PJOINT**	Parallel execution commands	PFORK label 1, label 2,…; Label 1: Process 1 JOINTO label X Label 2: Process 2 JOINTO label X Label • • Label X: PJOINT;	Executes the blocks designated by the labels in parallel. With a subroutine, a maximum of two labels can be designated. Also, a motion command cannot be used in the block designated by the second label. END and RET cannot be used during parallel execution processing.
	SFORK **JOINTO** **SJOINT**	Selective execution commands	SFORK conditional expression 1? label 1, Conditional expression 2? label 2,…; Label 1: Process 1 JOINTO label X Label 2: Process 2 JOINTO label X Label • • Label X: SJOINT;	Executes process 1 if conditional expression 1 is satisfied, and executes process 2 if the conditional expression 2 is satisfied.

B.2 Ladder Instructions and Standard System Functions

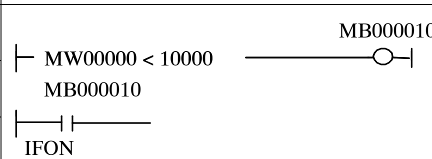
The following table shows a list of the ladder instructions and standard system functions.

Type	Name	Symbol	Abbreviated Instructions	Description
Program Control Instructions	Instructions with [ ]	–	–	–
	CHILD DRAWING CALL	SEE	SEE	Designate the child drawing number or the grandchild drawing number to be referenced after SEE. SEE H01
	DRAWING END	DEND	END	End of drawing (DWG)
	MOTION PROGRAM CALL	MSEE	MSEE	Designate the motion program number and the MSEE work register address to be referenced after MSEE. MSEE MPM001 DA00000
	FOR Structure	FOR : : FEND	FOR	Repeats execution statement 1 FOR V = a to b by c V: Can designate any integer register I or J. a, b, c: Can designate an any integer value (b > a > 0, c > 0). FEND: End of FOR instruction.
	WHILE Structure	WHILE : ON/OFF : WEND	WHILE ON OFF	Repeats execution statement 2 WEND: End of WHILE-ON/OFF instruction
	IF Structure -1, -2	IFON/IFOFF : ELSE : IEND	IFON IFOFF ELSE	Conditional execution statement IEND: End of IFON/IFOFF instruction
	FUNCTION CALL	FSTART	FSTART	Calls a function.
	FUNCTION INPUT	FIN	FIN	Function input instruction Stores input data from the designated input register in the function input register.
	FUNCTION OUTPUT	FOUT	FOUT	Function output instruction Stores output data from the function output register in the designated output register.
	COMMENT	"nnnnnnn"	"	A character string enclosed in quotation marks is treated as a comment.
	EXTENSION PROGRAM CALL	XCALL	XCALL	Calls an extension program.

Type	Name	Symbol	Abbreviated Instructions	Description
Direct I/O Instructions	INPUT STRAIGHT	INS	INS	INS MA00100 Executes the input and storage of data with interrupts disabled.
	OUTPUT STRAIGHT	OUTS	OUTS	OUTS MA00100 Executes the setting and output of data with interrupts disabled.
Relay Circuit Instruction	NO CONTACT		][	No limit in a series circuit. Bit designation of any register as a relay number is possible.
	NC CONTACT		]/	No limit in a series circuit. Bit designation of any register as a relay number is possible.
	COIL		@	MW0200 = 0001 MB000000 MB000000 IFON
	SET COIL	S	@S	MB000000 MB000010 S
	RESET COIL	R	@R	MB000020 MB000010 R
	RISING PULSE		]P	No limit in a series circuit. Bit designation of any register as a relay number is possible.
	FALLING PULSE		]N	No limit in a series circuit. Bit designation of any register as a relay number is possible.
	10-MS ON-DELAY TIMER	T	[ON	Set value: Timer register T Set value = any register or constant (setting unit: 10 ms) Timer register = M or D register
	10-MS OFF-DELAY TIMER	T	[OFF	
	1-S ON-DELAY TIMER	s	[SON	Set value: Timer register S Set value = any register or constant (setting unit: 10 ms) Timer register = M or D register
	1-S OFF-DELAY TIMER	s	[SOFF	
	Branching/convergence		,,.	A branching or convergence symbol can be connected to any of the above relay instructions.
Logic Operation Instructions	AND	<	&	Integer designation of any register or constant is possible.
	OR	>	|	Integer designation of any register or constant is possible.
	XOR	⊕	^	Integer designation of any register or constant is possible.

B

Type	Name	Symbol	Abbreviated Instructions	Description
Numeric Operation Instructions	INTEGER ENTRY	⊢	;	Starts an integer operation. ⊦ MW00280 + 00100 ⇒ MW00220
	REAL NUMBER ENTRY	⊩	;;	Starts a real number operation. ⊩MW00280 + 00100 ⇒ MW00220
	STORE	⇒	:	Stores the operation result in the designated register.
	ADDITION	+	+	Ordinary numeric addition (with operation error) ⊦ MW00280 +00100 ⇒ MW00220
	SUBTRACTION	–	–	Ordinary numeric subtraction (with operation error) ⊦ MW00280 –00100 ⇒ MW00220
	EXTENDED ADDITION	++	++	Closed numeric addition (without operation error) 0 → 32767 → –32768 → 0
	EXTENDED SUBTRACTION	– –	– –	Closed numeric subtraction (without operation error) 0 → 32768 → –32767 → 0
	MULTIPLICATION	×	*	For integer and double integers, use × and ÷ in combination.
	DIVISION	÷	/	
	MOD	MOD	MOD	Gets the remainder of the division result. ⊦ MW00100 × 0100 ÷ 00121 MOD ⇒ MW00101
	REM	REM	REM	Gets the remainder of the division result. MF00200 REM 1.5 ⇒ MF00202
	INCREMENT	INC	INC	Adds 1 to the designated register. INC MW00100
	DECREMENT	DEC	DEC	Subtracts 1 from the designated register. DEC MW00100
	ADD TIME	TMADD	TMADD	Addition of hours, minutes, and seconds TMADD MW00000, MW00100
	SUBTRACT TIME	TMSUB	TMSUB	Subtraction of hours, minutes, and seconds TMSUB MW00000, MW00100
	SPEND TIME	SPEND	SPEND	Calculates the elapsed time between two times. SPEND MW00000, MW00100

Type	Name	Symbol	Abbreviated Instructions	Description
Numeric Conversion Instructions	SIGN INVERSION	INV	INV	⊦ MW00100 INV If MW00100 = 99, the operation result = –99.
	1'S COMPLE-MENT	COM	COM	⊦ MW00100 COM If MW00100 = FFFFH, the operation result = 0000H.
	ABSOLUTE VAL-UE CONVERSION	ABS	ABS	⊦ MW00100 ABS If MW00100 = –99, the operation result = 99.
	BINARY CON-VERSION	BIN	BIN	⊦ MW00100 BIN If MW00100 = 1234H (hexadecimal), the operation result = 1234 (decimal).
	BCD CONVER-SION	BCD	BCD	⊦ MW00100 BCD If MW00100 = 1234 (decimal), the operation result = 1234H (hexadecimal).
	PARITY CON-VERSION	PARITY	PARITY	Calculates the number of binary bits that are ON. If MW00100 = F0F0H, the operation result = 8.
	ASCII CONVER-SION 1	ASCII	ASCII	The designated character string is converted to ASCII code and substituted in the register. MW00200 "ABCDEFG"
	ASCII CONVER-SION 2	BINASC	BINASC	Converts 16-bit binary data to 4-digit hexadecimal ASCII code. BINASC MW00100
	ASCII CONVER-SION 3	ASCBIN	ASCBIN	Converts the numeric value indicated by a 4-digit hexadeci-mal ASCII code to 16-bit binary data. ASCBIN MW00100
Numeric Comparison Instructions	<	<	<	
	≦	≦	< =	
	=	=	=	
	≠	≠	< >	
	≧	≧	> =	
	>	>	>	
	RANGE CHECK	RCHK	RCHK	Checks whether or not the value in the A register is in range. ⊦ MW00100 RCHK –1000, 1000

B

Type	Name	Symbol	Abbreviated Instructions	Description
Data Operation Instructions	BIT ROTATION LEFT and BIT ROTATION RIGHT	ROTR ROTL	ROTR ROTL	Example: ROTR Bit-addr Count Width ROTR MB00100A → N = 1 W = 20
	MOVE BITS	MOVB	MOVB	Source Desti. Width MOVB MB00100A → MB00200A W = 20
	MOVE WORD	MOVW	MOVW	Source Desti. Width MOVW MB00100 → MB00200 W = 20
	EXCHANGE	XCHG	XCHG	Source1 Source2 Width XCHG MB00100 → MB00200 W = 20
	SET WORDS	SETW	SETW	Desti. Data Width SETW MW00200 D = 00000 W = 20
	BYTE-TO-WORD EXPANSION	BEXTD	BEXTD	Expands the byte data stored in the word registers into words. BEXTD MW00100 to MW00200 B = 10
	WORD-TO-BYTE COMPRESSION	BPRESS	BPRESS	Collects the lower bytes of the word data stored in the word register area. BPRESS MW00100 to MW00200 B = 10
	BINARY SEARCH	BSRCH	BSRCH	Retrieves the register position that matches the data within the designated register range. BSRCH MW00000 W = 20 D = 100 R = MW00100
	SORT	SORT	SORT	Sorts registers within the designated register range. SORT MW00000 W = 100
	BIT SHIFT LEFT	SHFTL	SHFTL	Shifts the designated bit strings to the left. SHFTL MB00100A N = 1 W = 20
	BIT SHIFT RIGHT	SHFTR	SHFTR	Shifts the designated bit strings to the right. SHFTR MB00100A N = 1 W = 2
	COPY WORD	COPYW	COPYW	Copies the designated register range. COPYW MW00100 → MW00200 W = 20
	BYTE SWAP	BSWAP	BSWAP	The upper and lower bytes of the designated word are swapped. BSWAP MW00100

Type	Name	Symbol	Abbreviated Instructions	Description
Basic Function Instructions	SQUARE ROOT	SQRT	SQRT	Taking the square root of a negative number will result in the square root of the absolute value multiplied by –1. MF00100 SQRT |⊢
	SINE	SIN	SIN	Input = degrees |⊢MF00100 SIN
	COSINE	COS	COS	Input = degrees |⊢MF00100 COS
	TANGENT	TAN	TAN	Input = degrees |⊢MF00100 TAN
	ARC SINE	ASIN	ASIN	|⊢MF00100 ASIN
	ARC COSINE	ACOS	ACOS	|⊢MF00100 ACOS
	ARC TANGENT	ATAN	ATAN	|⊢ MF00100 ATAN
	EXPONENT	EXP	EXP	|⊢MF00100 EXP e MF00100
	NATURAL LOG-ARITHM	LN	LN	|⊢MF00100 LN $\log_e$ (FM00100)
	COMMON LOG-ARITHM	LOG	LOG	|⊢MF00100 LOG $\log_{10}$ (FM00100)

B

Type	Name	Symbol	Abbreviated Instructions	Description
DDC Instructions	DEAD ZONE A	DZA	DZA	⊦ MW00100 DZA 00100
	DEAD ZONE B	DZB	DZB	⊦ MW00100 DZB 00100
	UPPER/LOWER LIMIT	LIMIT	LIMIT	⊦ MW00100 LIMIT –00100 00100
	PI CONTROL	PI	PI	⊦ MW00100 PI MA00200
	PD CONTROL	PD	PD	⊦ MW00100 PD MA00200
	PID CONTROL	PID	PID	⊦ MW00100 PID MA00200
	FIRST-ORDER LAG	LAG	LAG	⊦ MW00100 LAG MA00200
	PHASE LEAD/ LAG	LLAG	LLAG	⊦ MW00100 LLAG MA00200
	FUNCTION GEN-ERATOR	FGN	FGN	⊦ MW00100 FGN MA00200
	INVERSE FUNC-TION GENERA-TOR	IFGN	IFGN	⊦ MW00100 IFGN MA00200
	LINEAR ACCEL-ERATOR/DECEL-ERATOR 1	LAU	LAU	⊦ MW00100 LAU MA00200
	LINEAR ACCEL-ERATOR/DECEL-ERATOR 2	SLAU	SLAU	⊦ MW00100 SLAU MA00200
	PULSE WIDTH MODULATION	PWM	PWM	⊦ MW00100 PWM MA00200

Type	Name	Symbol	Abbreviated Instructions	Description
Table Data Operation Instructions	TABLE READ	TBLBR	TBLBR	TBLBR TBL1, MA00000, MA00100
	TABLE WRITE	TBLBW	TBLBW	TBLBW TBL1, MA00000, MA00100
	ROW SEARCH	TBLSRL	TBLSRL	TBLSRL TBL1, MA00000, MA00100
	COLUMN SEARCH	TBLSRC	TBLSRC	TBLSRC TBL1, MA00000, MA00100
	TABLE CLEAR	TBLCL	TBLCL	TBLCL TBL1, MA00000
	TABLE BLOCK MOVE	TBLMV	TBLMV	TBLMV TBL1, TBL2, MA00000
	QUEUE TABLE READ	QTBLR	QTBLR	QTBLR TBL1, MA00000, MA00100
	QUEUE TABLE READ AND IN-CREMENT	QTBLRI	QTBLRI	QTBLRI TBL1, MA00000, MA00100
	QUEUE TABLE WRITE	QTBLW	QTBLW	QTBLW TBL1, MA00000, MA00100
	QUEUE TABLE WRITE AND IN-CREMENT	QTBLWI	QTBLWI	QTBLWI TBL1, MA00000, MA00100
	QUEUE POINTER CLEAR	QTBLCL	QTBLCL	QTBLCL TBL1

B

Type	Name	Symbol	Abbreviated Instructions	Description
Standard System Functions	DATA TRACE READ	DTRC-RD	DTRC-RD	Data readout from data trace memory to user memory
	TRACE	TRACE	TRACE	Data trace execution control
	FAILURE TRACE READOUT	FTRC-RD	FTRC-RD	Data readout from failure trace memory to user memory
	SEND MESSAGE	MSG-SND	MSG-SND	Sending a message from a Communications Module
	RECEIVE MES-SAGE	MSG-RCV	MSG-RCV	Receiving a message from a Communications Module
	COUNTER	COUNTER	COUNTER	Increments or decrements a counter.
	FIRST-IN FIRST-OUT	FINFOUT	FINFOUT	First-in, first-out
	INVERTER TRACE READ	ITRC-RD	ITRC-RD	Reads inverter trace data to store it in user register.
	INVERTER CONSTANT WRITE	ICNS-WR	ICNS-WR	Writes inverter constant.
	INVERTER CONSTANT READ	ICNS-RD	ICNS-RD	Reads inverter constant to register.

C

Parameter Lists

C

The table in Appendix C shows the fixed parameters, setting parameters, and monitor parameters.

C

C.1 Fixed Parameter List

Fixed parameters are parameters that cannot be changed from the program.

The following table shows a list of the fixed parameters.

No.	Name	Size	Setting Range	Meaning	Initial Value
1	Axis selection	1W	–	Not used.	0
2	Not used				
3	Encoder selection	1W	0, 1	0: Incremental encoder 1: Absolute encoder	0
4	Not used				
5	Pulse selection	1W	4 to 6	4: A/B pulse 1 X 5: A/B pulse 2 X 6: A/B pulse 4 X	6
6	Not used				
7	Rated speed	1W	1 to 32000	1 r/min	3000
8	Number of FB pulses per motor revolution	1W	Multiple of 4 in a range of 4 to 65535	1 = 1pulse	2048
9 to 15	Not used				
16	Simulation mode selection	1W	0, 1	0: Normal operation mode 1: Simulation mode	0
17	Servo module function selection flag	1W			1
18	Number of digits after the decimal point	1W	0, 1, 2, 3, 4, 5	The minimum reference unit is determined by this parameter and the reference unit selected.	3
19	One machine revolution/reference unit	2W	1 to 2^{31}–1	1 = 1 reference unit	10000
20	Gear ratio (motor side)	1W	1 to 65535		1
21	Gear ratio (load side)	1W	1 to 65535		1
22	Maximum value of infinite-length counter (posmax)	2W	1 to 2^{31}–1	1 = 1 reference unit	360000
23	Absolute encoder maximum rotation	2W	1 to 2^{31}–1	1 = 1 rotation Set according to the encoder specifications.	99999
24	Positive stored stroke limit	2W	–2147483648 to 2147483647	1 = 1 reference unit	2^{31}–1
25	Negative stored stroke limit	2W	–2147483648 to 2147483647	1 = 1 reference unit	-2^{31}
26	Zero point return method	1W	0 to 3	0: DEC signal + phase C 1: ZERO signal 2: ZERO signal + phase C 3: Phase C	0
27	–	1W		Not used.	

C.2 Setting Parameter List

The following table shows a list of the setting parameters.

No.	Name	Register Number	Setting Range	Meaning	Initial Value
1	Operation mode setting	OWxx00	Set for each bit		
2	Servo drive operation reference setting	OWxx01	Set for each bit		
3 to 6	Not used				
7	Zero point offset setting	OLxx06	0 to $\pm 2^{31}-1$	1 = 1 reference unit	0
8 to 10	Not used				
11	Linear acceleration time setting	OWxx0C	0 to 32767	1 = 1 ms	0
12 to 14	Not used				
15	Position loop gain setting	OWxx10	1 to 32767	1 = 0.1/S	400
16	Feed forward gain setting	OWxx11	0 to 200	1 = 1%	0
17	Position reference pulse setting	OLxx12	-2^{31} to $2^{31}-1$	1 = 1 reference unit	0
18	Average rotation setting	OWxx14	0 to 65535	1 = 1 ms	0
19	Speed reference setting	OWxx15	0 to ±32767	1 = 0.01%	0
20 to 25	Not used				
26	Speed loop gain setting	OWxx1D	1 to 32767	1 = 0.1 HZ	400
27	Not used				
28	Motion command code	OWxx20	0 or more	Command number designated for servo amplifier	
29	Motion command control flag	OWxx21	Set for each bit		
30	Rapid traverse speed	OLxx22	0 to $2^{31}-1$	1 = 10^n reference unit/min (n: Number of digits after the decimal point)	0
31	External positioning travel distance	OLxx24	-2^{31} to $2^{31}-1$	1 = 1 reference unit	0
32	Stopping distance	OLxx26	-2^{31} to $2^{31}-1$	1 = 1 reference unit	0
33	STEP travel distance	OLxx28	0 to $2^{31}-1$	1 = 1 reference unit	1000
34	Zero point return final travel distance	OLxx2A	-2^{31} to $2^{31}-1$	1 = 1 reference unit	0
35	Override	OWxx2C	0 to 32767	1 = 0.01%	10000
36	Position control flag	OWxx2D			
37	Work coordinate offset	OLxx2E	-2^{31} to $2^{31}-1$	1 = 1 reference unit	0
38	Preset data of number of POSMAX turns	OLxx30	-2^{31} to $2^{31}-1$	1 = 1 reference unit	0
39	Second in-position width	OWxx32	0 to 65535	1 = 1 reference unit	0

No.	Name	Register Number	Setting Range	Meaning	Initial Value
40	Zero point output width	OWxx33	0 to 65535	1 = 1 reference unit	10
41	Positioning completion check time	OWxx34	0 to 65535	1 = 1 msec	0
42	MECHATROLINK Servo user constant number	OWxx35	b0 to b11: 1 to 4095	User constant Cn	0
			b12 to b15: 1 to 2	Number of words	0
43	MECHATROLINK Servo user constant set value	OLxx36	-2^{31} to 2^{31}-1		0
44	Encoder position at power OFF (low)	OLxx38	-2^{63} to 2^{63}-1	1 = 1 pulse	0
45	Encoder position at power OFF (high)	OLxx3A			
46	Pulse absolute position at power OFF (low)	OLxx3C	-2^{63} to 2^{63}-1	1 = 1 pulse	0
47	Pulse absolute position at power OFF (high)	OLxx3E			

C

C.3 Monitor Parameter List

The following table shows a list of the monitor parameters.

No.	Name	Register Number	Setting Range	Meaning	Initial Value
1	Operation status	IWxx00	Each bit		
2	MECHATROLINK servo status	IWxx01	Each bit		
3	Machine coordinate system calculation position monitor	ILxx02	-2^{31} to $2^{31}-1$	1 = 1 reference unit	CPOS
4	Target position incremental mode	ILxx04	0 to $\pm 2^{16}$	1 = 1 reference unit	
5	Machine coordinate system latch position monitor	ILxx06	-2^{31} to $2^{31}-1$	1 = 1 reference unit	LPOS
6	Machine coordinate system feedback position monitor	ILxx08	-2^{31} to $2^{31}-1$	1 = 1 reference unit	APOS
7 to 10	Not used	IWxx0A to IWxx0E			
11	Range exceeded parameter No.	IWxx0F	1 to 47 101 to 127		
12 13	Not used	IWxx10 to IWxx13			
14	Motion command response code	IWxx14	0 to 65535	Motion command currently being executed	
15	Motion command status	IWxx15	Each bit		
16	Number of digits after the decimal point	IWxx16	0 to 5	Same as fixed parameter 18 (“number of digits after decimal point”)	
17	Position control status	IWxx17			
18	Machine coordinate system reference position	ILxx18	-2^{31} to $2^{31}-1$	1 = 1 reference unit	MPOS
19	Not used	ILxx1A			
20	POSMAX monitor	ILxx1C	1 to $2^{31}-1$	1 = 1 reference unit	
21	Number of POSMAX turns	ILxx1E	-2^{31} to $2^{31}-1$	1 = 1 rotation	
22	Servo driver user monitor information	ILxx20	-2^{31} to $2^{31}-1$		
23	Alarm	ILxx22	Each bit	Servo-related alarm	
24	Servo driver ALARM code	IWxx24		Servopack alarm code	
25	Servo driver I/O monitor	IWxx25	Each bit		
26	Speed reference output value monitor	ILxx26	-2^{31} to $2^{31}-1$	1 = 1 reference unit/sec	

No.	Name	Register Number	Setting Range	Meaning	Initial Value
27	MECHATROLINK servo user constant monitor	ILxx28	-2^{31} to $2^{31}-1$		
28	Not used	IWxx30 to IWxx37			
29	Encoder position at power OFF (low)	ILxx38	-2^{31} to $2^{31}-1$	1 = 1 pulse	For ABS system infinite-length position control
30	Encoder position at power OFF (high)	ILxx3A			
31	Pulse absolute position at power OFF (low)	ILxx3C	-2^{31} to $2^{31}-1$	1 = 1 pulse	For ABS system infinite-length position control
32	Pulse absolute position at power OFF (high)	ILxx3E			

D

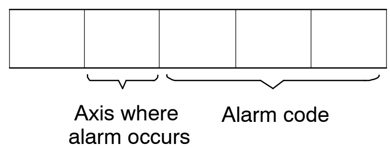
Monitor Parameter Alarm List

The tables in this appendix show the monitor parameter alarms, together with their causes and the action to be taken.

D.1 Motion Program Error

The following shows the display location of the motion alarms stored in the alarm output registers set by group definition.

Set the display mode to HEX (H).



The following table shows the alarm codes related motion program and the corrective actions to be taken.

Alarm Code	Alarm Name	Description	Action
02h	Division error	Data was divided by 0.	Recheck the motion program.
10h	Alarm by circumference specification for radius specification	The number of turns (T) was specified for circular/helical interpolation instruction with radius specification	Change the radius specification to center coordinate specification, and execute circular/helical interpolation instruction. Do not specify the number of turns.
11h	Interpolation feed speed exceeded	Specified interpolation feed speed exceeds the range specified by the FMX instruction.	Change the interpolation feed speed specification of the interpolation instruction.
12h	Interpolation feed speed not specified	The interpolation feed speed has never been specified. (Once the feed speed is specified, it can be omitted in the same motion program.)	Specify an interpolation feed speed in an interpolation instruction.
13h	Out of range after changing acceleration parameters	The acceleration parameter set in the indirect setting mode is out of the set range.	Change the value of the register to execute an indirect setting.
14h	Circular length exceeds LONG_MAX	In the circular/helical interpolation instruction, the specified circular length exceeds the setting range.	Check the circular length specification in the circular/helical interpolation instruction.
15h	No vertical axis specified for circular plane specification	A vertical axis is not specified in the circular /helical interpolation instruction.	Specify a vertical axis by PLN instruction.
16h	No horizontal axis specified for circular plane specification	A horizontal axis is not specified in the circular/helical interpolation instruction.	Specify a horizontal axis by PLN instruction.
17h	The specified number of axes exceeded	The number of axes specified in the circular interpolation instruction (up to 2 axes)/helical interpolation instruction (up to 3 axes) exceeds the set range.	Change the setting for axis specification in the circular/helical interpolation instruction.
18h	The specified number of turns exceeded	The number of turns specified in the circular /helical interpolation instruction exceeds the set range.	Change the setting for the number of turns in the circular/helical interpolation instruction.

Alarm Code	Alarm Name	Description	Action
19h	Radius exceeds LONG_MAX.	The radius specified in the circular/helical interpolation instruction exceeds the set range.	Check the radius specification in the circular /helical interpolation instruction.
1Ah	Center point specification error	The center point specification in the circular /helical interpolation instruction is incorrect.	Specify correctly the center point in the circular/helical interpolation instruction.
1Bh	Executing emergency stop reference	The axis move command was stopped by a request to stop the program.	Turn OFF the request of the motion program control signal to stop the program, and reset the MP930 alarm.
1Ch	Linear interpolation travel distance exceeds LONG_MAX.	The travel distance specified in the linear interpolation instruction exceeds the set range.	Check the travel distance specified in the linear interpolation instruction.
1Dh	FMX not defined	The FMX instruction is not executed in the motion program with an interpolation instruction.	Execute the FMX instruction necessary for a motion program with interpolation instruction.
1Eh	Address T out of range	In the IAC/IDC/FMX instruction, a specification exceeds its set range.	Check the settings in the IAC/IDC/FMX instruction.
1Fh	Address P out of range	In the IFP instruction, a specification exceeds its set range.	Check the settings in the IFP instruction.
80h	Logic axis use prohibited	Axis selection in the fixed parameter is set to “Not used”.	Set the selection of axis to use in the fixed parameter to “Used”.
81h	Value exceeding POSMAX specified at infinite length axis specification	The specified travel distance exceeded the POSMAX set at infinite-length axis specification.	Change the setting of the maximum value of the infinite-length counter (POSMAX) in the fixed parameter. Check the motion program.
82h	The axis move distance exceeded LONG_MAX.	The specified axis travel distance exceeded the set range.	Check the motion program.
84h	Motion commands duplicated	Multiple instructions were executed for one axis.	Check the ladder program.
85h	Motion command responses duplicated	Inappropriate response for a motion command was returned.	Check the ladder program.
87h	Out of the VEL set data range	In the VEL instruction, a specified instruction exceeded the set range.	Check the VEL instruction.
88h	Out of the INP set data range	In the INP instruction, a specified instruction exceeded the set range.	Check the INP instruction.
89h	Out of the ACC/SCC/DCC set data range	In the ACC/SCC/DCC instruction, a specified instruction exceeded the set range.	Check the ACC/SCC/DCC instruction.
8Ah	No time specification in MVT instruction	In the MVT instruction, time (T) specification is 0.	Check the MVT instruction.

D

D.2 Setting Errors in Fixed and Setting Parameters

The following table shows the parameter No.1 operation status (bit 1 and bit 2).

Register Number	Bit	Alarm Name	Cause	Action
IWxx00	**b1: PRMERR**	Setting parameter setting error	A setting that exceeds the setting range has been made in a setting parameter.	Reset the setting parameter.
	b2: FPRMERR	Fixed parameter setting error	A setting that exceeds the setting range has been made in a fixed parameter.	Reset the fixed parameter.

The parameter number in which the setting range error has occurred can be identified using IWxx0 (Range Exceeded Parameter No.).

Parameter Number	Name	Register Number	Description	Remarks
11	Range exceeded parameter No.	IWxx0	• When a setting that exceeds the setting range has been made in a fixed parameter or a setting parameter, the last parameter number in which a setting range error occurred is stored. • Setting parameters: 1 to 47 • Fixed parameters: 101 to 127	

D.3 Monitor Parameter Number 23 Alarms

The following table shows the servo-related alarms for each axis.

Register Number	Bit	Alarm Name	Cause	Action
ILxx22	**b0: SVERR**	Servopack error	• A SERVOPACK alarm was detected. • See IWxx24 for the alarm details.	• Check for Servopack errors. • Contact your Yaskawa representative if the problem recurs even after the Servopack is reset.
	b1: OTF	Forward overtravel	• Forward overtravel of the Servopack was detected (P-OT signal ON).	• Check the overtravel limit switch, and bring it back in the opposite direction after resetting the alarm. • Check the overtravel input signal. • Check the parameters relating to overtravel alarm detection.
	b2: OTR	Reverse overtravel	• Reverse overtravel of the Servopack was detected (N-OT signal ON).	• Check the overtravel limit switch, and bring it back in the opposite direction after resetting the alarm. • Check the overtravel input signal. • Check the parameters relating to overtravel alarm detection.
	b3: SOTF	Stored positive stroke limit	• Movement towards the positive stored stroke limit area was detected.	• After checking the program and operation, reset the alarm and bring the switch back in the opposite direction. • Check the parameters relating to the stored stroke limit.
	b4: SOTR	Stored negative stroke limit	• Movement towards the negative stored stroke limit area was detected.	• After checking the program and operation, reset the alarm and bring the switch back in the opposite direction. • Check the parameters relating to the stored stroke limit.
	b5: SVOFF	Servo power OFF	• A move command was generated, but the servo power was OFF.	• After resetting the alarm, turn ON the servo power.
	b6: TIMEOVER	Positioning time exceeded	• After the command was executed, positioning could not be completed in the time set in OWxx34 (“positioning completion check time”).	• Check the parameters relating to the servo characteristics (each type of gain). • Check the connections between the Servopack and motors.

D

Register Number	Bit	Alarm Name	Cause	Action
ILxx22	**b7: DISTOVER**	Positioning travel distance exceeded	• A move command was executed that exceeded the positioning travel distance limiting value.	• Check the program axis movement.
	b8: FILTYPERR	Filter type change error	• The filter type was changed while the command was being executed.	• Check the filter type change timing, and correct it accordingly.
	b9: FILTIMERR	Filter time constant change error	• The filter time constant was changed while the command was being executed.	• Check the filter time constant change timing, and correct it accordingly.
	b11: SET_NRDY	Zero point not set	• No zero point has been set.	• Set the zero point.
	b12:	–	Not used.	–
	b13:	–	Not used.	–
	b14: WDT_ERR	MECHATROLINK servo synchronous communications error	• A synchronous communications error was detected in the MECHATROLINK servo.	• Check the connection between the MP930 Controller and the Servopack. • Check for Servopack errors. • Reset the alarm. • Turn ON the MP930 again.
	b15: COM_ERR	MECHATROLINK servo communications error	• Two consecutive MECHATROLINK servo communications errors were detected.	• Check the connection between the MP930 Controller and the Servopack. • Check for Servopack errors. • Reset the alarm. • Turn ON the MP930 again.
	b16: SVTIMOUT	MECHATROLINK servo command timeout error	• The MECHATROLINK servo command was not completed within the stipulated time.	• Check the connection between the MP930 Controller and the Servopack. • Check for Servopack errors. • Reset the alarm.
	b17: ABSOVER	ABS encoder rotation exceeded	• The ABS encoder rotation exceeded the range that could be handled by the MP930.	• Set the home position again.

When ILxx22 bit 0 (Servopack Error) is ON, a MECHATROLINK Servopack alarm will be generated. For the alarm details, refer to the *MECHATROLINK Servo Alarm Codes*.

The following table shows a list of the alarm codes and corrective actions.

Alarm Code	Alarm Name	Description	Action
99h	Normal	–	–
94h	User constant setting error	Out of the set range for MECHATROLINK communications setting, or incorrect setting order	Change the setting. Check the setting order.
95h	MECHATROLINK command alarm	A fault occurred in a MECHATROLINK command. (For example, the servo was not turned ON by a Servo ON command.)	Set the motor encoder according to the Servopack encoder.
96h	MECHATROLINK communications error alarm	A synchronous error or a communications error occurred.	Correct the connection. Take noise prevension measures.
00h	Absolute data error	The absolute encoder malfunctioned or the wiring was incorrect.	Setup the absolute encoder. Correct the connection.
02h	User constant destroyed	EEPROM data in Servopack faulty	Replace the Servopack.
10h	Overcurrent	Overcurrent was in the main circuit.	Replace the Servopack.
11h	Grounding	The motor power line was grounded.	Correct motor connection. Replace the motor. Replace the Servopack.
40h	Overvoltage	Overvoltage was applied to the main circuit due to regenerative error, etc.	Check the inertia on the load side by the motor axis formula. Replace the Servopack.
51h	Overspeed	The motor speed exceeded the max. value.	Correct the motor connection. Check the phase-A, phase-B and phase-C pulses at 2CN, and correct them.
71h	Overload (instantaneous)	The motor rotated for several seconds or tens of seconds with the torque largely exceeding its rating.	Reduce the load torque or inertia. Choose a motor with a larger capacity.
72h	Overload (continuous)	Continuous operation was made with the torque exceeding the rating.	Reduce the load torque or inertia. Replace the motor with a larger capacity motor.
80h	Absolute encoder error	The absolute encoder is abnormal.	Check the encoder connection. Reset the number of pulses of the encoder. Separate the encoder wiring from the main circuit. Resets the alarm.
81h	Absolute encoder backup error	All the power supplies to the encoder were shut down, and the position data were cleared.	Setup the absolute encoder. Replace the Servopack.

D

Alarm Code	Alarm Name	Description	Action
82h	Absolute encoder checksum error	The result of [checksum] for the absolute encoder memory is abnormal.	Setup the absolute encoder. Replace the Servopack.
83h	Absolute encoder battery error	The backup battery voltage for the absolute encoder was lowered.	Check the battery connection. Replace the battery. Replace the Servopack.
84h	Absolute encoder data error	The received absolute data are abnormal.	Turn ON the power supply for the MP930 again and the Servopack after turning OFF. Reset the Servopack alarm. Replace the Servopack.
85h	Absolute encoder overspeed	At power ON, the encoder rotated at a high speed.	Turn OFF the power supply and then ON again without running the motor. Replace the Servopack.
B1h	Gate array 1 error	Gate array 1 is faulty.	Replace the Servopack.
B2h	Gate array 2 error	Gate array 2 is faulty.	Replace the Servopack.
B3h	Power supply feedback phase U error	An error occurred in the power supply feedback phase U.	Replace the Servopack.
B4h	Power supply feedback phase V error	An error occurred in the power supply feedback phase V.	Replace the Servopack.
B5h	Watchdog detector error	The watchdog detector fault occurred.	Replace the Servopack.
C1h	Runaway detected	The motor ran away.	Correct the motor connection. Correct the encoder wiring.
C2h	Encoder phase error detected	The encoder phase was shifted.	Correct the encoder connection. Take noise prevension measures. Replace the encoder.
C3h	Encoder phase A phase B disconnection	The signal of encoder phase A phase B is disconnected.	Correct the encoder signal line.
C4h	Encoder phase C disconnection	The signal of encoder phase C is disconnected.	Correct the encoder signal line.
C5h	Incremental encoder initial pulse error	The incremental encoder initial pulse error occurred.	Correct the encoder connection. Take noise prevention measures. Replace the incremental encoder.
D0h	Position deviation overflow	The number of pulses in the deviation counter exceeded the set value.	Correct the encoder signal line. Reconsider the load. Replace the Servopack.
E5h	MECHATROLINK synchronous error	A MECHATROLINK servo synchronous communications error occurred.	Take noise prevention measures.
E6h	MECHATROLINK communications error	A MECHATROLINK communications error occurred twice.	Correct the connector wiring. Take noise prevention measures.
F3h	Power loss	A power loss was detected.	Turn ON the power again.

E

List of System Registers

This appendix outlines the system (S) registers that contain MP930 operation status and error information.

E.1 System (S) Register Allocation

SW00000	System service register
SW00030	System status*
SW00050	System error status*
SW00080	User operation status*
SW00090	System service execution status
SW00100	Interrupt input error status
SW00110	User operation error status (details)*
SW00200	System I/O error status
SW00424	Reserved for system
SW00500	Status for system analysis
SW00530	Reserved for system
SW00600	System operation error status
SW00620	Reserved for system
SW00800 SW01023	Reserved for Optional Modules

* See *9.2.4 System Register Configuration* for details.

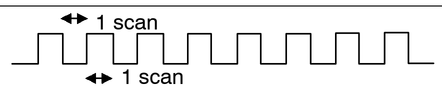
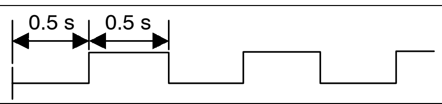
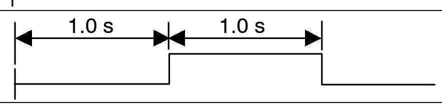
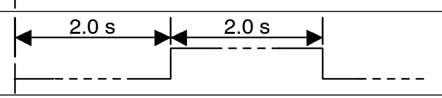
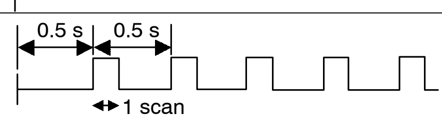
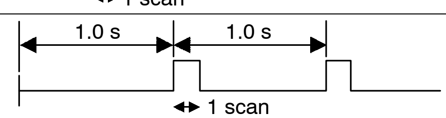
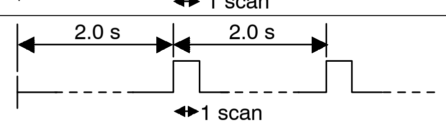
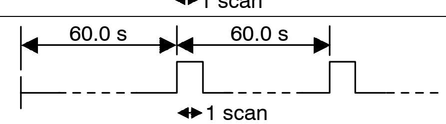
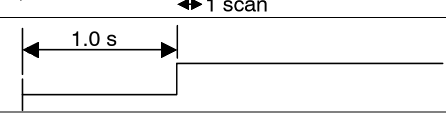
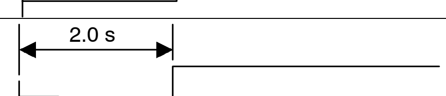
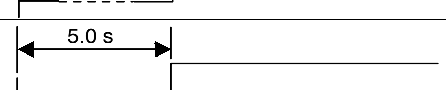
E.2 System Service Registers

■ Registers Common to All Drawings

Name	Register Number	Remarks
First scan (high-speed)	SB000001	ON for only the first scan after high-speed scan is started.
First scan (low-speed)	SB000003	ON for only the first scan after low-speed scan is started.
Always ON	SB000004	–

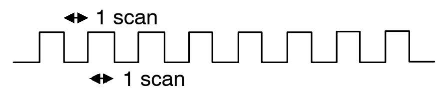
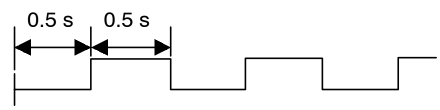
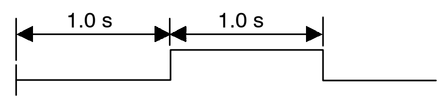
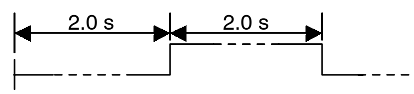
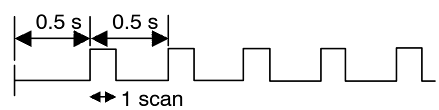
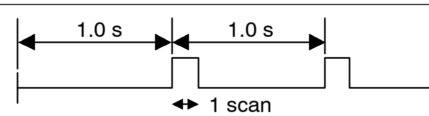
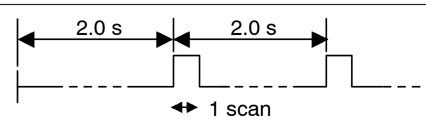
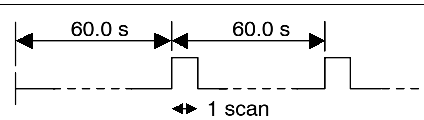
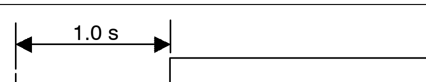
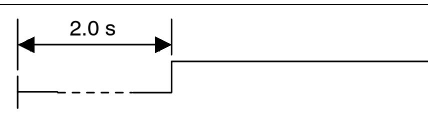
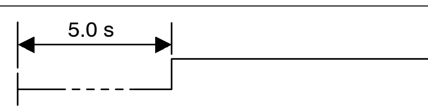
■ Registers Specific to DWG.H

These registers are set when HSCAN starts.

Name	Register Number	Remarks
1-scan flicker relay	SB000010	
0.5-s flicker relay	SB000011	
1.0-s flicker relay	SB000012	
2.0-s flicker relay	SB000013	
0.5-s sampling relay	SB000014	
1.0-s sampling relay	SB000015	
2.0-s sampling relay	SB000016	
60.0-s sampling relay	SB000017	
1.0-s-after-start-of-scan-process relay	SB000018	
2.0-s-after-start-of-scan-process relay	SB000019	
5.0-s-after-start-of-scan-process relay	SB00001A	

Registers Specific to DWG.L

These registers are set when LSCAN starts.

Name	Register Number	Remarks
1-scan flicker relay	SB000030	
0.5-s flicker relay	SB000031	
1.0-s flicker relay	SB000032	
2.0-s flicker relay	SB000033	
0.5-s sampling relay	SB000034	
1.0-s sampling relay	SB000035	
2.0-s sampling relay	SB000036	
60.0-s sampling relay	SB000037	
1.0-s-after-start-of-scan-process relay	SB000038	
2.0-s-after-start-of-scan-process relay	SB000039	
5.0-s-after-start-of-scan-process relay	SB00003A	

Scan Execution Status and Calendar

Name	Register Number	Remarks
High-speed scan set value	SW000004	High-speed scan set value (0.1 ms)
High-speed scan current value	SW000005	High-speed scan current value (0.1 ms)
High-speed scan maximum value	SW000006	High-speed scan maximum value (0.1 ms)
Reserved for system	SW000007 to SW00009	Not used
Low-speed scan set value	SW000010	Low-speed scan set value (0.1 ms)
Low-speed scan current value	SW000011	Low-speed scan current value (0.1 ms)
Low-speed scan maximum value	SW000012	Low-speed scan maximum value (0.1 ms)
Reserved for system	SW000013	Not used
Executing scan current value	SW000014	Current value of the scan being executed (0.1 ms)
Calendar: Year	SW000015	Year 1999: 0099 (BCD) (lower 2 digits only)
Calendar: Month/day	SW000016	December 31: 1231 (BCD)
Calendar: Hours/minutes	SW000017	23 hours 59 minutes: 2359 (BCD)
Calendar: Seconds	SW000018	59 seconds: 59 (BCD)
Calendar: Week	SW000019	0 to 6: Sun., Mon. to Sat.

Other Informations

Name	Register Number	Remarks
System program software number	SW000020	Sxxxx (xxxx is stored as a BCD value)
System number	SW000021 to SW00025	Not used
Remaining program memory capacity	SW000026	In bytes
Total module memory capacity	SW000028	In bytes

Machine Controller MP930

USER'S MANUAL DESIGN AND MAINTENANCE

IRUMA BUSINESS CENTER
480, Kamifujisawa, Iruma, Saitama 358-8555, Japan
Phone 81-4-2962-5696 Fax 81-4-2962-6138

YASKAWA ELECTRIC AMERICA, INC.
2121 Norman Drive South, Waukegan, IL 60085, U.S.A.
Phone 1-847-887-7000 Fax 1-847-887-7370

MOTOMAN INC. HEADQUARTERS
805 Liberty Lane West Carrollton, OH 45449, U.S.A.
Phone 1-937-847-6200 Fax 1-937-847-6277

YASKAWA ELÉTRICO DO BRASIL COMÉRCIO LTD.A.
Avenida Fagundes Filho, 620 Bairro Saude-Sao Päulo-SP, Brazil CEP: 04304-000
Phone 55-11-5071-2552 Fax 55-11-5581-8795

YASKAWA ELECTRIC EUROPE GmbH
Am Kronberger Hang 2, 65824 Schwalbach, Germany
Phone 49-6196-569-300 Fax 49-6196-569-312

Motoman Robotics Europe AB
Box 504 S38525 Torsås, Sweden
Phone 46-486-48800 Fax 46-486-41410

Motoman Robotec GmbH
Kammerfeldstraβe1, 85391 Allershausen, Germany
Phone 49-8166-90-100 Fax 49-8166-90-103

YASKAWA ELECTRIC UK LTD.
1 Hunt Hill Orchardton Woods Cumbernauld, G68 9LF, United Kingdom
Phone 44-1236-735000 Fax 44-1236-458182

YASKAWA ELECTRIC KOREA CORPORATION
Kfpa Bldg #1201, 35-4 Youido-dong, Yeongdungpo-Ku, Seoul 150-010, Korea
Phone 82-2-784-7844 Fax 82-2-784-8495

YASKAWA ELECTRIC (SINGAPORE) PTE. LTD.
151 Lorong Chuan, #04-01, New Tech Park Singapore 556741, Singapore
Phone 65-6282-3003 Fax 65-6289-3003

YASKAWA ELECTRIC (SHANGHAI) CO., LTD.
No.18 Xizang Zhong Road. Room 1805, Harbour Ring Plaza Shanghai 20000, China
Phone 86-21-5385-2200 Fax 86-21-5385-3299

YATEC ENGINEERING CORPORATION
4F., No.49 Wu Kong 6 Rd, Wu-Ku Industrial Park, Taipei, Taiwan
Phone 886-2-2298-3676 Fax 886-2-2298-3677

YASKAWA ELECTRIC (HK) COMPANY LIMITED
Rm. 2909-10, Hong Kong Plaza, 186-191 Connaught Road West, Hong Kong
Phone 852-2803-2385 Fax 852-2547-5773

BEIJING OFFICE
Room No. 301 Office Building of Beijing International Club, 21
Jianguomenwai Avenue, Beijing 100020, China
Phone 86-10-6532-1850 Fax 86-10-6532-1851

TAIPEI OFFICE
9F, 16, Nanking E. Rd., Sec. 3, Taipei, Taiwan
Phone 886-2-2502-5003 Fax 886-2-2505-1280

SHANGHAI YASKAWA-TONGJI M & E CO., LTD.
27 Hui He Road Shanghai China 200437
Phone 86-21-6553-6060 Fax 86-21-5588-1190

BEIJING YASKAWA BEIKE AUTOMATION ENGINEERING CO., LTD.
30 Xue Yuan Road, Haidian, Beijing P.R. China Post Code: 100083
Phone 86-10-6233-2782 Fax 86-10-6232-1536

SHOUGANG MOTOMAN ROBOT CO., LTD.
7, Yongchang-North Street, Beijing Economic Technological Investment & Development Area,
Beijing 100076, P.R. China
Phone 86-10-6788-0551 Fax 86-10-6788-2878

YASKAWA ELECTRIC CORPORATION

YASKAWA

In the event that the end user of this product is to be the military and said product is to be employed in any weapons systems or the manufacture thereof, the export will fall under the relevant regulations as stipulated in the Foreign Exchange and Foreign Trade Regulations. Therefore, be sure to follow all procedures and submit all relevant documentation according to any and all rules, regulations and laws that may apply.

Specifications are subject to change without notice
for ongoing product modifications and improvements.

MANUAL NO. SIEZ-C887-1.1

© Printed in Japan March 2004 98-8 ④
04-1⑤