MITSUBISHI ELECTRIC

Position Board
SSCNETⅢ/H Interface

MR-MC200/MR-MC300 Series Position Board User's Manual (API Library)

-MR-MC210
-MR-MC211
-MR-MC220U3
-MR-MC220U6
-MR-MC240
-MR-MC241
-MR-MC341

● SAFETY PRECAUTIONS ●

(Please read these instructions before using this equipment.)

Before using this product, please read this manual and the relevant manuals introduced in this manual carefully and pay full attention to safety to handle the product correctly.
These precautions apply only to this product.
In this manual, the safety instructions are ranked as "DANGER" and "CAUTION".

⚠DANGER	Indicates that incorrect handling may cause hazardous conditions, resulting in death or severe injury.
⚠CAUTION	Indicates that incorrect handling may cause hazardous conditions, resulting in medium or slight personal injury or physical damage.

Depending on circumstances, procedures indicated by ⚠ CAUTION may also be linked to serious results.
In any case, it is important to follow the directions for usage.

Please save this manual to make it accessible when required and always forward it to the end user.

For Safe Operations

1. Prevention of electric shocks

⚠DANGER

- Never open the front case or terminal covers of the servo amplifier while the power is ON or the unit is running, as this may lead to electric shocks.
- Never run the unit with the front case or terminal cover of the servo amplifier removed. The high voltage terminal and charged sections will be exposed and may lead to electric shocks.
- Never open the front case or terminal cover of the servo amplifier at times other than wiring work or periodic inspections even if the power is OFF. The insides of the position board and servo amplifier are charged and may lead to electric shocks.
- Completely turn off the externally supplied power used in the system before mounting or removing the position board, performing wiring work, or inspections. Failing to do so may lead to electric shocks.
- When performing wiring work or inspections, turn the power OFF, wait at least ten minutes, and then check the voltage with a tester, etc. Failing to do so may lead to electric shocks.
- Be sure to ground the controller incorporating the position board, servo amplifier and servo motor. (Ground resistance : 100 Ω or less) Do not ground commonly with other devices.
- The wiring work and inspections must be done by a qualified technician.
- Wire the units after installing the position board, servo amplifier and servo motor. Failing to do so may lead to electric shocks or damage.
- Never operate the switches with wet hands, as this may lead to electric shocks.
- Do not damage, apply excessive stress, place heavy things on or sandwich the cables, as this may lead to electric shocks.
- Do not touch the position board, servo amplifier or servo motor terminal blocks while the power is ON, as this may lead to electric shocks.
- Do not touch the built-in power supply, built-in grounding or signal wires of the position board and servo amplifier, as this may lead to electric shocks.

2. For fire prevention

⚠CAUTION

- Install the position board, servo amplifier, servo motor and regenerative resistor on incombustible. Installing them directly or close to combustibles will lead to fire.
- If a fault occurs in the position board or servo amplifier, shut the power OFF at the servo amplifier's power source. If a large current continues to flow, fire may occur.
- When using a regenerative resistor, shut the power OFF with an error signal. The regenerative resistor may abnormally overheat due to a fault in the regenerative transistor, etc., and may lead to fire.
- Always take heat measures such as flame proofing for the inside of the control panel where the servo amplifier or regenerative resistor is installed and for the wires used. Failing to do so may lead to fire.
- Do not damage, apply excessive stress, place heavy things on or sandwich the cables, as this may lead to fire.

3. For injury prevention

⚠CAUTION
● Do not apply a voltage other than that specified in this manual and the instruction manual of the product you are using on any terminal. Doing so may lead to destruction or damage. ● Do not mistake the terminal connections, as this may lead to destruction or damage. ● Do not mistake the polarity (+ / -), as this may lead to destruction or damage. ● Do not touch the heat radiating fins of position board or servo amplifier, regenerative resistor and servo motor, etc., while the power is ON and for a short time after the power is turned OFF. In this timing, these parts become very hot and may lead to burns. ● Always turn the power OFF before touching the servo motor shaft or coupled machines, as these parts may lead to injuries. ● Do not go near the machine during test operations or during operations such as teaching. Doing so may lead to injuries.

4. Various precautions

Strictly observe the following precautions.

Mistaken handling of the unit may lead to faults, injuries or electric shocks.

(1) System structure

⚠CAUTION
● Always install a leakage breaker on the controller incorporating the position board and servo amplifier power source. ● If installation of an electromagnetic contactor for power shut off during an error, etc., is specified in the instruction manual for the servo amplifier, etc., always install the electromagnetic contactor. ● Install the emergency stop circuit externally so that the operation can be stopped immediately and the power shut off. ● Use the position board, servo amplifier, servo motor and regenerative resistor with the correct combinations listed in the instruction manual. Other combinations may lead to fire or faults. ● If safety standards (ex., robot safety rules, etc.,) apply to the system using the position board, servo amplifier and servo motor, make sure that the safety standards are satisfied. ● Construct a safety circuit externally of the position board or servo amplifier if the abnormal operation of the position board or servo amplifier differ from the safety directive operation in the system. ● In systems where coasting of the servo motor will be a problem during the forced stop, emergency stop, servo OFF or power supply OFF, use dynamic brakes. ● Make sure that the system considers the coasting amount even when using dynamic brakes. ● In systems where perpendicular shaft dropping may be a problem during the forced stop, emergency stop, servo OFF or power supply OFF, use both dynamic brakes and electromagnetic brakes.

⚠CAUTION

- The dynamic brakes must be used only on errors that cause the forced stop, emergency stop, or servo OFF. These brakes must not be used for normal braking.
- The brakes (electromagnetic brakes) assembled into the servo motor are for holding applications, and must not be used for normal braking.
- The system must have a mechanical allowance so that the machine itself can stop even if the stroke limits switch is passed through at the max. speed.
- Use wires and cables that have a wire diameter, heat resistance and bending resistance compatible with the system.
- Use wires and cables within the length of the range described in the instruction manual.
- The ratings and characteristics of the parts (other than position board, servo amplifier and servo motor) used in a system must be compatible with the position board, servo amplifier and servo motor.
- Install a cover on the shaft so that the rotary parts of the servo motor are not touched during operation.
- There may be some cases where holding by the electromagnetic brakes is not possible due to the life or mechanical structure (when the ball screw and servomotor are connected with a timing belt, etc.). Install a stopping device to ensure safety on the machine side.

(2) Parameter settings and programming

⚠CAUTION

- Set the parameter values to those that are compatible with the position board, servo amplifier, servo motor and regenerative resistor model and the system application. The protective functions may not function if the settings are incorrect.
- The regenerative resistor model and capacity parameters must be set to values that conform to the operation mode and servo amplifier. The protective functions may not function if the settings are incorrect.
- Set the mechanical brake output and dynamic brake output validity parameters to values that are compatible with the system application. The protective functions may not function if the settings are incorrect.
- Set the stroke limit input validity parameter to a value that is compatible with the system application. The protective functions may not function if the setting is incorrect.
- Set the servo motor encoder type (increment, absolute position type, etc.) parameter to a value that is compatible with the system application. The protective functions may not function if the setting is incorrect.
- Set the servo motor capacity and type (standard, low-inertia, flat, etc.) parameter to values that are compatible with the system application. The protective functions may not function if the settings are incorrect.
- Set the servo amplifier capacity and type parameters to values that are compatible with the system application. The protective functions may not function if the settings are incorrect.
- Use the program commands for the program with the conditions specified in the instruction manual.

(3) Transportation and installation

⚠CAUTION

- Transport the product with the correct method according to the mass.
- Use the servo motor suspension bolts only for the transportation of the servo motor. Do not transport the servo motor with machine installed on it.
- Do not stack products past the limit.
- When transporting, installing, and removing the position board, never touch the print board inner part and electronic components. Hold the front panel or edge of the print board.
- When transporting the position board or servo amplifier, never hold the connected wires or cables.
- When transporting the servo motor, never hold the cables, shaft or detector.
- When transporting the position board or servo amplifier, never hold the front case as it may fall off.
- When transporting, installing or removing the position board or servo amplifier, never hold the edges.
- Install the unit according to the instruction manual in a place where the mass can be withstood.
- Do not get on or place heavy objects on the product.
- Always observe the installation direction.
- Mount the position board to a connector or slot that is compatible with standards, and keep the designated clearance between the position board and other boards.
- Keep the designated clearance between the position board or servo amplifier and control panel inner surface or the position board and servo amplifier, position board or servo amplifier and other devices.
- Do not install or operate position board, servo amplifiers or servo motors that are damaged or that have missing parts.
- Do not block the intake/outtake ports of the servo amplifier and servo motor with cooling fan.
- Do not allow conductive matter such as screw or cutting chips or combustible matter such as oil enter the position board, servo amplifier or servo motor.
- The position board, servo amplifier and servo motor are precision machines, so do not drop or apply strong impacts on them.
- Securely fix the position board, servo amplifier and servo motor to the machine according to the instruction manual. If the fixing is insufficient, these may come off during operation.

⚠CAUTION

- Always install the servo motor with reduction gears in the designated direction. Failing to do so may lead to oil leaks.
- Store and use the unit in the following environmental conditions.

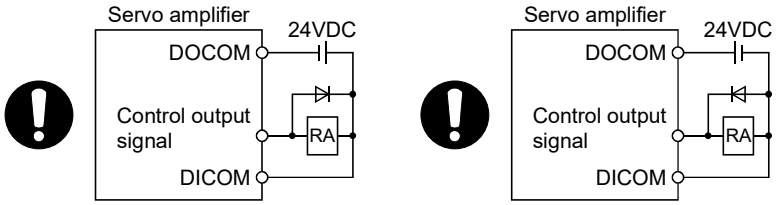
Environment	Conditions	
	Position board/Servo amplifier	Servomotor
Ambient temperature	According to each instruction manual.	0°C to +40°C (With no freezing) (32°F to +104°F)
Ambient humidity	According to each instruction manual.	80% RH or less (With no dew condensation)
Storage temperature	According to each instruction manual.	-20°C to +65°C (-4°F to +149°F)
Atmosphere	Indoors (where not subject to direct sunlight). No corrosive gases, flammable gases, oil mist or dust must exist	
Altitude	According to each instruction manual	
Vibration	According to each instruction manual	

- When coupling with the synchronous encoder or servo motor shaft end, do not apply impact such as by hitting with a hammer. Doing so may lead to detector damage.
- Do not apply a load larger than the tolerable load onto the synchronous encoder and servo motor shaft. Doing so may lead to shaft breakage.
- When not using for a long time, disconnect the power line from the servo amplifier.
- Place the position board and servo amplifier in static electricity preventing vinyl bags and store.
- When storing for a long time, please contact with our sales representative.
  Also, execute a trial operation.
- When fumigants that contain halogen materials such as fluorine, chlorine, bromine, and iodine are used for disinfecting and protecting wooden packaging from insects, they cause malfunction when entering our products.
  Please take necessary precautions to ensure that remaining materials from fumigant do not enter our products, or treat packaging with methods other than fumigation (heat method).
  Additionally, disinfect and protect wood from insects before packing products.

(4) Wiring

⚠CAUTION

- Correctly and securely wire the wires. Reconfirm the connections for mistakes and the terminal screws for tightness after wiring. Failing to do so may lead to run away of the servo motor.
- After wiring, install the protective covers such as the terminal covers to the original positions.
- Do not install a phase advancing capacitor, surge absorber or radio noise filter (option FR-BIF) on the output side of the servo amplifier.
- Correctly connect the output side (terminal U, V, W) and ground. Incorrect connections will lead the servo motor to operate abnormally.
- Do not connect a commercial power supply to the servo motor, as this may lead to trouble.
- Do not mistake the direction of the surge absorbing diode installed on the DC relay for the control signal output of brake signals, etc. Incorrect installation may lead to signals not being output when trouble occurs or the protective functions not functioning.



For the sink output interface　　For the source output interface

- Do not connect or disconnect the connection cables between each unit or the encoder cable while the power is ON.
- Securely tighten the cable connector fixing screws and fixing mechanisms. Insufficient fixing may lead to the cables coming off during operation.
- Do not bundle the power line or cables.

(5) Trial operation and adjustment

⚠CAUTION

- Confirm and adjust the program and each parameter before operation. Unpredictable movements may occur depending on the machine.
- Extreme adjustments and changes may lead to unstable operation, so never make them.
- When using the absolute position system function, on starting up, and when the position board or absolute position motor has been replaced, always perform a home position return.
- Before starting test operation, set the parameter speed limit value to the slowest value, and make sure that operation can be stopped immediately by the forced stop, etc. if a hazardous state occurs.

(6) Usage methods

⚠CAUTION

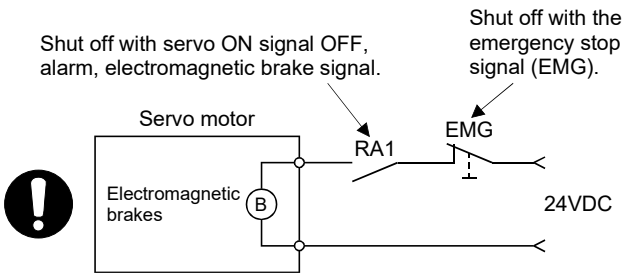
- Immediately turn OFF the power if smoke, abnormal sounds or odors are emitted from the position board, servo amplifier or servo motor.
- Always execute a test operation before starting actual operations after the program or parameters have been changed or after maintenance and inspection.
- Do not attempt to disassemble and repair the units excluding a qualified technician whom our company recognized.
- Do not make any modifications to the unit.
- Keep the effect or electromagnetic obstacles to a minimum by installing a noise filter or by using wire shields, etc. Electromagnetic obstacles may affect the electronic devices used near the position board or servo amplifier.
- When using the CE Mark-compliant equipment, refer to this manual for the position boards and refer to the corresponding EMC guideline information for the servo amplifiers, inverters and other equipment.
- Use the units with the following conditions.

Item	Conditions
Input power	According to each instruction manual.
Input frequency	According to each instruction manual.
Tolerable momentary power failure	According to each instruction manual.

(7) Corrective actions for errors

⚠CAUTION

- If an error occurs in the self diagnosis of the position board or servo amplifier, confirm the check details according to the instruction manual, and restore the operation.
- If a dangerous state is predicted in case of a power failure or product failure, use a servo motor with electromagnetic brakes or install a brake mechanism externally.
- Use a double circuit construction so that the electromagnetic brake operation circuit can be operated by emergency stop signals set externally.



- If an error occurs, remove the cause, secure the safety and then resume operation after alarm release.
- The unit may suddenly resume operation after a power failure is restored, so do not go near the machine. (Design the machine so that personal safety can be ensured even if the machine restarts suddenly.)

(8) Maintenance, inspection and part replacement

⚠CAUTION

- Perform the daily and periodic inspections according to the instruction manual.
- Perform maintenance and inspection after backing up the program and parameters for the position board and servo amplifier.
- Do not place fingers or hands in the clearance when opening or closing any opening.
- Periodically replace consumable parts such as batteries according to the instruction manual.
- Do not touch the lead sections such as ICs or the connector contacts.
- Before touching the position board, always touch grounded metal, etc. to discharge static electricity from human body. Failure to do so may cause the position board to fail or malfunction.
- Do not directly touch the position board's conductive parts and electronic components. Touching them could cause an operation failure or give damage to the position board.
- Do not place the position board or servo amplifier on metal that may cause a power leakage or wood, plastic or vinyl that may cause static electricity buildup.
- Do not perform a megger test (insulation resistance measurement) during inspection.
- When replacing the position board or servo amplifier, always set the new position board settings correctly.
- When the position board or absolute value motor has been replaced, carry out a home position return operation from the user program. Failing to do so may cause position displacement.
- After maintenance and inspections are completed, confirm that the position detection of the absolute position detector function is correct.
- Do not drop or impact the battery installed to the module. Doing so may damage the battery, causing battery liquid to leak in the battery. Do not use the dropped or impacted battery, but dispose of it.
- Do not short circuit, charge, overheat, incinerate or disassemble the batteries.
- The electrolytic capacitor will generate gas during a fault, so do not place your face near the position board or servo amplifier.
- The electrolytic capacitor and fan will deteriorate. Periodically replace these to prevent secondary damage from faults. Replacements can be made by our sales representative.
- Lock the control panel and prevent access to those who are not certified to handle or install electric equipment.
- Do not burn or break a position board and servo amplifier. Doing so may cause a toxic gas.

(9) About processing of waste

When you discard position board, servo amplifier, a battery (primary battery) and other option articles, please follow the law of each country (area).

⚠CAUTION

- This product is not designed or manufactured to be used in equipment or systems in situations that can affect or endanger human life.
- When considering this product for operation in special applications such as machinery or systems used in passenger transportation, medical, aerospace, atomic power, electric power, or submarine repeating applications, please contact your nearest Mitsubishi Electric sales representative.
- Although this product was manufactured under conditions of strict quality control, you are strongly advised to install safety devices to forestall serious accidents when it is used in facilities where a breakdown in the product is likely to cause a serious accident.

(10) General cautions

- All drawings provided in the instruction manual show the state with the covers and safety partitions removed to explain detailed sections. When operating the product, always return the covers and partitions to the designated positions, and operate according to the instruction manual.

REVISIONS

* The manual number is given on the bottom left of the back cover.

Print Date	* Manual Number	Revision
Dec., 2013	IB(NA)-0300225-A	First edition
Dec., 2014	IB(NA)-0300225-B	[Additional model] MR-MC240, MR-MC241 [Additional function] Parameter functions (sscCheckSvPrmChangeNumEx), Continuous operation to torque control data functions (sscSetPressData, sscGetPressData), Operating functions (sscChangeControlMode), High speed monitor functions (sscGetPositionDroopFast), Mark detection functions (sscGetMarkDetectionData, sscGetMarkDetectionCounter, sscClearMarkDetectionData), Interface mode functions (ssclfmGetMaximumBufferNumberEx, ssclfmRenewLatestBufferEx, ssclfmCheckLatestBufferEx, ssclfmGetTransmitBufferEx, ssclfmTrqSetSpeedLimit, ssclfmSetControlMode, ssclfmGetControlMode), PRESS_DATA structure [Additional correction] Conditions for use, bit definition (system status bit, axis command bit, axis status bit), interrupt event factor (axis interrupt), detailed error code
Jul., 2015	IB(NA)-0300225-C	[Additional function] Point table functions (sscSetLatestPointNumber), Interface mode functions (ssclfmGetEventStatusBits), I/O device functions (sscGetInputDeviceBit, sscGetInputDeviceWord, sscSetOutputDeviceBit, sscSetOutputDeviceWord, sscGetOutputDeviceBit, sscGetOutputDeviceWord), Transient transmit functions (sscSendReceiveTransientData), TRANSIENT_CMD structure, TRANSIENT_STS structure [Additional correction] Parameter functions (sscChangeParameter, sscChange2Parameter, sscCheckParameter, sscCheck2Parameter), System functions (sscReconnectSSCNET, sscDisconnectSSCNET), Command/Status functions (sscSetCommandBitSignalEx, sscGetStatusBitSignalEx, sscWaitStatusBitSignalEx), Alarm functions (sscGetAlarm, sscResetAlarm), General monitor functions (sscSetMonitor, sscStopMonitor, sscGetMonitor), Interface mode functions (ssclfmRenewLatestBuffer, ssclfmRenewLatestBufferEx), Interrupt functions (sscResetIntEvent, sscSetIntEvent, sscWaitIntEvent, sscResetIntEventMulti, sscSetIntEventMulti, sscWaitIntEventMulti), PNT_DATA_EX structure, INT_CB_DATA structure, bit definition (system status bit, axis status bit, station command bit, station status bit), Interrupt event factor (system interrupt, station interrupt), detailed error code
Feb., 2017	IB(NA)-0300225-D	[Additional model] MR-MC220U3, MR-MC220U6 [Additional correction] For safe operations, Operating functions (sscLinearStart), Change functions (sscChangeLinearPosition), Interface mode functions (ssclfmGetEventStatusBits), I/O device functions (sscSetOutputDeviceWord), Transient transmit functions (sscSendReceiveTransientData), PNT_DATA_EX structure, OAS_DATA structure, Bit definition (system status bit), detailed error code, Warranty

Print Date	✻ Manual Number	Revision
Sep., 2017	IB(NA)-0300225-E	[Additional correction] Conditions for use
Mar., 2018	IB(NA)-0300225-F	[Additional model] MR-MC341 [Additional function] Information functions (sscGetBoardSerialNumber), System functions (sscReconnectSSCNETEx, sscDisconnectSSCNETEx, sscGetControllingAxis), Operating functions (sscInterpolationStart), Interface mode functions (sscIfmGetEventStatusBitsEx), SLAVE_INFO structure [Additional correction] Manual Page Organization, Conditions for use, Device functions (sscOpen), Parameter functions (sscLoadAllParameterFromFlashROM, sscSaveAllParameterToFlashROM), System functions (sscReboot), Point table functions (sscSetPointDataEx, sscCheckPointDataEx), Operating functions (sscSetDriveMode, sscGetDriveMode), Change functions (sscChangeLinearPosition), Interrupt functions (sscResetIntPassPosition, sscSetIntPassPosition), PNT_DATA_EX structure, OAS_DATA structure, SMP_ERR structure, SMP_DATA structure, INT_CB_DATA structure, Bit definition (System command bit, System status bit, Axis command bit, Axis status bit), Detailed error codes
Sep., 2018	IB(NA)-0300225-G	[Additional correction] Operating functions (sscChangeControlMode), PNT_DATA_EX structure, Differences between MR-MC2□□ and MR-MC3□□
Jun., 2022	IB(NA)-0300225-H	[Additional correction] Conditions for use

Japanese Manual Number IB(NA)-0300224

INTRODUCTION

Thank you for choosing the Mitsubishi Electric position board MR-MC210/MR-MC211/MR-MC220U3 /MR-MC220U6/MR-MC240/MR-MC241/MR-MC341.
Before using the equipment, please read this manual carefully to develop full familiarity with the functions and performance of the position board you have purchased, so as to ensure correct use.

CONTENTS

About Manuals

The following manuals are also related to this product.

When necessary, order them by quoting the details in the tables below.

Related Manuals

(1) Position Board

Manual Name	Manual Number (Model Code)
MR-MC200/MR-MC300 Series Position Board User's Manual (Details) This manual explains specifications of the position board, information on how to establish a system, maintenance/inspection, trouble shooting, functions for the positioning control of the position board, programming, dual port memory and others.	IB-0300223 (1XB968)
MR-MC200/MR-MC300 Series Position Board User's Manual (API Library) This manual explains the library of functions and others that the host controller uses to control the position board.	IB-0300225 (1XB970)

(2) Servo amplifier

Manual Name	Manual Number (Model Code)
SSCNETⅢ/H interface AC Servo MR-J4_B(-RJ)/MR-J4_B4(-RJ)/MR-J4_B1(-RJ) Servo amplifier Instruction Manual This manual explains the I/O signals, parts names, parameters, start-up procedure and others for AC Servo MR-J4_B(-RJ)/MR-J4_B4(-RJ)/MR-J4_B1(-RJ) Servo amplifier.	SH-030106 (1CW805)
SSCNETⅢ/H interface Multi-axis AC Servo MR-J4W2-_B/MR-J4W3-_B Servo amplifier Instruction Manual This manual explains the I/O signals, parts names, parameters, start-up procedure and others for Multi-axis AC Servo MR-J4W2-_B/MR-J4W3-_B Servo amplifier.	SH-030105 (1CW806)
SSCNETⅢ interface MR-J3-□B Servo amplifier Instruction Manual This manual explains the I/O signals, parts names, parameters, start-up procedure and others for MR-J3-□B Servo amplifier.	SH-030051 (1CW202)
SSCNETⅢ Compatible Linear Servo MR-J3-□B-RJ004U□ Instruction Manual This manual explains the I/O signals, parts names, parameters, start-up procedure and others for Linear Servo MR-J3-□B-RJ004U□ Servo amplifier.	SH-030054 (1CW943)
SSCNETⅢ Compatible Fully Closed Loop Control MR-J3-□B-RJ006 Servo amplifier Instruction Manual This manual explains the I/O signals, parts names, parameters, start-up procedure and others for Fully Closed Loop Control MR-J3-□B-RJ006 Servo amplifier.	SH-030056 (1CW304)
SSCNETⅢ interface 2-axis AC Servo AmplifierMR-J3W-0303BN6/MR-J3W-□B Servo amplifier Instruction Manual This manual explains the I/O signals, parts names, parameters, start-up procedure and others for 2-axis AC Servo Amplifier MR-J3W-0303BN6/MR-J3W-□B Servo amplifier.	SH-030073 (1CW604)
SSCNETⅢ Interface Direct Drive Servo MR-J3-□B-RJ080W Servo amplifier Instruction Manual This manual explains the I/O signals, parts names, parameters, start-up procedure and others for Direct Drive Servo MR-J3-□B-RJ080W Servo amplifier.	SH-030079 (1CW601)
SSCNETⅢ interface Drive Safety integrated MR-J3-□B Safety Servo amplifier Instruction Manual This manual explains the I/O signals, parts names, parameters, start-up procedure and others for safety integrated MR-J3-□B Safety Servo amplifier.	SH-030084 (1CW205)

Manual Page Organization

The symbols used in this manual are shown below.

Symbol	Description
MC200	Symbol that indicates correspondence to only MR-MC210/MR-MC211/MR-MC220U3/ MR-MC220U6/MR-MC240/MR-MC241.
MC300	Symbol that indicates correspondence to only MR-MC341.

1. SUMMARY

This position board API library is a collection of API functions for creating applications on the host controller which control our position boards shown below.

・PCI bus compatible position board (MR-MC210/MR-MC211)
・CompactPCI bus compatible position board (MR-MC220U3/MR-MC220U6)
・PCI Express bus compatible position board (MR-MC240/MR-MC241/MR-MC341)

By using these functions, it is possible to open and close communication with the position board, initialize communication with the servo amplifier, change parameters, start operations in each operating mode and monitor.
When using these API functions, up to 4 position boards can be used simultaneously.

In this manual, the following abbreviations are used.

Generic term/Abbreviation	Description
MR-MC2□□	General name for PCI bus compatible position board MR-MC210/MR-MC211/ CompactPCI bus compatible position board MR-MC220U3/MR-MC220U6/ PCI Express bus compatible position board MR-MC240/MR-MC241.
MR-MC3□□	General name for PCI Express bus compatible position board MR-MC341.
Position board	General name for MR-MC2□□ and MR-MC3□□.
Host controller	General name for computer equipped with position board and operates user program.
MR-J4(W□)-□B	Servo amplifier model MR-J4-□B/MR-J4W□-□B.
Servo amplifier	General name for SSCNETⅢ/H compatible servo amplifier.
Utility software	General name for the Position Board Utility2 (MRZJW3-MC2-UTL) which includes test tool for start-up and examination, and the API library for position board.
Test tool	Abbreviation for start-up and examination tool for position board
API library	General name for the library of functions for positioning control that the host controller uses to control the position board.
MR Configurator2	Abbreviation for the Servo set-up software MR Configurator2 version 1.10L or later.
User program	Program created by the user that operates on the host controller.
System program	Internal program that controls the position board.
SSCNETⅢ/H(Note)	High-speed synchronized network between the position board and the servo amplifier.
Board Ver.	System version of position board.
API Ver.	Software version of the API library for position board.
Station No.	Station number on the position board.

Note. SSCNET: Servo System Controller NETwork

(1) Functional limitation depending on software version

Available functions are limited depending on the software version of the position board API library.

Function/Item name	MR-MC2□□		MR-MC3□□		Change details
	API Ver.	Board Ver.	API Ver.	Board Ver.	
Digital input/output	Ver.1.02	A1	—	—	The function is added.
Digital output signal control for the other axes start	Ver.1.02	A1	—	—	The structure member is added.
Pass position interrupt	Ver.1.02	A1	—	—	The function is added.
Interrupt call back	Ver.1.02	–	—	—	The function is added.
Interrupt event notification	Ver.1.02	–	—	—	The function is added.
64-bit operating system compatible	Ver.1.10	–	—	—	The supported operating system is added.
Alarm history function	Ver.1.50	A3	—	—	The function is added.
Interface mode	Ver.1.50	A3	—	—	The function is added.
Servo parameter change number	Ver. 1.60	A0	—	—	The function is added.
Speed-torque control (interface mode only)	Ver.1.60	A4	—	—	The function is added.
Addition of position droop to high speed monitor (interface mode only)	Ver.1.60	A4	—	—	The function is added.
Mark detection function compatible	Ver.1.60	A5	—	—	The function is added.
Continuous operation to torque control compatible (automatic operation in standard mode only)	Ver.1.60	A5	—	—	The function is added.
Windows®8 compatible	Ver.1.60	–	—	—	The supported operating system is added.
Point table loop method compatible	Ver.1.80	A6	—	—	The function is added.
Remote I/O module compatible	Ver.1.80	A8	—	—	Extension of axis numbers.
I/O device compatible	Ver.1.80	A8	—	—	The function is added.
Transient transmit compatible	Ver.1.80	A8	—	—	The function is added.
Changeable interpolation group	Ver.1.90	A9	—	—	The function is added.
Sensing module compatible	Ver.1.90	B1	—	—	–
Windows®10 compatible	Ver.2.00	–	—	—	The supported operating system is added.
Position board MR-MC341 compatible	Not supported	Not supported	—	—	The function is added.
Circular interpolation compatible	Not supported	Not supported	Ver. 1.10	A1	The structure member is added.

—: No restriction by version.

POINT

• The API Ver. can be checked by the file version in the property dialog of the files in the table below.

(Example) For Ver. 1.00

The file version is shown as "1.0.0.0".

Position board	File
MR-MC2□□	• mc2xxstd.dll • mc2xxstd_x64.dll
MR-MC3□□	• mc3xxstd.dll • mc3xxstd_x64.dll

• Refer to "MR-MC200/MR-MC300 Series Position Board User's Manual (Details)" for how to check the Board Ver.

2. CONDITIONS FOR USE

The following conditions when using these API functions apply.

(1) These API functions are assumed to be used with a compiler that runs on the following operating systems.

Position board	Operating system	Compiler
MR-MC2□□	• Windows® 10 (32-bit/64-bit) • Windows® 8.1 (32-bit/64-bit) • Windows® 8 (32-bit/64-bit) • Windows® 7 (32-bit/64-bit) Service Pack1	• Microsoft® Visual C++® 2015/2013/2012/2010/2008/2005 • Microsoft® Visual C#® 2015/2013/2012/2010/2008/2005 • Microsoft® Visual Basic® 2015/2013/2012/2010/2008/2005 • Embarcadero® C++ Builder® 2010/2009/2007
MR-MC3□□	• Windows® 10 (32-bit/64-bit) • Windows® 7 (32-bit/64-bit) Service Pack1	• Microsoft® Visual C++® 2015/2013/2012/2010 • Microsoft® Visual C#® 2015/2013/2012/2010

(2) These API functions use WinDriver produced by Jungo Software Technologies in order to access the PCI bus.

(3) These API functions provide the following library.

Position board	32-bit operating system compatible library	64-bit operating system compatible library
MR-MC2□□	• mc2xxstd.dll • mc2xxstd.lib (COFF format)	• mc2xxstd_x64.dll • mc2xxstd_x64.lib (COFF format)
MR-MC3□□	• mc3xxstd.dll • mc3xxstd.lib (COFF format)	• mc3xxstd_x64.dll • mc3xxstd_x64.lib (COFF format)

(4) These API functions support the following.

Position board	Bus specification	Model
MR-MC2□□	• PCI bus compatible position board	• MR-MC210 • MR-MC211
	• CompactPCI bus compatible position board	• MR-MC220U3 • MR-MC220U6
	• PCI Express bus compatible position board	• MR-MC240 • MR-MC241
MR-MC3□□	• PCI Express bus compatible position board	• MR-MC341

POINT

- Use __stdcall as the calling convention of this API function.
- The following operating system and library combinations are not supported.
  - 32-bit operating system + 64-bit operating system compatible library (mc2xxstd_x64.dll, mc3xxstd_x64.dll)
  - 64-bit operating system + 32-bit operating system compatible library (mc2xxstd.dll, mc3xxstd.dll)
- Be sure to test the user program thoroughly when incorporating with user equipment.
- The MR-MC2□□ library and MR-MC3□□ library cannot be used together on one user program.

MEMO

3. LIST OF API FUNCTIONS

Function Type	Function Name	Function Content	Usable/Unusable		Reference Section
			MR-MC2□□	MR-MC3□□	
Support Functions	sscGetLastError	Gets the detailed error codes.	○	○	4.1.1
	sscGetMountChannel	Gets the mount channel information.	○	○	4.1.2
Device Functions	sscOpen	Opens memory access port.	○	○	4.2.1
	sscClose	Closes memory access port.	○	○	4.2.2
Information Functions	sscGetControlCycle	Gets control cycle status.	○	○	4.3.1
	sscGetBoardVersion	Gets position board system version information.	○	○	4.3.2
	sscGetDriverVersion	Gets the version information for the driver.	○	○	4.3.3
	sscGetOperationCycleMonitor	Gets operation cycle monitor data.	○	○	4.3.4
	sscClearOperationCycleMonitor	Clears operation cycle monitor data.	○	○	4.3.5
	sscGetBoardSerialNumber	Gets position board serial No. information	×	○	4.3.6
Parameter Functions	sscResetAllParameter	Sets the initial values in all parameters before system startup.	○	○	4.4.1
	sscChangeParameter	Writes the parameter.	○	○	4.4.2
	sscChange2Parameter	Writes the parameters (for 2 parameters).	○	○	4.4.3
	sscCheckParameter	Reads the parameter set value.	○	○	4.4.4
	sscCheck2Parameter	Reads the parameter set values (for 2 parameters).	○	○	4.4.5
	sscLoadAllParameterFromFlashROM	Loads all the parameters from a flash ROM before system startup.	○	○	4.4.6
	sscSaveAllParameterToFlashROM	Saves all the parameters into a flash ROM before system startup.	○	○	4.4.7
	sscCheckSvPrmChangeNumEx	Gets parameter change number.	○	○	4.4.8
System Functions	sscReboot	Reboots the system.	○	○	4.5.1
	sscSystemStart	Starts the system.	○	○	4.5.2
	sscGetSystemStatusCode	Gets the system status code.	○	○	4.5.3
	sscReconnectSSCNET	Reconnects the SSCNET communication.	○	○	4.5.4
	sscReconnectSSCNETEx	Reconnects the SSCNET communication.	×	○	4.5.5
	sscDisconnectSSCNET	Disconnects the SSCNET communication.	○	○	4.5.6
	sscDisconnectSSCNETEx	Disconnects the SSCNET communication.	×	○	4.5.7
	sscGetControllingAxis	Gets the controlling axis information and controlling station information.	×	○	4.5.8
Command/ Status Functions	sscSetCommandBitSignalEx	Arbitrarily sets the command bit.	○	○	4.6.1
	sscGetStatusBitSignalEx	Arbitrarily gets the status bit.	○	○	4.6.2
	sscWaitStatusBitSignalEx	Waits until the specified bit turns on/off.	○	○	4.6.3

○: Usable, ×: Unusable

Function Type	Function Name	Function Content	Usable/Unusable		Reference Section
			MR-MC2□□	MR-MC3□□	
Point Table Functions	sscSetPointDataEx	Sets the point data.	○	○	4.7.1
	sscCheckPointDataEx	Gets the point data.	○	○	4.7.2
	sscSetPointOffset	Sets the point number offset.	○	○	4.7.3
	sscCheckPointOffset	Gets the point number offset.	○	○	4.7.4
	sscGetDrivingPointNumber	Gets the operation point number.	○	○	4.7.5
	sscSetLatestPointNumber	Sets the latest command point number.	○	○	4.7.6
Continuous Operation to Torque Control Data Functions	sscSetPressData	Sets the continuous operation to torque control data.	○	○	4.8.1
	sscGetPressData	Gets the continuous operation to torque control data.	○	○	4.8.2
Operating Functions	sscJogStart	Starts JOG operation.	○	○	4.9.1
	sscJogStop	Stops JOG operation.	○	○	4.9.2
	sscJogStopNoWait	Stops JOG operation. (No wait function)	○	○	4.9.3
	sscIncStart	Starts incremental feed.	○	○	4.9.4
	sscAutoStart	Starts automatic operation.	○	○	4.9.5
	sscHomeReturnStart	Starts home position return.	○	○	4.9.6
	sscLinearStart	Starts linear interpolation.	○	○	4.9.7
	sscInterpolationStart	Starts interpolation operation.	×	○	4.9.8
	sscDataSetStart	Starts the home position reset (data set).	○	○	4.9.9
	sscDriveStop	Stops operation.	○	○	4.9.10
	sscDriveStopNoWait	Stops operation. (No wait function)	○	○	4.9.11
	sscDriveRapidStop	Stops operation rapidly.	○	○	4.9.12
	sscDriveRapidStopNoWait	Stops operation rapidly. (No wait function)	○	○	4.9.13
	sscSetDriveMode	Switches the operation mode.	○	○	4.9.14
	sscGetDriveMode	Gets the operation mode status.	○	○	4.9.15
	sscGetDriveFinStatus	Gets the operation completion status.	○	○	4.9.16
	sscChangeControlMode	Switches the control mode of the servo amplifier.	○	○	4.9.17
Change Functions	sscChangeManualPosition	Changes position during incremental feed.	○	○	4.10.1
	sscChangeAutoPosition	Changes position during automatic operation.	○	○	4.10.2
	sscChangeLinearPosition	Changes position during linear interpolation.	○	○	4.10.3
	sscChangeManualSpeed	Changes speed of "JOG operation" or "incremental feed".	○	○	4.10.4
	sscChangeAutoSpeed	Changes speed of "automatic operation" or "linear interpolation"MC200/interpolation operationMC300.	○	○	4.10.5
	sscChangeManualAccTime	Changes acceleration time constant of "JOG operation" or "incremental feed".	○	○	4.10.6

○: Usable, ×: Unusable

Function Type	Function Name	Function Content	Usable/Unusable		Reference Section
			MR-MC2□□	MR-MC3□□	
Change Functions	sscChangeAutoAccTime	Changes acceleration time constant of "automatic operation" or "linear interpolation" MC200 /interpolation operation MC300.	○	○	4.10.7
	sscChangeManualDecTime	Changes deceleration time constant of "JOG operation" or "incremental feed".	○	○	4.10.8
	sscChangeAutoDecTime	Changes deceleration time constant of "automatic operation" or "linear interpolation" MC200 /interpolation operation MC300.	○	○	4.10.9
Alarm Functions	sscGetAlarm	Gets the alarm number.	○	○	4.11.1
	sscResetAlarm	Resets the alarm.	○	○	4.11.2
General Monitor Functions	sscSetMonitor	Starts monitoring.	○	○	4.12.1
	sscStopMonitor	Stops monitoring.	○	○	4.12.2
	sscGetMonitor	Gets monitoring data.	○	○	4.12.3
High Speed Monitor Functions	sscGetCurrentCmdPositionFast	Gets the current command position. (High speed monitor function)	○	○	4.13.1
	sscGetCurrentFbPositionFast	Gets the current feedback position. (High speed monitor function)	○	○	4.13.2
	sscGetIoStatusFast	Gets the external signal status. (High speed monitor function)	○	○	4.13.3
	sscGetCmdSpeedFast	Gets the moving speed. (High speed monitor function)	○	○	4.13.4
	sscGetFbSpeedFast	Gets the feedback moving speed. (High speed monitor function)	○	○	4.13.5
	sscGetCurrentFbFast	Gets the current feedback. (High speed monitor function)	○	○	4.13.6
	sscGetPositionDroopFast	Gets the position droop. (High speed monitor function)	○	○	4.13.7
User Watchdog Functions	sscWdEnable	Enables the user watchdog function.	○	○	4.14.1
	sscWdDisable	Disables the user watchdog function.	○	○	4.14.2
	sscChangeWdCounter	Updates the watchdog counter.	○	○	4.14.3
Other Axes Start Functions	sscSetOtherAxisStartData	Sets the data for starting other axes.	○	○	4.15.1
	sscGetOtherAxisStartData	Gets the data for starting other axes.	○	○	4.15.2
	sscOtherAxisStartAbortOn	Turns the other axes start cancel signal to ON.	○	○	4.15.3
	sscOtherAxisStartAbortOff	Turns the other axes start cancel signal to OFF	○	○	4.15.4
	sscGetOtherAxisStartStatus	Gets the other axes start status.	○	○	4.15.5
Pass Position Interrupt Functions	sscSetIntPassPositionData	Sets the pass position interrupt condition data.	○	○	4.16.1
	sscCheckIntPassPositionData	Gets the pass position interrupt condition data.	○	○	4.16.2
	sscSetStartingPassNumber	Sets the pass position condition start and end numbers.	○	○	4.16.3
	sscGetExecutingPassNumber	Gets the running pass position condition number.	○	○	4.16.4

○: Usable, ×: Unusable

Function Type	Function Name	Function Content	Usable/Unusable		Reference Section
			MR-MC2□□	MR-MC3□□	
Sampling Functions	sscStartSampling	Starts sampling.	○	○	4.17.1
	sscStopSampling	Stops sampling.	○	○	4.17.2
	sscSetSamplingParameter	Writes the sampling parameters.	○	○	4.17.3
	sscGetSamplingParameter	Reads the sampling parameters.	○	○	4.17.4
	sscGetSamplingError	Gets the sampling error.	○	○	4.17.5
	sscGetSamplingStatus	Gets the sampling execution information.	○	○	4.17.6
	sscGetSamplingData	Gets the sampling data.	○	○	4.17.7
Log Functions	sscStartLog	Starts the log.	○	○	4.18.1
	sscStopLog	Stops the log.	○	○	4.18.2
	sscCheckLogStatus	Gets the running status of the log.	○	○	4.18.3
	sscCheckLogEventNum	Gets the number of valid log data events.	○	○	4.18.4
	sscReadLogData	Reads the log data.	○	○	4.18.5
	sscClearLogData	Clears (initializes) the log data.	○	○	4.18.6
	sscGetAlarmHistoryData	Gets alarm history data.	○	○	4.18.7
	sscCheckAlarmHistoryEventNum	Gets the number of valid alarm history data events.	○	○	4.18.8
	sscClearAlarmHistoryData	Clears (initializes) the alarm history data.	○	○	4.18.9
Digital Input/Output Functions	sscGetDigitalInputDataBit	Gets the DI data of the designated digital input in 1-point basis.	○	○	4.19.1
	sscGetDigitalInputDataWord	Gets the DI data of the designated digital input in 16-point basis.	○	○	4.19.2
	sscSetDigitalOutputDataBit	Sets the DO data of the designated digital output in 1-point basis.	○	○	4.19.3
	sscSetDigitalOutputDataWord	Sets the DO data of the designated digital output in 16-point basis.	○	○	4.19.4
	sscGetDigitalOutputDataBit	Gets the DO data of the designated digital output in 1-point basis.	○	○	4.19.5
	sscGetDigitalOutputDataWord	Gets the DO data of the designated digital output in 16-point basis.	○	○	4.19.6
Mark Detection Functions	sscGetMarkDetectionData	Gets mark detection data.	○	○	4.20.1
	sscGetMarkDetectionCounter	Gets mark detection counter.	○	○	4.20.2
	sscClearMarkDetectionData	Clears (initializes) the mark detection data.	○	○	4.20.3
Interface Mode Functions	sscIfmGetReadErrorCount	Gets read error counter.	○	○	4.21.1
	sscIfmSetHomePosition	Performs home position set.	○	○	4.21.2
	sscIfmGetMaximumBufferNumber	Gets maximum buffer number. (Only position control mode)	○	×	4.21.3
	sscIfmGetMaximumBufferNumberEx	Gets maximum buffer number of the designated control mode.	○	○	4.21.4
	sscIfmRenewLatestBuffer	Renews the latest command buffer number and data. (Only position control mode)	○	×	4.21.5
	sscIfmRenewLatestBufferEx	Renews the latest command buffer number and data of the designated control mode.	○	○	4.21.6

○: Usable, ×: Unusable

Function Type	Function Name	Function Content	Usable/Unusable		Reference Section
			MR-MC2□□	MR-MC3□□	
Interface Mode Functions	sscIfmCheckLatestBuffer	Gets the latest command buffer number and data. (Only position control mode)	○	×	4.21.7
	sscIfmCheckLatestBufferEx	Gets the latest command buffer number and data of the designated control mode.	○	○	4.21.8
	sscIfmGetTransmitBuffer	Gets the transmit buffer number and data. (Only position control mode)	○	×	4.21.9
	sscIfmGetTransmitBufferEx	Gets the transmit buffer number and data of the designated control mode.	○	○	4.21.10
	sscIfmTrqSetSpeedLimit	Sets the speed limit value for torque control.	○	○	4.21.11
	sscIfmSetControlMode	Sets the control mode.	○	○	4.21.12
	sscIfmGetControlMode	Gets the control mode.	○	○	4.21.13
	sscIfmGetEventStatusBits	Gets the status bit information of all axes for the designated status signal using the event detect function.	○	×	4.21.14
	sscIfmGetEventStatusBitsEx	Gets the status bit information of all axes for the designated status signal using the event detect function.	×	○	4.21.15
Interrupt Functions	sscIntStart	Starts up the interrupt driver.	○	○	4.22.1
	sscIntEnd	Closes the interrupt driver.	○	○	4.22.2
	sscIntEnable	Enables interrupt output.	○	○	4.22.3
	sscIntDisable	Disables interrupt output.	○	○	4.22.4
	sscRegisterIntCallback	Registers the interrupt callback function.	○	○	4.22.5
	sscUnregisterIntCallback	Unregisters the interrupt callback function.	○	○	4.22.6
	sscResetIntEvent	Sets the interrupt event signal status to nonsignaled.	○	○	4.22.7
	sscSetIntEvent	Sets the interrupt event signal status to signaled.	○	○	4.22.8
	sscWaitIntEvent	Waits until the interrupt event status becomes signaled.	○	○	4.22.9
	sscResetIntEventMulti	Sets the statuses of the multiple interrupt events to nonsignaled.	○	○	4.22.10
	sscSetIntEventMulti	Sets the statuses of the multiple interrupt events to signaled.	○	○	4.22.11
	sscWaitIntEventMulti	Waits until the statuses of the multiple interrupt events become signaled.	○	○	4.22.12
	sscResetIntOasEvent	Sets the status of the other axes start interrupt event to nonsignaled.	○	○	4.22.13
	sscSetIntOasEvent	Sets the status of the other axes start interrupt event to signaled.	○	○	4.22.14
	sscWaitIntOasEvent	Waits until the status of the other axes start interrupt event becomes signaled.	○	○	4.22.15
	sscResetIntPassPosition	Sets the status of the pass position interrupt event to nonsignaled.	○	○	4.22.16

○: Usable, ×: Unusable

Function Type	Function Name	Function Content	Usable/Unusable		Reference Section
			MR-MC2□□	MR-MC3□□	
Interrupt Functions	sscSetIntPassPosition	Sets the status of the pass position interrupt event to signaled.	○	○	4.22.17
	sscWaitIntPassPosition	Waits until the status of the pass position interrupt event becomes signaled.	○	○	4.22.18
	sscResetIntDriveFin	Sets the status of the operation completion interrupt event to nonsignaled.	○	○	4.22.19
	sscSetIntDriveFin	Sets the status of the operation completion interrupt event to signaled.	○	○	4.22.20
	sscWaitIntDriveFin	Waits until the status of the operation completion interrupt event becomes signaled.	○	○	4.22.21
I/O Device Functions	sscGetInputDeviceBit	Gets the designated input bit device in 1-point basis.	○	○	4.23.1
	sscGetInputDeviceWord	Gets the designated input word device in 1-word basis.	○	○	4.23.2
	sscSetOutputDeviceBit	Sets the designated output bit device in 1-point basis.	○	○	4.23.3
	sscSetOutputDeviceWord	Sets the designated output word device in 1-word basis.	○	○	4.23.4
	sscGetOutputDeviceBit	Gets the designated output bit device in 1-point basis.	○	○	4.23.5
	sscGetOutputDeviceWord	Gets the designated output word device in 1-word basis.	○	○	4.23.6
Transient Transmit Functions	sscSendReceiveTransientData	Sends and receives the specified transient transmit data for axes or stations connected to SSCNET.	○	○	4.24.1

○: Usable, ×: Unusable

4. API FUNCTION DETAILS

4.1 Support functions

4.1.1 sscGetLastError

For each function, if an error occurs (return value is "SSC_NG"), the detailed error codes will be got by calling up that function.

```
int  sscGetLastError (
   void
);
```

Argument

None.

Return value

Latest error code

Detailed error code

None.

Point

• When the return value is "SSC_UNOPEN", the detailed error code is not set.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

None.

4.1.2 sscGetMountChannel

The mount channel information will be got.

```
int  sscGetMountChannel (
   int  board_id,
   short  *mountch
);
```

Argument

board_id [in]

Board ID number (0 to 3)

mountch [out]

Pointer to 2-byte variable (2 bytes × 1) which stores the mount channel information

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

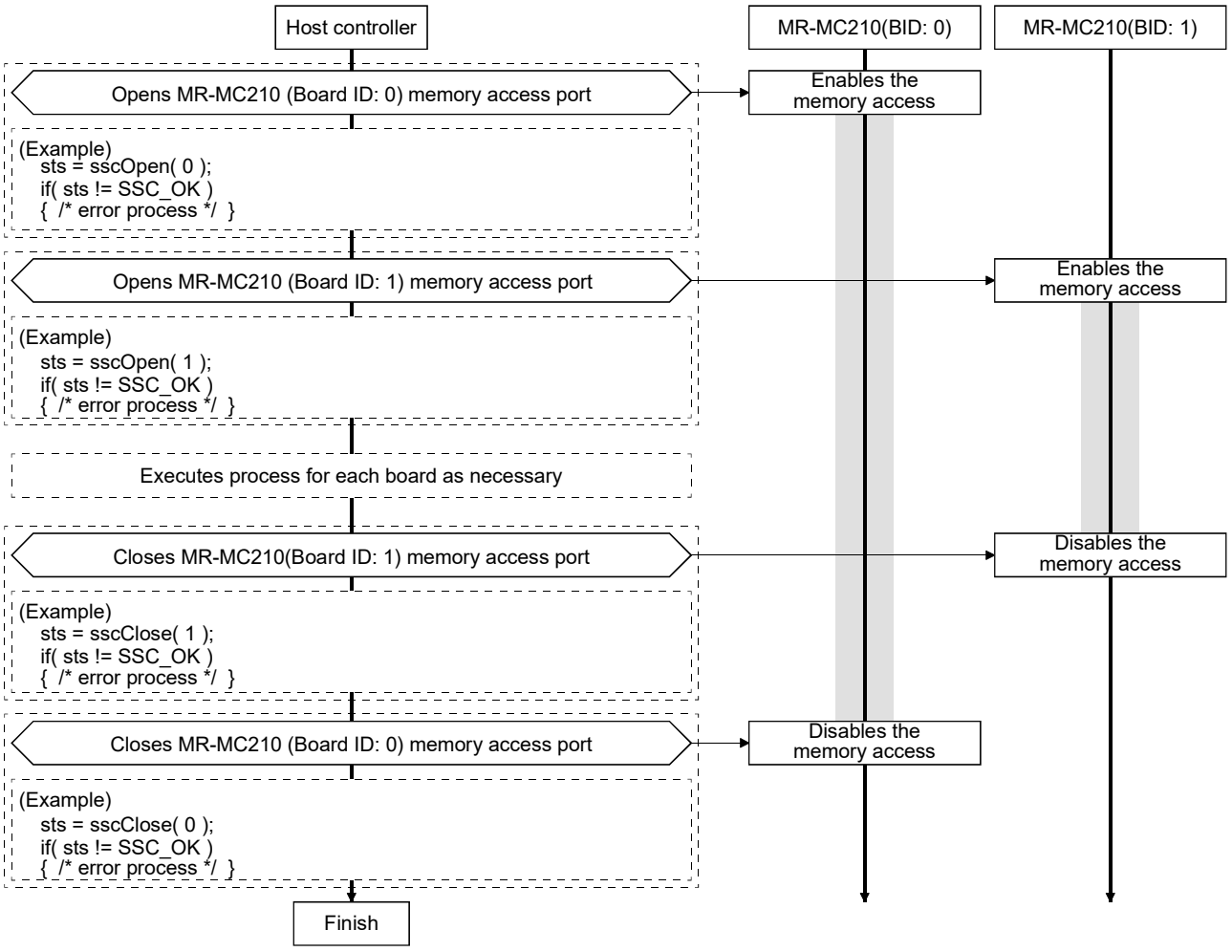
Reference

None.

4.2 Device functions

(1) Processing procedure

An example of device processing procedure for memory access when MR-MC210 (Board ID: 0) and MR-MC210 (Board ID: 1) are connected to the host controller is below.



POINT

- When the same device (Board ID) is not used, the memory access port can be opened at the same time.
- Do not call the sscOpen/sscClose function sequentially.
- By organizing open/close of a memory access port at the beginning and the end of user program process, an error with unopened memory access port when calling API function can be prevented.

  (Example) User program process

  - After turning on the power, open the memory access port in the user program initial process.

    ↓

  - Execute arbitrary process while energizing.

    ↓

  - Before turning off the power, close the memory access port in the user program end process.

4.2.1 sscOpen

The memory access port will be opened.

```
int  sscOpen (
   int  board_id
);
```

Argument

board_id [in]

Board ID number (0 to 3)

Return value

SSC_OK Function succeeded.

SSC_NG Function failed. (To confirm the detailed error code, use the sscGetLastError function.)

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_REOPEN	The sscOpen function is already called.
SSC_FUNC_ERR_DEVICE_DRIVER	An error occurred with a call of the device driver. Confirm that the device driver is installed.
SSC_FUNC_ERR_GET_CHANNEL_NUM	The mount channel information cannot be got. The operating system may not recognize the position board properly. Confirm that the position board is properly mounted using the device manager.
SSC_FUNC_ERR_CREATE_SEMAPHORE	• An error occurred in the CreateSemaphore function (Win32API). Call the GetLastError function of Win32API and confirm the error details. • When connecting the position board from the position board test tool, check that "Confirm the hardware interrupts" is unchecked. MC300
SSC_FUNC_ERR_NOT_FOUND_BOARD	The position board which has the designated board ID could not be found. Confirm the board ID selection (dip switch) of the position board.
SSC_FUNC_ERR_UNSUPPORT_DEVICE_DRIVER	The device driver is not a supported version. Use a API library that combines with the device driver contained in the utility software.

Point

• Call each API function after calling the sscOpen function which corresponds to the used board ID.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscClose

4.2.2 sscClose

The memory access port will be closed.

```
int  sscClose (
   int  board_id
);
```

Argument

board_id [in]

Board ID number (0 to 3)

Return value

SSC_OK Function succeeded.

SSC_NG Function failed. (To confirm the detailed error code, use the sscGetLastError function.)

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_UNOPEN	The sscOpen function has not been called.
SSC_FUNC_ERR_DEVICE_DRIVER	An error occurred with a call of the device driver. Confirm that the device driver is installed.
SSC_FUNC_ERR_DELETE_SEMAPHORE	An error occurred in the CloseHandle function (Win32API). Call the GetLastError function of Win32API and confirm the error details.

Point

• Call sscClose which corresponds to the board ID where the memory access port is already open before finishing the user program.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscOpen

4.3 Information functions

4.3.1 sscGetControlCycle

The control cycle status will be got.

```
int  sscGetControlCycle (
   int  board_id,
   int  channel,
   short *ctrl_cycle
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

ctrl_cycle [out]

Pointer to 2-byte variable (2 bytes × 1) which stores the control cycle status

Value	Description
SSC_CTRL_CYCLE_ERROR	Before system startup
SSC_CTRL_CYCLE_888	0.888ms
SSC_CTRL_CYCLE_444	0.444ms
SSC_CTRL_CYCLE_222	0.222ms

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

None.

4.3.2 sscGetBoardVersion

The system version of the position board will be got.

```
int  sscGetBoardVersion (
   int  board_id,
   int  channel,
   char *version
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

version [out]

Pointer to 16-byte array (1 byte × 16) which stores the system version information

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscGetDriverVersion

4.3.3 sscGetDriverVersion

The driver version information will be got.

```
int  sscGetDriverVersion (
   int  board_id,
   int  channel,
   int  axnum,
   char  *version
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32 MC200 /1 to 64 MC300)

version [out]

Pointer to 16-byte array (1 byte × 16) which stores the drive version information

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscGetBoardVersion

4.3.4 sscGetOperationCycleMonitor

The operation cycle monitor data will be got.

```
int  sscGetOperationCycleMonitor (
   int  board_id,
   int  channel,
   short *now,
   short *max,
   short *over,
   char *status
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

now [out]

Pointer to 2-byte variable which stores the operation cycle current time [μs]

max [out]

Pointer to 2-byte variable which stores the operation cycle maximum time [μs]

over [out]

Pointer to 2-byte variable which stores the operation cycle over time

status [out]

Pointer to 1-byte variable (1 byte × 1) which stores the operation cycle status

The got data is set in the logical sum of each value.

Value	Description
SSC_BIT_OCME	Operation cycle alarm
SSC_BIT_OCMW	Operation cycle warning

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscClearOperationCycleMonitor

4.3.5 sscClearOperationCycleMonitor

The operation cycle monitor data will be cleared (initialized).

```
int sscClearOperationCycleMonitor (
  int board_id,
  int channel
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_RUNNING_CHANNEL	The system is in the status of before system startup. Start the system with the sscSystemStart function.
SSC_FUNC_ERR_TIMEOUT_0□ □ = 1 to 2: Timeout location	During the confirmation of response after executing the command to the position board, the timeout time (1 second) has elapsed.
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	A system error (E□□□h) occurred. Get the system status code with the sscGetSystemStatusCode function and remove the cause.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscGetOperationCycleMonitor

4.3.6 sscGetBoardSerialNumber MC300

Gets the position board serial number.

```
int  sscGetBoardSerialNumber (
   int  board_id,
   char  *serialnum
);
```

Argument

board_id [in]

Board ID number (0 to 3)

serialnum [out]

Pointer to 16-byte array (1 byte × 16) which stores the serial number.

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.

Point

• There is no NULL code at the end of the serial number.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

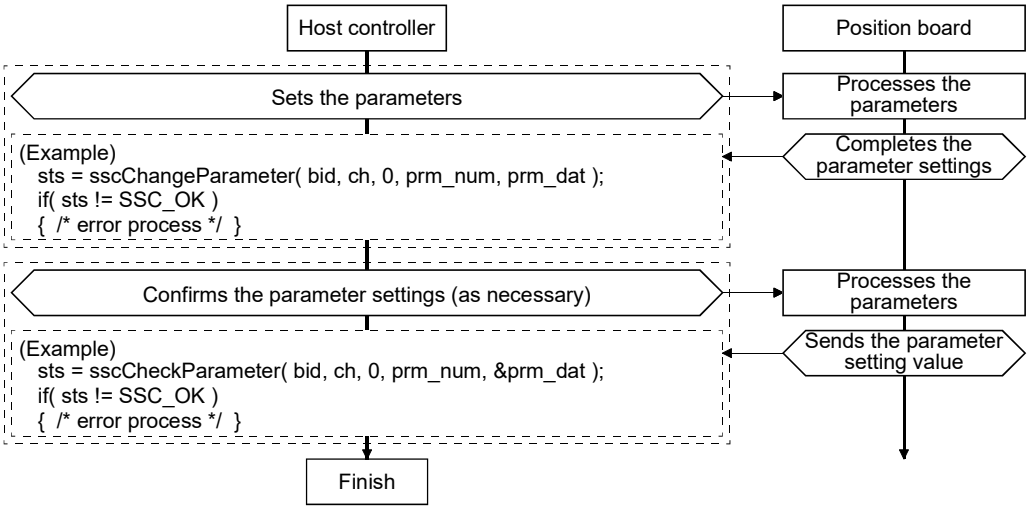
sscGetBoardVersion

4.4 Parameter functions

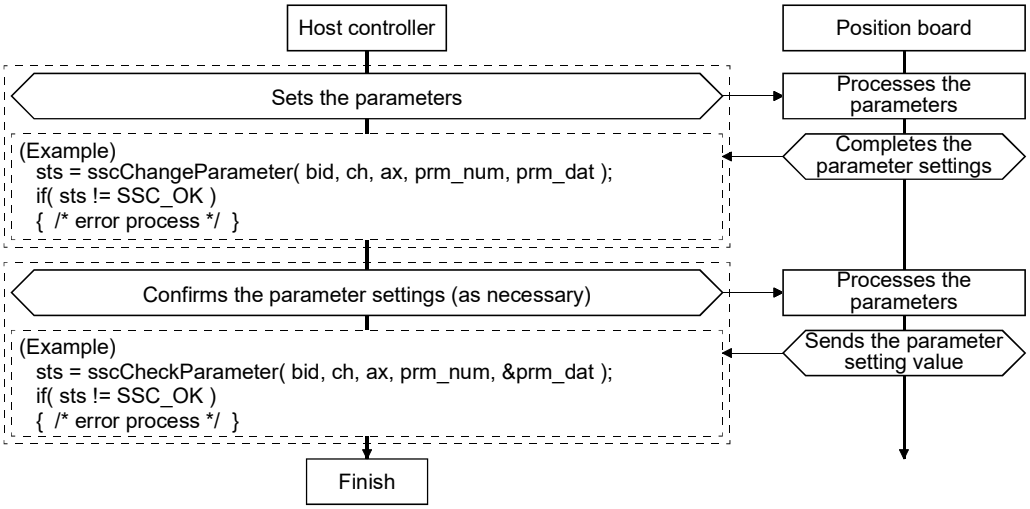
(1) Processing procedure

POINT
• Parameters different from initial values should be changed after all initializing with sscResetAllParameter function.

(a) Example of parameter processing procedure for setting the system parameters



(b) Example of parameter processing procedure for setting the control/servo parameters



4.4.1 sscResetAllParameter

All parameters will be set to the initial values before system startup (system preparation completion).

```
int sscResetAllParameter (
  int board_id,
  int channel,
  int timeout
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

timeout [in]

Timeout time[ms] (0 to 65535)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_UNREADY_CHANNEL	The system is in the status other than system preparation completion. Reboot the system with the sscReboot function.
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	A system error (E□□□h) occurred. Get the system status code with the sscGetSystemStatusCode function and remove the cause.
SSC_FUNC_ERR_TIMEOUT_0□ □ = 1 to 2: Timeout location	During the confirmation of response after executing the command to the position board, the designated timeout time has elapsed.

Point

• When a value 2 seconds (2000ms) or less is designated as the timeout time, the timeout will be 2 seconds (2000ms).

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

None.

4.4.2 sscChangeParameter

Each of the parameters will be written.

```
int  sscChangeParameter (
   int  board_id,
   int  channel,
   int  axnum,
   short prmnum,
   short prmdata
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (-4 to 32 MC200 /-16 to 64 MC300)

0: System parameter

1 to 64: Axis parameter

-16 to -1: Station parameter (-1: Station 1, -2: Station 2, -3: Station 3, -4: Station 4 • • • • • -15: Station 15, -16: Station 16)

prmnum [in]

Parameter write number

prmdata [in]

Parameter write data

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	A system error (E □ □ □ h) occurred. Get the system status code with the sscGetSystemStatusCode function and remove the cause.
SSC_FUNC_ERR_TIMEOUT_0□ □ = 1 to 2: Timeout location	During the confirmation of response after executing the command to the position board, the timeout time (1 second) has elapsed.
SSC_FUNC_ERR_STS_BIT_PWEN1	A parameter write number error occurred. • A value outside the range is set in the parameter write number 1. • The axis number and the parameter write number do not correspond. (Example: When "System parameter" is set to the axis number and "Axis parameter" is set to the parameter write number, etc.)
SSC_FUNC_ERR_STS_BIT_PWED1	A value outside the range is set in the parameter write data 1.
SSC_FUNC_ERR_MISMATCH_PARAM_WRITE_NUM1	The command and the status of the parameter write number 1 do not correspond.
SSC_FUNC_ERR_MISMATCH_PARAM_WRITE_DATA1	The command and the status of the parameter write data 1 do not correspond.

Point

- This function can write the parameters either before or while the system is running.
- It takes about 2 to 10 control cycles to write parameters because of the time it takes to confirm the response of the position board.
- When a number of parameters are written to multiple axes sequentially, the writing time can be shortened by calling this function in each thread of "system" or "axis".

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.80	A8	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscChange2Parameter, sscCheckParameter, sscCheck2Parameter

4.4.3 sscChange2Parameter

Two of the parameters will be written.

```
int  sscChange2Parameter (
   int  board_id,
   int  channel,
   int  axnum,
   short *prmnum,
   short *prmdata,
   char *status
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (-4 to 32 MC200 /-16 to 64 MC300)

0: System parameter

1 to 64: Axis parameter

-16 to -1: Station parameter (-1: Station 1, -2: Station 2, -3: Station 3, -4: Station 4 • • • • • -15: Station 15, -16: Station 16)

prmnum [in]

Pointer to 4-byte array (2 bytes × 2) which stores the parameter write numbers (for 2 numbers)

prmdata [in]

Pointer to 4-byte array (2 bytes × 2) which stores the parameter write data (for 2 data)

status [out]

Pointer to 2-byte array (1 byte × 2) which stores the parameter write statuses (for 2 statuses)

The got data is set in the logical sum of each value.

Value	Description
SSC_BIT_PWFIN	Parameter write complete
SSC_BIT_PWEN	Parameter number error
SSC_BIT_PWED	Outside range of parameter data

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	A system error (E □□□ h) occurred. Get the system status code with the sscGetSystemStatusCode function and remove the cause.
SSC_FUNC_ERR_TIMEOUT_0□ □ = 1 to 2: Timeout location	During the confirmation of response after executing the command to the position board, the timeout time (1 second) has elapsed.
SSC_FUNC_ERR_STS_BIT_PWEN□ □ = 1 to 2: Array number of the parameter write numbers (for 2 numbers)	A parameter write number error occurred. • A value outside the range is set in the parameter write number. • The axis number and the parameter write number do not correspond. (Example: When "System parameter" is set to the axis number and "Axis parameter" is set to the parameter write number, etc.)
SSC_FUNC_ERR_STS_BIT_PWED□ □ = 1 to 2: Array number of the parameter write data (for 2 data)	A value outside the range is set in the parameter write data.
SSC_FUNC_ERR_MISMATCH_PARAM_WRITE_NUM□ □ = 1 to 2: Array number of the parameter write numbers (for 2 numbers)	The command and the status of the parameter write number do not correspond.
SSC_FUNC_ERR_MISMATCH_PARAM_WRITE_DATA□ □ = 1 to 2: Array number of the parameter write data (for 2 data)	The command and the status of the parameter write data do not correspond.

Point

- This function can write the parameters either before or while the system is running.
- To change only 1 parameter, set 0 in the other parameters.
- It takes about 2 to 10 control cycles to write two parameters because of the time it takes to confirm the response of the position board.
- When a number of parameters are written to multiple axes sequentially, the writing time can be shortened by calling this function in each thread of "system" or "axis".

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.80	A8	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscChangeParameter, sscCheckParameter, sscCheck2Parameter

4.4.4 sscCheckParameter

The set value of the designated parameter will be read.

```
int  sscCheckParameter (
   int  board_id,
   int  channel,
   int  axnum,
   short prmnum,
   short *prmdata
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (-4 to 32 MC200 /-16 to 64 MC300)

0: System parameter

1 to 64: Axis parameter

-16 to -1: Station parameter (-1: Station 1, -2: Station 2, -3: Station 3, -4: Station 4 • • • • • -15: Station 15, -16: Station 16)

prmnum [in]

Parameter read number

prmdata [out]

Pointer to 2-byte variable (2 bytes × 1) which stores the parameter read data

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_TIMEOUT_0□ □ = 1 to 2: Timeout location	During the confirmation of response after executing the command to the position board, the timeout time (1 second) has elapsed.
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	A system error (E □□□ h) occurred. Get the system status code with the sscGetSystemStatusCode function and remove the cause.
SSC_FUNC_ERR_STS_BIT_PREN1	A parameter read error occurred. • A value outside the range is set in the parameter read number. • The axis number and the parameter read number do not correspond. (Example: When "System parameter" is set to the axis number and "Axis parameter" is set to the parameter read number, etc.)
SSC_FUNC_ERR_MISMATCH_PARAM_READ_NUM1	The command and the status of the parameter read number 1 do not correspond.

Point

- This function can read the parameters either before or while the system is running.
- It takes about 2 to 10 control cycles to read the set value of the designated parameter because of the time it takes to confirm the response of the position board.
- When a number of multiple parameters are read from multiple axes continuously, the reading time can be shortened by calling this function in each thread of "system" or "axis".

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.80	A8	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscCheck2Parameter, sscChangeParameter, sscChange2Parameter

4.4.5 sscCheck2Parameter

Two set values for the designated parameters will be read.

```
int  sscCheck2Parameter (
   int  board_id,
   int  channel,
   int  axnum,
   short *prmnum,
   short *prmdata,
   char *status
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (-4 to 32 MC200 /-16 to 64 MC300)

0: System parameter

1 to 64: Axis parameter

-16 to -1: Station parameter (-1: Station 1, -2: Station 2, -3: Station 3, -4: Station 4 • • • • • -15: Station 15, -16: Station 16)

prmnum [in]

Pointer to 4-byte array (2 bytes × 2) which stores the parameter read numbers (for 2 numbers)

prmdata [out]

Pointer to 4-byte array (2 bytes × 2) which stores the parameter read data (for 2 data)

status [out]

Pointer to 2-byte array (1 byte × 2) which stores the parameter got statuses (for 2 statuses)

The got data is set in the logical sum of each value.

Value	Description
SSC_BIT_PRFIN	Parameter read complete
SSC_BIT_PREN	Parameter number error

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0☐ ☐ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_TIMEOUT_0☐ ☐ = 1 to 2: Timeout location	During the confirmation of response after executing the command to the position board, the timeout time (1 second) has elapsed.
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	A system error (E☐☐☐h) occurred. Get the system status code with the sscGetSystemStatusCode function and remove the cause.
SSC_FUNC_ERR_STS_BIT_PREN☐ ☐ = 1 to 2: Array number of the parameter read numbers (for 2 numbers)	A parameter read error occurred. • A value outside the range is set in the parameter read number. • The axis number and the parameter read number do not correspond. (Example: When "System parameter" is set to the axis number and "Axis parameter" is set to the parameter read number, etc.)
SSC_FUNC_ERR_MISMATCH_PARAM_READ_NUM☐ ☐ = 1 to 2: Array number of the parameter read numbers (for 2 numbers)	The command and the status of the parameter read number do not correspond.

Point

- This function can read the parameters either before or while the system is running.
- To read only 1 parameter, set 0 in the other parameters.
- It takes about 2 to 10 control cycles to read two set values for the designated parameters because of the time it takes to confirm the response of the position board.
- When a number of multiple parameters are read from multiple axes continuously, the writing time can be shortened by calling this function in each thread of "system" or "axis".

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.80	A8	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscChangeParameter, sscChange2Parameter, sscCheckParameter

4.4.6 sscLoadAllParameterFromFlashROM

All parameters before the system startup (system preparation completion) will be read from the flash ROM.

```
int sscLoadAllParameterFromFlashROM (
  int board_id,
  int channel,
  int timeout
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

timeout [in]

Timeout time[ms] (0 to 65535)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error codes

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_UNREADY_CHANNEL	The system is in the status other than system preparation completion. Reboot the system with the sscReboot function.
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	A system error (E□□□h) occurred. Get the system status code with the sscGetSystemStatusCode function and remove the cause.
SSC_FUNC_ERR_TIMEOUT_0□ □ = 1 to 2: Timeout location	During the confirmation of response after executing the command to the position board, the designated timeout time has elapsed.
SSC_FUNC_ERR_FLASHROM_PARAM_LOAD	The flash ROM parameter read error occurred.

Point

• When a value 2 seconds (2000ms) MC200 /5 seconds (5000ms) MC300 or less is designated as the timeout time, the timeout will be 2 seconds (2000ms) MC200 /5 seconds (5000ms) MC300 .

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscSaveAllParameterToFlashROM

4.4.7 sscSaveAllParameterToFlashROM

All parameters will be saved to the flash ROM.

```
int sscSaveAllParameterToFlashROM (
  int board_id,
  int channel,
  int timeout
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

timeout [in]

Timeout time[ms] (0 to 65535 MC200 /0 to 600000 MC300)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	A system error (E□□□h) occurred. Get the system status code with the sscGetSystemStatusCode function and remove the cause.
SSC_FUNC_ERR_TIMEOUT_0□ □ = 1 to 3: Timeout location	During the confirmation of response after executing the command to the position board, the designated timeout time has elapsed.
SSC_FUNC_ERR_STS_BIT_FRNG	The flash ROM transfer preparation error (FRNG) occurred.
SSC_FUNC_ERR_STS_BIT_FSNG	The flash ROM transfer error (FSNG) occurred.

Point

- When a value 10 seconds (10000ms) MC200 /300 seconds (300000ms) MC300 or less is designated as the timeout time, the timeout will be 10 seconds (10000ms) MC200 /300 seconds (300000ms) MC300 .
- To change the parameter contents backed up in a flash ROM to the initial value, call this function after initializing all parameters with the sscResetAllParameter function.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscLoadAllParameterFromFlashROM

4.4.8 sscCheckSvPrmChangeNumEx

The servo parameter change number will be got.

```
int  sscCheckSvPrmChangeNumEx (
   int  board_id,
   int  channel,
   int  axnum,
   short  *prmnum
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32 MC200 /1 to 64 MC300)

prmnum [out]

Pointer to 8-byte array (2 bytes × 4) which stores the servo parameter change number

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.

Point

- For SSCNETⅢ communication method, a value is not stored in the upper 6 bytes of the servo parameter change number.
- Use monitor for detailed information of the servo parameter change number.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.60	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

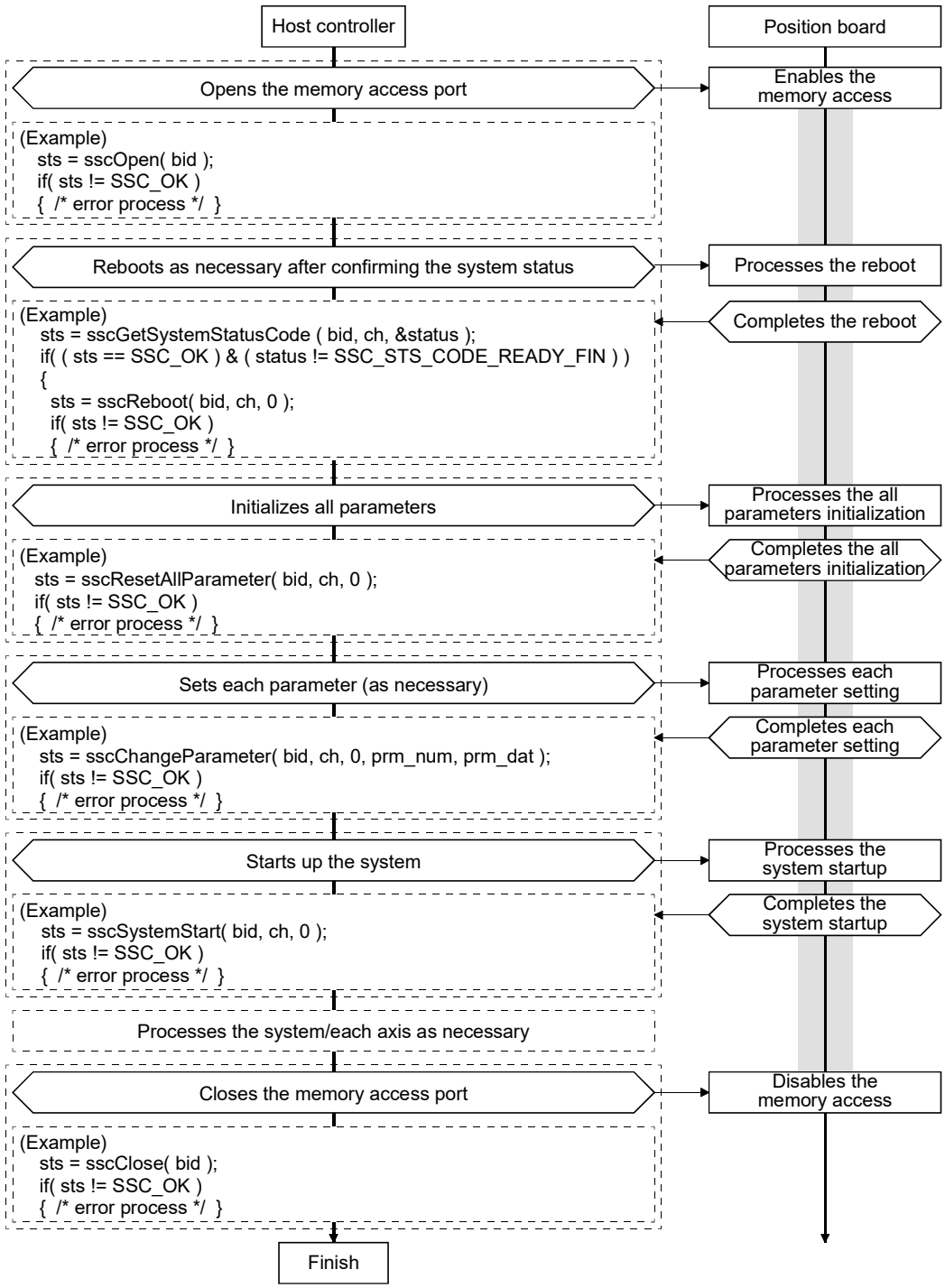
Reference

None.

4.5 System functions

(1) Processing procedure

An example of system processing procedure until starting up/shutting down the system is below.



4.5.1 sscReboot

The system will be rebooted (system running → system preparation completion).
The function will wait internally until the system preparation completion.

```
int  sscReboot (
   int  board_id,
   int  channel,
   int  timeout
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

timeout [in]

Timeout time[ms] (0 to 65535)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_TIMEOUT_0□ □ = 1 to 2: Timeout location	During the response after executing the command to the position board, the designated timeout time has elapsed.

Point

- When a value 10 seconds (10000ms) MC200 /20 seconds (20000ms) MC300 or less is designated as the timeout time, the timeout will be 10 seconds (10000ms) MC200 /20 seconds (20000ms) MC300 .
- The function will reboot even if the function executes in the system preparation is complete.
- Do not open device from the position board test tool during reboot.
  It may not operate correctly. MC300

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscGetSystemStatusCode

4.5.2 sscSystemStart

The system will start after servo amplifier communication initialization (system preparation completion → system running).

```
int  sscSystemStart (
   int  board_id,
   int  channel,
   int  timeout
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

timeout [in]

Timeout time[ms] (0 to 65535)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_UNREADY_CHANNEL	The system is in the status other than system preparation completion. Reboot the system with the sscReboot function.
SSC_FUNC_ERR_TIMEOUT_01	During the confirmation of response after executing the command to the position board, the designated timeout time has elapsed. Confirm that the SSCNETⅢ cable on the position board side is connected properly. Or, the SSCNET communication method is not correct.
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	A system error (E□□□h) occurred. Get the system status code with the sscGetSystemStatusCode function and remove the cause.

Point

- When a value 10 seconds (10000ms) or less is designated as the timeout time, the timeout will be 10 seconds (10000ms).
- Reboot when restarting the system.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

None.

4.5.3 sscGetSystemStatusCode

The system status code will be got.

```
int  sscGetSystemStatusCode (
   int  board_id,
   int  channel,
   short  *statuscode
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

statuscode [out]

Pointer to 2-byte variable (2 bytes × 1) which stores the system status code

Value	Description
SSC_STS_CODE_READY_FIN	System preparation completion
SSC_STS_CODE_RUNNING	System running

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error codes

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0☐ ☐ = 1 to 9: Argument location	The argument is outside the set range.

Point

• For the system status codes other than "System preparation completion" and "System running", refer to "MR-MC200/MR-MC300 Series Position Board User's Manual (Details)".

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2☐☐	Ver.1.00	A0	mc2xxstd.h
MR-MC3☐☐	Ver.1.00	A0	mc3xxstd.h

Reference

None.

4.5.4 sscReconnectSSCNET

SSCNET communication with the non-communicating axes designated as control axes will be started.

```
int  sscReconnectSSCNET (
   int  board_id,
   int  channel,
   unsigned long long *ctrl_axbit,
   unsigned short  *err_code
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

ctrl_axbit [out]

Pointer to 8-byte variable (8 bytes × 1) which stores the controlling axis information bit and controlling station information bit (0 to 0F000000FFFFFFFFh)

bit0 → Axis 1, bit1 → Axis 2 ... bit31 → Axis 32, bit56 → Station 1, bit57 → Station 2, bit58 → Station 3, bit59 → Station 4

err_code [out]

Pointer to 2-byte variable (2 bytes × 1) which stores the error code of reconnection/disconnection

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_TIMEOUT_0□ □ = 1 to 2: Timeout location	During the confirmation of response after executing the command to the position board, the timeout time (10 second) has elapsed.
SSC_FUNC_ERR_STS_BIT_RCE	The reconnection error (RCE) occurred.

Point

• For the "Error code of reconnection/disconnection", refer to "MR-MC200/MR-MC300 Series Position Board User's Manual (Details)".

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.80	A8	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscDisconnectSSCNET

4.5.5 sscReconnectSSCNETEx MC300

SSCNET communication with the non-communicating axes designated as control axes will be started.

```
int  sscReconnectSSCNETEx (
   int  board_id,
   int  channel,
   SLAVE_INFO  *pSlaveInfo,
   unsigned short  *err_code
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

pSlaveInfo [out]

Pointer to 80-byte structure (80 bytes × 1) which stores the slave information (controlling axis information and controlling station information)

Refer to "5.11 SLAVE_INFO structure" for the slave information structure.

err_code [out]

Pointer to 2-byte variable (2 bytes × 1) which stores the error code of reconnection/disconnection

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_TIMEOUT_0□ □ = 1 to 2: Timeout location	During the confirmation of response after executing the command to the position board, the timeout time (10 seconds) has elapsed.
SSC_FUNC_ERR_STS_BIT_RCE	The reconnection error (RCE) occurred.

Point

• For the "Error code of reconnection/disconnection", refer to "MR-MC200/MR-MC300 Series Position Board User's Manual (Details)".

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscDisconnectSSCNETEx

4.5.6 sscDisconnectSSCNET

SSCNET communication which is connected to the slave devices in SSCNET communication (such as servo amplifiers) set by the disconnection axis number or later will be disconnected.

```
int  sscDisconnectSSCNET (
   int  board_id,
   int  channel,
   int  com_num,
   unsigned long long *ctrl_axbit,
   unsigned short  *err_code
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

com_num [in]

Disconnection axis number (-4 to 32)

1 to 32: Axis number

-4 to -1: Station number (-1: Station 1, -2: Station 2, -3: Station 3, -4: Station 4)

ctrl_axbit [out]

Pointer to 8-byte variable (8 bytes × 1) which stores the controlling axis information bit and controlling station information bit (0 to 0F000000FFFFFFFFh)

bit0 → Axis 1, bit1 → Axis 2 ... bit31 → Axis 32, bit56 → Station 1, bit57 → Station 2, bit58 → Station 3, bit59 → Station 4

err_code [out]

Pointer to 2-byte variable (2 bytes × 1) which stores the error code of reconnection/disconnection

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_TIMEOUT_0□ □ = 1 to 2: Timeout location	During the confirmation of response after executing the command to the position board, the timeout time (10 second) has elapsed.
SSC_FUNC_ERR_STS_BIT_CCE	The disconnection error (CCE) occurred.

Point

• For the "Error code of reconnection/disconnection", refer to "MR-MC200/MR-MC300 Series Position Board User's Manual (Details)".

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.80	A8	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscReconnectSSCNET

4.5.7 sscDisconnectSSCNETEx MC300

SSCNET communication which is connected to the slave devices in SSCNET communication (such as servo amplifiers) set by the disconnection axis number or later will be disconnected.

```
int  sscDisconnectSSCNETEx (
   int  board_id,
   int  channel,
   int  com_num,
   SLAVE_INFO  *pSlaveInfo,
   unsigned short  *err_code
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

com_num [in]

Disconnection axis number (-16 to 64)

1 to 64: Axis number

-16 to -1: Station number (-1: Station 1, -2: Station 2, -3: Station 3, -4: Station 4 • • • • •-15: Station 15, -16: Station 16)

pSlaveInfo [out]

Pointer to 80-byte structure (80 bytes × 1) which stores the slave information (controlling axis information and controlling station information)

Refer to "5.11 SLAVE_INFO structure" for the slave information structure.

err_code [out]

Pointer to 2-byte variable (2 bytes × 1) which stores the error code of reconnection/disconnection

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_TIMEOUT_0□ □ = 1 to 2: Timeout location	During the confirmation of response after executing the command to the position board, the timeout time (10 seconds) has elapsed.
SSC_FUNC_ERR_STS_BIT_CCE	The disconnection error (CCE) occurred.

Point

• For the "Error code of reconnection/disconnection" , refer to "MR-MC200/MR-MC300 Series Position Board User's Manual (Details)".

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscReconnectSSCNETEx

4.5.8 sscGetControllingAxis MC300

Gets the controlling axis information and controlling station information.

```
int sscGetControllingAxis (
  int board_id,
  int channel,
  SLAVE_INFO *pSlaveInfo
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

pSlaveInfo [out]

Pointer to 80-byte structure (80 bytes × 1) which stores the slave information (controlling axis information and controlling station information)

Refer to "5.11 SLAVE_INFO structure" for the slave information structure.

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

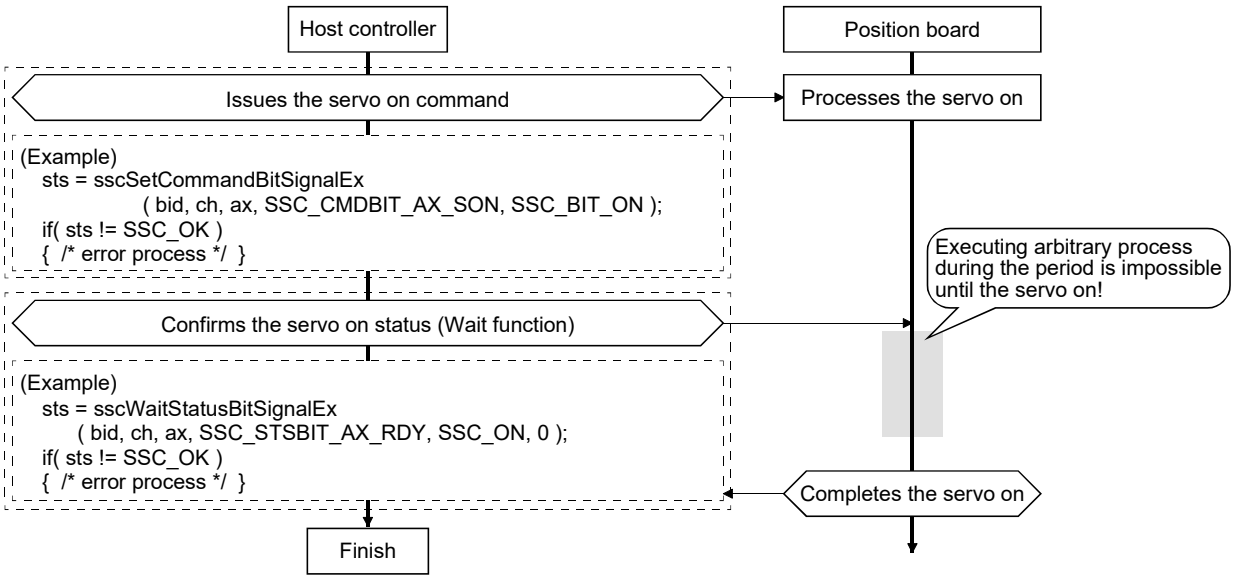
Reference

None.

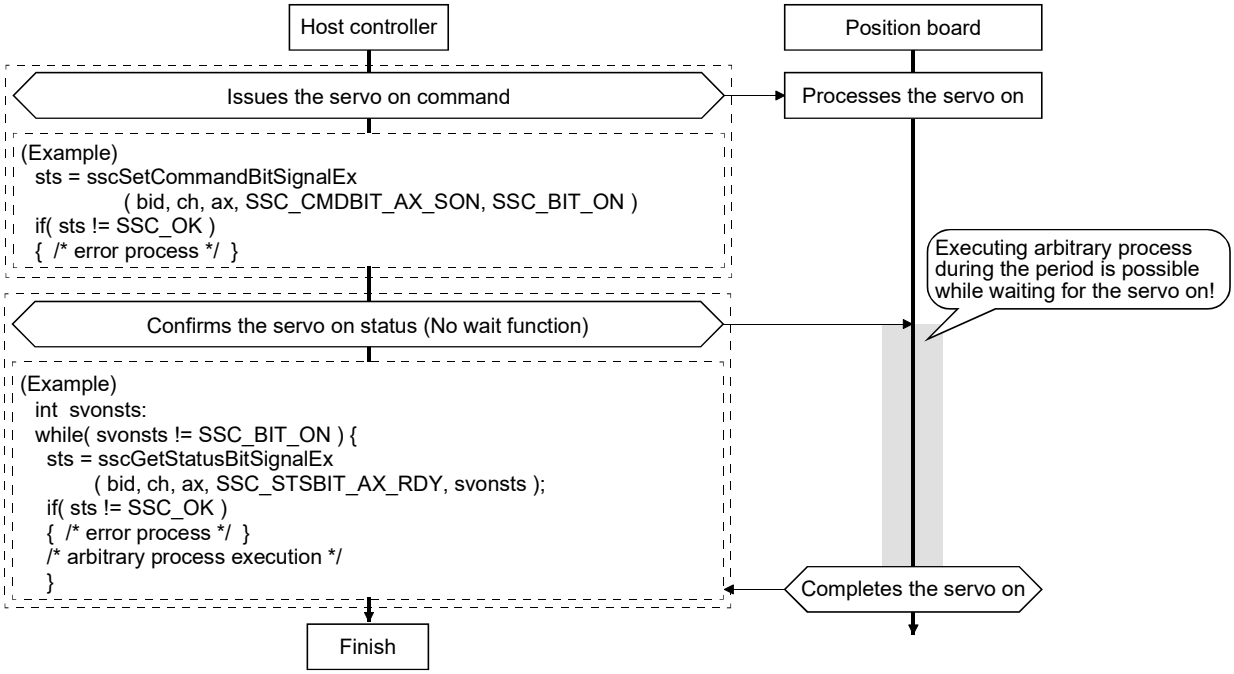
4.6 Command/Status functions

(1) Processing procedure

(a) Example of command/status processing procedure when turning on the servo with wait function



(b) Example of command/status processing procedure when turning on the servo with no wait function



4.6.1 sscSetCommandBitSignalEx

The designated command bit will be turned on or off.

```
int  sscSetCommandBitSignalEx (
   int  board_id,
   int  channel,
   int  axnum,
   int  bitnum,
   int  bitdata
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (-4 to 32 MC200 /-16 to 64 MC300)

0: System command bit

1 to 64: Axis command bit

-16 to -1: Station command bit (-1: Station 1, -2: Station 2, -3: Station 3, -4: Station 4 • • • • •-15: Station 15, -16: Station 16)

bitnum [in]

Command bit number

Refer to "Chapter 6 BIT DEFINITION LIST" for the command bit number.

bitdata [in]

Command bit data

Value	Description
SSC_BIT_OFF	Bit OFF
SSC_BIT_ON	Bit ON

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0☐ ☐ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_TIMEOUT_01	During the confirmation of response after executing the command to the position board, the timeout time (1 second) has elapsed.
SSC_FUNC_ERR_ARGUMENT_MISMATCH	The axis number and the command bit number do not correspond. (Example: When "0" is set in the axis number and "SSC_CMDBIT_AX_SON" is set in the command bit number, etc.)

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.80	A8	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscGetStatusBitSignalEx, sscWaitStatusBitSignalEx

4.6.2 sscGetStatusBitSignalEx

The designated status bit will be got.

```
int  sscGetStatusBitSignalEx (
   int  board_id,
   int  channel,
   int  axnum,
   int  bitnum,
   int  *bitstatus
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (-4 to 32 MC200 /-16 to 64 MC300)

0: System command bit

1 to 64: Axis command bit

-16 to -1: Station command bit (-1: Station 1, -2: Station 2, -3: Station 3, -4: Station 4 • • • • •-15: Station 15, -16: Station 16)

bitnum [in]

Status bit number

Refer to "Chapter 6 BIT DEFINITION LIST" for the status bit number.

bitstatus [out]

Pointer to 4-byte variable (4 bytes × 1) which stores the status bit data

Value	Description
SSC_BIT_OFF	Bit OFF
SSC_BIT_ON	Bit ON

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_ARGUMENT_MISMATCH	The axis number and the status bit number do not correspond. (Example: When "0" is set to the axis number and "SSC_STSBIT_AX_RDY" is set to the status bit number, etc.)

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.80	A8	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscSetCommandBitSignalEx, sscWaitStatusBitSignalEx

4.6.3 sscWaitStatusBitSignalEx

This function waits until the designated status bit becomes to the designated state.

```
int sscWaitStatusBitSignalEx (
  int board_id,
  int channel,
  int axnum,
  int bitnum,
  int waitstatus,
  int timeout
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (-4 to 32 MC200 /-16 to 64 MC300)

0: System command bit

1 to 64: Axis command bit

-16 to -1: Station command bit (-1: Station 1, -2: Station 2, -3: Station 3, -4: Station 4 • • • • •-15: Station 15, -16: Station 16)

bitnum [in]

Status bit number

Refer to "Chapter 6 BIT DEFINITION LIST" for the status bit number.

waitstatus [in]

Bit status to be waited

Value	Description
SSC_BIT_OFF	Bit OFF
SSC_BIT_ON	Bit ON

timeout [in]

Timeout time[ms] (0 to 65535)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_TIMEOUT_01	During the confirmation of response after executing the command to the position board, the designated timeout time has elapsed.
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	A system error (E□□□h) occurred. Get the system status code with the sscGetSystemStatusCode function and remove the cause.
SSC_FUNC_ERR_ARGUMENT_MISMATCH	The axis number and the status bit number do not correspond. (Example: When "0" is set to the axis number and "SSC_STSBIT_AX_RDY" is set to the status bit number, etc.)

Point

• When SSC_INFINITE is designated as the timeout time, timeout is not checked. Instead, this function infinitely waits until the status bit becomes to the designated "Bit status to be waited".

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.80	A8	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscSetCommandBitSignalEx, sscGetStatusBitSignalEx

4.7 Point table functions

4.7.1 sscSetPointDataEx

The point data stored in the structure variable designated by the pointer will be set.

```
int  sscSetPointDataEx (
   int  board_id,
   int  channel,
   int  axnum,
   int  pntnum,
   PNT_DATA_EX *pPntDataEx
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32 MC200 /1 to 64 MC300)

pntnum [in]

Point number (0 to 319 MC200 /0 to 2047 MC300)

pPntDataEx [in]

Pointer to 32-byte structure (32 bytes × 1) MC200 /48-byte structure (48 bytes × 1) MC300 which stores the point data

Refer to "5.1 PNT_DATA_EX structure" for the point data structure.

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0☐ ☐ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_POINT_NUMBER_OVER	Designated point number + point number offset value exceeded the point table range.

Point

• A check of the set point data contents will not be performed.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2☐☐	Ver.1.00	A0	mc2xxstd.h
MR-MC3☐☐	Ver.1.00	A0	mc3xxstd.h

Reference

sscSetPointOffset

4.7.2 sscCheckPointDataEx

The point data will be got.

```
int  sscCheckPointDataEx (
   int  board_id,
   int  channel,
   int  axnum,
   int  pntnum,
   PNT_DATA_EX  *pPntDataEx
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32 MC200 /1 to 64 MC300)

pntnum [in]

Point number (0 to 319 MC200 /0 to 2047 MC300)

pPntDataEx [in]

Pointer to 32-byte structure (32 bytes × 1) MC200 /48-byte structure (48 bytes × 1) MC300 which stores the point data

Refer to "5.1 PNT_DATA_EX structure" for the point data structure.

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_POINT_NUMBER_OVER	Designated point number + point number offset value exceeded the point table range.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscCheckPointOffset

4.7.3 sscSetPointOffset

The point number offset will be set.

```
int  sscSetPointOffset (
   int  board_id,
   int  channel,
   int  axnum,
   short  offset
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32 MC200 /1 to 64 MC300)

offset [in]

Point number offset value (0 to 319 MC200 /0 to 2047 MC300)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscSetPointDataEx

4.7.4 sscCheckPointOffset

The point number offset will be got.

```
int  sscCheckPointOffset (
   int  board_id,
   int  channel,
   int  axnum,
   short  *offset
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32 MC200 /1 to 64 MC300)

offset [out]

Pointer to 2-byte variable (2 bytes × 1) which stores the point table offset

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscCheckPointDataEx

4.7.5 sscGetDrivingPointNumber

The operation point number will be got.

```
int  sscGetDrivingPointNumber (
   int  board_id,
   int  channel,
   int  axnum,
   short  *driving_pnt
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32 MC200 /1 to 64 MC300)

driving_pnt [out]

Pointer to 2-byte variable (2 bytes × 1) which stores the operation point number

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_MISMATCH_DRIVE_MODE	The operation mode is other than the "automatic operation" or "linear interpolation MC200 "/"interpolation operation MC300 ".

Point

• The got operation point number is stored after 1 is added. When the operation is stopped, 0 will be stored.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

None.

4.7.6 sscSetLatestPointNumber

Sets the latest command point number. Used in point table loop method.

```
int  sscSetLatestPointNumber (
   int  board_id,
   int  channel,
   int  axnum,
   short  latest_point
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32 MC200 /1 to 64 MC300)

latest_point [in]

Latest command point number (1 to 320 MC200 /1 to 2048 MC300)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.

Point

• Set the latest command point number to the value of the point number + 1.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.80	A6	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscAutoStart, sscSetPointDataEx

4.8 Continuous operation to torque control data functions

4.8.1 sscSetPressData

The continuous operation to torque control data stored in the structure variable designated by the pointer will be set.

```
int  sscSetPressData (
   int  board_id,
   int  channel,
   int  axnum,
   PRESS_DATA  *pPressData
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32 MC200 /1 to 64 MC300)

pPressData [in]

Pointer to 32-byte structure (32 bytes × 1) which stores the continuous operation to torque control data

Refer to "5.3 PRESS_DATA structure" for the continuous operation to torque control data structure.

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.

Point

• A check of the set continuous operation to torque control data contents will not be performed.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.60	A5	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscGetPressData

4.8.2 sscGetPressData

The continuous operation to torque control data will be got.

```
int  sscGetPressData (
   int  board_id,
   int  channel,
   int  axnum,
   PRESS_DATA  *pPressData
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32 MC200 /1 to 64 MC300)

pPressData [out]

Pointer to 32-byte structure (32 bytes × 1) which stores the continuous operation to torque control data

Refer to "5.3 PRESS_DATA structure" for the continuous operation to torque control data structure.

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.60	A5	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscSetPressData

4.9 Operating functions

4.9.1 sscJogStart

JOG operation will be started.
After performing the necessary settings for operation and changing to JOG operation mode, the start operation signal (ST) will be turned on.

```
int  sscJogStart (
   int  board_id,
   int  channel,
   int  axnum,
   long  speed,
   short  actime,
   short  dctime,
   char  dir
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32 MC200 /1 to 64 MC300)

speed [in]

Manual feed speed [speed unit] (0 to 2147483647)

actime [in]

Manual feed speed acceleration time constant [ms] (0 to 20000)

dctime [in]

Manual feed speed deceleration time constant [ms] (0 to 20000)

dir [in]

Movement direction

Value	Description
SSC_DIR_PLUS	+ direction
SSC_DIR_MINUS	- direction

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_NOW_DRIVING_READY	During operation startup preparation (until the position board receives the signal after the start operation is requested).
SSC_FUNC_ERR_NOW_DRIVING	During operation.
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	A system error (E□□□h) occurred. Get the system status code with the sscGetSystemStatusCode function and remove the cause.
SSC_FUNC_ERR_NOW_ALARM_SERVO	A servo alarm is occurring. Get the alarm No. with the sscGetAlarm function and remove the cause.
SSC_FUNC_ERR_NOW_ALARM_DRIVE	An operation alarm is occurring. Get the alarm No. with the sscGetAlarm function and remove the cause.
SSC_FUNC_ERR_TIMEOUT_01	During the confirmation of response after executing the command to the position board, the timeout time (1 second) has elapsed.

Point

• The response is not confirmed after the start operation signal (ST) is turned on.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscJogStop, sscJogStopNoWait, sscDriveStop, sscDriveStopNoWait, sscDriveRapidStop, sscDriveRapidStopNoWait

4.9.2 sscJogStop

JOG operation will be stopped.
The function will wait internally from when the start operation signal (ST) is turned off until the during operation signal (OP) is OFF (maximum 20s).
If the during operation signal (OP) is already OFF, the function will immediately terminate.

```
int sscJogStop (
  int board_id,
  int channel,
  int axnum
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32 MC200 /1 to 64 MC300)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_MISMATCH_DRIVE_MODE	The operation mode is other than the "JOG operation".
SSC_FUNC_ERR_TIMEOUT_01	During the confirmation of response is being waited after requesting the command to the position board, the timeout time (20 seconds) has elapsed.
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	A system error (E□□□h) occurred. Get the system status code with the sscGetSystemStatusCode function and remove the cause.

Point

• In order to prevent this function from waiting internally, the sscJogStopNoWait function should be used.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscJogStart, sscJogStopNoWait, sscDriveStop, sscDriveStopNoWait, sscDriveRapidStop, sscDriveRapidStopNoWait

4.9.3 sscJogStopNoWait

JOG operation will be stopped.
The start operation signal (ST) is turned off and the during operation signal (OP) will be returned.

```
int  sscJogStopNoWait (
   int  board_id,
   int  channel,
   int  axnum,
   short  *stpsts
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32 MC200 /1 to 64 MC300)

stpsts [out]

Pointer to 2-byte variable (2 bytes × 1) which stores the stop complete status

Value	Description
SSC_DRIVING	During operation
SSC_DRIVE_FIN	Stop complete

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0☐ ☐ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_MISMATCH_DRIVE_MODE	The operation mode is other than the "JOG operation".

Point

• In order to make use of wait inside this function, the sscJogStop function should be used.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2☐☐	Ver.1.00	A0	mc2xxstd.h
MR-MC3☐☐	Ver.1.00	A0	mc3xxstd.h

Reference

sscJogStart, sscJogStop, sscDriveStop, sscDriveStopNoWait, sscDriveRapidStop, sscDriveRapidStopNoWait

4.9.4 sscIncStart

Incremental feed will be started.
After performing the necessary settings for operation and changing to incremental feed mode, the fast start operation signal (FST) will be turned on.

```
int sscIncStart (
   int board_id,
   int channel,
   int axnum,
   long distance,
   long speed,
   short actime,
   short dctime
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32 MC200 /1 to 64 MC300)

distance [in]

Amount of incremental feed movement [command unit] (-2147483647 to 2147483647)

speed [in]

Manual feed speed [speed unit] (0 to 2147483647)

actime [in]

Manual feed speed acceleration time constant [ms] (0 to 20000)

dctime [in]

Manual feed speed deceleration time constant [ms] (0 to 20000)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_NOW_DRIVING_READY	During operation startup preparation (until the position board receives the signal after the start operation is requested).
SSC_FUNC_ERR_NOW_DRIVING	During operation.
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	A system error (E□□□h) occurred. Get the system status code with the sscGetSystemStatusCode function and remove the cause.
SSC_FUNC_ERR_NOW_ALARM_SERVO	A servo alarm is occurring. Get the alarm No. with the sscGetAlarm function and remove the cause.
SSC_FUNC_ERR_NOW_ALARM_DRIVE	An operation alarm is occurring. Get the alarm No. with the sscGetAlarm function and remove the cause.
SSC_FUNC_ERR_TIMEOUT_01	During the confirmation of response after executing the command to the position board, the timeout time (1 second) has elapsed.

Point

- If a positive value is designated for the movement distance, movement will be in the + direction, and if a negative value is designated, movement will be in the - direction.
- The response is not confirmed after the fast start operation signal (FST) is turned on.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscDriveStop, sscDriveStopNoWait, sscDriveRapidStop, sscDriveRapidStopNoWait

4.9.5 sscAutoStart

Automatic operation will be started.
After performing the necessary settings for operation and changing to automatic operation mode, the fast start operation signal (FST) will be turned on.

```
int  sscAutoStart (
   int  board_id,
   int  channel,
   int  axnum,
   int  point_s,
   int  point_e
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32 MC200 /1 to 64 MC300)

point_s [in]

Start point number (0 to 319 MC200 /0 to 2047 MC300)

point_e [in]

End point number (0 to 319 MC200 /0 to 2047 MC300)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_NOW_DRIVING_READY	During operation startup preparation (until the position board receives the signal after the start operation is requested).
SSC_FUNC_ERR_NOW_DRIVING	During operation.
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	A system error (E□□□h) occurred. Get the system status code with the sscGetSystemStatusCode function and remove the cause.
SSC_FUNC_ERR_NOW_ALARM_SERVO	A servo alarm is occurring. Get the alarm No. with the sscGetAlarm function and remove the cause.
SSC_FUNC_ERR_NOW_ALARM_DRIVE	An operation alarm is occurring. Get the alarm No. with the sscGetAlarm function and remove the cause.
SSC_FUNC_ERR_TIMEOUT_01	During the confirmation of response after executing the command to the position board, the timeout time (1 second) has elapsed.

Point

• The response is not confirmed after the fast start operation signal (FST) is turned on.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscDriveStop, sscDriveStopNoWait, sscDriveRapidStop, sscDriveRapidStopNoWait

4.9.6 sscHomeReturnStart

Home position return will be started.
After performing the necessary settings for operation and changing to home position return mode, the fast start operation signal (FST) will be turned on.

```
int  sscHomeReturnStart (
   int  board_id,
   int  channel,
   int  axnum
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32 MC200 /1 to 64 MC300)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_NOW_DRIVING_READY	During operation startup preparation (until the position board receives the signal after the start operation is requested).
SSC_FUNC_ERR_NOW_DRIVING	During operation.
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	A system error (E□□□h) occurred. Get the system status code with the sscGetSystemStatusCode function and remove the cause.
SSC_FUNC_ERR_NOW_ALARM_SERVO	A servo alarm is occurring. Get the alarm No. with the sscGetAlarm function and remove the cause.
SSC_FUNC_ERR_NOW_ALARM_DRIVE	An operation alarm is occurring. Get the alarm No. with the sscGetAlarm function and remove the cause.
SSC_FUNC_ERR_TIMEOUT_01	During the confirmation of response after executing the command to the position board, the timeout time (1 second) has elapsed.

Point

• The response is not confirmed after the fast start operation signal (FST) is turned on.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscDriveStop, sscDriveStopNoWait, sscDriveRapidStop, sscDriveRapidStopNoWait

4.9.7 sscLinearStart

Linear interpolation will be started.
After performing the necessary settings for operation and changing to linear interpolation mode, the fast start operation signal (FST) will be turned on.

```
int  sscLinearStart (
    int  board_id,
    int  channel,
    int  axnum,
    int  grpnum,
    int  point_s,
    int  point_e
);
```

Argument

board_id [in]
Board ID number (0 to 3)

channel [in]
Channel number (1)

axnum [in]
Primary axis number (1 to 32)

grpnum [in]
Interpolation group number (0 to 8)
0 : Changeable interpolation group enabled
1 to 8 : Changeable interpolation group disabled

point_s [in]
Start point number (0 to 319)

point_e [in]
End point number (0 to 319)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_POINT_NUMBER_OVER	Designated point number + point number offset value exceeded the point table range.
SSC_FUNC_ERR_STS_BIT_IPCH_ON	Changeable interpolation group setting is enabled. Set 0 to interpolation group number.
SSC_FUNC_ERR_STS_BIT_IPCH_OFF	Changeable interpolation group setting is disabled. Set 1 to 8 to interpolation group number.
SSC_FUNC_ERR_SUB_AXIS_NUM	When changeable interpolation group is enabled, the interpolation axis No. set to the point table is an incorrect value. • The interpolation axis No. is outside the setting range • The interpolation axis No. is the same as the primary axis No. • The interpolation axis No. is the same as another interpolation axis No.
SSC_FUNC_ERR_NOW_DRIVING_READY	During operation startup preparation (until the position board receives the signal after the start operation is requested).
SSC_FUNC_ERR_NOW_DRIVING	During operation.
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	A system error (E□□□h) occurred. Get the system status code with the sscGetSystemStatusCode function and remove the cause.
SSC_FUNC_ERR_NOW_ALARM_SERVO	A servo alarm is occurring. Get the alarm No. with the sscGetAlarm function and remove the cause.
SSC_FUNC_ERR_NOW_ALARM_DRIVE	An operation alarm is occurring. Get the alarm No. with the sscGetAlarm function and remove the cause.
SSC_FUNC_ERR_TIMEOUT_01	During the confirmation of response after executing the command to the position board, the timeout time (1 second) has elapsed.

Point

• The response is not confirmed after the fast start operation signal (FST) is turned on.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.90	A9	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscDriveStop, sscDriveStopNoWait, sscDriveRapidStop, sscDriveRapidStopNoWait

4.9.8 sscInterpolationStart MC300

Interpolation operation will be started.
After performing the necessary settings for operation and changing to interpolation operation mode, the fast start operation signal (FST) will be turned on.

```
int  sscInterpolationStart (
   int  board_id,
   int  channel,
   int  axnum,
   int  grpnum,
   int  point_s,
   int  point_e
);
```

Argument

board_id [in]
Board ID number (0 to 3)

channel [in]
Channel number (1)

axnum [in]
Primary axis number (1 to 64)

grpnum [in]
Interpolation group number (0 to 16)
0 : Changeable interpolation group enabled
1 to 16 : Changeable interpolation group disabled

point_s [in]
Start point number (0 to 2047)

point_e [in]
End point number (0 to 2047)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_POINT_NUMBER_OVER	Designated point number + point number offset value exceeded the point table range.
SSC_FUNC_ERR_STS_BIT_IPCH_ON	Changeable interpolation group setting is enabled. Set 0 to interpolation group number.
SSC_FUNC_ERR_STS_BIT_IPCH_OFF	Changeable interpolation group setting is disabled. Set 1 to 16 to interpolation group number.
SSC_FUNC_ERR_SUB_AXIS_NUM	When changeable interpolation group is enabled, the interpolation axis No. set to the point table is an incorrect value. • The interpolation axis No. is outside the setting range • The interpolation axis No. is the same as the primary axis No. • The interpolation axis No. is the same as another interpolation axis No.
SSC_FUNC_ERR_NOW_DRIVING_READY	During operation startup preparation (until the position board receives the signal after the start operation is requested).
SSC_FUNC_ERR_NOW_DRIVING	During operation.
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	A system error (E□□□h) occurred. Get the system status code with the sscGetSystemStatusCode function and remove the cause.
SSC_FUNC_ERR_NOW_ALARM_SERVO	A servo alarm is occurring. Get the alarm No. with the sscGetAlarm function and remove the cause.
SSC_FUNC_ERR_NOW_ALARM_DRIVE	An operation alarm is occurring. Get the alarm No. with the sscGetAlarm function and remove the cause.
SSC_FUNC_ERR_TIMEOUT_01	During the confirmation of response after executing the command to the position board, the timeout time (1 second) has elapsed.

Point

• The response is not confirmed after the fast start operation signal (FST) is turned on.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscDriveStop, sscDriveStopNoWait, sscDriveRapidStop, sscDriveRapidStopNoWait

4.9.9 sscDataSetStart

Home position reset (data set) will be started.
After performing the necessary settings for the operation and changing to home position reset (data set) mode, the fast start operation signal (FST) will be turned on.

```
int  sscDataSetStart (
   int  board_id,
   int  channel,
   int  axnum
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32 MC200 /1 to 64 MC300)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_NOW_DRIVING_READY	During operation startup preparation (until the position board receives the signal after the start operation is requested).
SSC_FUNC_ERR_NOW_DRIVING	During operation.
SSC_FUNC_ERR_NOW_ALARM_SERVO	A servo alarm is occurring. Get the alarm No. with the sscGetAlarm function and remove the cause.
SSC_FUNC_ERR_NOW_ALARM_DRIVE	An operation alarm is occurring. Get the alarm No. with the sscGetAlarm function and remove the cause.
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	A system error (E□□□h) occurred. Get the system status code with the sscGetSystemStatusCode function and remove the cause.
SSC_FUNC_ERR_TIMEOUT_01	During the confirmation of response after executing the command to the position board, the timeout time (1 second) has elapsed.

Point

• The response is not confirmed after the fast start operation signal (FST) is turned on.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscDriveStop, sscDriveStopNoWait, sscDriveRapidStop, sscDriveRapidStopNoWait

4.9.10 sscDriveStop

The operations will be stopped.
After the start operation signal (ST) is turned off and the stop operation signal (STP) turned on, the function will wait internally until the during operation signal (OP) is OFF.
After the during operation signal (OP) is confirmed to be OFF, the stop operation signal (STP) will be turned off.

```
int sscDriveStop (
  int board_id,
  int channel,
  int axnum,
  int timeout
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32 MC200 /1 to 64 MC300)

timeout [in]

Timeout time[ms] (0 to 65535)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_TIMEOUT_0□ □ = 1 to 2: Timeout location	During the confirmation of response after executing the command to the position board, the designated timeout time has elapsed.
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	A system error (E□□□h) occurred. Get the system status code with the sscGetSystemStatusCode function and remove the cause.

Point

• When 0 is designated as the timeout time, the timeout will be 20 seconds (20000ms).

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscDriveStopNoWait, sscDriveRapidStop, sscDriveRapidStopNoWait

4.9.11 sscDriveStopNoWait

The operations will be stopped.
The stop signal (STP) is turned on and the stop complete status is returned.

```
int  sscDriveStopNoWait (
   int  board_id,
   int  channel,
   int  axnum,
   short  *stpsts
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32 MC200 /1 to 64 MC300)

stpsts [out]

Pointer to 2-byte variable (2 bytes × 1) which stores the stop complete status

Value	Description
SSC_DRIVING	During operation
SSC_DRIVE_FIN	Stop complete

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_TIMEOUT_01	During the confirmation of response after executing the command to the position board, the timeout time (1 second) has elapsed.
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	A system error (E□□□h) occurred. Get the system status code with the sscGetSystemStatusCode function and remove the cause.

Point

- In order to make use of wait inside this function, the sscDriveStop function should be used.
- If the stop complete status storage variable has not been confirmed to be SSC_DRIVE_FIN, the stop operation signal (STP) will remain ON.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscDriveStop, sscDriveRapidStop, sscDriveRapidStopNoWait

4.9.12 sscDriveRapidStop

Operations will be stopped rapidly.
After the start operating signal (ST) is turned off and the rapid stop signal (RSTP) is turned on, the function will wait internally until the operating signal (OP) is turned off.
After the operating signal (OP) is confirmed to be OFF, the rapid stop signal (RSTP) will be turned off.

```
int  sscDriveRapidStop (
   int  board_id,
   int  channel,
   int  axnum,
   int  timeout
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32 MC200 /1 to 64 MC300)

timeout [in]

Timeout time[ms] (0 to 65535)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_TIMEOUT_0□ □ = 1 to 2: Timeout location	During the confirmation of response after executing the command to the position board, the designated timeout time has elapsed.
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	A system error (E□□□h) occurred. Get the system status code with the sscGetSystemStatusCode function and remove the cause.

Point

- When 0 is designated as the timeout time, the timeout will be 20 seconds (20000ms).
- In order to prevent this function from waiting internally, the sscDriveRapidStopNoWait function should be used.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscDriveStop, sscDriveStopNoWait, sscDriveRapidStopNoWait

4.9.13 sscDriveRapidStopNoWait

Operations will be stopped rapidly.
The start operation signal (ST) is turned off and the during operation signal (OP) is returned.

```
int  sscDriveRapidStopNoWait (
   int  board_id,
   int  channel,
   int  axnum,
   short  *stpsts
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32 MC200 /1 to 64 MC300)

stpsts [out]

Pointer to the stop complete status variable

Pointer to 2-byte variable (2 bytes × 1) which stores the stop complete status

Value	Description
SSC_DRIVING	During operation
SSC_DRIVE_FIN	Stop complete

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_TIMEOUT_01	During the confirmation of response after executing the command to the position board, the timeout time (1 second) has elapsed.
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	A system error (E□□□h) occurred. Get the system status code with the sscGetSystemStatusCode function and remove the cause.

Point

- In order to make use of wait inside this function, the sscDriveRapidStop function should be used.
- If the stop complete status storage variable has not been confirmed to be SSC_DRIVE_FIN, the rapid stop operation signal (RSTP) will remain ON.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscDriveStop, sscDriveStopNoWait, sscDriveRapidStop

4.9.14 sscSetDriveMode

The operation mode will be switched.
This function is used for setting the operation mode of the axis for which the other axes start is performed by the other axes start function.

```
int  sscSetDriveMode (
   int  board_id,
   int  channel,
   int  axnum,
   int  drv_mode
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32 MC200 /1 to 64 MC300)

drv_mode [in]

Operation mode (1 to 32)

Operation mode	Description
SSC_DRV_MODE_AUTO	Automatic operation mode
SSC_DRV_MODE_HOME	Home position return mode
SSC_DRV_MODE_JOG	JOG operation mode
SSC_DRV_MODE_INC	Incremental feed mode
SSC_DRV_MODE_LINEAR	Linear interpolation mode MC200
SSC_DRV_MODE_INTERP	Interpolation operation mode MC300
SSC_DRV_MODE_DST	Home position reset mode

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	A system error (E□□□h) occurred. Get the system status code with the sscGetSystemStatusCode function and remove the cause.
SSC_FUNC_ERR_NOW_DRIVING	During operation.
SSC_FUNC_ERR_NOW_ALARM_SERVO	A servo alarm is occurring. Get the alarm No. with the sscGetAlarm function and remove the cause.
SSC_FUNC_ERR_NOW_ALARM_DRIVE	An operation alarm is occurring. Get the alarm No. with the sscGetAlarm function and remove the cause.

Point

- To switch the operation mode of an axis for which the other axes start is performed in advance, use this function.
- Since the functions which start operations (such as sscAutoStart) switches the operation mode inside the function, this function is not needed to be called.
- Since the position board imports the operation mode when the acceptance of the start operation signal (ST) is completed, it does not need to wait until the operation mode is switched after this function is called.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscGetDriveMode, sscSetOtherAxisStartData

4.9.15 sscGetDriveMode

The operation mode will be got.

```
int  sscGetDriveMode (
   int  board_id,
   int  channel,
   int  axnum,
   int  *drv_mode
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32 MC200 /1 to 64 MC300)

drv_mode [out]

Pointer to 4-byte variable (4 bytes × 1) which stores the operation mode status

Operation mode	Description
SSC_DRV_MODE_NONE	Operation mode not set
SSC_DRV_MODE_AUTO	In automatic operation mode
SSC_DRV_MODE_HOME	In home position return mode
SSC_DRV_MODE_JOG	In JOG operation mode
SSC_DRV_MODE_INC	In incremental feed mode
SSC_DRV_MODE_LINEAR	In linear interpolation mode MC200
SSC_DRV_MODE_INTERP	Interpolation operation mode MC300
SSC_DRV_MODE_DST	In home position reset mode

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.

Point

• When the operation mode of axis for which the other axes start is performed needs to be switched in advance, use this function to check the operation mode.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscSetDriveMode, sscSetOtherAxisStartData

4.9.16 sscGetDriveFinStatus

The completion of operation status will be got.

```
int  sscGetDriveFinStatus (
   int  board_id,
   int  channel,
   int  axnum,
   int  fin_type,
   int  *fin_status,
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32 MC200 /1 to 64 MC300)

fin_type [in]

Operation completion type

Value	Description
SSC_FIN_TYPE_SMZ	Completion of operation by smoothing stop
SSC_FIN_TYPE_CPO	Completion of operation by rough match
SSC_FIN_TYPE_INP	Completion of operation by in-position stop

fin_status [out]

Pointer to 4-byte variable (4 bytes × 1) which stores the completion of operation status

Value	Description
SSC_FIN_STS_RDY	Before start up acceptance
SSC_FIN_STS_STP	Completion of operation
SSC_FIN_STS_MOV	During operation
SSC_FIN_STS_ALM_STP	Alarm occurrence (stop complete)
SSC_FIN_STS_ALM_MOV	Alarm occurrence (during deceleration stop)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0☐ ☐ = 1 to 9: Argument location	The argument is outside the set range.

Point

- When the operation completion type is set to "Stop by rough match" in other than the automatic operation and the linear interpolation MC200/interpolation operation MC300, the completion of operation confirmation type is judged by the "Smoothing stop" unconditionally.
- When the operation completion type is set to "In-position stop", the operation mode is always "During the operation" when the in-position signal is OFF.
- For the interlock stop or interference check standby, the completion of operation status is "During the operation".
- When the "deceleration check system" is set to in-position stop in the automatic operation or the linear interpolation MC200/ interpolation operation MC300, the "Operation completion type" is "In-position stop" even though the smoothing stop is set.
- The system alarm and the system error are not confirmed.
- Since the completion of operation status of this function is judged by the operation completion type, it may differ from the during • operation signal (OP) and the completion of operation signal (OPF) of the position board.
- For JOG operation, the completion of operation status is not "Before start up acceptance".

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

None.

4.9.17 sscChangeControlMode

The control mode of the servo amplifier will be switched.

```
int  sscChangeControlMode (
   int  board_id,
   int  channel,
   int  axnum,
   unsigned short  ctrl_mode
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32 MC200 /1 to 64 MC300)

ctrl_mode [in]

Control mode

Operation mode	Description
SSC_CTRL_MODE_POSITION	Position control mode
SSC_CTRL_MODE_PRESS	Continuous operation to torque control mode

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_STS_BIT_IFMO	Position board is in interface mode. When changing control mode in interface mode, use the sscIfmSetControlMode function.
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	A system error (E□□□h) occurred. Get the system status code with the sscGetSystemStatusCode function and remove the cause.
SSC_FUNC_ERR_NOW_ALARM_SERVO	A servo alarm is occurring. Get the alarm No. with the sscGetAlarm function and remove the cause.
SSC_FUNC_ERR_NOW_ALARM_DRIVE	An operation alarm is occurring. Get the alarm No. with the sscGetAlarm function and remove the cause.
SSC_FUNC_ERR_TIMEOUT_0□ □ = 1 to 2: Timeout location	During the confirmation of response after executing the command to the position board, the timeout time (1 second) has elapsed.
SSC_FUNC_ERR_STS_BIT_CTLMCE	The control mode switch error signal (CLTMCE) turned ON.

Point

• It takes approximately six control cycles + 4 to 6ms to switch the control mode of the servo amplifier becuase of the time it takes to confirm the response of the position board.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.60	A5	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscSetPressData

4.10 Change functions

4.10.1 sscChangeManualPosition

The position will be changed in the incremental feed.
The position change signal (PCHG) will be turned on, and the completion of preparation for changing position signal (PCF) or the position change error signal (PCE) will be confirmed to be on.

```
int  sscChangeManualPosition (
   int  board_id,
   int  channel,
   int  axnum,
   long  position
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32 MC200 /1 to 64 MC300)

position [in]

Amount of incremental feed movement after change [command unit] (-2147483647 to 2147483647)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_MISMATCH_DRIVE_MODE	The operation mode is other than the "Incremental feed".
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	A system error (E□□□h) occurred. Get the system status code with the sscGetSystemStatusCode function and remove the cause.
SSC_FUNC_ERR_TIMEOUT_0□ □ = 1 to 2: Timeout location	During the confirmation of response after executing the command to the position board, the timeout time (1 second) has elapsed.
SSC_FUNC_ERR_STS_BIT_PCE	The position change error signal (PCE) turned on.
SSC_FUNC_ERR_CHG_POS_DIR	The movement direction differs between before and after the position change.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscChangeManualSpeed, sscChangeManualAccTime, sscChangeManualDecTime

4.10.2 sscChangeAutoPosition

The automatic operation position will be changed.
The position change signal (PCHG) will be turned on, and the completion of preparation for changing position signal (PCF) or the position change error signal (PCE) will be confirmed to be on.

```
int  sscChangeAutoPosition (
   int  board_id,
   int  channel,
   int  axnum,
   int  pntnum,
   long  position
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32 MC200 /1 to 64 MC300)

pntnum [in]

Point number (0 to 319 MC200 /0 to 2047 MC300)

position [in]

Position data after change [Command unit]

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_MISMATCH_DRIVE_MODE	The operation mode is other than the "Automatic operation".
SSC_FUNC_ERR_POINT_NUMBER_OVER	Designated point number + point number offset value exceeded the point table range.
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	A system error (E□□□h) occurred. Get the system status code with the sscGetSystemStatusCode function and remove the cause.
SSC_FUNC_ERR_TIMEOUT_0□ □ = 1 to 2: Timeout location	During the confirmation of response after executing the command to the position board, the timeout time (1 second) has elapsed.
SSC_FUNC_ERR_STS_BIT_PCE	The position change error signal (PCE) turned on.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscGetDrivingPointNumber, sscChangeAutoSpeed, sscChangeAutoAccTime, sscChangeAutoDecTime

4.10.3 sscChangeLinearPosition

The linear interpolation position will be changed.

```
int  sscChangeLinearPosition (
   int  board_id,
   int  channel,
   int  axnum,
   int  grpnum,
   int  pnt_num,
   long  *pPosition
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Primary axis number (1 to 32 MC200 /1 to 64 MC300)

grpnum [in]

Interpolation group number (0 to 8 MC200 /0 to 16 MC300)

0 : Changeable interpolation group enabled

1 to 16 : Changeable interpolation group disabled

pnt_num [in]

Point number (0 to 319 MC200 /0 to 2047 MC300)

pPosition [in]

Pointer to 16-byte array (4 bytes × 4) which stores the position data [command units] after change

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_MISMATCH_DRIVE_MODE	The operation mode is other than the "Linear interpolation MC200 / interpolation operation MC300 ".
SSC_FUNC_ERR_STS_BIT_IPCH_ON	Changeable interpolation group setting is enabled. Set 0 to interpolation group number.
SSC_FUNC_ERR_STS_BIT_IPCH_OFF	Changeable interpolation group setting is disabled. Set 1 to 8 MC200 / 1 to 16 MC300 to interpolation group number.
SSC_FUNC_ERR_NOT_LIP_DRIVING	Linear interpolation is not in operation.
SSC_FUNC_ERR_POINT_NUMBER_OVER	Designated point number + point number offset value exceeded the point table range.
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	A system error (E□□□h) occurred. Get the system status code with the sscGetSystemStatusCode function and remove the cause.
SSC_FUNC_ERR_TIMEOUT_0□ □ = 1 to 2: Timeout location	During the confirmation of response after executing the command to the position board, the timeout time (1 second) has elapsed.
SSC_FUNC_ERR_STS_BIT_PCE	The position change error signal (PCE) turned on.

Point

• Set the axis number in ascending order of the linear interpolation target axes for the changed position data array. (In the linear interpolation of axis 1 and 3, set the changed position data of axis 1 to pPosition[0] and the changed position data of axis 3 to pPosition[1]. pPosition[2] and pPosition[3] are not used.)

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.90	A9	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscGetDrivingPointNumber, sscChangeAutoSpeed, sscChangeAutoAccTime, sscChangeAutoDecTime

4.10.4 sscChangeManualSpeed

The speed will be changed in the JOG operation or incremental feed.
The change speed signal (SCHG) will be turned on, and the completion of preparation for changing speed signal (SCF) or the speed change error signal (SCE) will be confirmed to be on.

```
int  sscChangeManualSpeed (
   int  board_id,
   int  channel,
   int  axnum,
   long  speed
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32 MC200 /1 to 64 MC300)

speed [in]

Manual feed speed after change [speed unit] (0 to 2147483647)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_MISMATCH_DRIVE_MODE	The operation mode is other than the "JOG operation" or "Incremental feed".
SSC_FUNC_ERR_TIMEOUT_0□ □ = 1 to 2: Timeout location	During the confirmation of response after executing the command to the position board, the timeout time (1 second) has elapsed.
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	A system error (E□□□h) occurred. Get the system status code with the sscGetSystemStatusCode function and remove the cause.
SSC_FUNC_ERR_STS_BIT_SCE	The speed change error signal (SCE) turned on.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscChangeManualPosition, sscChangeManualAccTime, sscChangeManualDecTime

4.10.5 sscChangeAutoSpeed

The automatic operation or linear interpolation MC200/interpolation operation MC300 speed will be changed. The change speed signal (SCHG) will be turned on, and the completion of preparation for changing speed signal (SCF) or the speed change error signal (SCE) will be confirmed to be on.

```
int  sscChangeAutoSpeed (
   int  board_id,
   int  channel,
   int  axnum,
   int  pntnum,
   long  speed
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32 MC200/1 to 64 MC300)

pntnum [in]

Point number (0 to 319 MC200/0 to 2047 MC300)

speed [in]

Feed speed after change [speed unit] (0 to 2147483647)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_MISMATCH_DRIVE_MODE	The operation mode is other than the "Automatic operation" or "Linear interpolation MC200/inteprolation operation MC300".
SSC_FUNC_ERR_POINT_NUMBER_OVER	Designated point number + point number offset value exceeded the point table range.
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	A system error (E□□□h) occurred. Get the system status code with the sscGetSystemStatusCode function and remove the cause.
SSC_FUNC_ERR_TIMEOUT_0□ □ = 1 to 2: Timeout location	During the confirmation of response after executing the command to the position board, the timeout time (1 second) has elapsed.
SSC_FUNC_ERR_STS_BIT_SCE	The speed change error signal (SCE) turned on.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscGetDrivingPointNumber, sscChangeAutoPosition, sscChangeAutoAccTime, sscChangeAutoDecTime

4.10.6 sscChangeManualAccTime

The JOG operation or incremental feed acceleration time constant will be changed.
The change acceleration time constant signal (TACHG) will be turned on, and the completion of preparation for changing acceleration time constant signal (TACF) or the acceleration time constant change error signal (TACE) will be confirmed to be on.

```
int  sscChangeManualAccTime (
   int  board_id,
   int  channel,
   int  axnum,
   short  acctime
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32 MC200 /1 to 64 MC300)

acctime [in]

Manual feed speed acceleration time constant after change [ms] (0 to 20000)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_MISMATCH_DRIVE_MODE	The operation mode is other than the "JOG operation" or "Incremental feed".
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	A system error (E□□□h) occurred. Get the system status code with the sscGetSystemStatusCode function and remove the cause.
SSC_FUNC_ERR_TIMEOUT_0□ □ = 1 to 2: Timeout location	During the confirmation of response after executing the command to the position board, the timeout time (1 second) has elapsed.
SSC_FUNC_ERR_STS_BIT_TACE	The acceleration time constant change error signal (TACE) turned on.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscChangeManualPosition, sscChangeManualSpeed, sscChangeManualDecTime

4.10.7 sscChangeAutoAccTime

The automatic operation or linear interpolation MC200/interpolation operation MC300 acceleration time constant will be changed.
The change acceleration time constant signal (TACHG) will be turned on, and the completion of preparation for changing acceleration time constant signal (TACF) or the acceleration time constant change error signal (TACE) will be confirmed to be on.

```
int  sscChangeAutoAccTime (
   int  board_id,
   int  channel,
   int  axnum,
   int  pntnum,
   short  acctime
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32 MC200/1 to 64 MC300)

pntnum [in]

Point number (0 to 319 MC200/0 to 2047 MC300)

acctime [in]

Acceleration time constant after change [ms] (0 to 20000)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_MISMATCH_DRIVE_MODE	The operation mode is other than the "Automatic operation" or "Linear interpolation MC200/interpolation operation MC300".
SSC_FUNC_ERR_POINT_NUMBER_OVER	Designated point number + point number offset value exceeded the point table range.
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	A system error (E□□□h) occurred. Get the system status code with the sscGetSystemStatusCode function and remove the cause.
SSC_FUNC_ERR_TIMEOUT_0□ □ = 1 to 2: Timeout location	During the confirmation of response after executing the command to the position board, the timeout time (1 second) has elapsed.
SSC_FUNC_ERR_STS_BIT_TACE	The acceleration time constant change error signal (TACE) turned on.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscGetDrivingPointNumber, sscChangeAutoPosition, sscChangeAutoSpeed, sscChangeAutoDecTime

4.10.8 sscChangeManualDecTime

The JOG operation or incremental feed deceleration time constant will be changed.
The change deceleration time constant signal (TDCHG) will be turned on, and the completion of preparation for changing deceleration time constant signal (TDCF) or the deceleration time constant change error signal (TDCE) will be confirmed to be on.

```
int sscChangeManualDecTime (
  int board_id,
  int channel,
  int axnum,
  short dectime
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32 MC200 /1 to 64 MC300)

dectime [in]

Manual feed speed deceleration time constant after change [ms] (0 to 20000)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_MISMATCH_DRIVE_MODE	The operation mode is other than the "JOG operation" or "Incremental feed".
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	A system error (E□□□h) occurred. Get the system status code with the sscGetSystemStatusCode function and remove the cause.
SSC_FUNC_ERR_TIMEOUT_0□ □ = 1 to 2: Timeout location	During the confirmation of response after executing the command to the position board, the timeout time (1 second) has elapsed.
SSC_FUNC_ERR_STS_BIT_TDCE	The deceleration time constant change error signal (TDCE) turned on.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscChangeManualPosition, sscChangeManualSpeed, sscChangeManualAccTime

4.10.9 sscChangeAutoDecTime

The automatic operation or linear interpolation MC200 /interpolation operation MC300 deceleration time constant will be changed.
The change deceleration time constant signal (TDCHG) will be turned on, and the completion of preparation for changing deceleration time constant signal (TDCF) or the deceleration time constant change error signal (TDCE) will be confirmed to be on.

```
int  sscChangeAutoDecTime (
   int  board_id,
   int  channel,
   int  axnum,
   int  pntnum,
   short  dectime
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32 MC200 /1 to 64 MC300)

pntnum [in]

Point number (0 to 319 MC200 /0 to 2047 MC300)

dectime [in]

Deceleration time constant after change [ms] (0 to 20000)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_MISMATCH_DRIVE_MODE	The operation mode is other than the "Automatic operation" or "Linear interpolation" MC200 /interpolation operation MC300 .
SSC_FUNC_ERR_POINT_NUMBER_OVER	Designated point number + point number offset value exceeded the point table range.
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	A system error (E□□□h) occurred. Get the system status code with the sscGetSystemStatusCode function and remove the cause.
SSC_FUNC_ERR_TIMEOUT_0□ □ = 1 to 2: Timeout location	During the confirmation of response after executing the command to the position board, the timeout time (1 second) has elapsed.
SSC_FUNC_ERR_STS_BIT_TDCE	The deceleration time constant change error signal (TDCE) turned on.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscGetDrivingPointNumber, sscChangeAutoPosition, sscChangeAutoSpeed, sscChangeAutoAccTime

4.11 Alarm functions

4.11.1 sscGetAlarm

The alarm number and the specific alarm number will be got.

```
int  sscGetAlarm (
   int  board_id,
   int  channel,
   int  axnum,
   int  alarm_type,
   unsigned short  *code,
   unsigned short  *detail_code
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (-4 to 32 MC200 /1 to 64 MC300)

0: System alarm

1 to 64: Axis alarm

-16 to -1: Station alarm (-1: Station 1, -2: Station 2, -3: Station 3, -4: Station 4 • • • • • -15: Station 15, -16: Station 16)

alarm_type [in]

Alarm type

Value	Description
SSC_ALARM_SYSTEM	System alarm
SSC_ALARM_SERVO	Servo alarm
SSC_ALARM_OPERATION	Operation alarm
SSC_ALARM_UNIT	RIO module alarm
SSC_ALARM_UNIT_CTRL	RIO control alarm

code [out]

Pointer to 2-byte variable (2 bytes × 1) which stores the alarm number variable

detail_code [out]

Pointer to 2-byte variable (2 bytes × 1) which stores the specific alarm number variable

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_ARGUMENT_MISMATCH	The axis number and the alarm type do not correspond. (Example: When "0" is set to the axis number and "SSC_ALARM_OPERATION" is set to the alarm type, etc.)

Point

• If an alarm has not occurred, 0 will be got as the alarm number and the specific alarm number.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.80	A8	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscResetAlarm

4.11.2 sscResetAlarm

The alarm will be reset.
After the alarm reset signal is turned on and the alarm signal is confirmed to be OFF, the alarm reset signal will be turned off.

```
int  sscResetAlarm (
   int  board_id,
   int  channel,
   int  axnum,
   int  alarm_type,
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (-4 to 32 MC200 /1 to 64 MC300)

0: System alarm

1 to 64: Axis alarm

-16 to -1: Station alarm (-1: Station 1, -2: Station 2, -3: Station 3, -4: Station 4 • • • • • -15: Station 15, -16: Station 16)

alarm_type [in]

Alarm type

Value	Description
SSC_ALARM_SYSTEM	System alarm
SSC_ALARM_SERVO	Servo alarm
SSC_ALARM_OPERATION	Operation alarm
SSC_ALARM_UNIT	RIO module alarm
SSC_ALARM_UNIT_CTRL	RIO control alarm

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0☐ ☐ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	A system error (E☐☐☐h) occurred. Get the system status code with the sscGetSystemStatusCode function and remove the cause.
SSC_FUNC_ERR_TIMEOUT_0☐ ☐ = 1 to 2: Timeout location	An alarm which cannot be reset occurred. Take proper countermeasures after confirming the cause of the alarm.
SSC_FUNC_ERR_ARGUMENT_MISMATCH	The axis number and the alarm type do not correspond. (Example: When "0" is set to the axis number and "SSC_ALARM_OPERATION" is set to the alarm type, etc.)

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.80	A8	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

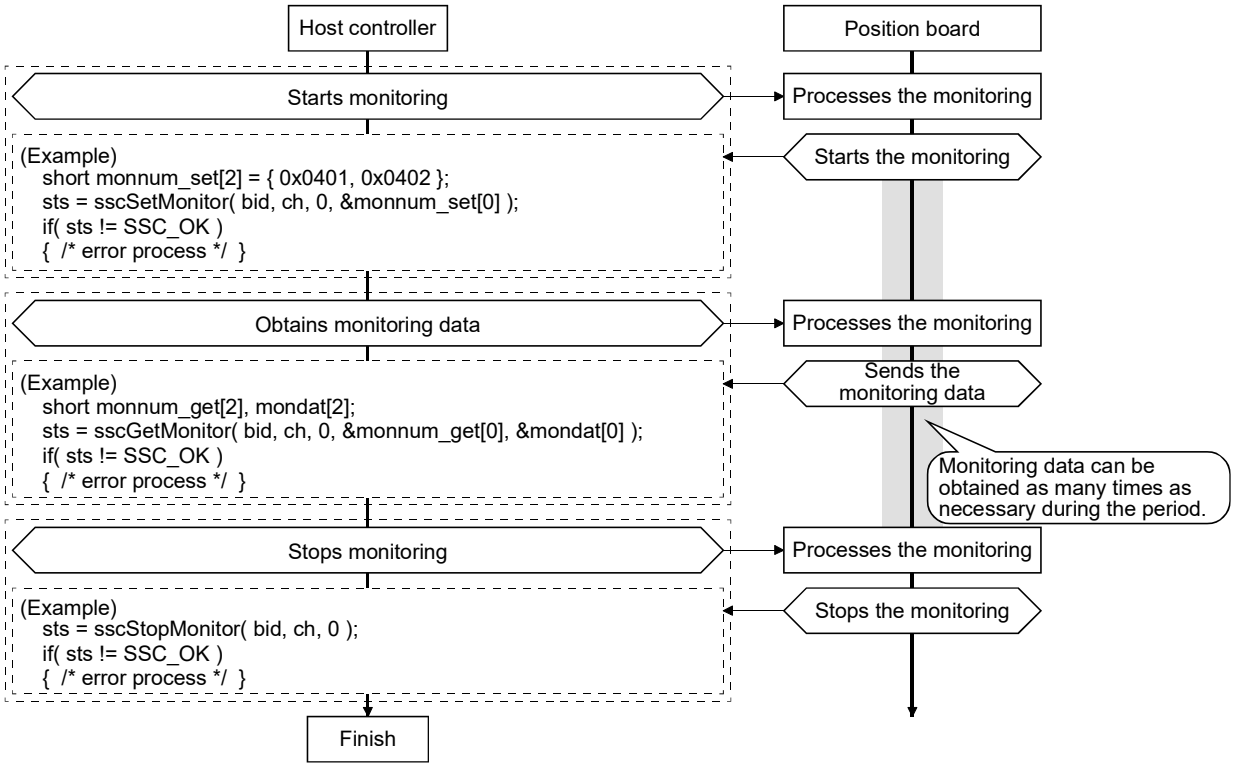
Reference

sscGetAlarm

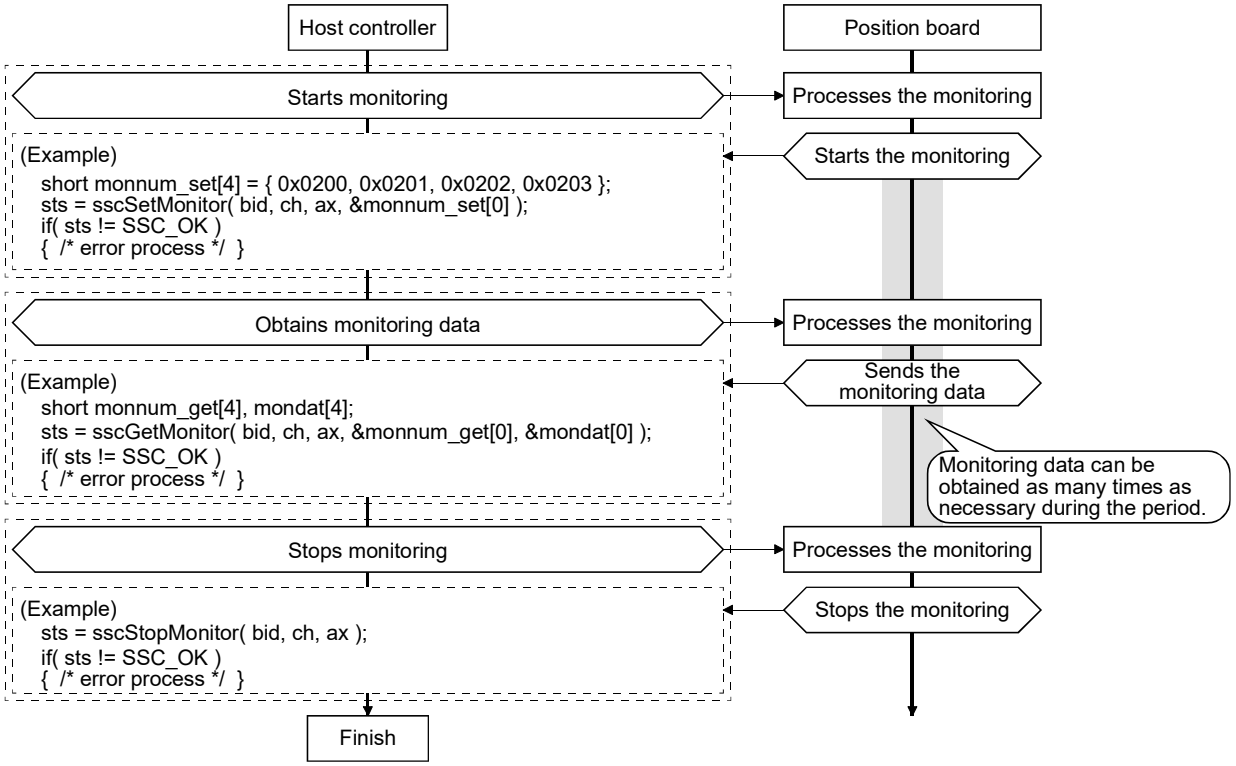
4.12 General monitor functions

(1) Processing procedure

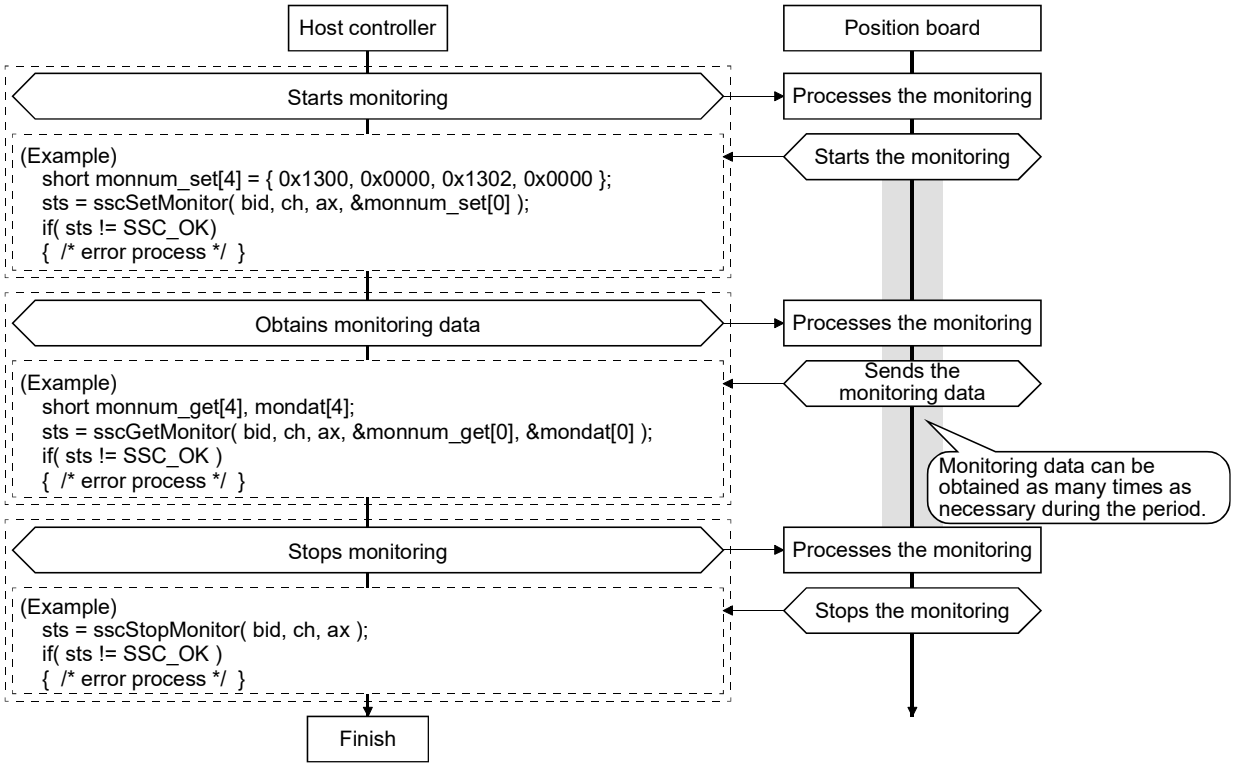
(a) Example of general monitor processing procedure for obtaining system information



(b) Example of general monitor processing procedure for obtaining servo information/operation information



(c) An example of general monitor processing procedure for obtaining operation information (double word) is below.



4.12.1 sscSetMonitor

The monitor will be started.
After setting 4 monitor numbers during axis monitoring and 2 monitor numbers during system monitoring, monitoring will begin by turning the monitor command signal (MON) ON, and the monitor output signal (MOUT) will be confirmed to be on.
If the monitor output signal (MOUT) is already on, monitoring will restart after the monitor command signal (MON) is OFF.

```
int  sscSetMonitor (
   int  board_id,
   int  channel,
   int  axnum,
   short  *monnum
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (-4 to 32 MC200 /-16 to 64 MC300)

0: System monitor

1 to 64: Axis monitor

-16 to -1: Station monitor (-1: Station 1, -2: Station 2, -3: Station 3, -4: Station 4 • • • • • -15: Station 15, -16: Station 16)

monnum [in]

Pointer to 8-byte array (2 bytes × 4) which stores the monitor numbers (for 4 numbers)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0☐ ☐ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_STS_BIT_MER☐ ☐ = 1 to 4: Array number of the monitor numbers (for 4 numbers)	A monitor number error occurred. • A value outside the range is set in the monitor number. • The axis number and the monitor number do not correspond. (Example: When "System monitor" is set to the axis number and "Axis monitor" is set to the monitor number, etc.)
SSC_FUNC_ERR_STS_BIT_MESV	The servo information was set as a monitor number when a servo amplifier was not connected.
SSC_FUNC_ERR_TIMEOUT_0☐ ☐ = 1 to 2: Timeout location	During the confirmation of response after executing the command to the position board, the timeout time (1 second) has elapsed.

Point

- When monitoring operating information (double word), monitor numbers 1 and 3 should be used, and monitor numbers 2 and 4 should be set to 0.
- Set 0 for the system monitor since the monitor number 3 and 4 are not used.
- Monitor numbers which will not be used should be set to 0.
- Depending on the control status of the position board, it takes a control cycle to several ms to start monitor because of the time it takes to confirm the response of the position board.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.80	A8	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscStopMonitor, sscGetMonitor

4.12.2 sscStopMonitor

The monitor will be stopped.
The monitor command signal (MON) will be turned off, and the monitor output signal (MOUT) will be confirmed to be OFF.

```
int  sscStopMonitor (
   int  board_id,
   int  channel,
   int  axnum
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (-4 to 32 MC200 /-16 to 64 MC300)

0: System monitor

1 to 64: Axis monitor

-16 to -1: Station monitor (-1: Station 1, -2: Station 2, -3: Station 3, -4: Station 4 • • • • • -15: Station 15, -16: Station 16)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0☐ ☐ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_ALREADY_MONITOR_STOP	The monitor has already stopped.
SSC_FUNC_ERR_TIMEOUT_01	During the confirmation of response after executing the command to the position board, the timeout time (1 second) has elapsed.

Point

• Depending on the control status of the position board, it takes a control cycle to several ms to stop monitor because of the time it takes to confirm the response of the position board.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2☐☐	Ver.1.80	A8	mc2xxstd.h
MR-MC3☐☐	Ver.1.00	A0	mc3xxstd.h

Reference

sscSetMonitor

4.12.3 sscGetMonitor

Monitor data will be got.

```
int  sscGetMonitor (
   int  board_id,
   int  channel,
   int  axnum,
   short  *monnum,
   short  *mondata
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (-4 to 32 MC200 /-16 to 64 MC300)

0: System monitor

1 to 64: Axis monitor

-16 to -1: Station monitor (-1: Station 1, -2: Station 2, -3: Station 3, -4: Station 4 • • • • • -15: Station 15, -16: Station 16)

monnum [out]

Pointer to 8-byte array (2 bytes × 4) which stores the monitor numbers

mondata [out]

Pointer to 8-byte array (2 bytes × 4) which stores the monitor data

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_NOT_START_MONITOR	The monitor has not been started.

Point

- For the system monitor, the third and fourth monitor number and monitor data are not stored.
- To obtain a current command position, current feedback position, external signal status, moving speed, feedback moving speed, or electrical current feedback, use high speed monitor functions to obtain the data quickly.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.80	A8	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscSetMonitor

4.13 High speed monitor functions

4.13.1 sscGetCurrentCmdPositionFast

The current command position will be got.

```
int  sscGetCurrentCmdPositionFast (
   int  board_id,
   int  channel,
   int  axnum,
   long  *position
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32 MC200 /1 to 64 MC300)

position [out]

Pointer to 4-byte variable (4 bytes × 1) which stores the current command position [command units]

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0☐ ☐ = 1 to 9: Argument location	The argument is outside the set range.

Point

• The position board "High speed monitor function" is used to get the current command position.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2☐☐	Ver.1.00	A0	mc2xxstd.h
MR-MC3☐☐	Ver.1.00	A0	mc3xxstd.h

Reference

sscGetCurrentFbPositionFast

4.13.2 sscGetCurrentFbPositionFast

The current feedback position will be got.

```
int  sscGetCurrentFbPositionFast (
   int  board_id,
   int  channel,
   int  axnum,
   long  *position
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32 MC200 /1 to 64 MC300)

position [out]

Pointer to 4-byte (4 bytes × 1) which stores the current feedback position variable [command units]

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.

Point

• The position board "High speed monitor function" is used to get the current feedback position.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscGetCurrentCmdPositionFast

4.13.3 sscGetIoStatusFast

The external signal (LSP, LSN, DOG signal) status will be got.

```
int  sscGetIoStatusFast (
   int  board_id,
   int  channel,
   int  axnum,
   short  *din
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32 MC200 /1 to 64 MC300)

din [out]

Pointer to 2-byte variable (2 bytes × 1) which stores the external signal status

The got data is set in the logical sum of each value.

Value	Description
SSC_BIT_LSP	+ side limit switch signal (LSP) is ON
SSC_BIT_LSN	− side limit switch signal (LSN) is ON
SSC_BIT_DOG	Proximity dog input signal (DOG) is ON

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.

Point

• The position board "High speed monitor function" is used to get the external signal status.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

None.

4.13.4 sscGetCmdSpeedFast

The moving speed will be got.

```
int  sscGetCmdSpeedFast (
   int  board_id,
   int  channel,
   int  axnum,
   long  *speed
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32 MC200 /1 to 64 MC300)

speed [out]

Pointer to 4-byte variable (4 bytes × 1) which stores the moving speed [speed units]

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.

Point

• The position board "High speed monitor function" is used to get the moving speed.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscGetFbSpeedFast

4.13.5 sscGetFbSpeedFast

The feedback moving speed will be got.

```
int  sscGetFbSpeedFast (
   int  board_id,
   int  channel,
   int  axnum,
   long  *speed
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32 MC200 /1 to 64 MC300)

speed [out]

Pointer to 4-byte variable (4 bytes × 1) which stores the feedback moving speed [speed units]

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.

Point

• The position board "High speed monitor function" is used to get the feedback moving speed.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscGetCmdSpeedFast

4.13.6 sscGetCurrentFbFast

The current feedback will be got.

```
int  sscGetCurrentFbFast (
   int  board_id,
   int  channel,
   int  axnum,
   short  *currentFb
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32 MC200 /1 to 64 MC300)

currentFb [out]

Pointer to 2-byte variable (2 bytes × 1) which stores the current feedback [0.1%]

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.

Point

• The position board "High speed monitor function" is used to get the current feedback.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

None.

4.13.7 sscGetPositionDroopFast

The position droop will be got.

```
int  sscGetPositionDroopFast (
   int  board_id,
   int  channel,
   int  axnum,
   long  *position_dp
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32 MC200 /1 to 64 MC300)

position_dp [out]

Pointer to 4-byte variable (4 bytes × 1) which stores the position droop [pulse]

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.

Point

- The position board "High speed monitor function" is used to get the position droop.
- This function is supported in interface mode only.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.60	A4	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

None.

4.14 User watchdog functions

4.14.1 sscWdEnable

The user watchdog function will be enabled.

```
int  sscWdEnable (
   int  board_id,
   int  channel,
   unsigned short  wdcnt
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

wdcnt [in]

Watchdog timer start counter

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_ALREADY_ENABLE_WDT	The user watchdog function has been already valid.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscWdDisable

4.14.2 sscWdDisable

The user watchdog function will be disabled.

```
int  sscWdDisable (
   int  board_id,
   int  channel
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_ALREADY_DISABLE_WDT	The user watchdog function has been already invalid.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscWdEnable

4.14.3 sscChangeWdCounter

The watchdog counter will be updated (+1).

```
int  sscChangeWdCounter (
   int  board_id,
   int  channel
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscWdEnable

4.15 Other axes start functions

4.15.1 sscSetOtherAxisStartData

The other axes start data will be set.

```
int  sscSetOtherAxisStartData (
   int  board_id,
   int  channel,
   int  oas_num,
   OAS_DATA  *pOasData
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

oas_num [in]

Other axes start table number (1 to 32 MC200 /1 to 64 MC300)

pOasData [in]

Pointer to 104-byte structure (104 bytes × 1) MC200 /128-byte structure (128 bytes × 1) MC300 which stores the other axes start data

Refer to "5.2 OAS_DATA structure" for the other axes start data structure.

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.

Point

• A check of the set other axes start data contents will not be performed.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscGetOtherAxisStartData

4.15.2 sscGetOtherAxisStartData

The other axes start table data will be got.

```
int  sscGetOtherAxisStartData (
   int  board_id,
   int  channel,
   int  oas_num,
   OAS_DATA  *pOasData
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

oas_num [in]

Other axes start table number (1 to 32 MC200 /1 to 64 MC300)

pOasData [out]

Pointer to 104-byte structure (104 bytes × 1) MC200 /128-byte structure (128 bytes × 1) MC300 which stores the other axes start data

Refer to "5.2 OAS_DATA structure" for the other axes start data structure.

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscSetOtherAxisStartData

4.15.3 sscOtherAxisStartAbortOn

The other axes start cancel signal (OSSTP□) will be turned on and the other axes start will be canceled.

```
int sscOtherAxisStartAbortOn (
  int board_id,
  int channel,
  int oas_num
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

oas_num [in]

Other axes start table number (1 to 32 MC200 /1 to 64 MC300)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_TIMEOUT_01	During the confirmation of response after executing the command to the position board, the timeout time (1 second) has elapsed.

Point

- The response is not confirmed after the other axes start cancel signal (OSSTP□) is turned on.
- To confirm the other axes start status, call the sscGetOtherAxisStartStatus function.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscOtherAxisStartAbortOff, sscGetOtherAxisStartStatus

4.15.4 sscOtherAxisStartAbortOff

The other axes start cancel signal (OSSTP☐) will be turned off and the other axes start will be canceled.

```
int sscOtherAxisStartAbortOff (
  int board_id,
  int channel,
  int oas_num
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

oas_num [in]

Other axes start table number (1 to 32 MC200 /1 to 64 MC300)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0☐ ☐ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_TIMEOUT_01	During the confirmation of response after executing the command to the position board, the timeout time (1 second) has elapsed.

Point

• To confirm the other axes start status, call the sscGetOtherAxisStartStatus function.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2☐☐	Ver.1.00	A0	mc2xxstd.h
MR-MC3☐☐	Ver.1.00	A0	mc3xxstd.h

Reference

sscOtherAxisStartAbortOn, sscGetOtherAxisStartStatus

4.15.5 sscGetOtherAxisStartStatus

The other axes start status (other axes start status bit) will be got.

```
int  sscGetOtherAxisStartStatus (
   int  board_id,
   int  channel,
   int  oas_num,
   short  *status
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

oas_num [in]

Other axes start table number (1 to 32 MC200 /1 to 64 MC300)

status [out]

Pointer to 2-byte variable (2 bytes × 1) which stores the other axes start status bit

The got data is set in the logical sum of each value.

Value	Description
SSC_BIT_OSOP	Other axes start notice
SSC_BIT_OSFIN	Other axes start complete
SSC_BIT_OSERR	Other axes start incomplete

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

None.

4.16 Pass position interrupt functions

4.16.1 sscSetIntPassPositionData

The pass position interrupt condition data will be set.

```
int  sscSetIntPassPositionData (
   int  board_id,
   int  channel,
   int  pass_num,
   unsigned long  pass_option,
   long  pass_data
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

pass_num [in]

Pass position condition number (1 to 64 MC200 /1 to 128 MC300)

pass_option [in]

Pass position option

Set data in the logical sum of each value.

Value	Description	
SSC_PASS_DIR_PLUS	Pass direction	+ direction pass position interrupt output
SSC_PASS_DIR_MINUS		- direction pass position interrupt output
SSC_PASS_JUDGE_CMD_POS	Judgment condition	Current position
SSC_PASS_JUDGE_FB_POS		Feedback position

pass_data [in]

Pass position data [command unit] (-2147483648 to 2147483647)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.

Point

• Only the judgment condition for the pass position condition start number is valid for the pass position option.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.02	A1	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscCheckIntPassPositionData

4.16.2 sscCheckIntPassPositionData

The pass position interrupt condition data will be got.

```
int  sscCheckIntPassPositionData (
   int  board_id,
   int  channel,
   int  pass_num,
   unsigned long  *pass_option,
   long  *pass_position
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

pass_num [in]

Pass position condition number (1 to 64 MC200 /1 to 128 MC300)

pass_option [out]

Pointer to 4-byte variable (4 bytes × 1) which stores the pass position option

pass_position [out]

Pointer to 4-byte variable (4 bytes × 1) which stores the pass position data [command units]

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.02	A1	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscSetIntPassPositionData

4.16.3 sscSetStartingPassNumber

The pass position condition start and end numbers will be set.

```
int  sscSetStartingPassNumber (
   int  board_id,
   int  channel,
   int  axnum,
   int  pass_start,
   int  pass_end
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32 MC200 /1 to 64 MC300)

pass_start [in]

Pass position condition start number (1 to 64 MC200 /1 to 128 MC300)

pass_end [in]

Pass position condition end number (1 to 64 MC200 /1 to 128 MC300)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.02	A1	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

None.

4.16.4 sscGetExecutingPassNumber

The running pass position condition number will be got.

```
int  sscGetExecutingPassNumber (
   int  board_id,
   int  channel,
   int  axnum,
   short  *executing_pass
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32 MC200 /1 to 64 MC300)

executing_pass [out]

Pointer to 2-byte variable (2 bytes × 1) which stores the running pass position condition number

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.

Point

- The end number of the pass position condition will be stored in the executing pass position condition number after the pass position condition completion.
- The canceled pass position condition number will be stored in the executing pass position condition number after the pass position condition incompletion.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.02	A1	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

None.

4.17 Sampling functions

4.17.1 sscStartSampling

Sampling will be started.
The sampling start signal (SMPS) will be turned on.

```
int  sscStartSampling (
   int  board_id,
   int  channel
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0☐ ☐ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_ALREADY_START_SAMPLING	The sampling start signal (SMPS) is ON. Stop the sampling with the sscStopSampling function.

Point

• The response is not confirmed after the sampling start signal (SMPS) is turned on.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2☐☐	Ver.1.00	A0	mc2xxstd.h
MR-MC3☐☐	Ver.1.00	A0	mc3xxstd.h

Reference

sscStopSampling, sscSetSamplingParameter, sscGetSamplingStatus

4.17.2 sscStopSampling

Sampling will be stopped.
The sampling start signal (SMPS) will be turned off, and the function will wait until all sampling status signals [waiting for sampling trigger (SMPW), sampling is being performed (SMPO), sampling is complete (SMPF), and sampling error (SMPE)] are turned off.

```
int  sscStopSampling (
   int  board_id,
   int  channel
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_TIMEOUT_01	During the confirmation of response after executing the command to the position board, the timeout time (1 second) has elapsed.
SSC_FUNC_ERR_ALREADY_STOP_SAMPLING	The sampling has already stopped.

Point

• Stop the sampling with this function after the sampling is complete signal (SMPF) or the sampling error signal (SMPE) turns on.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscStartSampling

4.17.3 sscSetSamplingParameter

The sampling parameters will be written.

```
int sscSetSamplingParameter (
  int board_id,
  int channel,
  short prm_num,
  long prm_data
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

prm_num [in]

Sampling setting write number

prm_data [in]

Sampling setting write data

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_TIMEOUT_0□ □ = 1 to 2: Timeout location	During the confirmation of response after executing the command to the position board, the timeout time (1 second) has elapsed.
SSC_FUNC_ERR_STS_BIT_SWEN	A value outside the range is set in the sampling setting write number.
SSC_FUNC_ERR_STS_BIT_SWED	A value outside the range is set in the sampling setting write data.
SSC_FUNC_ERR_MISMATCH_SMP_PARAM_WRITE_NUM	The command and the status of the sampling setting write number do not correspond.
SSC_FUNC_ERR_MISMATCH_SMP_PARAM_WRITE_DATA	The command and the status of the sampling write data do not correspond.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscGetSamplingParameter

4.17.4 sscGetSamplingParameter

The sampling parameters will be read.

```
int  sscGetSamplingParameter (
   int  board_id,
   int  channel,
   short  prm_num,
   long  *prm_data
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

prm_num [in]

Sampling setting read number

prm_data [out]

Pointer to 4-byte variable (4 bytes × 1) which stores the sampling setting read data

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_TIMEOUT_0□ □ = 1 to 2: Timeout location	During the confirmation of response after executing the command to the position board, the timeout time (1 second) has elapsed.
SSC_FUNC_ERR_STS_BIT_SREN	A value outside the range is set in the sampling setting read number.
SSC_FUNC_ERR_MISMATCH_SMP_PARAM_READ_NUM	The command and the status of the sampling setting read number do not correspond.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscSetSamplingParameter

4.17.5 sscGetSamplingError

The sampling error information will be got.

```
int  sscGetSamplingError (
   int  board_id,
   int  channel,
   SMP_ERR  *pSmpErr
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

pSmpErr [out]

Pointer to 32-byte structure (32 bytes × 1) MC200 /48-byte structure (48 bytes × 1) MC300 which stores the sampling error information

Refer to "5.4 SMP_ERR structure" for the sampling error information structure.

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscGetSamplingStatus

4.17.6 sscGetSamplingStatus

The sampling status (sampling status bit and sampling complete page number) will be got.

```
int sscGetSamplingStatus (
  int board_id,
  int channel,
  char *status,
  short *fin_page
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

status [out]

Pointer to 1-byte variable (1 byte × 1) which stores the sampling status bit

Value	Description
SSC_BIT_SMPW	Waiting for sampling trigger
SSC_BIT_SMPO	Sampling is being performed
SSC_BIT_SMPF	Sampling is complete
SSC_BIT_SMPE	Sampling error

fin_page [out]

Pointer to 2-byte variable (2 byte × 1) which stores the sampling completion page number

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscGetSamplingError, sscGetSamplingData

4.17.7 sscGetSamplingData

The sampling read enabled points and the sampling data (for 32 points MC200 /128 points MC300) will be got. The sampling read enabled points are not confirmed.

```
int  sscGetSamplingData (
   int  board_id,
   int  channel,
   int  page_num,
   short  *valid_num,
   SMP_DATA  *pSmpData
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

page_num [in]

Sampling read page number (0 to 256 MC200 /0 to 512 MC300)

valid_num [out]

Pointer to the sampling read enabled points

pSmpData [out]

Pointer to 4224-byte structure (132 bytes × 32) MC200 /16896-byte structure (132 bytes × 128) MC300 which stores the sampling data

Refer to "5.5 SMP_DATA structure" for the sampling data information structure.

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_TIMEOUT_01	During the confirmation of response after executing the command to the position board, the timeout time (1 second) has elapsed.

Point

- Set the sampling read page number to 0 in this function before starting sampling.
- For MR-MC3□□ API Ver. 1.00, the sampling data from 32769 points (page number 257) and after cannot be gotten.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscStartSampling, sscGetSamplingStatus

4.18 Log functions

4.18.1 sscStartLog

Logging will be started.
Logging will be started by turning the log command signal (LOGC) on, and the log operation being performed signal (LOGO) will be confirmed to be on.

```
int  sscStartLog (
   int  board_id,
   int  channel
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_TIMEOUT_01	During the confirmation of response after executing the command to the position board, the timeout time (1 second) has elapsed.
SSC_FUNC_ERR_ALREADY_START_LOG	The log command (LOGC) is ON. Stop the logging with the sscStopLog function.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscStopLog

4.18.2 sscStopLog

Logging will be stopped.
Logging will be stopped using log command signal (LOGC) OFF, and the log operation being performed signal (LOGO) will be confirmed to be OFF.

```
int  sscStopLog (
   int  board_id,
   int  channel
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_TIMEOUT_01	During the confirmation of response after executing the command to the position board, the timeout time (1 second) has elapsed.
SSC_FUNC_ERR_ALREADY_STOP_LOG	Logging has already been stopped.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscStartLog

4.18.3 sscCheckLogStatus

The log operation status will be got.

```
int  sscCheckLogStatus (
   int  board_id,
   int  channel,
   int  *status
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

status [out]

Pointer to 4-byte variable (4 bytes × 1) which stores the log operation status

Value	Description
SSC_LOGO_OFF	The log operation being performed signal (LOGO) is OFF.
SSC_LOGO_ON	The log operation being performed signal (LOGO) is ON.

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscStartLog

4.18.4 sscCheckLogEventNum

The number of valid log data events recorded in the log data will be got.

```
int sscCheckLogEventNum (
  int board_id,
  int channel,
  int *eventnum
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

eventnum [out]

Pointer to 4-byte variable (4 bytes × 1) which stores the number of valid log data events

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscStartLog

4.18.5 sscReadLogData

16 events of log data will be read.

```
int  sscReadLogData (
   int  board_id,
   int  channel,
   int  page_num,
   LOG_DATA  *pLogData
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

page_num [in]

Log data read page number (1 to 256)

pLogData [out]

Pointer to 256-byte structure (16 bytes × 16) which stores the log data

Refer to "5.6 LOG_DATA structure" for the log data information structure.

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_TIMEOUT_0□ □ = 1 to 2: Timeout location	During the confirmation of response after executing the command to the position board, the timeout time (1 second) has elapsed.
SSC_FUNC_ERR_NOW_LOGGING	The log operation being performed signal (LOGO) is ON.
SSC_FUNC_ERR_STS_BIT_LOGRE	The reading of log data error signal (LOGRE) is turned on.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscClearLogData

4.18.6 sscClearLogData

The log data will be cleared (initialized).

```
int sscClearLogData (
  int board_id,
  int channel
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_NOW_LOGGING	The log operation being performed signal (LOGO) is ON.
SSC_FUNC_ERR_TIMEOUT_0□ □ = 1 to 2: Timeout location	During the confirmation of response after executing the command to the position board, the timeout time (1 second) has elapsed.
SSC_FUNC_ERR_STS_BIT_LOGIE	The log data initialization error signal (LOGIE) is turned on.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscReadLogData

4.18.7 sscGetAlarmHistoryData

4 events of alarm history data will be read

```
int  sscGetAlarmHistoryData (
   int  board_id,
   int  channel,
   int  page_num,
   ALH_DATA  *pAlhData
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

page_num [in]

Alarm history data read page number (1 to 512)

pAlhData [out]

Pointer to 256-byte structure (64 bytes × 4) which stores the alarm history data

Refer to "5.7 ALH_DATA structure" for the alarm history data information structure.

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_UNSUPPORT_ALH	Alarm history function is not supported.
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_TIMEOUT_0□ □ = 1 to 2: Timeout location	During the confirmation of response after executing the command to the position board, the timeout time (1 second) has elapsed.
SSC_FUNC_ERR_STS_BIT_ALHRE	The alarm history read error signal (ALHRE) is turned on.

Point

• Check the got checksum with a user program.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.50	A3	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscClearAlarmHistoryData

4.18.8 sscCheckAlarmHistoryEventNum

The number of valid alarm history data events recorded in the alarm history data will be got.

```
int  sscCheckAlarmHistoryEventNum (
   int  board_id,
   int  channel,
   int  *eventnum
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

eventnum [out]

Pointer to 4-byte variable (4 bytes × 1) which stores the number of valid alarm history data events

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_UNSUPPORT_ALH	Alarm history function is not supported.
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.50	A3	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

None.

4.18.9 sscClearAlarmHistoryData

Clears (initializes) alarm history data.

```
int sscClearAlarmHistoryData (
  int board_id,
  int channel,
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_UNSUPPORT_ALH	Alarm history function is not supported.
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_TIMEOUT_0□ □ = 1 to 2: Timeout location	During the confirmation of response after executing the command to the position board, the timeout time (2 seconds MC200 /10 seconds MC300) has elapsed.
SSC_FUNC_ERR_STS_BIT_ALHIE	The alarm history initialization error signal (ALHIE) is turned on.

Point

• As there is a restriction on the number of times for writing to the position board flash ROM, keep calls of the alarm history initialize function to the minimum amount necessary.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.50	A3	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscGetAlarmHistoryData

4.19 Digital input/output functions

4.19.1 sscGetDigitalInputDataBit

The DI data of the designated digital input will be got in 1-point basis.

```
int  sscGetDigitalInputDataBit (
   int  board_id,
   int  channel,
   int  din_num,
   int  *din
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

din_num [in]

Digital input number (0 to 1023)

din [out]

Pointer to 4-byte variable (4 bytes × 1) which stores the DI data status of the digital input

Value	Description
SSC_BIT_OFF	Input signal OFF
SSC_BIT_ON	Input signal ON

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.02	A1	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscGetDigitalInputDataWord

4.19.2 sscGetDigitalInputDataWord

The DI data of the designated digital input will be got in 16-point basis.

```
int  sscGetDigitalInputDataWord (
   int  board_id,
   int  channel,
   int  din_word_num,
   unsigned short  *din
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

din_ word_num [in]

Digital input word number (0 to 63)

din [out]

Pointer to 2-byte variable (2 bytes × 1) which stores the DI data status of the digital input

(0: Output OFF 1: Output ON for each bit)

bit0 → DI_□□0,···bit15 → DI_□□F

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.02	A1	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscGetDigitalInputDataBit

4.19.3 sscSetDigitalOutputDataBit

The DO data of the designated digital output will be set in 1-point basis.

```
int sscSetDigitalOutputDataBit (
   int  board_id,
   int  channel,
   int  dout_num,
   int  dout
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

dout_num [in]

Digital output number (0 to 1023)

dout [in] (0 to 1)

DO data of the digital output

Value	Description
SSC_BIT_OFF	Output signal OFF
SSC_BIT_ON	Output signal ON

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0☐ ☐ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_TIMEOUT_0☐ ☐ = 1 to 5: Timeout location	During the confirmation of response after executing the command to the postion board, the timeout time (1 second) has elapsed.

Point

• This function sets the digital output with the exclusive control function of the position board.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2☐☐	Ver.1.02	A1	mc2xxstd.h
MR-MC3☐☐	Ver.1.00	A0	mc3xxstd.h

Reference

sscSetDigitalOutputDataWord, sscGetDigitalOutputDataBit

4.19.4 sscSetDigitalOutputDataWord

The DO data of the designated digital output will be set in 16-point basis.

```
int  sscSetDigitalOutputDataWord (
   int  board_id,
   int  channel,
   int  dout_word_num,
   unsigned short  dout
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

dout_word_num [in]

Digital output word number (0 to 63)

dout [in] (0000h to FFFFh)

DO data of the digital output

(0: Output OFF 1: Output ON for each bit)

bit0 → DO_□□0,··· bit15 → DO_□□F

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_TIMEOUT_0□ □ = 1 to 5: Timeout location	During the confirmation of response after executing the command to the postion board, the timeout time (1 second) has elapsed.

Point

• This function sets the digital output with the exclusive control function of the position board.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.02	A1	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscSetDigitalOutputDataBit, sscGetDigitalOutputDataWord

4.19.5 sscGetDigitalOutputDataBit

The DO data of the designated digital output will be got in 1-point basis.

```
int sscGetDigitalOutputDataBit (
  int  board_id,
  int  channel,
  int  dout_num,
  int  *dout
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

dout_num [in]

Digital output number (0 to 1023)

dout [out]

Pointer to 4-byte variable (4 bytes × 1) which stores the DO data status of the digital output

Value	Description
SSC_BIT_OFF	Output signal OFF
SSC_BIT_ON	Output signal ON

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.02	A1	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscSetDigitalOutputDataBit, sscGetDigitalOutputDataWord

4.19.6 sscGetDigitalOutputDataWord

The DO data of the designated digital output will be got in 16-point basis.

```
int  sscGetDigitalOutputDataWord (
   int  board_id,
   int  channel,
   int  dout_word_num,
   unsigned short  *dout
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

dout_word_num [in]

Digital output word number (0 to 63)

dout [out]

Pointer to 2-byte variable (2 bytes × 1) which stores the DO data status of the digital output

(0: Output OFF 1: Output ON for each bit)

bit0 → DO_□□0,··· bit15 → DO_□□F

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.02	A1	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscSetDigitalOutputDataWord, sscGetDigitalOutputDataBit

4.20 Mark detection functions

4.20.1 sscGetMarkDetectionData

Mark detection data will be got.

```
int  sscGetMarkDetectionData (
   int  board_id,
   int  channel,
   int  axnum,
   int  mark_num,
   int  *read_fin_num,
   int  *edge,
   int  *position
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32 MC200 /1 to 64 MC300)

mark_num [in]

Mark detection setting number (1 to 2)

read_fin_num [out]

Pointer to 4-byte variable (4 bytes × 1) which stores the read complete buffer number (1 to 255)

edge [out]

Pointer to 1-byte structure (1 byte × 1) which stores the mark detection edge data

Value	Description
SSC_MARK_EDGE_OFF	OFF edge
SSC_MARK_EDGE_ON	ON edge

position [out]

Pointer to 4-byte structure (4 bytes × 1) which stores the mark detection positioning data

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_MARK_DETECT_UNUSABLE	Mark detection function is disabled. • Mark detection function is not supported. • Mark detection function has been disabled by the settings.
SSC_FUNC_ERR_MARK_DETECT_UNDETECTED	There is no mark detection data that can be got. After checking that the mark detection count of the position board has been renewed, call the sscGetMarkDetectionData function.

Point

- After getting mark detection data, the read complete buffer number is renewed (+1). (For continuous detection mode, the read complete buffer number is set to 1 when the read complete buffer number exceeds 255. For ring buffer mode, the read complete buffer number is set to 1 when the number of continuous latch data storages is exceeded.)
- The same mark detection data can only be got once. Therefore it is necessary to hold the got data with a user program.
- Data is not got for arguments that designate a NULL pointer (read complete buffer number, mark detection edge data, mark detection positioning data).

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.60	A5	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscGetMarkDetectionCounter

4.20.2 sscGetMarkDetectionCounter

Mark detection counter will be got.

```
int sscGetMarkDetectionCounter (
  int board_id,
  int channel,
  int axnum,
  int mark_num,
  int *detected_counter
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32 MC200 /1 to 64 MC300)

mark_num [in]

Mark detection setting number (1 to 2)

detected_counter [out]

Pointer to 4-byte variable (4 bytes × 1) which stores the mark detection counter (1 to 255)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.60	A5	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscGetMarkDetectionData

4.20.3 sscClearMarkDetectionData

Mark detection data will be cleared (initialized).

```
int sscClearMarkDetectionData (
  int board_id,
  int channel,
  int axnum,
  int mark_num,
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32 MC200 /1 to 64 MC300)

mark_num [in]

Mark detection setting number (1 to 2)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0☐ ☐ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_TIMEOUT_0☐ ☐ = 1 to 2: Timeout location	During the confirmation of response after executing the command to the postion board, the timeout time (1 second) has elapsed.

Point

• The read complete buffer number is set to 0 after mark detection data is cleared.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2☐☐	Ver.1.60	A5	mc2xxstd.h
MR-MC3☐☐	Ver.1.00	A0	mc3xxstd.h

Reference

sscGetMarkDetectionData

4.21 Interface mode functions

4.21.1 ssclfmGetReadErrorCount

Read error counter will be got.

```
int ssclfmGetReadErrorCount (
   int  board_id,
   int  channel,
   short  *err_cnt
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

err_cnt [out]

Pointer to 2-byte variable (2 bytes × 1) which stores the read error counter

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.50	A3	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

None.

4.21.2 ssclfmSetHomePosition

Home position will be set, and after completion [home position multiple revolution data (parameter No.024D) and home position within 1 revolution position (parameter No.024E, parameter No.024F)] will be got.

```
int ssclfmSetHomePosition (
  int  board_id,
  int  channel,
  int  axnum,
  int  mode,
  short  *param
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32 MC200 /1 to 64 MC300)

mode [in]

In-position signal check mode

Value	Description
SSC_IFM_CHK_INP_WAIT	Waits until the in-position signal is ON.
SSC_IFM_CHK_INP_NOWAIT	Does not wait until the in-position signal is ON.

param [out]

Pointer to 6-byte array (2 bytes × 3) which stores the home position data

Array number	Description
0	Home position multiple revolution data (parameter No.024D)
1	Home position within 1 revolution position (parameter No.024E)
2	Home position within 1 revolution position (parameter No.024F)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	A system error (E□□□h) occurred. Get the system status code with the sscGetSystemStatusCode function and remove the cause.
SSC_FUNC_ERR_TIMEOUT_0□ □ = 1 to 3: Timeout location	During the confirmation of response after executing the command to the position board, the timeout time (10 seconds) has elapsed.
SSC_FUNC_ERR_IFM_INP_OFF	In-position signal is OFF. When setting "Does not wait until the in-position signal is ON" during in-position signal check mode, call the ssclfmSetHomePosition function when the in-position signal is ON.
SSC_FUNC_ERR_STS_BIT_ZSE	The home position set error (ZSE) occurred.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.50	A3	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

None.

4.21.3 sscIfmGetMaximumBufferNumber MC200

The maximum buffer number for position control mode will be got.

```
int sscIfmGetMaximumBufferNumber (
  int  board_id,
  int  channel,
  int  axnum,
  short  *bufnum
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32)

bufnum [out]

Pointer to 2-byte variable (2 bytes × 1) which stores the maximum buffer number

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.

Point

For "API Ver. 1.60 or later", use the sscIfmGetMaximumBufferNumberEx function.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.50	A3	mc2xxstd.h

Reference

None.

4.21.4 ssclfmGetMaximumBufferNumberEx

The maximum buffer number for the designated control mode will be got.

```
int ssclfmGetMaximumBufferNumberEx (
   int  board_id,
   int  channel,
   int  axnum,
   unsigned short  ctrl_mode,
   short  *bufnum
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32 MC200 /1 to 64 MC300)

ctrl_mode [in]

Control mode

Operation mode	Description
SSC_IFM_CTRL_MODE_POSITION	Position control mode
SSC_IFM_CTRL_MODE_SPEED	Speed control mode
SSC_IFM_CTRL_MODE_TORQUE	Torque control mode

bufnum [out]

Pointer to 2-byte variable (2 bytes × 1) which stores the maximum buffer number

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.60	A4	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

None.

4.21.5 sscIfmRenewLatestBuffer MC200

Latest command buffer number and data for position control mode will be renewed, and the renewed latest command buffer number will be got.

```
int  sscIfmRenewLatestBuffer (
   int  board_id,
   int  channel,
   int  axnum,
   long  bufdata,
   short  *bufnum
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32)

bufdata [in]

Renew data

bufnum [out]

Pointer to 2-byte variable (2 bytes × 1) which stores the latest command buffer number

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_IFM_CMD_BUF_FULL	There is no free space in the position command buffer. After checking that the position board transmit buffer number has been renewed, call the sscIfmRenewLatestBuffer function or sscIfmRenewLatestBufferEx function.

Point

- For "API Ver. 1.60 or later", use the sscIfmRenewLatestBufferEx function.
- For "API Ver. 1.61 or later", when a NULL pointer is designated as the latest command buffer number, data will not be got.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.50	A3	mc2xxstd.h

Reference

sscIfmCheckLatestBuffer, sscIfmGetTransmitBuffer

4.21.6 sscIfmRenewLatestBufferEx

Latest command buffer number and data for the designated control mode will be renewed, and the renewed latest command buffer number will be got.

```
int  sscIfmRenewLatestBufferEx (
   int  board_id,
   int  channel,
   int  axnum,
   unsigned short  ctrl_mode,
   long  bufdata,
   short  *bufnum
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32 MC200 /1 to 64 MC300)

ctrl_mode [in]

Control mode

Operation mode	Description
SSC_IFM_CTRL_MODE_POSITION	Position control mode
SSC_IFM_CTRL_MODE_SPEED	Speed control mode
SSC_IFM_CTRL_MODE_TORQUE	Torque control mode

bufdata [in]

Refresh data

bufnum [out]

Pointer to 2-byte variable (2 bytes × 1) which stores the latest command buffer number

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_IFM_CMD_BUF_FULL	There is no free space in the position command buffer. After checking that the position board transmit buffer number has been renewed, call the sscIfmRenewLatestBuffer function or sscIfmRenewLatestBufferEx function.

Point

• For "API Ver. 1.61 or later", when a NULL pointer is designated as the latest command buffer number, data will not be got.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.60	A4	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscIfmCheckLatestBufferEx, sscIfmGetTransmitBufferEx

4.21.7 sscIfmCheckLatestBuffer MC200

Latest command buffer number and latest command buffer data for position control mode will be got.

```
int  sscIfmCheckLatestBuffer (
   int  board_id,
   int  channel,
   int  axnum,
   short  *bufnum,
   long  *bufdata
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32)

bufnum [out]

Pointer to 2-byte variable (2 bytes × 1) which stores the latest command buffer number

bufdata [out]

Pointer to 2-byte variable (2 bytes × 1) which stores the latest command buffer data

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.

Point

- When a NULL pointer is designated as the latest command buffer data, only the latest command buffer number will be got.
- For "API Ver. 1.60 or later", use the sscIfmCheckLatestBufferEx function.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.50	A3	mc2xxstd.h

Reference

sscIfmGetTransmitBuffer

4.21.8 ssclfmCheckLatestBufferEx

Latest command buffer number and latest command buffer data for the designated control mode will be got.

```
int  ssclfmCheckLatestBufferEx (
   int  board_id,
   int  channel,
   int  axnum,
   unsigned short  ctrl_mode,
   short  *bufnum,
   long  *bufdata
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32 MC200 /1 to 64 MC300)

ctrl_mode [in]

Control mode

Operation mode	Description
SSC_IFM_CTRL_MODE_POSITION	Position control mode
SSC_IFM_CTRL_MODE_SPEED	Speed control mode
SSC_IFM_CTRL_MODE_TORQUE	Torque control mode

bufnum [out]

Pointer to 2-byte variable (2 bytes × 1) which stores the latest command buffer number

bufdata [out]

Pointer to 2-byte variable (2 bytes × 1) which stores the latest command buffer data

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.

Point

• When a NULL pointer is designated as the latest command buffer data, only the latest command buffer number will be got.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.60	A4	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

ssclfmGetTransmitBufferEx

4.21.9 sscIfmGetTransmitBuffer MC200

Transmit buffer number and transmit buffer data for position control mode will be got.

```
int  sscIfmGetTransmitBuffer (
   int  board_id,
   int  channel,
   int  axnum,
   short  *bufnum,
   long  *bufdata
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32)

bufnum [out]

Pointer to 2-byte variable (2 bytes × 1) which stores the transmit buffer number

bufdata [out]

Pointer to 4-byte variable (4 bytes × 1) which stores the transmit buffer data

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.

Point

- When a NULL pointer is designated as the transmit buffer data, only the transmit buffer number will be got.
- For "API Ver. 1.60 or later", use the sscIfmGetTransmitBufferEx function.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.50	A3	mc2xxstd.h

Reference

sscIfmCheckLatestBuffer

4.21.10 sscIfmGetTransmitBufferEx

Transmit buffer number and transmit buffer data for the designated control mode will be got.

```
int  sscIfmGetTransmitBufferEx (
   int  board_id,
   int  channel,
   int  axnum,
   unsigned short  ctrl_mode,
   short  *bufnum,
   long  *bufdata
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32 MC200 /1 to 64 MC300)

ctrl_mode [in]

Control mode

Operation mode	Description
SSC_IFM_CTRL_MODE_POSITION	Position control mode
SSC_IFM_CTRL_MODE_SPEED	Speed control mode
SSC_IFM_CTRL_MODE_TORQUE	Torque control mode

bufnum [out]

Pointer to 2-byte variable (2 bytes × 1) which stores the transmit buffer number

bufdata [out]

Pointer to 4-byte variable (4 bytes × 1) which stores the transmit buffer data

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.

Point

• When a NULL pointer is designated as the transmit buffer data, only the transmit buffer number will be got.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.60	A4	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscIfmCheckLatestBufferEx

4.21.11 sscIfmTrqSetSpeedLimit

The speed limit value for torque control will be set.

```
int sscIfmTrqSetSpeedLimit (
  int board_id,
  int channel,
  int axnum,
  long speed
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32 MC200 /1 to 64 MC300)

speed [in]

Speed limit value for torque control [0.01r/min] (0 to 1000000000)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.

Point

• Call this function before changing to torque control mode.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.60	A4	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscIfmSetControlMode

4.21.12 sscIfmSetControlMode

The control mode will be set.

```
int  sscIfmSetControlMode (
   int  board_id,
   int  channel,
   int  axnum,
   unsigned short  ctrl_mode,
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32 MC200 /1 to 64 MC300)

ctrl_mode [in]

Control mode

Operation mode	Description
SSC_IFM_CTRL_MODE_POSITION	Position control mode
SSC_IFM_CTRL_MODE_SPEED	Speed control mode
SSC_IFM_CTRL_MODE_TORQUE	Torque control mode

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_NOW_ALARM_SYSTEM	A system error (E□□□h) occurred. Get the system status code with the sscGetSystemStatusCode function and remove the cause.

Point

• It takes approximately several ms until the control mode is switched because of the time it takes to confirm the response of the servo amplifier.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.60	A4	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscIfmGetControlMode

4.21.13 sscIfmGetControlMode

The control mode will be got.

```
int  sscIfmGetControlMode (
   int  board_id,
   int  channel,
   int  axnum,
   unsigned short  *ctrl_mode,
   char  *status
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32 MC200 /1 to 64 MC300)

ctrl_mode [out]

Pointer to 2-byte variable (2 bytes × 1) which stores the control mode

Operation mode	Description
SSC_IFM_CTRL_MODE_POSITION	Position control mode
SSC_IFM_CTRL_MODE_SPEED	Speed control mode
SSC_IFM_CTRL_MODE_TORQUE	Torque control mode

status [out]

Pointer to 1-byte variable (1 byte × 1) which stores the control mode switch incorrect status

Operation mode	Description
SSC_IFM_CTRL_MODE_ERR_OFF	Control mode switch incorrect is OFF
SSC_IFM_CTRL_MODE_ERR_ON	Control mode switch incorrect is ON

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.

Point

• When control mode switch in incorrect, set the control mode to the control mode before the switch command was made and call the sscIfmSetControlMode function.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.60	A4	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscIfmSetControlMode

4.21.14 sscIfmGetEventStatusBits MC200

Gets the status bit information of all axes for the designated status signal using the event detect function.

```
int  sscIfmGetEventStatusBits (
   int  board_id,
   int  channel,
   int  bitnum,
   unsigned long  *status_bits
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

bitnum [in]

Status bit number

Value	Description
SSC_EVENT_AX_RDY	Servo ready
SSC_EVENT_AX_INP	In-position
SSC_EVENT_AX_ZSP	Zero speed
SSC_EVENT_AX_TLC	Torque limit effective
SSC_EVENT_AX_SALM	Servo alarm
SSC_EVENT_AX_SWRN	Servo warning
SSC_EVENT_AX_ABSE	Absolute position disappearance
SSC_EVENT_AX_OALM	Operation alarm
SSC_EVENT_AX_MAK1	Mark detection 1
SSC_EVENT_AX_MAK2	Mark detection 2
SSC_EVENT_AX_LSP	+ side limit switch
SSC_EVENT_AX_LSN	- side limit switch
SSC_EVENT_AX_DOG	Proximity dog

status_bits [out]

Pointer to 4-byte variable (4 bytes × 1) which stores the status bit data of all axes.

bit0 → Axis 1, bit1 → Axis 2 ... bit31 → Axis 32

Bit ON: 1, Bit OFF: 0 (For non-controlled axes, the status bit is permanently OFF)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_DISABLE_EVENT_DETECT	Event detect function is not enabled.

Point

- When interrupt processing is ended by the sscIntDisable function or sscIntEnd function, the correct status bit value cannot be got because the event notification function does not operate.
- Because status bits are got using the event detect function, this function reads faster compared to the sscGetStatusBitSignalEx function.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.80	A6	mc2xxstd.h

Reference

sscGetStatusBitSignalEx

4.21.15 sscIfmGetEventStatusBitsEx MC300

Gets the status bit information of all axes for the designated status signal using the event detect function.

```
int  sscIfmGetEventStatusBitsEx (
   int  board_id,
   int  channel,
   int  bitnum,
   SLAVE_INFO  *pSlaveInfo
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

bitnum [in]

Status bit number

Value	Description
SSC_EVENT_AX_RDY	Servo ready
SSC_EVENT_AX_INP	In-position
SSC_EVENT_AX_ZSP	Zero speed
SSC_EVENT_AX_TLC	Torque limit effective
SSC_EVENT_AX_SALM	Servo alarm
SSC_EVENT_AX_SWRN	Servo warning
SSC_EVENT_AX_ABSE	Absolute position disappearance
SSC_EVENT_AX_OALM	Operation alarm
SSC_EVENT_AX_MAK1	Mark detection 1
SSC_EVENT_AX_MAK2	Mark detection 2
SSC_EVENT_AX_LSP	+ side limit switch
SSC_EVENT_AX_LSN	- side limit switch
SSC_EVENT_AX_DOG	Proximity dog

pSlaveInfo [out]

Pointer to 80-byte structure (80 bytes × 1) which stores the slave information (status bit data of all axes)

Refer to "5.11 SLAVE_INFO structure" for the slave information structure. (For non-controlled axes and non-controlled stations information, the status bit is permanently OFF)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0☐ ☐ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_DISABLE_EVENT_DETECT	Event detect function is not enabled.

Point

- The event detect function needs to be enabled (0001h) with interface mode option 2 (parameter No.0010).
- When interrupt processing is ended by the sscIntDisable function or sscIntEnd function, the correct status bit value cannot be got because the event notification function does not operate.
- Because status bits are got using the event detect function, this function reads faster compared to the sscGetStatusBitSignalEx function.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

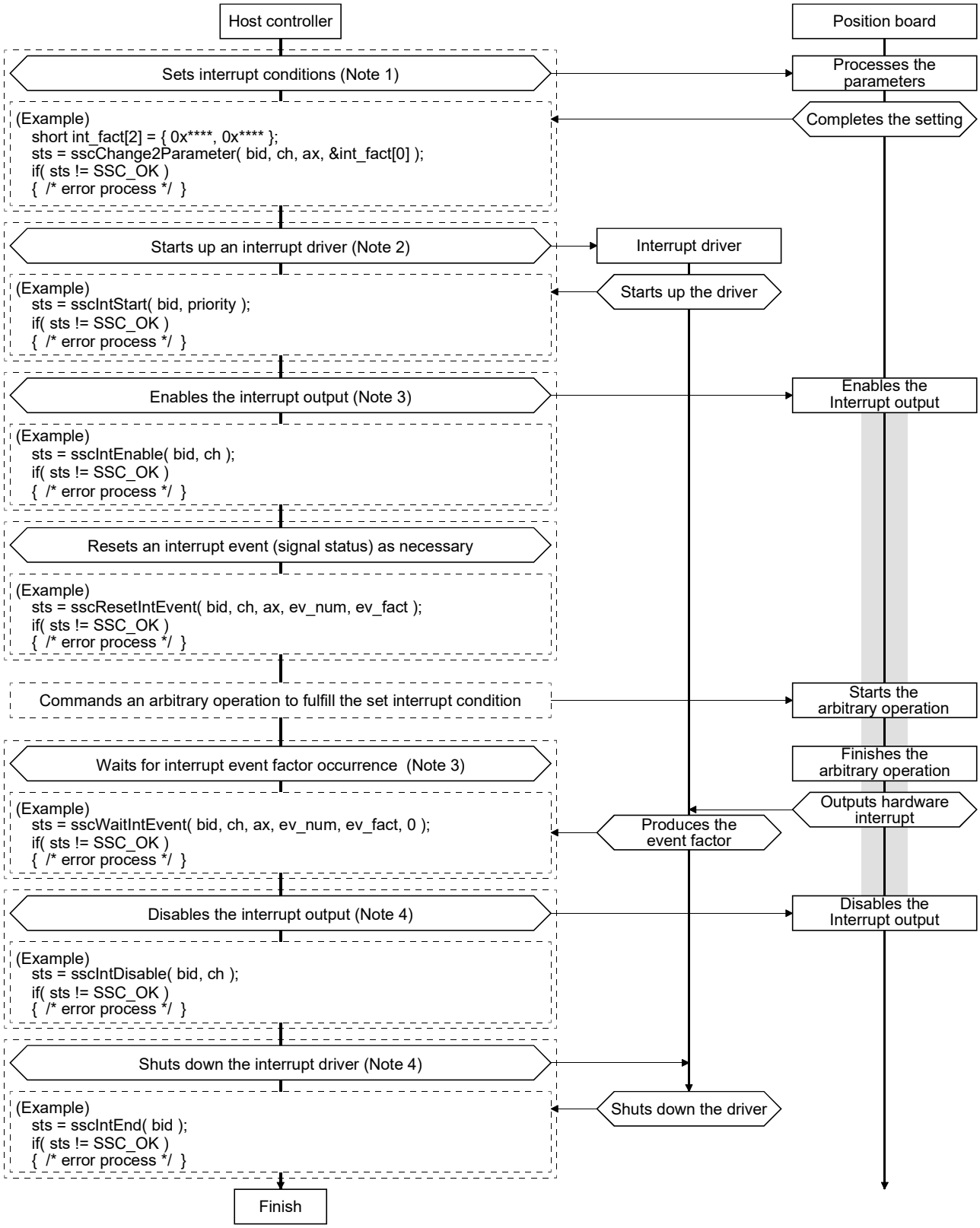
Reference

None.

4.22 Interrupt functions

(1) Processing procedure

An example of processing procedure for using interrupt functions is below.



Note 1. Only when changing the current set interrupt condition parameters (Pr.204 or 205), call the function.
2. Always enable interrupt output after starting up an interrupt driver. (When hardware interrupt is outputted while the interrupt driver is not operating properly, the host controller may hang-up because the hardware interrupt cannot be canceled.)
3. After starting interrupt, an interrupt event wait can be executed with the interrupt event wait functions.
4. Always shut down the interrupt driver after disabling the interrupt output. (For the same reason of the note 2 above)

4.22.1 sscIntStart

The interrupt driver will be started up.
This function is used when performing interrupt monitoring by using the functions related to interrupt event wait.

```
int  sscIntStart (
   int  board_id,
   int  priority
);
```

Argument

board_id [in]

Board ID number (0 to 3)

priority [in]

Priority number

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0☐ ☐ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_ALREADY_START_INT_DRIVER	The interrupt driver has already been started up.
SSC_FUNC_ERR_INT_DISABLE_MASK	The interrupt output mask selection (dip switch) is valid.
SSC_FUNC_ERR_CREATE_EVENT	An error occurred in the CreateEvent function (Win32API). Call the GetLastError function of Win32API and confirm the error details.
SSC_FUNC_ERR_CREATE_THREAD	An error occurred in the CreateThread function (Win32API). Call the GetLastError function of Win32API and confirm the error details.
SSC_FUNC_ERR_THREAD_PRIORITY	An error occurred in the SetThreadPriority function (Win32API). Call the GetLastError function of Win32API and confirm the error details.
SSC_FUNC_ERR_RESUME_THREAD	An error occurred in the ResumeThread function (Win32API). Call the GetLastError function of Win32API and confirm the error details.
SSC_FUNC_ERR_CLEAR_INT	The writing of the interrupt signal clear register is failed.
SSC_FUNC_ERR_ALREADY_OTHER_PROCESS_INT	The interrupt driver has already been started up in other processing.
SSC_FUNC_ERR_DEVICE_DRIVER	An error occurred with a call of the device driver.

Point

- The interrupt driver priority number is set using the SetThreadPriority function (Win32API).
- Refer to the reference manual for operating system for details about the priority set values.
- After calling up this function, interrupt should be enabled using the sscIntEnable function.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2☐☐	Ver.1.00	A0	mc2xxstd.h
MR-MC3☐☐	Ver.1.00	A0	mc3xxstd.h

Reference

sscIntEnd, sscIntEnable

4.22.2 sscIntEnd

The interrupt driver will be closed.

```
int  sscIntEnd (
   int  board_id
);
```

Argument

board_id [in]

Board ID number (0 to 3)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_ALREADY_END_INT_DRIVER	The interrupt driver has already been closed.
SSC_FUNC_ERR_DEVICE_DRIVER	An error occurred with a call of the device driver.
SSC_FUNC_ERR_SET_EVENT	An error occurred in the SetEvent function (Win32API). Call the GetLastError function of Win32API and confirm the error details.
SSC_FUNC_ERR_TIMEOUT_01	While the discard of interrupt handler is being waited, the timeout time (1 second) has elapsed.
SSC_FUNC_ERR_DELETE_THREAD	An error occurred in the CloseHandle function (Win32API). Call the GetLastError function of Win32API and confirm the error details.
SSC_FUNC_ERR_DELETE_EVENT	An error occurred in the CloseHandle function (Win32API). Call the GetLastError function of Win32API and confirm the error details.
SSC_FUNC_ERR_GET_EXIT_CODE_THREAD	An error occurred in the GetExitCodeThread function (Win32API). Call the GetLastError function of Win32API and confirm the error details.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscIntStart

4.22.3 sscIntEnable

The interrupt output valid signal (ITS) will be turned on and interrupt output will be enabled.

```
int  sscIntEnable (
   int  board_id,
   int  channel
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_RESET_EVENT	An error occurred in the ResetEvent function (Win32API). Call the GetLastError function of Win32API and confirm the error details.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscIntDisable, sscIntStart

4.22.4 sscIntDisable

The interrupt output valid signal (ITS) will be turned off and interrupt output will be disabled.

```
int  sscIntDisable (
   int  board_id,
   int  channel
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_RESET_EVENT	An error occurred in the ResetEvent function (Win32API). Call the GetLastError function of Win32API and confirm the error details.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscIntEnable

4.22.5 sscRegisterIntCallback

The interrupt callback function will be registered.
The registered function will be called back from the interrupt driver started by the sscIntStart function when an interrupt occurs.

```
int  sscRegisterIntCallback (
   int  board_id,
   int  channel
   void  *cbfunc
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

cbfunc [in]

Callback function pointer

Refer to "5.8 INT_CB_DATA structure" for the callback structure.

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_ALREADY_REREGISTER_CALLBACK	The interrupt callback function has already been registered. To change the interrupt callback function, call the sscUnregisterIntCallback function.

Point

- When using the C++ language, write the _stdcall declaration for the callback function.
- The update processing of the interrupt factor by the interrupt process end signal (ITE) and the outputting with factor of interrupt signal (ITO) is unnecessary in the callback function.
- The callback function is called back from the interrupt driver, therefore, write a minimum code without the infinite waiting processing.
- The callback function is called back before the interrupt factor occurrence waiting functions such as the sscWaitIntEvent function.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.02	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscIntStart, sscIntEnable, sscUnRegisterIntCallback

4.22.6 sscUnregisterIntCallback

The interrupt call back function will be unregistered.

```
int  sscUnregisterIntCallback (
   int  board_id,
   int  channel
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_ALREADY_UNREREGISTER_CALLBACK	The interrupt callback function has already been unregistered.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.02	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscRegisterIntCallback

4.22.7 sscResetIntEvent

Interrupt event signal status will be nonsignaled.
This function is used if interrupt events occurring prior to calling up the sscWaitIntEvent function are to be disabled.

```
int  sscResetIntEvent (
   int  board_id,
   int  channel,
   int  axnum,
   int  eventnum,
   int  eventfactor
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (-4 to 32 MC200 /-16 to 64 MC300)

0: System interrupt event

1 to 64: Axis interrupt event

-16 to -1: Station interrupt event (-1: Station 1, -2: Station 2, -3: Station 3, -4: Station 4 • • • • • -15: Station 15, -16: Station 16)

eventnum [in]

Event wait number (0 to 1)

eventfactor [in]

Event factor

Refer to “Chapter 7 INTERRUPT EVENT FACTOR LIST” for the event factors.

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_NOT_START_INT_DRIVER	The interrupt driver is closed. Call the sscIntStart function.
SSC_FUNC_ERR_RESET_EVENT	An error occurred in the ResetEvent function (Win32API). Call the GetLastError function of Win32API and confirm the error details.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.80	A8	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscWaitIntEvent

4.22.8 sscSetIntEvent

Interrupt event signal status will be signaled.
This function is used to release the standby status with the sscWaitIntEvent function at the timing of the user program, not the interrupt event of the position board.

```
int  sscSetIntEvent (
   int  board_id,
   int  channel,
   int  axnum,
   int  eventnum,
   int  eventfactor
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (-4 to 32 MC200 /-16 to 64 MC300)

0: System interrupt event

1 to 64: Axis interrupt event

-16 to -1: Station interrupt event (-1: Station 1, -2: Station 2, -3: Station 3, -4: Station 4 • • • • • -15: Station 15, -16: Station 16)

eventnum [in]

Event wait number (0 to 1)

eventfactor [in]

Event factor

Refer to "Chapter 7 INTERRUPT EVENT FACTOR LIST" for the event factors.

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_NOT_START_INT_DRIVER	The interrupt driver is closed. Call the sscIntStart function.
SSC_FUNC_ERR_SET_EVENT	An error occurred in the SetEvent function (Win32API). Call the GetLastError function of Win32API and confirm the error details.

Point

• When the interrupt standby status is released by calling this function, an error occurs in the sscWaitIntEvent function.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.80	A8	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscWaitIntEvent

4.22.9 sscWaitIntEvent

This function waits until the interrupt event status becomes signaled.
This function is used to wait for the interrupt from the position board for the designated event factor.
It is possible to wait for 2 interrupt events to occur from the same cause by changing the event wait number.

```
int  sscWaitIntEvent (
   int  board_id,
   int  channel,
   int  axnum,
   int  eventnum,
   int  eventfactor,
   int  timeout
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (-4 to 32 MC200 /-16 to 64 MC300)

0: System interrupt event

1 to 64: Axis interrupt event

-16 to -1: Station interrupt event (-1: Station 1, -2: Station 2, -3: Station 3, -4: Station 4 • • • • • -15: Station 15, -16: Station 16)

eventnum [in]

Event wait number (0 to 1)

eventfactor [in]

Event factor

Refer to "Chapter 7 INTERRUPT EVENT FACTOR LIST" for the event factors.

timeout [in]

Timeout time[ms] (0 to 65535)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_NOT_START_INT_DRIVER	The interrupt driver is closed. Call the sscIntStart function.
SSC_FUNC_ERR_WAIT_EVENT	An error occurred in the WaitForSingleObject function (Win32API). Call the GetLastError function of Win32API and confirm the error details.
SSC_FUNC_ERR_TERMINATE_INT_DRIVER	The sscIntEnd function was called while the interrupt for the designated event factor was being confirmed.
SSC_FUNC_ERR_TERMINATE_NOTIFY_EVENT	An error occurred in the interrupt event notification thread while the interrupt for the designated event factor was being confirmed.
SSC_FUNC_ERR_TIMEOUT_01	While the interrupt for the designated event factor was being waited, the designated timeout time elapsed.
SSC_FUNC_ERR_SET_HOST_APPLICATION_EVENT	While the interrupt for the designated event factor was being waited, a function which releases the standby status was called from the user program.

Point

• When SSC_INFINITE is designated as the timeout time, timeout is not checked. Instead, this function infinitely waits until the interrupt event occurs.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.80	A8	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscResetIntEvent, sscSetIntEvent, sscWaitIntEventMulti

4.22.10 sscResetIntEventMulti

Multiple interrupt event signal status will be nonsignaled.
This function is used if multiple interrupt events occurring prior to calling up the sscWaitIntEventMulti function are to be disabled.

```
int  sscResetIntEventMulti (
   int  board_id,
   int  channel,
   int  axnum
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (-4 to 32 MC200 /-16 to 64 MC300)

0: System interrupt event

1 to 64: Axis interrupt event

-16 to -1: Station interrupt event (-1: Station 1, -2: Station 2, -3: Station 3, -4: Station 4 • • • • • -15: Station 15, -16: Station 16)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_NOT_START_INT_DRIVER	The interrupt driver is closed. Call the sscIntStart function.
SSC_FUNC_ERR_RESET_EVENT	An error occurred in the ResetEvent function (Win32API). Call the GetLastError function of Win32API and confirm the error details.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.80	A8	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscWaitIntEventMulti

4.22.11 sscSetIntEventMulti

Multiple interrupt event signal status will be signaled.
This function is used to release the standby status with the sscWaitIntEventMulti function at the timing of the user program, not the interrupt event of the position board.

```
int  sscSetIntEventMulti (
   int  board_id,
   int  channel,
   int  axnum
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (-4 to 32 MC200 /-16 to 64 MC300)

0: System interrupt event

1 to 64: Axis interrupt event

-16 to -1: Station interrupt event (-1: Station 1, -2: Station 2, -3: Station 3, -4: Station 4 • • • • • -15: Station 15, -16: Station 16)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0☐ ☐ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_NOT_START_INT_DRIVER	The interrupt driver is closed. Call the sscIntStart function.
SSC_FUNC_ERR_SET_EVENT	An error occurred in the SetEvent function (Win32API). Call the GetLastError function of Win32API and confirm the error details.

Point

• When the interrupt standby status is released by calling this function, an error occurs in the sscWaitIntEventMulti function.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2☐☐	Ver.1.80	A8	mc2xxstd.h
MR-MC3☐☐	Ver.1.00	A0	mc3xxstd.h

Reference

sscWaitIntEventMulti

4.22.12 sscWaitIntEventMulti

This function waits until the multiple interrupt event status becomes signaled.
This function is used to wait for the interrupt from the position board for any multiple event factors.
The multiple event factors which occurred will be stored in the variable designated by the pointer.

```
int  sscWaitIntEventMulti (
   int  board_id,
   int  channel,
   int  axnum,
   int  timeout,
   unsigned long  *eventcode
);
```

Argument

board_id [in]
Board ID number (0 to 3)
channel [in]
Channel number (1)
axnum [in]
Axis number (-4 to 32 MC200 /-16 to 64 MC300)
0: System interrupt event
1 to 64: Axis interrupt event
-16 to -1: Station interrupt event (-1: Station 1, -2: Station 2, -3: Station 3, -4: Station 4 • • • • • -15: Station 15, -16: Station 16)
timeout [in]
Timeout time[ms] (0 to 65535)
eventcode [out]
Pointer to 4-byte variable (4 bytes × 1) which stores the multiple event factors
Refer to "Chapter 7 INTERRUPT EVENT FACTOR LIST" for the event factors.

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_NOT_START_INT_DRIVER	The interrupt driver is closed. Call the sscIntStart function.
SSC_FUNC_ERR_WAIT_EVENT_MULTI	An error occurred in the WaitForMultipleObjects function (Win32API). Call the GetLastError function of Win32API and confirm the error details.
SSC_FUNC_ERR_TERMINATE_INT_DRIVER	The sscIntEnd function was called while the interrupt for the designated event factor was being confirmed.
SSC_FUNC_ERR_TERMINATE_NOTIFY_EVENT	An error occurred in the interrupt event notification thread while the interrupt for the designated event factor was being confirmed.
SSC_FUNC_ERR_TIMEOUT_01	While the interrupt for the designated event factor was being waited, the designated timeout time elapsed.
SSC_FUNC_ERR_SET_HOST_APPLICATION_EVENT	While the interrupt for the designated event factor was being waited, a function which releases the standby status was called from the user program.

Point

- When SSC_INFINITE is designated as the timeout time, timeout is not checked. Instead, this function infinitely waits until the multiple interrupt events occur.
- When the multiple interrupt events occurred simultaneously, the smallest event factor will be stored.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.80	A8	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscResetIntEventMulti, sscSetIntEventMulti, sscWaitIntEvent

4.22.13 sscResetIntOasEvent

Other axes start interrupt event signal status will be nonsignaled.
This function is used if other axes start interrupt event occurring prior to calling up the sscWaitIntOasEvent function are to be disabled.

```
int  sscResetIntOasEvent (
   int  board_id,
   int  channel,
   int  axnum,
   int  oas_num
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number of other axes start table (1 to 32 MC200 /1 to 64 MC300)

oas_num [in]

Other axes start table number (1 to 32 MC200 /1 to 64 MC300)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_NOT_START_INT_DRIVER	The interrupt driver is closed. Call the sscIntStart function.
SSC_FUNC_ERR_RESET_EVENT	An error occurred in the ResetEvent function (Win32API). Call the GetLastError function of Win32API and confirm the error details.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscWaitIntOasEvent

4.22.14 sscSetIntOasEvent

Other axes start interrupt event signal status will be signaled.
This function is used to release the standby status with the sscWaitIntOasEvent function at the timing of the user program, not the interrupt event of the position board.

```
int  sscSetIntOasEvent (
   int  board_id,
   int  channel,
   int  axnum,
   int  oas_num
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number of other axes start table (1 to 32 MC200 /1 to 64 MC300)

oas_num [in]

Other axes start table number (1 to 32 MC200 /1 to 64 MC300)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_NOT_START_INT_DRIVER	The interrupt driver is closed. Call the sscIntStart function.
SSC_FUNC_ERR_SET_EVENT	An error occurred in the SetEvent function (Win32API). Call the GetLastError function of Win32API and confirm the error details.

Point

• When the standby status is released by calling this function, an error occurs in the sscWaitIntOasEvent function.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.02	A1	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscWaitIntOasEvent

4.22.15 sscWaitIntOasEvent

This function waits until the other axes start interrupt event status becomes signaled.
This function is used to wait for the interrupt from the position board for the designated other axes start wait type.
The checked status will be stored in the variable designated by the pointer.

```
int  sscWaitIntOasEvent (
   int  board_id,
   int  channel,
   int  axnum,
   int  oas_num,
   int  oas_type,
   int  *oas_status,
   int  timeout
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number of other axes start table (1 to 32 MC200 /1 to 64 MC300)

oas_num [in]

Other axes start table number (1 to 32 MC200 /1 to 64 MC300)

oas_type [in]

Other axes start wait type

Value	Description
SSC_OAS_WAIT_TYPE_OP	Other axes start notice wait
SSC_OAS_WAIT_TYPE_FIN	Other axes start completion wait

oas_status [out]

Pointer to 4-byte variable (4 bytes × 1) which stores the other axes start status

Value	Description
SSC_OAS_STS_OP	Other axes start notice
SSC_OAS_STS_FIN	Other axes start complete
SSC_OAS_STS_ERR	Other axes start incomplete
SSC_OAS_STS_OP_ERR	Other axes start notice prior error

timeout [in]

Timeout time[ms] (0 to 65535)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_NOT_START_INT_DRIVER	The interrupt driver is closed. Call the sscIntStart function.
SSC_FUNC_ERR_WAIT_EVENT	An error occurred in the WaitForSingleObject function (Win32API). Call the GetLastError function of Win32API and confirm the error details.
SSC_FUNC_ERR_TERMINATE_INT_DRIVER	The sscIntEnd function was called while the interrupt for the designated event factor was being confirmed.
SSC_FUNC_ERR_TERMINATE_NOTIFY_EVENT	An error occurred in the interrupt event notification thread while the interrupt for the designated event factor was being confirmed.
SSC_FUNC_ERR_TIMEOUT_01	While the interrupt for the designated event factor was being waited, the designated timeout time elapsed.
SSC_FUNC_ERR_SET_HOST_APPLICATION_EVENT	While the interrupt for the designated event factor was being waited, a function which releases the standby status was called from the user program.

Point

- When the other axes start complete or the other axes start incomplete occurs during other axes start notice wait, this function returns from the standby status.
- When an alarm occurs in the axis of other axes start before the other axes start notice, "Other axes start notice prior error" occurs.
- The following interrupt conditions are used for this function. When using this function, make sure to set an applicable interrupt condition by sscChange2Parameter().
  - OASF (Factor of other axes start interrupt is being sent)
  - OPF (Completion of operation)
- When SSC_INFINITE is designated as this timeout time, timeout is not checked. Instead, this function infinitely waits until the event occurs.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscResetIntOasEvent, sscSetIntOasEvent

4.22.16 sscResetIntPassPosition

Pass position interrupt event status will be set to nonsignaled.
This function is used to disable the pass position interrupt event occurring prior to calling the sscResetIntPassPosition function.

```
int  sscResetIntPassPosition (
   int  board_id,
   int  channel,
   int  pass_start,
   int  pass_end
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

pass_start [in]

Pass position condition start number (1 to 64 MC200 /1 to 128 MC300)

pass_end [in]

Pass position condition end number (1 to 64 MC200 /1 to 128 MC300)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_NOT_START_INT_DRIVER	The interrupt driver is closed. Call the sscIntStart function.
SSC_FUNC_ERR_RESET_EVENT	An error occurred in the ResetEvent function (Win32API). Call the GetLastError function of Win32API and confirm the error details.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.02	A1	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscWaitIntPassPosition

4.22.17 sscSetIntPassPosition

Pass position interrupt event status will be signaled.
This function is used to release the standby status with the sscWaitIntPassPosition function at the timing of the user program, not the interrupt event of the position board.

```
int  sscSetIntPassPosition (
   int  board_id,
   int  channel,
   int  pass_start,
   int  pass_end
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

pass_start [in]

Pass position condition start number (1 to 64 MC200 /1 to 128 MC300)

pass_end [in]

Pass position condition end number (1 to 64 MC200 /1 to 128 MC300)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_NOT_START_INT_DRIVER	The interrupt driver is closed. Call the sscIntStart function.
SSC_FUNC_ERR_SET_EVENT	An error occurred in the SetEvent function (Win32API). Call the GetLastError function of Win32API and confirm the error details.

Point

• When the standby status is released by calling this function, an error occurs in the sscWaitIntPassPosition function.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.02	A1	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscWaitIntPassPosition

4.22.18 sscWaitIntPassPosition

This function waits until the pass position interrupt event status becomes signaled.
This function is used to wait for the interrupt from the position board for the designated pass position condition number.
The checked status will be stored in the variable designated by the pointer.

```
int  sscWaitIntPassPosition (
   int  board_id,
   int  channel,
   int  pass_num,
   int  *pass_status,
   int  timeout
);
```

Argument

board_id [in]
Board ID number (0 to 3)
channel [in]
Channel number (1)
pass_num [in]
Pass position condition number (1 to 64 MC200 /1 to 128 MC300)
pass_status [out]
Pointer to 4-byte variable (4 bytes × 1) which stores the pass position status

Value	Description
SSC_PASS_STS_FIN	Pass position interrupt complete
SSC_PASS_STS_ERR	Pass position interrupt incomplete

timeout [in]
Timeout time[ms] (0 to 65535)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_NOT_START_INT_DRIVER	The interrupt driver is closed. Call the sscIntStart function.
SSC_FUNC_ERR_WAIT_EVENT	An error occurred in the WaitForSingleObject function (Win32API). Call the GetLastError function of Win32API and confirm the error details.
SSC_FUNC_ERR_TERMINATE_INT_DRIVER	The sscIntEnd function was called while the interrupt for the designated event factor was being confirmed.
SSC_FUNC_ERR_TERMINATE_NOTIFY_EVENT	An error occurred in the interrupt event notification thread while the interrupt for the designated event factor was being confirmed.
SSC_FUNC_ERR_TIMEOUT_01	While the interrupt for the designated event factor was being waited, the designated timeout time elapsed.
SSC_FUNC_ERR_SET_HOST_APPLICATION_EVENT	While the interrupt for the designated event factor was being waited, a function which releases the standby status was called from the user program.

Point

- When SSC_INFINITE is designated as the timeout time, timeout is not checked. Instead, this function infinitely waits until the interrupt event turns on or off.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.02	A1	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscResetIntPassPosition, sscSetIntPassPosition

4.22.19 sscResetIntDriveFin

Operation completion interrupt event status will be nonsignaled.
This function is used if operation completion interrupt event occurring prior to calling up the sscWaitIntDriveFin function are to be disabled.

```
int  sscResetIntDriveFin (
   int  board_id,
   int  channel,
   int  axnum
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32 MC200 /1 to 64 MC300)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_NOT_START_INT_DRIVER	The interrupt driver is closed. Call the sscIntStart function.
SSC_FUNC_ERR_RESET_EVENT	An error occurred in the ResetEvent function (Win32API). Call the GetLastError function of Win32API and confirm the error details.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscWaitIntDriveFin

4.22.20 sscSetIntDriveFin

Operation completion interrupt event status will be signaled.
This function is used to release the standby status with the sscWaitIntDriveFin function at the timing of the user program, not the interrupt event of the position board.

```
int  sscSetIntDriveFin (
   int  board_id,
   int  channel,
   int  axnum
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32 MC200 /1 to 64 MC300)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_NOT_START_INT_DRIVER	The interrupt driver is closed. Call the sscIntStart function.
SSC_FUNC_ERR_SET_EVENT	An error occurred in the SetEvent function (Win32API). Call the GetLastError function of Win32API and confirm the error details.

Point

• When the standby status is released by calling this function, an error occurs in the sscWaitIntDriveFin function.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.02	A1	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscWaitIntDriveFin

4.22.21 sscWaitIntDriveFin

This function waits until the operation completion interrupt event status becomes signaled.
This function is used to wait for the interrupt from the position board for the designated operation completion type.
The checked status will be stored in the variable designated by the pointer.

```
int  sscWaitIntDriveFin (
   int  board_id,
   int  channel,
   int  axnum,
   int  fin_type,
   int  *fin_status,
   int  timeout
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (1 to 32 MC200 /1 to 64 MC300)

fin_type [in]

Operation completion type

Value	Description
SSC_FIN_TYPE_SMZ	Completion of operation by smoothing stop
SSC_FIN_TYPE_CPO	Completion of operation by rough match
SSC_FIN_TYPE_INP	Completion of operation by in-position stop

fin_status [out]

Pointer to 4-byte variable (4 bytes × 1) which stores the completion of operation status

Value	Description
SSC_FIN_STS_STP	Completion of operation
SSC_FIN_STS_MOV	During operation
SSC_FIN_STS_ALM_STP	Alarm occurrence (stop complete)
SSC_FIN_STS_ALM_MOV	Alarm occurrence (during deceleration stop)

timeout [in]

Timeout time[ms] (0 to 65535)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_NOT_START_INT_DRIVER	The interrupt driver is closed. Call the sscIntStart function.
SSC_FUNC_ERR_WAIT_EVENT	An error occurred in the WaitForSingleObject function (Win32API). Call the GetLastError function of Win32API and confirm the error details.
SSC_FUNC_ERR_TERMINATE_INT_DRIVER	The sscIntEnd function was called while the interrupt for the designated event factor was being confirmed.
SSC_FUNC_ERR_TERMINATE_NOTIFY_EVENT	An error occurred in the interrupt event notification thread while the interrupt for the designated event factor was being confirmed.
SSC_FUNC_ERR_TIMEOUT_01	While the interrupt for the designated event factor was being waited, the designated timeout time elapsed.
SSC_FUNC_ERR_SET_HOST_APPLICATION_EVENT	While the interrupt for the designated event factor was being waited, a function which releases the standby status was called from the user program.

Point

- The completion of operation check conditions are the operation completion type designated by the argument.
  - If [Completion of operation by smoothing stop/Completion of operation by rough match/Completion of operation by in-position stop], shutdown will occur after waiting for the designated conditions to be met. However, if an operation alarm occurs and operation is cancelled, an alarm will be returned after the stop is completed. In this case, event waiting will use the WaitForSingleObject function (Win32API).
  - If a stop operation signal (STP) is input during operation, the status will be "Completion of operation" after the stop is completed.
  - If the operation completion type is "Completion of operation by rough match", the status will be "Completion of operation" when the rough match signal (CPO) is ON in automatic operation mode or linear interpolation MC200/interpolation operation MC300 mode.
- The rough match signal is only output during automatic operation mode and linear interpolation MC200/interpolation operation MC300 mode. Therefore, if a completion of operation check is performed when operating in a mode other than automatic operation mode or linear interpolation MC200/interpolation operation MC300 mode, the operation completion type should be other than "Completion of operation by rough match".
- The following interrupt conditions are used for this function. When using this function, make sure to set an applicable interrupt condition by sscChange2Parameter().
  - INP (In-position) Unnecessary when the operation completion type "Completion of operation during in-position stop" is not used.
  - SALM (Servo alarm)
  - CPO (Rough match) Unnecessary when the operation completion type "Completion of operation by rough match" is not used.
  - OALM (Operation alarm)
  - OPF (Completion of operation)
  - SYSE (System error) Unnecessary when the operation completion type "Completion of operation during in-position" is not used.
- Set necessary interrupt conditions only. Unnecessary interrupt condition may deteriorate the performance of the user program.
- When SSC_INFINITE is designated as this timeout time, timeout is not checked. Instead, this function infinitely waits until the event occurs.
- When the "deceleration check system" is set to in-position stop in the automatic operation or the linear interpolation MC200/ interpolation operation MC300, the "Operation completion type" is "Completion of operation by in-position stop" even though the smoothing stop is set.
- Since the completion of operation status of this function is judged by the operation completion type, it may differ from the during operation signal (OP) and the completion of operation signal (OPF) of the position board.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscResetIntDriveFin, sscSetIntDriveFin, sscGetDriveFinStatus

4.23 I/O device functions

4.23.1 sscGetInputDeviceBit

Gets the designated input bit device in 1-point basis.

```
int  sscGetInputDeviceBit (
   int  board_id,
   int  channel,
   int  bit_num,
   int  *dev_in
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

bit_num [in]

Input bit device number (0000h to 0FFFh MC200 /0000h to 23FFh MC300)

dev_in [out]

Pointer to 4-byte variable (4 bytes × 1) which stores the status of input bit device

Value	Description
SSC_BIT_OFF	Input signal OFF
SSC_BIT_ON	Input signal ON

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.80	A8	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscGetInputDeviceWord

4.23.2 sscGetInputDeviceWord

Gets the designated input word device in 1-word basis.

```
int  sscGetInputDeviceWord (
   int  board_id,
   int  channel,
   int  word_num,
   int  word_cnt,
   unsigned short  *dev_in
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

word_num [in]

Input word device number (0000h to 00FFh MC200 /0000h to 023Fh MC300)

word_cnt [in]

Word points from the input word device number (0001h to 0100h MC200 /0001h to 0240h MC300)

dev_in [out]

Pointer to the array (2 bytes × word_cnt) which stores the status of input word device

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_DVI_TABLE_RANGE_OVER	The "word_num" + "word_cnt" designated by the argument exceeds the size of the input device table.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.80	A8	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscGetInputDeviceBit

4.23.3 sscSetOutputDeviceBit

Sets the designated output bit device in 1-point basis.

```
int  sscSetOutputDeviceBit (
   int  board_id,
   int  channel,
   int  bit_num,
   int  dev_out
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

bit_num [in]

Output bit device number (0000h to 0FFFh MC200 /0000h to 23FFh MC300)

dev_out [in] (0 to 1)

Setting data

Value	Description
SSC_BIT_OFF	Output signal OFF
SSC_BIT_ON	Output signal ON

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.

Point

• This function sets the output bit device with the exclusive control function of the position board. However, the exclusive control function cannot be used in interface mode because the other axes start function cannot be used.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.80	A8	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscSetOutputDeviceWord, sscGetOutputDeviceBit

4.23.4 sscSetOutputDeviceWord

Sets the designated output word device in 1-word basis.

```
int  sscSetOutputDeviceWord (
   int  board_id,
   int  channel,
   int  word_num,
   int  word_cnt,
   unsigned short  *dev_out
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

word_num [in]

Output word device number (0000h to 00FFh MC200 /0000h to 023Fh MC300)

word_cnt [in]

Word points from the ouput word device number (0001h to 0100h MC200 /0001h to 0240h MC300)

dev_out [in]

Pointer to the array (2 bytes × word_cnt) which stores the setting data

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_DVO_TABLE_RANGE_OVER	The "word_num" + "word_cnt" designated by the argument exceeds the size of the output device table.

Point

• When controlling output signals by other axes start function, data may become inconsistent if output word devices are set for the applicable output signals.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.80	A8	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscSetOutputDeviceBit, sscGetOutputDeviceWord

4.23.5 sscGetOutputDeviceBit

Gets the designated output bit device in 1-point basis.

```
int  sscGetOutputDeviceBit (
   int  board_id,
   int  channel,
   int  bit_num,
   int  *dev_out
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

bit_num [in]

Output bit device number (0000h to 0FFFh MC200 /0000h to 23FFh MC300)

dev_out [out]

Pointer to 4-byte variable (4 bytes × 1) which stores the status of output bit device

Value	Description
SSC_BIT_OFF	Output signal OFF
SSC_BIT_ON	Output signal ON

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.80	A8	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscGetOutputDeviceWord, sscSetOutputDeviceBit

4.23.6 sscGetOutputDeviceWord

Gets the designated output word device in 1-word basis.

```
int  sscGetOutputDeviceWord (
   int  board_id,
   int  channel,
   int  word_num,
   int  word_cnt,
   unsigned short  *dev_out
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

word_num [in]

Output word device number (0000h to 00FFh MC200 /0000h to 023Fh MC300)

word_cnt [in]

Word points from the output word device number (0001h to 0100h MC200 /0001h to 0240h MC300)

dev_out [out]

Pointer to the array (2 bytes × word_cnt) which stores the status of output word device

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_DVO_TABLE_RANGE_OVER	The "word_num" + "word_cnt" designated by the argument exceeds the size of the output device table.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.80	A8	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscGetOutputDeviceBit, sscSetOutputDeviceWord

4.24 Transient transmit functions

4.24.1 sscSendReceiveTransientData

Sends and receives the specified transient transmit data for axes or stations connected to SSCNET.

```
int  sscSendReceiveTransientData (
   int  board_id,
   int  channel,
   int  axnum,
   TRANSIENT_CMD  *pTransientCmd,
   TRANSIENT_STS  *pTransientSts,
   int  timeout
);
```

Argument

board_id [in]

Board ID number (0 to 3)

channel [in]

Channel number (1)

axnum [in]

Axis number (-4 to -1, 1 to 32 MC200 /-16 to -1, 1 to 64 MC300)

1 to 64: Axis

-16 to -1: Station (-1: Station 1, -2: Station 2, -3: Station 3, -4: Station 4 • • • • • -15: Station 15, -16: Station 16)

pTransientCmd [in]

Pointer to 16-byte structure (16 bytes × 1) which stores the transient transmit command data

Refer to "5.8 TRANSIENT_CMD structure" for the transient transmit command data structure.

pTransientSts [out]

Pointer to 16-byte structure (16 bytes × 1) which stores the transient transmit status data

Refer to "5.9 TRANSIENT_STS structure" for the transient transmit status data structure.

timeout [in]

Timeout time[ms] (0 to 65535)

Return value

SSC_OK	Function succeeded.
SSC_NG	Function failed. (To confirm the detailed error code, use the sscGetLastError function.)
SSC_UNOPEN	Before calling the sscOpen function.

Detailed error code

Value	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_TIMEOUT_01	After sending command to position board, the designated timeout time elapsed while confirming completion of transient processing.
SSC_FUNC_ERR_TRANSIENT_INVALID_DATA	Transient data is invalid.

Point

• When the timeout time is designated to less than 1 second (1000ms), the timeout is at 1 second (1000ms).

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.80	A8	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

None.

MEMO

5. STRUCTURE LIST

5.1 PNT_DATA_EX structure

This point data structure is used for the automatic operation and linear interpolation MC200/interpolation operation MC300.

5.1.1 PNT_DATA_EX structure (using MR-MC2□□)

```
typedef struct {
/* 0000h */
   long    position;
   unsigned long    speed;
   unsigned short   actime;
   unsigned short   dctime;
   unsigned short   dwell;
   unsigned short   subcmd;

/* 0010h */
   unsigned char   oas_num[2];
   unsigned char   reserve2[2];
   unsigned char   s_curve;
   unsigned char   reserve2[3];
   unsigned char   sub_axnum[3];
   unsigned char   reserve3[5];

/* 0020h */
} PNT_DATA_EX;
```

Member

position

Position data

speed

Feed speed (0 to 2147483647)

actime

Acceleration time constant [ms] (0 to 20000)

dctime

Deceleration time constant [ms] (0 to 20000)

dwell

Dwell [ms]

subcmd

Auxiliary command

Set data in the logical sum of each value.

Value	Description	
SSC_SUBCMD_POS_ABS	Position command system	Absolute position command
SSC_SUBCMD_POS_INC		Relative position command
SSC_SUBCMD_STOP_INP	Deceleration check system	In-position stop
SSC_SUBCMD_STOP_SMZ		Smoothing stop
SSC_SUBCMD_STOP_CONTINUE		Continue operation
SSC_SUBCMD_PNT_SWITCH_AFTER	Speed switching point specification	After point switching
SSC_SUBCMD_PNT_SWITCH_BEFORE		Before point switching
SSC_SUBCMD_DWELL	Dwell specification	Dwell
SSC_SUBCMD_PREDWELL		Predwell
SSC_SUBCMD_PASS_POS_DISABLE	Pass position interrupt specification	Disable
SSC_SUBCMD_PASS_POS_ENABLE		Enable
SSC_SUBCMD_PRESS_DISABLE	Continuous operation to torque control specification	Disable
SSC_SUBCMD_PRESS_ENABLE		Enable
SSC_SUBCMD_PNT_LOOP_DISABLE	Loop specification	Disable
SSC_SUBCMD_PNT_LOOP_START		Loop start point
SSC_SUBCMD_PNT_LOOP_END		Loop end point

oas_num[2]

Other axes start specification (0 to 32)

s_curve

S-curve ratio [%] (0 to 100)

sub_axnum[3]

Interpolation axis No. (0 to 32)

Set the axis Nos. of the auxiliary axes to be in the same interpolation group. When not using auxiliary axes, set 0.

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.90	A9	mc2xxstd.h

Reference

sscSetPointDataEx, sscCheckPointDataEx

5.1.2 PNT_DATA_EX structure (using MR-MC3□□)

```
typedef struct {
/* 0000h */
   long    position;
   unsigned long    speed;
   unsigned short   actime;
   unsigned short   dctime;
   unsigned short   dwell;
   unsigned short   subcmd;

/* 0010h */
   unsigned char   oas_num[4];
   unsigned char   s_curve;
   unsigned char   reserve1[3];
   unsigned char   sub_axnum[4];
   long   arc_coord;

/* 0020h */
   unsigned short   ac_dc_data[4];
   unsigned short   subcmd2;
   char   reserve3[6];

/* 0030h */
} PNT_DATA_EX;
```

Member

position

Position data [command units]

speed

Feed speed [speed units] (0 to 2147483647)

actime

Acceleration time constant [ms] (0 to 20000)

dctime

Deceleration time constant [ms] (0 to 20000)

dwell

Dwell [ms]

subcmd

Auxiliary command

Set data in the logical sum of each value.

Value	Description	
SSC_SUBCMD_POS_ABS	Position command system	Absolute position command
SSC_SUBCMD_POS_INC		Relative position command
SSC_SUBCMD_DAMPING_DISABLE	Vibration suppression command filter 1 command	Vibration suppression command filter 1 disable
SSC_SUBCMD_DAMPING_ENABLE		Vibration suppression command filter 1 enable
SSC_SUBCMD_STOP_INP	Deceleration check system	In-position stop
SSC_SUBCMD_STOP_SMZ		Smoothing stop
SSC_SUBCMD_STOP_CONTINUE		Continue operation
SSC_SUBCMD_PNT_SWITCH_AFTER	Speed switching point specification	After point switching
SSC_SUBCMD_PNT_SWITCH_BEFORE		Before point switching
SSC_SUBCMD_DWELL	Dwell specification	Dwell
SSC_SUBCMD_PREDWELL		Predwell
SSC_SUBCMD_PASS_POS_DISABLE	Pass position interrupt specification	Disable
SSC_SUBCMD_PASS_POS_ENABLE		Enable
SSC_SUBCMD_PRESS_DISABLE	Continuous operation to torque control specification	Disable
SSC_SUBCMD_PRESS_ENABLE		Enable
SSC_SUBCMD_PNT_LOOP_DISABLE	Loop specification	Disable
SSC_SUBCMD_PNT_LOOP_START		Loop start point
SSC_SUBCMD_PNT_LOOP_END		Loop end point
SSC_SUBCMD_INTERP_LINEAR	Interpolation system	Linear interpolation
SSC_SUBCMD_INTERP_ARC		Auxillary point-specified circular interpolation
SSC_SUBCMD_INTERP_ARC_CW		Central point-specified circular interpolation (CW)
SSC_SUBCMD_INTERP_ARC_CCW		Central point-specified circular interpolation (CCW)

oas_num[4]

Other axes start specification (0 to 64)

s_curve

S-curve ratio [%] (0 to 100)

sub_axnum[4]

Interpolation axis No. (0 to 64)

Set the axis Nos. of the auxiliary axes to be in the same interpolation group. When not using auxiliary axes, set 0.

arc_coord

Arc coordinates [command units]

Set the auxillary point of the arc or the coordinates of the central point. Settings vary according to the interpolation system.

ac_dc_data[4]

Acceleration/deceleration data [0.1%] (0 to 1000)

subcmd2

Auxiliary command 2

Set data in the logical sum of each value.

Value	Description	
SSC_SUBCMD2_ACCDEC_LINE_S	Acceleration/deceleration system	Linear acceleration/deceleration and S-curve acceleration/deceleration
SSC_SUBCMD2_ACCDEC_JERK		Jerk ratio acceleration/deceleration

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC3□□	Ver. 1.10	A1	mc3xxstd.h

Reference

sscSetPointDataEx, sscCheckPointDataEx

5.2 OAS_DATA structure

This other axis start data structure is used for the other axes start.

5.2.1 OAS_DATA structure (using MR-MC2□□)

```
typedef struct {
/* 0000h */
   unsigned long   opt_own;
   unsigned long   opt_observ;
   long   data_own;
   long   data_observ;

/* 0010h */
   char   reserve1[8];
   unsigned long long   st_axbit;


/* 0020h */
   unsigned short   st_pnt_s;
   unsigned short   st_pnt_e;
   char   reserve2[12];

/* 0030h */
   char   reserve3[40];
   unsigned char   dout_ctrl;
   unsigned char   dout_num;
   unsigned short   dout_ctrlbit;
   unsigned short   dout_data;
   char   reserve4[10];

/* 0068h */
} OAS_DATA;
```

Member

opt_own

Axis option

Set data in the logical sum of each value.

Value	Description	
SSC_OAS_OWN_REMAINING_DISTANCE	Axis judgment condition	Remaining distance specification
SSC_OAS_OWN_POSITION_PASS		Specified position pass specification
SSC_OAS_OWN_JUDGE_COORD_FB	Axis judgment coordinate	Current feedback position
SSC_OAS_OWN_JUDGE_COORD_CMD		Command position

opt_observ

Observed axis option

Set data in the logical sum of each value.

Value	Description	
SSC_OAS_OBSERV_DISABLE	Observed axis specification	Disable
SSC_OAS_OBSERV_ENABLE		Enable
SSC_OAS_OBSERV_POSITION_PASS	Observed axis judgment condition	Observed axis specified position pass specification
SSC_OAS_OBSERV_JUDGE_COORD_FB	Observed axis judgment coordinate	Current feedback position
SSC_OAS_OBSERV_JUDGE_COORD_CMD		Command position
SSC_OAS_OBSERV_DATA_LESS	Observed axis specified position pass judgment condition	Condition is satisfied when observed axis position is less than or equal to observed axis specified position data
SSC_OAS_OBSERV_DATA_MORE		Condition is satisfied when observed axis position is more than or equal to observed axis specified position data

data_own

Axis remaining distance data (or axis pass position data) [command units]

data_observ

Observed axis remaining distance data (or observed axis pass position data) [command units]

st_axbit

Start axis designation

st_pnt_s

Start axis start point (0 to 319)

st_pnt_e

Start axis end point (0 to 319)

dout_ctrl

Digital output signal control/Output device signal control

Value	Description	
SSC_OAS_DO_DISABLE	Digital output signal control	Disable
SSC_OAS_DO_ENABLE		Enable

dout_num

Digital output signal number/Output device signal number (00 to 3Fh)

dout_ctrlbit

Digital output signal enable selection/Output device signal enable selection

dout_data

Digital output signal command/Output device signal command

Point

• Always set Observed axis specified position pass specification (SSC_OAS_OBSERV_POSITION_PASS) when Observed axis specification is enabled (SSC_OAS_OBSERV_ENABLE).

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.02	A1	Mc2xxstd.h

Reference

sscSetOtherAxisStartData, sscGetOtherAxisStartData

5.2.2 OAS_DATA structure (using MR-MC3□□)

```
typedef struct {
/* 0000h */
   unsigned long   opt_own;
   unsigned long   opt_observ;
   long   data_own;
   long   data_observ;

/* 0010h */
   unsigned char   reserve1[24];
   unsigned long long   st_axbit;

/* 0030h */
   unsigned short   st_pnt_s;
   unsigned short   st_pnt_e;
   char   reserve2[12];

/* 0040h */
   unsigned char   reserve3[40];
   unsigned char   dout_ctrl;
   unsigned char   dout_num;
   unsigned short   dout_ctrlbit;
   unsigned short   dout_data;
   char   reserve4[2];

/* 0070h */
   char   reserve5[16];

/* 0080h */
} OAS_DATA;
```

Member

opt_own

Axis option

Set data in the logical sum of each value.

Value	Description	
SSC_OAS_OWN_REMAINING_DISTANCE	Axis judgment condition	Remaining distance specification
SSC_OAS_OWN_POSITION_PASS		Specified position pass specification
SSC_OAS_OWN_JUDGE_COORD_FB	Axis judgment coordinate	Current feedback position
SSC_OAS_OWN_JUDGE_COORD_CMD		Command position

opt_observ

Observed axis option

Set data in the logical sum of each value.

Value	Description	
SSC_OAS_OBSERV_DISABLE	Observed axis specification	Disable
SSC_OAS_OBSERV_ENABLE		Enable
SSC_OAS_OBSERV_POSITION_PASS	Observed axis judgment condition	Observed axis specified position pass specification
SSC_OAS_OBSERV_JUDGE_COORD_FB	Observed axis judgment coordinate	Current feedback position
SSC_OAS_OBSERV_JUDGE_COORD_CMD		Command position
SSC_OAS_OBSERV_DATA_LESS	Observed axis specified position pass judgment condition	Condition is satisfied when observed axis position is less than or equal to observed axis specified position data
SSC_OAS_OBSERV_DATA_MORE		Condition is satisfied when observed axis position is more than or equal to observed axis specified position data

data_own

Axis remaining distance data (or axis pass position data) [command units]

data_observ

Observed axis remaining distance data (or observed axis pass position data) [command units]

st_axbit

Start axis designation

st_pnt_s

Start axis start point (0 to 2047)

st_pnt_e

Start axis end point (0 to 2047)

dout_ctrl

Digital output signal control/Output device signal control

Value	Description	
SSC_OAS_DO_DISABLE	Digital output signal control	Disable
SSC_OAS_DO_ENABLE		Enable

dout_num

Digital output signal number/Output device signal number (00 to 3Fh)

dout_ctrlbit

Digital output signal enable selection/Output device signal enable selection

dout_data

Digital output signal command/Output device signal command

Point

• Always set observed axis specified position pass specification (SSC_OAS_OBSERV_POSITION_PASS) when observed axis specification is enabled (SSC_OAS_OBSERV_ENABLE).

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscSetOtherAxisStartData, sscGetOtherAxisStartData

5.3 PRESS_DATA structure

This continuous operation to torque control data structure is used for automatic operation (continuous operation to torque control).

```
typedef struct {
/* 0000h */
   long   switch_position;
   long   position_limit;
   long   speed_limit;
   unsigned short   target_torque;
   unsigned short   continue_time;

/* 0010h */
   unsigned short   torque_settle_width;
   unsigned short   torque_settle_time;
   unsigned short   actime;
   unsigned short   dctime;
   unsigned short   condition;
   char   reserve1[6];

/* 0020h */
} PRESS_DATA;
```

Member

switch_position

Continuous operation to torque control switching position [command units]

position_limit

Press limit position [command units]

speed_limit

Continuous operation to torque control speed limit value [speed units] (1 to 2147483647)

target_torque

Target torque [0.1%] (0 to 32767)

continue_time

Press time [ms]

torque_settle_width

Torque settle width [0.1%]

torque_settle_time

Torque settle waiting time [ms]

actime

Continuous operation to torque control acceleration time constant [ms] (0 to 20000)

dctime

Continuous operation to torque control deceleration time constant [ms] (0 to 20000)

condition

Continuous operation to torque control operating conditions

Set data in the logical sum of each value.

Value	Description	
SSC_PRESS_START_AUTO_CMD	Start switch to continuous operation to torque control condition	Automatic switch (command position)
SSC_PRESS_START_AUTO_FB		Automatic switch (current feedback position)
SSC_PRESS_START_MANUAL		Manual switch
SSC_PRESS_END_AUTO	End switch to continuous operation to torque control condition	Automatic switch
SSC_PRESS_END_MANUAL		Manual switch

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.60	A5	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscSetPressData, sscGetPressData

5.4 SMP_ERR structure

This sampling error structure is used for the sampling.

5.4.1 SMP_ERR structure (using MR-MC2□□)

```
typedef struct {
/* 0000h */
   unsigned long long   err_ax;
   unsigned short   err_ut;
   char   reserve1[6];
   unsigned long   err_dat;
   char   reserve2[4];
   unsigned long   err_bit;
   char   reserve3[4];

/* 0020h */
} SMP_ERR;
```

Member

err_ax

Axis error information

err_ut

Station error information

err_dat

Data error information

err_bit

Bit error information

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h

Reference

sscGetSamplingError

5.4.2 SMP_ERR structure (using MR-MC3□□)

```
typedef struct {
/* 0000h */
   unsigned long long   err_ax;
   char   reserve1[8];

/* 0010h */
   unsigned short   err_ut;
   char   reserve2[6];
   unsigned long   err_dat;
   char   reserve3[4];

/* 0020h */
   unsigned long   err_bit;
   char   reserve4[4];
   char   reserve5[8];

/* 0030h */
} SMP_ERR;
```

Member

err_ax

Axis error information

err_ut

Station error information

err_dat

Data error information

err_bit

Bit error information

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscGetSamplingError

5.5 SMP_DATA structure

This sampling data structure is used for the sampling.

5.5.1 SMP_DATA structure (using MR-MC2□□)

```
typedef struct {
/* 0000h */
   long   smpdata[32];

/* 0080h */
   unsigned short   smpbit[1];
   unsigned char   reserve1[2];

/* 0084h */
} SMP_DATA;
```

Member

smpdata[32]

Data 1 to 32

smpbit[1]

Bit information

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h

Reference

sscGetSamplingData

5.5.2 SMP_DATA structure (using MR-MC3□□)

```
typedef struct {
/* 0000h */
   long   smpdata[32];

/* 0080h */
   unsigned long   smpbit[1];

/* 0084h */
} SMP_DATA;
```

Member

smpdata[32]

Data 1 to 32

smpbit[1]

Bit information

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscGetSamplingData

5.6 LOG_DATA structure

This log data structure is used for the log function.

```
typedef struct {
/* 0000h */
   short   axnum;
   short   eventcode;
   long    eventtime;
   short   eventdata[4];

/* 0010h */
} LOG_DATA;
```

Member

axnum

Axis number

eventcode

Event code

eventtime

Time stamp

eventdata[4]

Information for each event

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.00	A0	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscReadLogData

5.7 ALH_DATA structure

This alarm history data structure is used for the alarm history function.

```
typedef struct {
/* 0000h */
   long long  system_time;
   unsigned long  free_run_cnt;
   unsigned char  ctrl_cycle;
   unsigned char  event_code;
   char  reserve1[2];

/* 0010h */
   unsigned char  sscnet_type;
   unsigned char  ctrl_mode;
   char  reserve2[2];
   unsigned short  axnum;
   unsigned short  alarm_code;
   unsigned char  drive_mode;
   char  reserve3[3];
   long  cmd_pos;

/* 0020h */
   long  fb_pos;
   char  reserve4[27];
   unsigned char  check_sum;

/* 0040h */
} ALH_DATA;
```

Member

system_time
: System start-up time

free_run_cnt
: Free-run counter

ctrl_cycle
: Control cycle

event_code
: Event code

sscnet_type
: Communication mode

ctrl_mode
: Control mode

axnum
: Error axis number

alarm_code
: Alarm number

drive_mode
: Operation mode

cmd_pos
: Current position [command units]

fb_pos
: Feedback position [command units]

check_sum
: Checksum

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.50	A3	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscGetAlarmHistoryData

5.8 TRANSIENT_CMD structure

This transient transmit command data structure is used for sending and receiving transient transmit data.

```
typedef struct {
/* 0000h */
   unsigned short   cmd_req;
   unsigned short   command;
   unsigned short   req_data[4];
   unsigned short   reserve[2];

/* 0010h */
} TRANSIENT_CMD;
```

Member

cmd_req

Command send request

Value	Description
SSC_TRANSIENT_CMD_SINGLE	Transient request

command

Transient command

req_data[4]

Request data

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.80	A8	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscSendReceiveTransientData

5.9 TRANSIENT_STS structure

This transient transmit status data structure is used for sending and receiving transient transmit data.

```
typedef struct {
/* 0000h */
   unsigned short   status;
   unsigned short   reserve1;
   unsigned short   ans_data[4];
   unsigned short   reserve2[2];

/* 0010h */
} TRANSIENT_STS;
```

Member

status

Transient status

Obtained data is set with the logical sum of each value.

Value	Description
SSC_TRANSIENT_STS_WAITING	Waiting for transient command processing completion
SSC_TRANSIENT_STS_START	Transient request start
SSC_TRANSIENT_STS_RECEIVING	Transient receiving
SSC_TRANSIENT_STS_RECEIVE_FIN	Transient received successfully
SSC_TRANSIENT_STS_VALID_DATA	Transient data valid
SSC_TRANSIENT_STS_INVALID_DATA	Transient data invalid

ans_data[4]

Response data

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.80	A8	mc2xxstd.h
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscSendReceiveTransientData

5.10 INT_CB_DATA structure

This interrupt data structure is used for the interrupt callback function.

5.10.1 INT_CB_DATA structure (using MR-MC2□□)

```
typedef struct {
/* 0000h */
   int   board_id;
   int   channel;
   unsigned long   free_run_cnt;
   unsigned char   sys_factor_bit;
   char   reserve1;
   unsigned short   sys_factor;

/* 0010H */
   unsigned long long   axis_factor_bit;
   char   reserve2[8];
   unsigned long   axis_factor[48];

/* 00E0H */
   unsigned short   unit_factor_bit;
   char   reserve3[14];
   unsigned short   unit_factor[8];

/* 0100H */
   unsigned long   oas_factor_bit;
   char   reserve4[12];
   unsigned char   oas_factor[32];

/* 0130H */
   unsigned long long   pass_factor_bit;
   char   reserve5[8];
   unsigned char   pass_factor[64];

/* 0180H */
   char   reserve6[0x180];

/* 0300h */
} INT_CB_DATA;
```

Member

board_id
: Board ID number

channel
: Channel number

free_run_cnt
: Free-run counter

sys_factor_bit
: Bit for factor of system interrupt being sent

sys_factor
: Factor of system interrupt

axis_factor_bit
: Bit for factor of axes interrupt being sent

axis_factor[48]

Factor of axes interrupt

unit_factor_bit

Bit for factor of stations interrupt being sent

unit_factor[8]

Factor of stations interrupt

oas_factor_bit

Factor of other axes start interrupt

oas_factor[32]

Details for factor of other axes start interrupt

pass_factor_bit

Factor of pass position interrupt

pass_factor[64]

Details for factor of pass position interrupt

Point

• For API version 1.61 or later, if interface mode event detect is enabled and system is started, the following information is stored in bit for factor of axes interrupt being sent and factor of axes interrupt. These values are updated every time an event occurs.

• Factor of axes interrupt being sent : Turns ON when and event occurs at the corresponding axis.

• Factor of axes interrupt : Event detect information is stored.
Refer to "MR-MC200/MR-MC300 Series Position Board User's Manual (Details)" for description of the data stored.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC2□□	Ver.1.80	A8	mc2xxstd.h

Reference

sscRegisterIntCallback

5.10.2 INT_CB_DATA structure (using MR-MC3□□)

```
typedef struct {
/* 0000h */
   int   board_id;
   int   channel;
   unsigned long   free_run_cnt;
   unsigned char   sys_factor_bit;
   char   reserve1;
   unsigned short   sys_factor;

/* 0010H */
   unsigned long long   axis_factor_bit;
   char   reserve2[8];
   unsigned long   axis_factor[128];

/* 0220H */
   unsigned short   unit_factor_bit;
   char   reserve3[14];
   unsigned short   unit_factor[32];

/* 0270H */
   unsigned long long   oas_factor_bit;
   char   reserve4[8];
   unsigned char   oas_factor[64];

/* 02C0H */
   unsigned long long   pass_factor_bit[2];
   char   reserve5[16];
   unsigned char   pass_factor[128];

/* 0360H */
   unsigned long long   event_factor_bit;
   char   reserve6[8];
   unsigned long long   event_factor[64];

/* 0570H */
   unsigned short   latest_num[128];
   unsigned short   trans_num[128];

/* 0770H */
   char   reserve7[0x90];

/* 0800h */
} INT_CB_DATA;
```

Member

board_id

Board ID number

channel

Channel number

free_run_cnt

Free-run counter

sys_factor_bit

Bit for factor of system interrupt being sent

sys_factor

Factor of system interrupt

axis_factor_bit

Bit for factor of axes interrupt being sent

axis_factor[128]

Factor of axes interrupt

unit_factor_bit

Bit for factor of stations interrupt being sent

unit_factor[32]

Factor of stations interrupt

oas_factor_bit

Factor of other axes start interrupt

oas_factor[64]

Details for factor of other axes start interrupt

pass_factor_bit[2]

Factor of pass position interrupt

pass_factor[128]

Details for factor of pass position interrupt

event_factor_bit

Factor of event interrupt

event_factor[64]

Details for factor of event interrupt

latest_num[128]

Latest command buffer number

trans_num[128]

Transmit buffer number

Point

- If interface mode event detect is enabled and system is started, the following information is stored in bit for factor of axes interrupt being sent and factor of axes interrupt. These values are updated every time an event occurs.
  - Factor of axes interrupt being sent : Turns ON when and event occurs at the corresponding axis.
  - Factor of axes interrupt : Event detect information is stored.
    Refer to "MR-MC200/MR-MC300 Series Position Board User's Manual (Details)" for description of the data stored.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscRegisterIntCallback

5.11 SLAVE_INFO structure MC300

This slave information structure is used for SSCNET communication disconnect/reconnect functions and functions to get controlling axis information.

```
typedef struct {
/* 0000h */
   unsigned char   axis[64];

/* 0040h */
   unsigned char   unit[16];

/* 0050h */
} SLAVE_INFO;
```

Member

axis[64]

Axis information.

Value	Description
SSC_SLAVE_ON	Controlled axis or status bit ON
SSC_SLAVE_OFF	Non-controlled axis or status bit OFF

unit[16]

Station information.

Value	Description
SSC_SLAVE_ON	Controlled station or status bit ON
SSC_SLAVE_OFF	Non-controlled station or status bit OFF

Point

None.

Supported version

Position board	API Ver.	Board Ver.	Header file
MR-MC3□□	Ver.1.00	A0	mc3xxstd.h

Reference

sscReconnectSSCNETEx, sscDisconnectSSCNETEx, sscIfmGetEventStatusBitsEx

MEMO

6. BIT DEFINITION LIST

The following tables list the bit definitions to be specified in the sscSetCommandBitSignalEX function, sscGetStatusBitSignalEx function, and sscWaitStatusBitSignalEx function.

6.1 System command bit

Bit number (constant)	Signal name
SSC_CMDBIT_SYS_ITE	Interrupt processing complete
SSC_CMDBIT_SYS_ITS	Interrupt output valid
SSC_CMDBIT_SYS_03	Reserved
SSC_CMDBIT_SYS_04	
SSC_CMDBIT_SYS_HMA	During user program memory access
SSC_CMDBIT_SYS_06	Reserved
SSC_CMDBIT_SYS_07	
SSC_CMDBIT_SYS_08	

Bit number (constant)	Signal name
SSC_CMDBIT_SYS_SMPS	Sampling start
SSC_CMDBIT_SYS_10	Reserved
SSC_CMDBIT_SYS_11	
SSC_CMDBIT_SYS_12	
SSC_CMDBIT_SYS_13	
SSC_CMDBIT_SYS_14	
SSC_CMDBIT_SYS_15	
SSC_CMDBIT_SYS_16	

Bit number (constant)	Signal name
SSC_CMDBIT_SYS_SEMI	Software forced stop
SSC_CMDBIT_SYS_18	Reserved
SSC_CMDBIT_SYS_19	
SSC_CMDBIT_SYS_20	
SSC_CMDBIT_SYS_21	
SSC_CMDBIT_SYS_22	
SSC_CMDBIT_SYS_23	
SSC_CMDBIT_SYS_24	

Bit number (constant)	Signal name
SSC_CMDBIT_SYS_25	Reserved
SSC_CMDBIT_SYS_26	
SSC_CMDBIT_SYS_27	
SSC_CMDBIT_SYS_28	
SSC_CMDBIT_SYS_29	
SSC_CMDBIT_SYS_30	
SSC_CMDBIT_SYS_31	
SSC_CMDBIT_SYS_32	

Bit number (constant)	Signal name
SSC_CMDBIT_SYS_ITFE	Interrupt processing high speed complete
SSC_CMDBIT_SYS_34	Reserved
SSC_CMDBIT_SYS_35	
SSC_CMDBIT_SYS_36	
SSC_CMDBIT_SYS_37	
SSC_CMDBIT_SYS_38	
SSC_CMDBIT_SYS_39	
SSC_CMDBIT_SYS_40	

Bit number (constant)	Signal name
SSC_CMDBIT_SYS_41	Reserved
SSC_CMDBIT_SYS_42	
SSC_CMDBIT_SYS_43	
SSC_CMDBIT_SYS_44	
SSC_CMDBIT_SYS_45	
SSC_CMDBIT_SYS_46	
SSC_CMDBIT_SYS_47	
SSC_CMDBIT_SYS_48	

Bit number (constant)	Signal name
SSC_CMDBIT_SYS_ASYN1	Non-synchronous command (group 1)
SSC_CMDBIT_SYS_ASYN2	Non-synchronous command (group 2)
SSC_CMDBIT_SYS_ASYN3	Non-synchronous command (group 3)
SSC_CMDBIT_SYS_ASYN4	Non-synchronous command (group 4)
SSC_CMDBIT_SYS_ASYN5	Non-synchronous command (group 5)
SSC_CMDBIT_SYS_ASYN6	Non-synchronous command (group 6)
SSC_CMDBIT_SYS_ASYN7	Non-synchronous command (group 7)
SSC_CMDBIT_SYS_ASYN8	Non-synchronous command (group 8)

Bit number (constant)	Signal name
SSC_CMDBIT_SYS_57	Reserved
SSC_CMDBIT_SYS_58	
SSC_CMDBIT_SYS_59	
SSC_CMDBIT_SYS_60	
SSC_CMDBIT_SYS_61	
SSC_CMDBIT_SYS_62	
SSC_CMDBIT_SYS_63	
SSC_CMDBIT_SYS_64	

Bit number (constant)	Signal name
SSC_CMDBIT_SYS_RBR	Reboot preparation
SSC_CMDBIT_SYS_RBS	Execution of reboot
SSC_CMDBIT_SYS_CRST	System alarm reset
SSC_CMDBIT_SYS_68	Reserved
SSC_CMDBIT_SYS_SMON	Monitor command
SSC_CMDBIT_SYS_SMONR	Monitor latch command
SSC_CMDBIT_SYS_71	Reserved
SSC_CMDBIT_SYS_72	

Bit number (constant)	Signal name
SSC_CMDBIT_SYS_73	Reserved
SSC_CMDBIT_SYS_74	
SSC_CMDBIT_SYS_75	
SSC_CMDBIT_SYS_76	
SSC_CMDBIT_SYS_77	
SSC_CMDBIT_SYS_78	
SSC_CMDBIT_SYS_79	
SSC_CMDBIT_SYS_80	

Bit number (constant)	Signal name
SSC_CMDBIT_SYS_LOGC	Log command
SSC_CMDBIT_SYS_LOGR	Reading of log data command
SSC_CMDBIT_SYS_83	Reserved
SSC_CMDBIT_SYS_LOGI	Log data initialization command
SSC_CMDBIT_SYS_85	Reserved
SSC_CMDBIT_SYS_OCMC	Operation cycle monitor clear
SSC_CMDBIT_SYS_87	Reserved
SSC_CMDBIT_SYS_88	

Bit number (constant)	Signal name
SSC_CMDBIT_SYS_RCC	Reconnection command
SSC_CMDBIT_SYS_90	Reserved
SSC_CMDBIT_SYS_91	
SSC_CMDBIT_SYS_CCC	Disconnection command
SSC_CMDBIT_SYS_93	Reserved
SSC_CMDBIT_SYS_94	
SSC_CMDBIT_SYS_95	
SSC_CMDBIT_SYS_96	

Bit number (constant)	Signal name
SSC_CMDBIT_SYS_97	Reserved
SSC_CMDBIT_SYS_98	
SSC_CMDBIT_SYS_99	
SSC_CMDBIT_SYS_100	
SSC_CMDBIT_SYS_101	
SSC_CMDBIT_SYS_102	
SSC_CMDBIT_SYS_103	
SSC_CMDBIT_SYS_104	

Bit number (constant)	Signal name
SSC_CMDBIT_SYS_105	Reserved
SSC_CMDBIT_SYS_106	
SSC_CMDBIT_SYS_107	
SSC_CMDBIT_SYS_108	
SSC_CMDBIT_SYS_109	
SSC_CMDBIT_SYS_110	
SSC_CMDBIT_SYS_111	
SSC_CMDBIT_SYS_112	

Bit number (constant)	Signal name
SSC_CMDBIT_SYS_113	Reserved
SSC_CMDBIT_SYS_114	
SSC_CMDBIT_SYS_115	
SSC_CMDBIT_SYS_116	
SSC_CMDBIT_SYS_117	
SSC_CMDBIT_SYS_118	
SSC_CMDBIT_SYS_119	
SSC_CMDBIT_SYS_120	

Bit number (constant)	Signal name
SSC_CMDBIT_SYS_121	Reserved
SSC_CMDBIT_SYS_122	
SSC_CMDBIT_SYS_123	
SSC_CMDBIT_SYS_124	
SSC_CMDBIT_SYS_125	
SSC_CMDBIT_SYS_126	
SSC_CMDBIT_SYS_127	
SSC_CMDBIT_SYS_128	

Bit number (constant)	Signal name
SSC_CMDBIT_SYS_SPWRT	Parameter write command
SSC_CMDBIT_SYS_130	Reserved
SSC_CMDBIT_SYS_131	
SSC_CMDBIT_SYS_132	
SSC_CMDBIT_SYS_133	
SSC_CMDBIT_SYS_134	
SSC_CMDBIT_SYS_135	
SSC_CMDBIT_SYS_136	

Bit number (constant)	Signal name
SSC_CMDBIT_SYS_SPRD	Parameter read command
SSC_CMDBIT_SYS_138	Reserved
SSC_CMDBIT_SYS_139	
SSC_CMDBIT_SYS_140	
SSC_CMDBIT_SYS_141	
SSC_CMDBIT_SYS_142	
SSC_CMDBIT_SYS_143	
SSC_CMDBIT_SYS_144	

Bit number (constant)	Signal name
SSC_CMDBIT_SYS_SMPSW	Sampling setting write command
SSC_CMDBIT_SYS_146	Reserved
SSC_CMDBIT_SYS_147	
SSC_CMDBIT_SYS_148	
SSC_CMDBIT_SYS_SMPSR	Sampling setting read command
SSC_CMDBIT_SYS_150	Reserved
SSC_CMDBIT_SYS_151	
SSC_CMDBIT_SYS_152	

Bit number (constant)	Signal name
SSC_CMDBIT_SYS_153	Reserved
SSC_CMDBIT_SYS_154	
SSC_CMDBIT_SYS_155	
SSC_CMDBIT_SYS_156	
SSC_CMDBIT_SYS_157	
SSC_CMDBIT_SYS_158	
SSC_CMDBIT_SYS_159	
SSC_CMDBIT_SYS_160	

Bit number (constant)	Signal name
SSC_CMDBIT_SYS_161	Reserved
SSC_CMDBIT_SYS_162	
SSC_CMDBIT_SYS_163	
SSC_CMDBIT_SYS_164	
SSC_CMDBIT_SYS_165	
SSC_CMDBIT_SYS_166	
SSC_CMDBIT_SYS_167	
SSC_CMDBIT_SYS_168	

Bit number (constant)	Signal name
SSC_CMDBIT_SYS_169	Reserved
SSC_CMDBIT_SYS_170	
SSC_CMDBIT_SYS_171	
SSC_CMDBIT_SYS_172	
SSC_CMDBIT_SYS_173	
SSC_CMDBIT_SYS_174	
SSC_CMDBIT_SYS_175	
SSC_CMDBIT_SYS_176	

Bit number (constant)	Signal name
SSC_CMDBIT_SYS_FTR	Flash ROM transfer preparation
SSC_CMDBIT_SYS_FTS	Flash ROM transfer execution
SSC_CMDBIT_SYS_179	Reserved
SSC_CMDBIT_SYS_180	
SSC_CMDBIT_SYS_FIR	Flash ROM initialization preparation
SSC_CMDBIT_SYS_FIS	Flash ROM initialization execution
SSC_CMDBIT_SYS_183	Reserved
SSC_CMDBIT_SYS_184	

Bit number (constant)	Signal name
SSC_CMDBIT_SYS_ALHR	Alarm history read command
SSC_CMDBIT_SYS_186	Reserved
SSC_CMDBIT_SYS_ALHI	Alarm history initialization command
SSC_CMDBIT_SYS_188	Reserved
SSC_CMDBIT_SYS_189	
SSC_CMDBIT_SYS_190	
SSC_CMDBIT_SYS_191	
SSC_CMDBIT_SYS_192	

Bit number (constant)	Signal name
SSC_CMDBIT_SYS_193	Reserved
SSC_CMDBIT_SYS_194	
SSC_CMDBIT_SYS_195	
SSC_CMDBIT_SYS_196	
SSC_CMDBIT_SYS_197	
SSC_CMDBIT_SYS_198	
SSC_CMDBIT_SYS_199	
SSC_CMDBIT_SYS_200	

Bit number (constant)	Signal name
SSC_CMDBIT_SYS_201	Reserved
SSC_CMDBIT_SYS_202	
SSC_CMDBIT_SYS_203	
SSC_CMDBIT_SYS_204	
SSC_CMDBIT_SYS_205	
SSC_CMDBIT_SYS_206	
SSC_CMDBIT_SYS_207	
SSC_CMDBIT_SYS_208	

Bit number (constant)	Signal name
SSC_CMDBIT_SYS_209	Reserved
SSC_CMDBIT_SYS_210	
SSC_CMDBIT_SYS_211	
SSC_CMDBIT_SYS_212	
SSC_CMDBIT_SYS_213	
SSC_CMDBIT_SYS_214	
SSC_CMDBIT_SYS_215	
SSC_CMDBIT_SYS_216	

Bit number (constant)	Signal name
SSC_CMDBIT_SYS_217	Reserved
SSC_CMDBIT_SYS_218	
SSC_CMDBIT_SYS_219	
SSC_CMDBIT_SYS_220	
SSC_CMDBIT_SYS_221	
SSC_CMDBIT_SYS_222	
SSC_CMDBIT_SYS_223	
SSC_CMDBIT_SYS_224	

Bit number (constant)	Signal name
SSC_CMDBIT_SYS_225	Reserved
SSC_CMDBIT_SYS_226	
SSC_CMDBIT_SYS_227	
SSC_CMDBIT_SYS_228	
SSC_CMDBIT_SYS_229	
SSC_CMDBIT_SYS_230	
SSC_CMDBIT_SYS_231	
SSC_CMDBIT_SYS_232	

Bit number (constant)	Signal name
SSC_CMDBIT_SYS_233	Reserved
SSC_CMDBIT_SYS_234	
SSC_CMDBIT_SYS_235	
SSC_CMDBIT_SYS_236	
SSC_CMDBIT_SYS_237	
SSC_CMDBIT_SYS_238	
SSC_CMDBIT_SYS_239	
SSC_CMDBIT_SYS_240	

Bit number (constant)	Signal name
SSC_CMDBIT_SYS_241	Reserved
SSC_CMDBIT_SYS_242	
SSC_CMDBIT_SYS_243	
SSC_CMDBIT_SYS_244	
SSC_CMDBIT_SYS_245	
SSC_CMDBIT_SYS_246	
SSC_CMDBIT_SYS_247	
SSC_CMDBIT_SYS_248	

Bit number (constant)	Signal name
SSC_CMDBIT_SYS_249	Reserved
SSC_CMDBIT_SYS_250	
SSC_CMDBIT_SYS_251	
SSC_CMDBIT_SYS_252	
SSC_CMDBIT_SYS_253	
SSC_CMDBIT_SYS_254	
SSC_CMDBIT_SYS_255	
SSC_CMDBIT_SYS_256	

Bit number (constant)	Signal name
SSC_CMDBIT_SYS_257	Reserved
SSC_CMDBIT_SYS_258	
SSC_CMDBIT_SYS_259	
SSC_CMDBIT_SYS_260	
SSC_CMDBIT_SYS_261	
SSC_CMDBIT_SYS_262	
SSC_CMDBIT_SYS_263	
SSC_CMDBIT_SYS_264	

Bit number (constant)	Signal name
SSC_CMDBIT_SYS_265	Reserved
SSC_CMDBIT_SYS_266	
SSC_CMDBIT_SYS_267	
SSC_CMDBIT_SYS_268	
SSC_CMDBIT_SYS_269	
SSC_CMDBIT_SYS_270	
SSC_CMDBIT_SYS_271	
SSC_CMDBIT_SYS_272	

Bit number (constant)	Signal name
SSC_CMDBIT_SYS_273	Reserved
SSC_CMDBIT_SYS_274	
SSC_CMDBIT_SYS_275	
SSC_CMDBIT_SYS_276	
SSC_CMDBIT_SYS_277	
SSC_CMDBIT_SYS_278	
SSC_CMDBIT_SYS_279	
SSC_CMDBIT_SYS_280	

Bit number (constant)	Signal name
SSC_CMDBIT_SYS_281	Reserved
SSC_CMDBIT_SYS_282	
SSC_CMDBIT_SYS_283	
SSC_CMDBIT_SYS_284	
SSC_CMDBIT_SYS_285	
SSC_CMDBIT_SYS_286	
SSC_CMDBIT_SYS_287	
SSC_CMDBIT_SYS_288	

Bit number (constant)	Signal name
SSC_CMDBIT_SYS_289	Reserved
SSC_CMDBIT_SYS_290	
SSC_CMDBIT_SYS_291	
SSC_CMDBIT_SYS_292	
SSC_CMDBIT_SYS_293	
SSC_CMDBIT_SYS_294	
SSC_CMDBIT_SYS_295	
SSC_CMDBIT_SYS_296	

Bit number (constant)	Signal name
SSC_CMDBIT_SYS_297	Reserved
SSC_CMDBIT_SYS_298	
SSC_CMDBIT_SYS_299	
SSC_CMDBIT_SYS_300	
SSC_CMDBIT_SYS_301	
SSC_CMDBIT_SYS_302	
SSC_CMDBIT_SYS_303	
SSC_CMDBIT_SYS_304	

Bit number (constant)	Signal name
SSC_CMDBIT_SYS_305	Reserved
SSC_CMDBIT_SYS_306	
SSC_CMDBIT_SYS_307	
SSC_CMDBIT_SYS_308	
SSC_CMDBIT_SYS_309	
SSC_CMDBIT_SYS_310	
SSC_CMDBIT_SYS_311	
SSC_CMDBIT_SYS_312	

Bit number (constant)	Signal name
SSC_CMDBIT_SYS_313	Reserved
SSC_CMDBIT_SYS_314	
SSC_CMDBIT_SYS_315	
SSC_CMDBIT_SYS_316	
SSC_CMDBIT_SYS_317	
SSC_CMDBIT_SYS_318	
SSC_CMDBIT_SYS_319	
SSC_CMDBIT_SYS_320	

Bit number (constant)	Signal name
SSC_CMDBIT_SYS_321	Reserved
SSC_CMDBIT_SYS_322	
SSC_CMDBIT_SYS_323	
SSC_CMDBIT_SYS_324	
SSC_CMDBIT_SYS_325	
SSC_CMDBIT_SYS_326	
SSC_CMDBIT_SYS_327	
SSC_CMDBIT_SYS_328	

Bit number (constant)	Signal name
SSC_CMDBIT_SYS_329	Reserved
SSC_CMDBIT_SYS_330	
SSC_CMDBIT_SYS_331	
SSC_CMDBIT_SYS_332	
SSC_CMDBIT_SYS_333	
SSC_CMDBIT_SYS_334	
SSC_CMDBIT_SYS_335	
SSC_CMDBIT_SYS_336	

Bit number (constant)	Signal name
SSC_CMDBIT_SYS_337	Reserved
SSC_CMDBIT_SYS_338	
SSC_CMDBIT_SYS_339	
SSC_CMDBIT_SYS_340	
SSC_CMDBIT_SYS_341	
SSC_CMDBIT_SYS_342	
SSC_CMDBIT_SYS_343	
SSC_CMDBIT_SYS_344	

Bit number (constant)	Signal name
SSC_CMDBIT_SYS_345	Reserved
SSC_CMDBIT_SYS_346	
SSC_CMDBIT_SYS_347	
SSC_CMDBIT_SYS_348	
SSC_CMDBIT_SYS_349	
SSC_CMDBIT_SYS_350	
SSC_CMDBIT_SYS_351	
SSC_CMDBIT_SYS_352	

Bit number (constant)	Signal name
SSC_CMDBIT_SYS_353	Reserved
SSC_CMDBIT_SYS_354	
SSC_CMDBIT_SYS_355	
SSC_CMDBIT_SYS_356	
SSC_CMDBIT_SYS_357	
SSC_CMDBIT_SYS_358	
SSC_CMDBIT_SYS_359	
SSC_CMDBIT_SYS_360	

Bit number (constant)	Signal name
SSC_CMDBIT_SYS_361	Reserved
SSC_CMDBIT_SYS_362	
SSC_CMDBIT_SYS_363	
SSC_CMDBIT_SYS_364	
SSC_CMDBIT_SYS_365	
SSC_CMDBIT_SYS_366	
SSC_CMDBIT_SYS_367	
SSC_CMDBIT_SYS_368	

Bit number (constant)	Signal name
SSC_CMDBIT_SYS_369	Reserved
SSC_CMDBIT_SYS_370	
SSC_CMDBIT_SYS_371	
SSC_CMDBIT_SYS_372	
SSC_CMDBIT_SYS_373	
SSC_CMDBIT_SYS_374	
SSC_CMDBIT_SYS_375	
SSC_CMDBIT_SYS_376	

Bit number (constant)	Signal name
SSC_CMDBIT_SYS_377	Reserved
SSC_CMDBIT_SYS_378	
SSC_CMDBIT_SYS_379	
SSC_CMDBIT_SYS_380	
SSC_CMDBIT_SYS_381	
SSC_CMDBIT_SYS_382	
SSC_CMDBIT_SYS_383	
SSC_CMDBIT_SYS_384	

6.2 System status bit

Bit number (constant)	Signal name
SSC_STSBIT_SYS_ITO	Outputting with factor of interrupt
SSC_STSBIT_SYS_IITO	During interface mode interrupt valid
SSC_STSBIT_SYS_EVDO	Event detect enabled
SSC_STSBIT_SYS_HRIF	During highly response I/F valid
SSC_STSBIT_SYS_BMA	During system program memory access
SSC_STSBIT_SYS_PRINF	Continuous operation to torque control compatible information
SSC_STSBIT_SYS_07	Reserved
SSC_STSBIT_SYS_IFMO	In interface mode

Bit number (constant)	Signal name
SSC_STSBIT_SYS_SMPW	Waiting for sampling trigger
SSC_STSBIT_SYS_SMPO	Sampling is being performed
SSC_STSBIT_SYS_SMPF	Sampling is complete
SSC_STSBIT_SYS_SMPE	Sampling error
SSC_STSBIT_SYS_13	Reserved
SSC_STSBIT_SYS_AHINF	Alarm history information
SSC_STSBIT_SYS_15	Reserved
SSC_STSBIT_SYS_16	

Bit number (constant)	Signal name
SSC_STSBIT_SYS_EMIO	During forced stop
SSC_STSBIT_SYS_18	Reserved
SSC_STSBIT_SYS_TSTO	In test mode
SSC_STSBIT_SYS_20	Reserved
SSC_STSBIT_SYS_21	
SSC_STSBIT_SYS_22	
SSC_STSBIT_SYS_EMID	External forced stop disabled
SSC_STSBIT_SYS_24	Reserved

Bit number (constant)	Signal name
SSC_STSBIT_SYS_25	Reserved
SSC_STSBIT_SYS_26	
SSC_STSBIT_SYS_27	
SSC_STSBIT_SYS_28	
SSC_STSBIT_SYS_29	
SSC_STSBIT_SYS_30	
SSC_STSBIT_SYS_IPCH	Changeable interpolation group
SSC_STSBIT_SYS_32	Reserved

Bit number (constant)	Signal name
SSC_STSBIT_SYS_33	Reserved
SSC_STSBIT_SYS_34	
SSC_STSBIT_SYS_35	
SSC_STSBIT_SYS_36	
SSC_STSBIT_SYS_37	
SSC_STSBIT_SYS_38	
SSC_STSBIT_SYS_39	
SSC_STSBIT_SYS_40	

Bit number (constant)	Signal name
SSC_STSBIT_SYS_41	Reserved
SSC_STSBIT_SYS_42	
SSC_STSBIT_SYS_43	
SSC_STSBIT_SYS_44	
SSC_STSBIT_SYS_45	
SSC_STSBIT_SYS_46	
SSC_STSBIT_SYS_47	
SSC_STSBIT_SYS_48	

Bit number (constant)	Signal name
SSC_STSBIT_SYS_ASYO1	In non-synchronous mode (group 1)
SSC_STSBIT_SYS_ASYO2	In non-synchronous mode (group 2)
SSC_STSBIT_SYS_ASYO3	In non-synchronous mode (group 3)
SSC_STSBIT_SYS_ASYO4	In non-synchronous mode (group 4)
SSC_STSBIT_SYS_ASYO5	In non-synchronous mode (group 5)
SSC_STSBIT_SYS_ASYO6	In non-synchronous mode (group 6)
SSC_STSBIT_SYS_ASYO7	In non-synchronous mode (group 7)
SSC_STSBIT_SYS_ASYO8	In non-synchronous mode (group 8)

Bit number (constant)	Signal name
SSC_STSBIT_SYS_SYEO1	Synchronizing (group 1)
SSC_STSBIT_SYS_SYEO2	Synchronizing (group 2)
SSC_STSBIT_SYS_SYEO3	Synchronizing (group 3)
SSC_STSBIT_SYS_SYEO4	Synchronizing (group 4)
SSC_STSBIT_SYS_SYEO5	Synchronizing (group 5)
SSC_STSBIT_SYS_SYEO6	Synchronizing (group 6)
SSC_STSBIT_SYS_SYEO7	Synchronizing (group 7)
SSC_STSBIT_SYS_SYEO8	Synchronizing (group 8)

Bit number (constant)	Signal name
SSC_STSBIT_SYS_RBOK	Reboot preparation complete
SSC_STSBIT_SYS_RBNG	Reboot preparation error
SSC_STSBIT_SYS_CALM	Current system alarm
SSC_STSBIT_SYS_68	Reserved
SSC_STSBIT_SYS_SMOUT	Monitor output
SSC_STSBIT_SYS_SMRCH	Monitor latch
SSC_STSBIT_SYS_SMER1	Monitor number error 1
SSC_STSBIT_SYS_SMER2	Monitor number error 2

Bit number (constant)	Signal name
SSC_STSBIT_SYS_73	Reserved
SSC_STSBIT_SYS_74	
SSC_STSBIT_SYS_75	
SSC_STSBIT_SYS_76	
SSC_STSBIT_SYS_77	
SSC_STSBIT_SYS_78	
SSC_STSBIT_SYS_79	
SSC_STSBIT_SYS_80	

Bit number (constant)	Signal name
SSC_STSBIT_SYS_LOGO	Log operation being performed
SSC_STSBIT_SYS_LOGRF	Reading of log data complete
SSC_STSBIT_SYS_LOGRE	Reading of log data error
SSC_STSBIT_SYS_LOGIF	Log data initialization is complete
SSC_STSBIT_SYS_LOGIE	Log data initialization error
SSC_STSBIT_SYS_OCMCO	During operation cycle monitor clear
SSC_STSBIT_SYS_OCME	Operation cycle alarm
SSC_STSBIT_SYS_OCMW	Operation cycle warning

Bit number (constant)	Signal name
SSC_STSBIT_SYS_RCO	During reconnection processing
SSC_STSBIT_SYS_RCF	Reconnection complete
SSC_STSBIT_SYS_RCE	Reconnection error
SSC_STSBIT_SYS_CCO	During disconnection processing
SSC_STSBIT_SYS_CCF	Disconnection complete
SSC_STSBIT_SYS_CCE	Disconnection error
SSC_STSBIT_SYS_95	Reserved
SSC_STSBIT_SYS_96	

Bit number (constant)	Signal name
SSC_STSBIT_SYS_97	Reserved
SSC_STSBIT_SYS_98	
SSC_STSBIT_SYS_99	
SSC_STSBIT_SYS_100	
SSC_STSBIT_SYS_101	
SSC_STSBIT_SYS_102	
SSC_STSBIT_SYS_103	
SSC_STSBIT_SYS_104	

Bit number (constant)	Signal name
SSC_STSBIT_SYS_105	Reserved
SSC_STSBIT_SYS_106	
SSC_STSBIT_SYS_107	
SSC_STSBIT_SYS_108	
SSC_STSBIT_SYS_109	
SSC_STSBIT_SYS_110	
SSC_STSBIT_SYS_111	
SSC_STSBIT_SYS_112	

Bit number (constant)	Signal name
SSC_STSBIT_SYS_113	Reserved
SSC_STSBIT_SYS_114	
SSC_STSBIT_SYS_115	
SSC_STSBIT_SYS_116	
SSC_STSBIT_SYS_117	
SSC_STSBIT_SYS_118	
SSC_STSBIT_SYS_119	
SSC_STSBIT_SYS_120	

Bit number (constant)	Signal name
SSC_STSBIT_SYS_121	Reserved
SSC_STSBIT_SYS_122	
SSC_STSBIT_SYS_123	
SSC_STSBIT_SYS_124	
SSC_STSBIT_SYS_125	
SSC_STSBIT_SYS_126	
SSC_STSBIT_SYS_127	
SSC_STSBIT_SYS_128	

Bit number (constant)	Signal name
SSC_STSBIT_SYS_SPWFIN1	Parameter write complete 1
SSC_STSBIT_SYS_SPWEN1	Parameter number error 1
SSC_STSBIT_SYS_SPWED1	Parameter data out of bounds 1
SSC_STSBIT_SYS_132	Reserved
SSC_STSBIT_SYS_SPWFIN2	Parameter write complete 2
SSC_STSBIT_SYS_SPWEN2	Parameter number error 2
SSC_STSBIT_SYS_SPWED2	Parameter data out of bounds 2
SSC_STSBIT_SYS_136	Reserved

Bit number (constant)	Signal name
SSC_STSBIT_SYS_SPRFIN1	Parameter read complete 1
SSC_STSBIT_SYS_SPREN1	Parameter number error 1
SSC_STSBIT_SYS_SPRFIN2	Parameter read complete 2
SSC_STSBIT_SYS_SPREN2	Parameter number error 2
SSC_STSBIT_SYS_141	Reserved
SSC_STSBIT_SYS_142	
SSC_STSBIT_SYS_143	
SSC_STSBIT_SYS_144	

Bit number (constant)	Signal name
SSC_STSBIT_SYS_SWFIN	Sampling setting write complete
SSC_STSBIT_SYS_SWEN	Sampling setting number error
SSC_STSBIT_SYS_SWED	Sampling setting data out of bounds
SSC_STSBIT_SYS_148	Reserved
SSC_STSBIT_SYS_SRFIN	Sampling setting read complete
SSC_STSBIT_SYS_SREN	Sampling setting number error
SSC_STSBIT_SYS_151	Reserved
SSC_STSBIT_SYS_152	

Bit number (constant)	Signal name
SSC_STSBIT_SYS_153	Reserved
SSC_STSBIT_SYS_154	
SSC_STSBIT_SYS_155	
SSC_STSBIT_SYS_156	
SSC_STSBIT_SYS_157	
SSC_STSBIT_SYS_158	
SSC_STSBIT_SYS_159	
SSC_STSBIT_SYS_160	

Bit number (constant)	Signal name
SSC_STSBIT_SYS_161	Reserved
SSC_STSBIT_SYS_162	
SSC_STSBIT_SYS_163	
SSC_STSBIT_SYS_164	
SSC_STSBIT_SYS_165	
SSC_STSBIT_SYS_166	
SSC_STSBIT_SYS_167	
SSC_STSBIT_SYS_168	

Bit number (constant)	Signal name
SSC_STSBIT_SYS_169	Reserved
SSC_STSBIT_SYS_170	
SSC_STSBIT_SYS_171	
SSC_STSBIT_SYS_172	
SSC_STSBIT_SYS_173	
SSC_STSBIT_SYS_174	
SSC_STSBIT_SYS_175	
SSC_STSBIT_SYS_176	

Bit number (constant)	Signal name
SSC_STSBIT_SYS_FROK	Flash ROM transfer preparation complete
SSC_STSBIT_SYS_FRNG	Flash ROM transfer preparation error
SSC_STSBIT_SYS_FSOK	Flash ROM transfer complete
SSC_STSBIT_SYS_FSNG	Flash ROM transfer error
SSC_STSBIT_SYS_FIROK	Flash ROM initialization preparation complete
SSC_STSBIT_SYS_FIRNG	Flash ROM initialization preparation error
SSC_STSBIT_SYS_FIOK	Flash ROM initialization complete
SSC_STSBIT_SYS_FING	Flash ROM initialization error

Bit number (constant)	Signal name
SSC_STSBIT_SYS_ALHRF	Alarm history read complete
SSC_STSBIT_SYS_ALHRE	Alarm history read error
SSC_STSBIT_SYS_ALHIF	Alarm history initialization complete
SSC_STSBIT_SYS_ALHIE	Alarm history initialization error
SSC_STSBIT_SYS_189	Reserved
SSC_STSBIT_SYS_190	
SSC_STSBIT_SYS_191	
SSC_STSBIT_SYS_192	

Bit number (constant)	Signal name
SSC_STSBIT_SYS_193	Reserved
SSC_STSBIT_SYS_194	
SSC_STSBIT_SYS_195	
SSC_STSBIT_SYS_196	
SSC_STSBIT_SYS_197	
SSC_STSBIT_SYS_198	
SSC_STSBIT_SYS_199	
SSC_STSBIT_SYS_200	

Bit number (constant)	Signal name
SSC_STSBIT_SYS_201	Reserved
SSC_STSBIT_SYS_202	
SSC_STSBIT_SYS_203	
SSC_STSBIT_SYS_204	
SSC_STSBIT_SYS_205	
SSC_STSBIT_SYS_206	
SSC_STSBIT_SYS_207	
SSC_STSBIT_SYS_208	

Bit number (constant)	Signal name
SSC_STSBIT_SYS_209	Reserved
SSC_STSBIT_SYS_210	
SSC_STSBIT_SYS_211	
SSC_STSBIT_SYS_212	
SSC_STSBIT_SYS_213	
SSC_STSBIT_SYS_214	
SSC_STSBIT_SYS_215	
SSC_STSBIT_SYS_216	

Bit number (constant)	Signal name
SSC_STSBIT_SYS_217	Reserved
SSC_STSBIT_SYS_218	
SSC_STSBIT_SYS_219	
SSC_STSBIT_SYS_220	
SSC_STSBIT_SYS_221	
SSC_STSBIT_SYS_222	
SSC_STSBIT_SYS_223	
SSC_STSBIT_SYS_224	

Bit number (constant)	Signal name
SSC_STSBIT_SYS_225	Reserved
SSC_STSBIT_SYS_226	
SSC_STSBIT_SYS_227	
SSC_STSBIT_SYS_228	
SSC_STSBIT_SYS_229	
SSC_STSBIT_SYS_230	
SSC_STSBIT_SYS_231	
SSC_STSBIT_SYS_232	

Bit number (constant)	Signal name
SSC_STSBIT_SYS_233	Reserved
SSC_STSBIT_SYS_234	
SSC_STSBIT_SYS_235	
SSC_STSBIT_SYS_236	
SSC_STSBIT_SYS_237	
SSC_STSBIT_SYS_238	
SSC_STSBIT_SYS_239	
SSC_STSBIT_SYS_240	

Bit number (constant)	Signal name
SSC_STSBIT_SYS_241	Reserved
SSC_STSBIT_SYS_242	
SSC_STSBIT_SYS_243	
SSC_STSBIT_SYS_244	
SSC_STSBIT_SYS_245	
SSC_STSBIT_SYS_246	
SSC_STSBIT_SYS_247	
SSC_STSBIT_SYS_248	

Bit number (constant)	Signal name
SSC_STSBIT_SYS_249	Reserved
SSC_STSBIT_SYS_250	
SSC_STSBIT_SYS_251	
SSC_STSBIT_SYS_252	
SSC_STSBIT_SYS_253	
SSC_STSBIT_SYS_254	
SSC_STSBIT_SYS_255	
SSC_STSBIT_SYS_256	

Bit number (constant)	Signal name
SSC_STSBIT_SYS_257	Reserved
SSC_STSBIT_SYS_258	
SSC_STSBIT_SYS_259	
SSC_STSBIT_SYS_260	
SSC_STSBIT_SYS_261	
SSC_STSBIT_SYS_262	
SSC_STSBIT_SYS_263	
SSC_STSBIT_SYS_264	

Bit number (constant)	Signal name
SSC_STSBIT_SYS_265	Reserved
SSC_STSBIT_SYS_266	
SSC_STSBIT_SYS_267	
SSC_STSBIT_SYS_268	
SSC_STSBIT_SYS_269	
SSC_STSBIT_SYS_270	
SSC_STSBIT_SYS_271	
SSC_STSBIT_SYS_272	

Bit number (constant)	Signal name
SSC_STSBIT_SYS_273	Reserved
SSC_STSBIT_SYS_274	
SSC_STSBIT_SYS_275	
SSC_STSBIT_SYS_276	
SSC_STSBIT_SYS_277	
SSC_STSBIT_SYS_278	
SSC_STSBIT_SYS_279	
SSC_STSBIT_SYS_280	

Bit number (constant)	Signal name
SSC_STSBIT_SYS_281	Reserved
SSC_STSBIT_SYS_282	
SSC_STSBIT_SYS_283	
SSC_STSBIT_SYS_284	
SSC_STSBIT_SYS_285	
SSC_STSBIT_SYS_286	
SSC_STSBIT_SYS_287	
SSC_STSBIT_SYS_288	

Bit number (constant)	Signal name
SSC_STSBIT_SYS_289	Reserved
SSC_STSBIT_SYS_290	
SSC_STSBIT_SYS_291	
SSC_STSBIT_SYS_292	
SSC_STSBIT_SYS_293	
SSC_STSBIT_SYS_294	
SSC_STSBIT_SYS_295	
SSC_STSBIT_SYS_296	

Bit number (constant)	Signal name
SSC_STSBIT_SYS_297	Reserved
SSC_STSBIT_SYS_298	
SSC_STSBIT_SYS_299	
SSC_STSBIT_SYS_300	
SSC_STSBIT_SYS_301	
SSC_STSBIT_SYS_302	
SSC_STSBIT_SYS_303	
SSC_STSBIT_SYS_304	

Bit number (constant)	Signal name
SSC_STSBIT_SYS_305	Reserved
SSC_STSBIT_SYS_306	
SSC_STSBIT_SYS_307	
SSC_STSBIT_SYS_308	
SSC_STSBIT_SYS_309	
SSC_STSBIT_SYS_310	
SSC_STSBIT_SYS_311	
SSC_STSBIT_SYS_312	

Bit number (constant)	Signal name
SSC_STSBIT_SYS_313	Reserved
SSC_STSBIT_SYS_314	
SSC_STSBIT_SYS_315	
SSC_STSBIT_SYS_316	
SSC_STSBIT_SYS_317	
SSC_STSBIT_SYS_318	
SSC_STSBIT_SYS_319	
SSC_STSBIT_SYS_320	

Bit number (constant)	Signal name
SSC_STSBIT_SYS_321	Reserved
SSC_STSBIT_SYS_322	
SSC_STSBIT_SYS_323	
SSC_STSBIT_SYS_324	
SSC_STSBIT_SYS_325	
SSC_STSBIT_SYS_326	
SSC_STSBIT_SYS_327	
SSC_STSBIT_SYS_328	

Bit number (constant)	Signal name
SSC_STSBIT_SYS_329	Reserved
SSC_STSBIT_SYS_330	
SSC_STSBIT_SYS_331	
SSC_STSBIT_SYS_332	
SSC_STSBIT_SYS_333	
SSC_STSBIT_SYS_334	
SSC_STSBIT_SYS_335	
SSC_STSBIT_SYS_336	

Bit number (constant)	Signal name
SSC_STSBIT_SYS_337	Reserved
SSC_STSBIT_SYS_338	
SSC_STSBIT_SYS_339	
SSC_STSBIT_SYS_340	
SSC_STSBIT_SYS_341	
SSC_STSBIT_SYS_342	
SSC_STSBIT_SYS_343	
SSC_STSBIT_SYS_344	

Bit number (constant)	Signal name
SSC_STSBIT_SYS_345	Reserved
SSC_STSBIT_SYS_346	
SSC_STSBIT_SYS_347	
SSC_STSBIT_SYS_348	
SSC_STSBIT_SYS_349	
SSC_STSBIT_SYS_350	
SSC_STSBIT_SYS_351	
SSC_STSBIT_SYS_352	

Bit number (constant)	Signal name
SSC_STSBIT_SYS_353	Reserved
SSC_STSBIT_SYS_354	
SSC_STSBIT_SYS_355	
SSC_STSBIT_SYS_356	
SSC_STSBIT_SYS_357	
SSC_STSBIT_SYS_358	
SSC_STSBIT_SYS_359	
SSC_STSBIT_SYS_360	

Bit number (constant)	Signal name
SSC_STSBIT_SYS_361	Reserved
SSC_STSBIT_SYS_362	
SSC_STSBIT_SYS_363	
SSC_STSBIT_SYS_364	
SSC_STSBIT_SYS_365	
SSC_STSBIT_SYS_366	
SSC_STSBIT_SYS_367	
SSC_STSBIT_SYS_368	

Bit number (constant)	Signal name
SSC_STSBIT_SYS_369	Reserved
SSC_STSBIT_SYS_370	
SSC_STSBIT_SYS_371	
SSC_STSBIT_SYS_372	
SSC_STSBIT_SYS_373	
SSC_STSBIT_SYS_374	
SSC_STSBIT_SYS_375	
SSC_STSBIT_SYS_376	

Bit number (constant)	Signal name
SSC_STSBIT_SYS_377	Reserved
SSC_STSBIT_SYS_378	
SSC_STSBIT_SYS_379	
SSC_STSBIT_SYS_380	
SSC_STSBIT_SYS_381	
SSC_STSBIT_SYS_382	
SSC_STSBIT_SYS_383	
SSC_STSBIT_SYS_384	

6.3 Axis command bit

Bit number (constant)	Signal name
SSC_CMDBIT_AX_SON	Servo on
SSC_CMDBIT_AX_2	Reserved
SSC_CMDBIT_AX_3	
SSC_CMDBIT_AX_4	
SSC_CMDBIT_AX_TL	Torque limit
SSC_CMDBIT_AX_SRST	Servo alarm reset
SSC_CMDBIT_AX_7	Reserved
SSC_CMDBIT_AX_8	

Bit number (constant)	Signal name
SSC_CMDBIT_AX_ST	Start operation
SSC_CMDBIT_AX_DIR	Movement direction
SSC_CMDBIT_AX_STP	Stop operation
SSC_CMDBIT_AX_RSTP	Rapid stop
SSC_CMDBIT_AX_13	Reserved
SSC_CMDBIT_AX_ORST	Operation alarm reset
SSC_CMDBIT_AX_15	Reserved
SSC_CMDBIT_AX_16	

Bit number (constant)	Signal name
SSC_CMDBIT_AX_AUT	Automatic operation mode
SSC_CMDBIT_AX_ZRN	Home position return mode
SSC_CMDBIT_AX_JOG	JOG operation mode
SSC_CMDBIT_AX_S	Incremental feed mode
SSC_CMDBIT_AX_21	Reserved
SSC_CMDBIT_AX_LIP	Linear interpolation mode MC200 Interpolation operation mode MC300
SSC_CMDBIT_AX_DST	Home position reset mode
SSC_CMDBIT_AX_24	Reserved

Bit number (constant)	Signal name
SSC_CMDBIT_AX_25	Reserved
SSC_CMDBIT_AX_26	
SSC_CMDBIT_AX_27	
SSC_CMDBIT_AX_28	
SSC_CMDBIT_AX_29	
SSC_CMDBIT_AX_30	
SSC_CMDBIT_AX_31	
SSC_CMDBIT_AX_32	

Bit number (constant)	Signal name
SSC_CMDBIT_AX_ITL	Interlock
SSC_CMDBIT_AX_RMONR	High speed monitor latch command
SSC_CMDBIT_AX_35	Reserved
SSC_CMDBIT_AX_36	
SSC_CMDBIT_AX_LSPC	+ side limit switch input
SSC_CMDBIT_AX_LSNC	- side limit switch input
SSC_CMDBIT_AX_DOGC	Proximity dog input
SSC_CMDBIT_AX_40	Reserved

Bit number (constant)	Signal name
SSC_CMDBIT_AX_SCHG	Change speed
SSC_CMDBIT_AX_TACHG	Change acceleration time constant
SSC_CMDBIT_AX_TDCHG	Change deceleration time constant
SSC_CMDBIT_AX_PCHG	Position change
SSC_CMDBIT_AX_45	Reserved
SSC_CMDBIT_AX_46	
SSC_CMDBIT_AX_47	
SSC_CMDBIT_AX_48	

Bit number (constant)	Signal name
SSC_CMDBIT_AX_FST	Fast start operation
SSC_CMDBIT_AX_50	Reserved
SSC_CMDBIT_AX_51	
SSC_CMDBIT_AX_52	
SSC_CMDBIT_AX_53	
SSC_CMDBIT_AX_54	
SSC_CMDBIT_AX_55	
SSC_CMDBIT_AX_56	

Bit number (constant)	Signal name
SSC_CMDBIT_AX_PPISTP	Pass position interrupt cancel
SSC_CMDBIT_AX_58	Reserved
SSC_CMDBIT_AX_59	
SSC_CMDBIT_AX_60	
SSC_CMDBIT_AX_61	
SSC_CMDBIT_AX_62	
SSC_CMDBIT_AX_63	
SSC_CMDBIT_AX_64	

Bit number (constant)	Signal name
SSC_CMDBIT_AX_GAIN	Gain switching command
SSC_CMDBIT_AX_FCLS	Fully closed loop control change command
SSC_CMDBIT_AX_67	Reserved
SSC_CMDBIT_AX_CPC	PID control command
SSC_CMDBIT_AX_69	Reserved
SSC_CMDBIT_AX_70	
SSC_CMDBIT_AX_71	
SSC_CMDBIT_AX_72	

Bit number (constant)	Signal name
SSC_CMDBIT_AX_73	Reserved
SSC_CMDBIT_AX_74	
SSC_CMDBIT_AX_75	
SSC_CMDBIT_AX_76	
SSC_CMDBIT_AX_77	
SSC_CMDBIT_AX_78	
SSC_CMDBIT_AX_79	
SSC_CMDBIT_AX_80	

Bit number (constant)	Signal name
SSC_CMDBIT_AX_81	Reserved
SSC_CMDBIT_AX_82	
SSC_CMDBIT_AX_83	
SSC_CMDBIT_AX_84	
SSC_CMDBIT_AX_ZSC	Home position set command
SSC_CMDBIT_AX_86	Reserved
SSC_CMDBIT_AX_87	
SSC_CMDBIT_AX_88	

Bit number (constant)	Signal name
SSC_CMDBIT_AX_89	Reserved
SSC_CMDBIT_AX_MKC1	Mark detection clear command 1
SSC_CMDBIT_AX_MKD1	Mark detection disable command 1
SSC_CMDBIT_AX_MKSEN1	Mark detection setting enable command 1
SSC_CMDBIT_AX_93	Reserved
SSC_CMDBIT_AX_MKC2	Mark detection clear command 2
SSC_CMDBIT_AX_MKD2	Mark detection disable command 2
SSC_CMDBIT_AX_MKSEN2	Mark detection setting enable command 2

Bit number (constant)	Signal name
SSC_CMDBIT_AX_97	Reserved
SSC_CMDBIT_AX_98	
SSC_CMDBIT_AX_99	
SSC_CMDBIT_AX_100	
SSC_CMDBIT_AX_CTLMC	Control mode switch command
SSC_CMDBIT_AX_102	Reserved
SSC_CMDBIT_AX_103	
SSC_CMDBIT_AX_104	

Bit number (constant)	Signal name
SSC_CMDBIT_AX_105	Reserved
SSC_CMDBIT_AX_106	
SSC_CMDBIT_AX_107	
SSC_CMDBIT_AX_108	
SSC_CMDBIT_AX_109	
SSC_CMDBIT_AX_110	
SSC_CMDBIT_AX_111	
SSC_CMDBIT_AX_112	

Bit number (constant)	Signal name
SSC_CMDBIT_AX_113	Reserved
SSC_CMDBIT_AX_114	
SSC_CMDBIT_AX_115	
SSC_CMDBIT_AX_116	
SSC_CMDBIT_AX_117	
SSC_CMDBIT_AX_118	
SSC_CMDBIT_AX_119	
SSC_CMDBIT_AX_120	

Bit number (constant)	Signal name
SSC_CMDBIT_AX_121	Reserved
SSC_CMDBIT_AX_122	
SSC_CMDBIT_AX_123	
SSC_CMDBIT_AX_124	
SSC_CMDBIT_AX_125	
SSC_CMDBIT_AX_126	
SSC_CMDBIT_AX_127	
SSC_CMDBIT_AX_128	

Bit number (constant)	Signal name
SSC_CMDBIT_AX_MON	Monitor command
SSC_CMDBIT_AX_MONR	Monitor latch command
SSC_CMDBIT_AX_131	Reserved
SSC_CMDBIT_AX_132	
SSC_CMDBIT_AX_133	
SSC_CMDBIT_AX_134	
SSC_CMDBIT_AX_135	
SSC_CMDBIT_AX_136	

Bit number (constant)	Signal name
SSC_CMDBIT_AX_137	Reserved
SSC_CMDBIT_AX_138	
SSC_CMDBIT_AX_139	
SSC_CMDBIT_AX_140	
SSC_CMDBIT_AX_141	
SSC_CMDBIT_AX_142	
SSC_CMDBIT_AX_143	
SSC_CMDBIT_AX_144	

Bit number (constant)	Signal name
SSC_CMDBIT_AX_145	Reserved
SSC_CMDBIT_AX_146	
SSC_CMDBIT_AX_147	
SSC_CMDBIT_AX_148	
SSC_CMDBIT_AX_149	
SSC_CMDBIT_AX_150	
SSC_CMDBIT_AX_151	
SSC_CMDBIT_AX_152	

Bit number (constant)	Signal name
SSC_CMDBIT_AX_153	Reserved
SSC_CMDBIT_AX_154	
SSC_CMDBIT_AX_155	
SSC_CMDBIT_AX_156	
SSC_CMDBIT_AX_157	
SSC_CMDBIT_AX_158	
SSC_CMDBIT_AX_159	
SSC_CMDBIT_AX_160	

Bit number (constant)	Signal name
SSC_CMDBIT_AX_PWRT	Parameter write command
SSC_CMDBIT_AX_162	Reserved
SSC_CMDBIT_AX_163	
SSC_CMDBIT_AX_164	
SSC_CMDBIT_AX_165	
SSC_CMDBIT_AX_166	
SSC_CMDBIT_AX_167	
SSC_CMDBIT_AX_PSF	Servo parameter read complete

Bit number (constant)	Signal name
SSC_CMDBIT_AX_PRD	Parameter read command
SSC_CMDBIT_AX_170	Reserved
SSC_CMDBIT_AX_171	
SSC_CMDBIT_AX_172	
SSC_CMDBIT_AX_173	
SSC_CMDBIT_AX_174	
SSC_CMDBIT_AX_175	
SSC_CMDBIT_AX_176	

Bit number (constant)	Signal name
SSC_CMDBIT_AX_177	Reserved
SSC_CMDBIT_AX_178	
SSC_CMDBIT_AX_179	
SSC_CMDBIT_AX_180	
SSC_CMDBIT_AX_181	
SSC_CMDBIT_AX_182	
SSC_CMDBIT_AX_183	
SSC_CMDBIT_AX_184	

Bit number (constant)	Signal name
SSC_CMDBIT_AX_185	Reserved
SSC_CMDBIT_AX_186	
SSC_CMDBIT_AX_187	
SSC_CMDBIT_AX_188	
SSC_CMDBIT_AX_189	
SSC_CMDBIT_AX_190	
SSC_CMDBIT_AX_191	
SSC_CMDBIT_AX_192	

Bit number (constant)	Signal name
SSC_CMDBIT_AX_193	Reserved
SSC_CMDBIT_AX_194	
SSC_CMDBIT_AX_195	
SSC_CMDBIT_AX_196	
SSC_CMDBIT_AX_197	
SSC_CMDBIT_AX_198	
SSC_CMDBIT_AX_199	
SSC_CMDBIT_AX_200	

Bit number (constant)	Signal name
SSC_CMDBIT_AX_201	Reserved
SSC_CMDBIT_AX_202	
SSC_CMDBIT_AX_203	
SSC_CMDBIT_AX_204	
SSC_CMDBIT_AX_205	
SSC_CMDBIT_AX_206	
SSC_CMDBIT_AX_207	
SSC_CMDBIT_AX_208	

Bit number (constant)	Signal name
SSC_CMDBIT_AX_209	Reserved
SSC_CMDBIT_AX_210	
SSC_CMDBIT_AX_211	
SSC_CMDBIT_AX_212	
SSC_CMDBIT_AX_213	
SSC_CMDBIT_AX_214	
SSC_CMDBIT_AX_215	
SSC_CMDBIT_AX_216	

Bit number (constant)	Signal name
SSC_CMDBIT_AX_217	Reserved
SSC_CMDBIT_AX_218	
SSC_CMDBIT_AX_219	
SSC_CMDBIT_AX_220	
SSC_CMDBIT_AX_221	
SSC_CMDBIT_AX_222	
SSC_CMDBIT_AX_223	
SSC_CMDBIT_AX_224	

Bit number (constant)	Signal name
SSC_CMDBIT_AX_225	Reserved
SSC_CMDBIT_AX_226	
SSC_CMDBIT_AX_227	
SSC_CMDBIT_AX_228	
SSC_CMDBIT_AX_229	
SSC_CMDBIT_AX_230	
SSC_CMDBIT_AX_231	
SSC_CMDBIT_AX_232	

Bit number (constant)	Signal name
SSC_CMDBIT_AX_233	Reserved
SSC_CMDBIT_AX_234	
SSC_CMDBIT_AX_235	
SSC_CMDBIT_AX_236	
SSC_CMDBIT_AX_237	
SSC_CMDBIT_AX_238	
SSC_CMDBIT_AX_239	
SSC_CMDBIT_AX_240	

Bit number (constant)	Signal name
SSC_CMDBIT_AX_241	Reserved
SSC_CMDBIT_AX_242	
SSC_CMDBIT_AX_243	
SSC_CMDBIT_AX_244	
SSC_CMDBIT_AX_245	
SSC_CMDBIT_AX_246	
SSC_CMDBIT_AX_247	
SSC_CMDBIT_AX_248	

Bit number (constant)	Signal name
SSC_CMDBIT_AX_249	Reserved
SSC_CMDBIT_AX_250	
SSC_CMDBIT_AX_251	
SSC_CMDBIT_AX_252	
SSC_CMDBIT_AX_253	
SSC_CMDBIT_AX_254	
SSC_CMDBIT_AX_255	
SSC_CMDBIT_AX_256	

Bit number (constant)	Signal name
SSC_CMDBIT_AX_257	Reserved
SSC_CMDBIT_AX_258	
SSC_CMDBIT_AX_259	
SSC_CMDBIT_AX_260	
SSC_CMDBIT_AX_261	
SSC_CMDBIT_AX_262	
SSC_CMDBIT_AX_263	
SSC_CMDBIT_AX_264	

Bit number (constant)	Signal name
SSC_CMDBIT_AX_265	Reserved
SSC_CMDBIT_AX_266	
SSC_CMDBIT_AX_267	
SSC_CMDBIT_AX_268	
SSC_CMDBIT_AX_269	
SSC_CMDBIT_AX_270	
SSC_CMDBIT_AX_271	
SSC_CMDBIT_AX_272	

Bit number (constant)	Signal name
SSC_CMDBIT_AX_273	Reserved
SSC_CMDBIT_AX_274	
SSC_CMDBIT_AX_275	
SSC_CMDBIT_AX_276	
SSC_CMDBIT_AX_277	
SSC_CMDBIT_AX_278	
SSC_CMDBIT_AX_279	
SSC_CMDBIT_AX_280	

Bit number (constant)	Signal name
SSC_CMDBIT_AX_281	Reserved
SSC_CMDBIT_AX_282	
SSC_CMDBIT_AX_283	
SSC_CMDBIT_AX_284	
SSC_CMDBIT_AX_285	
SSC_CMDBIT_AX_286	
SSC_CMDBIT_AX_287	
SSC_CMDBIT_AX_288	

Bit number (constant)	Signal name
SSC_CMDBIT_AX_289	Reserved
SSC_CMDBIT_AX_290	
SSC_CMDBIT_AX_291	
SSC_CMDBIT_AX_292	
SSC_CMDBIT_AX_293	
SSC_CMDBIT_AX_294	
SSC_CMDBIT_AX_295	
SSC_CMDBIT_AX_296	

Bit number (constant)	Signal name
SSC_CMDBIT_AX_297	Reserved
SSC_CMDBIT_AX_298	
SSC_CMDBIT_AX_299	
SSC_CMDBIT_AX_300	
SSC_CMDBIT_AX_301	
SSC_CMDBIT_AX_302	
SSC_CMDBIT_AX_303	
SSC_CMDBIT_AX_304	

Bit number (constant)	Signal name
SSC_CMDBIT_AX_305	Reserved
SSC_CMDBIT_AX_306	
SSC_CMDBIT_AX_307	
SSC_CMDBIT_AX_308	
SSC_CMDBIT_AX_309	
SSC_CMDBIT_AX_310	
SSC_CMDBIT_AX_311	
SSC_CMDBIT_AX_312	

Bit number (constant)	Signal name
SSC_CMDBIT_AX_313	Reserved
SSC_CMDBIT_AX_314	
SSC_CMDBIT_AX_315	
SSC_CMDBIT_AX_316	
SSC_CMDBIT_AX_317	
SSC_CMDBIT_AX_318	
SSC_CMDBIT_AX_319	
SSC_CMDBIT_AX_320	

Bit number (constant)	Signal name
SSC_CMDBIT_AX_321	Reserved
SSC_CMDBIT_AX_322	
SSC_CMDBIT_AX_323	
SSC_CMDBIT_AX_324	
SSC_CMDBIT_AX_325	
SSC_CMDBIT_AX_326	
SSC_CMDBIT_AX_327	
SSC_CMDBIT_AX_328	

Bit number (constant)	Signal name
SSC_CMDBIT_AX_329	Reserved
SSC_CMDBIT_AX_330	
SSC_CMDBIT_AX_331	
SSC_CMDBIT_AX_332	
SSC_CMDBIT_AX_333	
SSC_CMDBIT_AX_334	
SSC_CMDBIT_AX_335	
SSC_CMDBIT_AX_336	

Bit number (constant)	Signal name
SSC_CMDBIT_AX_337	Reserved
SSC_CMDBIT_AX_338	
SSC_CMDBIT_AX_339	
SSC_CMDBIT_AX_340	
SSC_CMDBIT_AX_341	
SSC_CMDBIT_AX_342	
SSC_CMDBIT_AX_343	
SSC_CMDBIT_AX_344	

Bit number (constant)	Signal name
SSC_CMDBIT_AX_345	Reserved
SSC_CMDBIT_AX_346	
SSC_CMDBIT_AX_347	
SSC_CMDBIT_AX_348	
SSC_CMDBIT_AX_349	
SSC_CMDBIT_AX_350	
SSC_CMDBIT_AX_351	
SSC_CMDBIT_AX_352	

Bit number (constant)	Signal name
SSC_CMDBIT_AX_353	Reserved
SSC_CMDBIT_AX_354	
SSC_CMDBIT_AX_355	
SSC_CMDBIT_AX_356	
SSC_CMDBIT_AX_357	
SSC_CMDBIT_AX_358	
SSC_CMDBIT_AX_359	
SSC_CMDBIT_AX_360	

Bit number (constant)	Signal name
SSC_CMDBIT_AX_361	Reserved
SSC_CMDBIT_AX_362	
SSC_CMDBIT_AX_363	
SSC_CMDBIT_AX_364	
SSC_CMDBIT_AX_365	
SSC_CMDBIT_AX_366	
SSC_CMDBIT_AX_367	
SSC_CMDBIT_AX_368	

Bit number (constant)	Signal name
SSC_CMDBIT_AX_369	Reserved
SSC_CMDBIT_AX_370	
SSC_CMDBIT_AX_371	
SSC_CMDBIT_AX_372	
SSC_CMDBIT_AX_373	
SSC_CMDBIT_AX_374	
SSC_CMDBIT_AX_375	
SSC_CMDBIT_AX_376	

Bit number (constant)	Signal name
SSC_CMDBIT_AX_377	Reserved
SSC_CMDBIT_AX_378	
SSC_CMDBIT_AX_379	
SSC_CMDBIT_AX_380	
SSC_CMDBIT_AX_381	
SSC_CMDBIT_AX_382	
SSC_CMDBIT_AX_383	
SSC_CMDBIT_AX_384	

6.4 Axis status bit

Bit number (constant)	Signal name
SSC_STSBIT_AX_RDY	Servo ready
SSC_STSBIT_AX_INP	In-position
SSC_STSBIT_AX_ZSP	Zero speed
SSC_STSBIT_AX_ZPAS	Passed Z-phase
SSC_STSBIT_AX_TLC	Torque limit effective
SSC_STSBIT_AX_SALM	Servo alarm
SSC_STSBIT_AX_SWRN	Servo warning
SSC_STSBIT_AX_ABSE	Absolute position disappearance

Bit number (constant)	Signal name
SSC_STSBIT_AX_OP	During operation
SSC_STSBIT_AX_CPO	Rough match
SSC_STSBIT_AX_PF	Positioning finish
SSC_STSBIT_AX_ZP	Home position return complete
SSC_STSBIT_AX_SMZ	During smoothing of stopping
SSC_STSBIT_AX_OALM	Operation alarm
SSC_STSBIT_AX_OPF	Completion of operation
SSC_STSBIT_AX_PSW	Position switch

Bit number (constant)	Signal name
SSC_STSBIT_AX_AUTO	In automatic operation mode
SSC_STSBIT_AX_ZRNO	In home position return mode
SSC_STSBIT_AX_JO	In JOG operation mode
SSC_STSBIT_AX_SO	In incremental feed mode
SSC_STSBIT_AX_21	Reserved
SSC_STSBIT_AX_LIPO	In linear interpolation mode MC200 In interpolation operation mode MC300
SSC_STSBIT_AX_DSTO	In home position reset mode
SSC_STSBIT_AX_24	Reserved

Bit number (constant)	Signal name
SSC_STSBIT_AX_25	Reserved
SSC_STSBIT_AX_26	
SSC_STSBIT_AX_27	
SSC_STSBIT_AX_28	
SSC_STSBIT_AX_29	
SSC_STSBIT_AX_30	
SSC_STSBIT_AX_31	
SSC_STSBIT_AX_32	

Bit number (constant)	Signal name
SSC_STSBIT_AX_ISTP	Interlock stop
SSC_STSBIT_AX_RMRCH	High speed monitor is latched
SSC_STSBIT_AX_POV	Stop position over-bound
SSC_STSBIT_AX_STO	Start up acceptance complete
SSC_STSBIT_AX_37	Reserved
SSC_STSBIT_AX_38	
SSC_STSBIT_AX_ZREQ	Home position return request
SSC_STSBIT_AX_40	Reserved

Bit number (constant)	Signal name
SSC_STSBIT_AX_SCF	Completion of preparation for changing speed
SSC_STSBIT_AX_TACF	Completion of preparation for changing acceleration time constant
SSC_STSBIT_AX_TDCF (Note 1)	Completion of preparation for changing deceleration time constant
SSC_STSBIT_AX_PCF	Completion of preparation for changing position
SSC_STSBIT_AX_SCE	Speed change error
SSC_STSBIT_AX_TACE (Note 2)	Acceleration time constant change error
SSC_STSBIT_AX_TDCE	Deceleration time constant change error
SSC_STSBIT_AX_PCE	Position change error

Note 1. For MR-MC2□□ API Ver. 2.00 or earlier, the bit number is "SSC_STSBIT_AX_TACE".
2. For MR-MC2□□ API Ver. 2.00 or earlier, the bit number is "SSC_STSBIT_AX_TDCF".

Bit number (constant)	Signal name
SSC_STSBIT_AX_49	Reserved
SSC_STSBIT_AX_50	
SSC_STSBIT_AX_51	
SSC_STSBIT_AX_52	
SSC_STSBIT_AX_53	
SSC_STSBIT_AX_54	
SSC_STSBIT_AX_55	
SSC_STSBIT_AX_56	

Bit number (constant)	Signal name
SSC_STSBIT_AX_PPIOP	Pass position interrupt
SSC_STSBIT_AX_PPIFIN	Pass position interrupt complete
SSC_STSBIT_AX_PPIERR	Pass position interrupt incomplete
SSC_STSBIT_AX_60	Reserved
SSC_STSBIT_AX_61	
SSC_STSBIT_AX_62	
SSC_STSBIT_AX_63	
SSC_STSBIT_AX_AUTLO	In point table loop

Bit number (constant)	Signal name
SSC_STSBIT_AX_GAINO	During gain switching
SSC_STSBIT_AX_FCLSO	Fully closed loop control changing
SSC_STSBIT_AX_TLSO	Selecting torque limit
SSC_STSBIT_AX_SPC	During PID control
SSC_STSBIT_AX_69	Reserved
SSC_STSBIT_AX_70	
SSC_STSBIT_AX_71	
SSC_STSBIT_AX_PRSMO	During continuous operation to torque control

Bit number (constant)	Signal name
SSC_STSBIT_AX_IWT	Interference check standby
SSC_STSBIT_AX_SINP	Servo amplifier in-position
SSC_STSBIT_AX_75	Reserved
SSC_STSBIT_AX_76	
SSC_STSBIT_AX_77	
SSC_STSBIT_AX_78	
SSC_STSBIT_AX_79	
SSC_STSBIT_AX_80	

Bit number (constant)	Signal name
SSC_STSBIT_AX_81	Reserved
SSC_STSBIT_AX_82	
SSC_STSBIT_AX_83	
SSC_STSBIT_AX_ZSF	Home position set complete
SSC_STSBIT_AX_ZSE	Home position set error
SSC_STSBIT_AX_86	Reserved
SSC_STSBIT_AX_87	
SSC_STSBIT_AX_88	

Bit number (constant)	Signal name
SSC_STSBIT_AX_MKIF1	Mark detection compatible information 1
SSC_STSBIT_AX_MKCF1	Mark detection clear complete 1
SSC_STSBIT_AX_MKDO1	Mark detection disabled 1
SSC_STSBIT_AX_MKSEF1	Mark detection setting enable complete 1
SSC_STSBIT_AX_MKIF2	Mark detection compatible information 2
SSC_STSBIT_AX_MKCF2	Mark detection clear complete 2
SSC_STSBIT_AX_MKDO2	Mark detection disabled 2
SSC_STSBIT_AX_MKSEF2	Mark detection setting enable complete 2

Bit number (constant)	Signal name
SSC_STSBIT_AX_97	Reserved
SSC_STSBIT_AX_98	
SSC_STSBIT_AX_99	
SSC_STSBIT_AX_CTLMCF	Control mode switch complete
SSC_STSBIT_AX_CTLMCE	Control mode switch error
SSC_STSBIT_AX_102	Reserved
SSC_STSBIT_AX_103	
SSC_STSBIT_AX_104	

Bit number (constant)	Signal name
SSC_STSBIT_AX_105	Reserved
SSC_STSBIT_AX_106	
SSC_STSBIT_AX_107	
SSC_STSBIT_AX_108	
SSC_STSBIT_AX_109	
SSC_STSBIT_AX_110	
SSC_STSBIT_AX_111	
SSC_STSBIT_AX_112	

Bit number (constant)	Signal name
SSC_STSBIT_AX_113	Reserved
SSC_STSBIT_AX_114	
SSC_STSBIT_AX_115	
SSC_STSBIT_AX_116	
SSC_STSBIT_AX_117	
SSC_STSBIT_AX_118	
SSC_STSBIT_AX_119	
SSC_STSBIT_AX_120	

Bit number (constant)	Signal name
SSC_STSBIT_AX_121	Reserved
SSC_STSBIT_AX_122	
SSC_STSBIT_AX_123	
SSC_STSBIT_AX_124	
SSC_STSBIT_AX_125	
SSC_STSBIT_AX_126	
SSC_STSBIT_AX_127	
SSC_STSBIT_AX_128	

Bit number (constant)	Signal name
SSC_STSBIT_AX_MOUT	Monitor output
SSC_STSBIT_AX_MRCH	Monitor latch
SSC_STSBIT_AX_MER1	Monitor number error 1
SSC_STSBIT_AX_MER2	Monitor number error 2
SSC_STSBIT_AX_MER3	Monitor number error 3
SSC_STSBIT_AX_MER4	Monitor number error 4
SSC_STSBIT_AX_MESV	Servo amplifier is not connected
SSC_STSBIT_AX_136	Reserved

Bit number (constant)	Signal name
SSC_STSBIT_AX_137	Reserved
SSC_STSBIT_AX_138	
SSC_STSBIT_AX_139	
SSC_STSBIT_AX_140	
SSC_STSBIT_AX_141	
SSC_STSBIT_AX_142	
SSC_STSBIT_AX_143	
SSC_STSBIT_AX_144	

Bit number (constant)	Signal name
SSC_STSBIT_AX_145	Reserved
SSC_STSBIT_AX_146	
SSC_STSBIT_AX_147	
SSC_STSBIT_AX_148	
SSC_STSBIT_AX_149	
SSC_STSBIT_AX_150	
SSC_STSBIT_AX_151	
SSC_STSBIT_AX_152	

Bit number (constant)	Signal name
SSC_STSBIT_AX_153	Reserved
SSC_STSBIT_AX_154	
SSC_STSBIT_AX_155	
SSC_STSBIT_AX_156	
SSC_STSBIT_AX_157	
SSC_STSBIT_AX_158	
SSC_STSBIT_AX_159	
SSC_STSBIT_AX_160	

Bit number (constant)	Signal name
SSC_STSBIT_AX_PWFIN1	Parameter write complete 1
SSC_STSBIT_AX_PWEN1	Parameter number error 1
SSC_STSBIT_AX_PWED1	Parameter data out of bounds 1
SSC_STSBIT_AX_164	Reserved
SSC_STSBIT_AX_PWFIN2	Parameter write complete 2
SSC_STSBIT_AX_PWEN2	Parameter number error 2
SSC_STSBIT_AX_PWED2	Parameter data out of bounds 2
SSC_STSBIT_AX_PSCHG	Changes to servo parameters exist

Bit number (constant)	Signal name
SSC_STSBIT_AX_PRFIN1	Parameter read complete 1
SSC_STSBIT_AX_PREN1	Parameter number error 1
SSC_STSBIT_AX_PRFIN2	Parameter read complete 2
SSC_STSBIT_AX_PREN2	Parameter number error 2
SSC_STSBIT_AX_173	Reserved
SSC_STSBIT_AX_174	
SSC_STSBIT_AX_175	
SSC_STSBIT_AX_176	

Bit number (constant)	Signal name
SSC_STSBIT_AX_177	Reserved
SSC_STSBIT_AX_178	
SSC_STSBIT_AX_179	
SSC_STSBIT_AX_180	
SSC_STSBIT_AX_181	
SSC_STSBIT_AX_182	
SSC_STSBIT_AX_183	
SSC_STSBIT_AX_184	

Bit number (constant)	Signal name
SSC_STSBIT_AX_185	Reserved
SSC_STSBIT_AX_186	
SSC_STSBIT_AX_187	
SSC_STSBIT_AX_188	
SSC_STSBIT_AX_189	
SSC_STSBIT_AX_190	
SSC_STSBIT_AX_191	
SSC_STSBIT_AX_192	

Bit number (constant)	Signal name
SSC_STSBIT_AX_193	Reserved
SSC_STSBIT_AX_194	
SSC_STSBIT_AX_195	
SSC_STSBIT_AX_196	
SSC_STSBIT_AX_197	
SSC_STSBIT_AX_198	
SSC_STSBIT_AX_199	
SSC_STSBIT_AX_200	

Bit number (constant)	Signal name
SSC_STSBIT_AX_201	Reserved
SSC_STSBIT_AX_202	
SSC_STSBIT_AX_203	
SSC_STSBIT_AX_204	
SSC_STSBIT_AX_205	
SSC_STSBIT_AX_206	
SSC_STSBIT_AX_207	
SSC_STSBIT_AX_208	

Bit number (constant)	Signal name
SSC_STSBIT_AX_209	Reserved
SSC_STSBIT_AX_210	
SSC_STSBIT_AX_211	
SSC_STSBIT_AX_212	
SSC_STSBIT_AX_213	
SSC_STSBIT_AX_214	
SSC_STSBIT_AX_215	
SSC_STSBIT_AX_216	

Bit number (constant)	Signal name
SSC_STSBIT_AX_217	Reserved
SSC_STSBIT_AX_218	
SSC_STSBIT_AX_219	
SSC_STSBIT_AX_220	
SSC_STSBIT_AX_221	
SSC_STSBIT_AX_222	
SSC_STSBIT_AX_223	
SSC_STSBIT_AX_224	

Bit number (constant)	Signal name
SSC_STSBIT_AX_225	Reserved
SSC_STSBIT_AX_226	
SSC_STSBIT_AX_227	
SSC_STSBIT_AX_228	
SSC_STSBIT_AX_229	
SSC_STSBIT_AX_230	
SSC_STSBIT_AX_231	
SSC_STSBIT_AX_232	

Bit number (constant)	Signal name
SSC_STSBIT_AX_233	Reserved
SSC_STSBIT_AX_234	
SSC_STSBIT_AX_235	
SSC_STSBIT_AX_236	
SSC_STSBIT_AX_237	
SSC_STSBIT_AX_238	
SSC_STSBIT_AX_239	
SSC_STSBIT_AX_240	

Bit number (constant)	Signal name
SSC_STSBIT_AX_241	Reserved
SSC_STSBIT_AX_242	
SSC_STSBIT_AX_243	
SSC_STSBIT_AX_244	
SSC_STSBIT_AX_245	
SSC_STSBIT_AX_246	
SSC_STSBIT_AX_247	
SSC_STSBIT_AX_248	

Bit number (constant)	Signal name
SSC_STSBIT_AX_249	Reserved
SSC_STSBIT_AX_250	
SSC_STSBIT_AX_251	
SSC_STSBIT_AX_252	
SSC_STSBIT_AX_253	
SSC_STSBIT_AX_254	
SSC_STSBIT_AX_255	
SSC_STSBIT_AX_256	

Bit number (constant)	Signal name
SSC_STSBIT_AX_257	Reserved
SSC_STSBIT_AX_258	
SSC_STSBIT_AX_259	
SSC_STSBIT_AX_260	
SSC_STSBIT_AX_261	
SSC_STSBIT_AX_262	
SSC_STSBIT_AX_263	
SSC_STSBIT_AX_264	

Bit number (constant)	Signal name
SSC_STSBIT_AX_265	Reserved
SSC_STSBIT_AX_266	
SSC_STSBIT_AX_267	
SSC_STSBIT_AX_268	
SSC_STSBIT_AX_269	
SSC_STSBIT_AX_270	
SSC_STSBIT_AX_271	
SSC_STSBIT_AX_272	

Bit number (constant)	Signal name
SSC_STSBIT_AX_273	Reserved
SSC_STSBIT_AX_274	
SSC_STSBIT_AX_275	
SSC_STSBIT_AX_276	
SSC_STSBIT_AX_277	
SSC_STSBIT_AX_278	
SSC_STSBIT_AX_279	
SSC_STSBIT_AX_280	

Bit number (constant)	Signal name
SSC_STSBIT_AX_281	Reserved
SSC_STSBIT_AX_282	
SSC_STSBIT_AX_283	
SSC_STSBIT_AX_284	
SSC_STSBIT_AX_285	
SSC_STSBIT_AX_286	
SSC_STSBIT_AX_287	
SSC_STSBIT_AX_288	

Bit number (constant)	Signal name
SSC_STSBIT_AX_289	Reserved
SSC_STSBIT_AX_290	
SSC_STSBIT_AX_291	
SSC_STSBIT_AX_292	
SSC_STSBIT_AX_293	
SSC_STSBIT_AX_294	
SSC_STSBIT_AX_295	
SSC_STSBIT_AX_296	

Bit number (constant)	Signal name
SSC_STSBIT_AX_297	Reserved
SSC_STSBIT_AX_298	
SSC_STSBIT_AX_299	
SSC_STSBIT_AX_300	
SSC_STSBIT_AX_301	
SSC_STSBIT_AX_302	
SSC_STSBIT_AX_303	
SSC_STSBIT_AX_304	

Bit number (constant)	Signal name
SSC_STSBIT_AX_305	Reserved
SSC_STSBIT_AX_306	
SSC_STSBIT_AX_307	
SSC_STSBIT_AX_308	
SSC_STSBIT_AX_309	
SSC_STSBIT_AX_310	
SSC_STSBIT_AX_311	
SSC_STSBIT_AX_312	

Bit number (constant)	Signal name
SSC_STSBIT_AX_313	Reserved
SSC_STSBIT_AX_314	
SSC_STSBIT_AX_315	
SSC_STSBIT_AX_316	
SSC_STSBIT_AX_317	
SSC_STSBIT_AX_318	
SSC_STSBIT_AX_319	
SSC_STSBIT_AX_320	

Bit number (constant)	Signal name
SSC_STSBIT_AX_321	Reserved
SSC_STSBIT_AX_322	
SSC_STSBIT_AX_323	
SSC_STSBIT_AX_324	
SSC_STSBIT_AX_325	
SSC_STSBIT_AX_326	
SSC_STSBIT_AX_327	
SSC_STSBIT_AX_328	

Bit number (constant)	Signal name
SSC_STSBIT_AX_329	Reserved
SSC_STSBIT_AX_330	
SSC_STSBIT_AX_331	
SSC_STSBIT_AX_332	
SSC_STSBIT_AX_333	
SSC_STSBIT_AX_334	
SSC_STSBIT_AX_335	
SSC_STSBIT_AX_336	

Bit number (constant)	Signal name
SSC_STSBIT_AX_337	Reserved
SSC_STSBIT_AX_338	
SSC_STSBIT_AX_339	
SSC_STSBIT_AX_340	
SSC_STSBIT_AX_341	
SSC_STSBIT_AX_342	
SSC_STSBIT_AX_343	
SSC_STSBIT_AX_344	

Bit number (constant)	Signal name
SSC_STSBIT_AX_345	Reserved
SSC_STSBIT_AX_346	
SSC_STSBIT_AX_347	
SSC_STSBIT_AX_348	
SSC_STSBIT_AX_349	
SSC_STSBIT_AX_350	
SSC_STSBIT_AX_351	
SSC_STSBIT_AX_352	

Bit number (constant)	Signal name
SSC_STSBIT_AX_353	Reserved
SSC_STSBIT_AX_354	
SSC_STSBIT_AX_355	
SSC_STSBIT_AX_356	
SSC_STSBIT_AX_357	
SSC_STSBIT_AX_358	
SSC_STSBIT_AX_359	
SSC_STSBIT_AX_360	

Bit number (constant)	Signal name
SSC_STSBIT_AX_361	Reserved
SSC_STSBIT_AX_362	
SSC_STSBIT_AX_363	
SSC_STSBIT_AX_364	
SSC_STSBIT_AX_365	
SSC_STSBIT_AX_366	
SSC_STSBIT_AX_367	
SSC_STSBIT_AX_368	

Bit number (constant)	Signal name
SSC_STSBIT_AX_369	Reserved
SSC_STSBIT_AX_370	
SSC_STSBIT_AX_371	
SSC_STSBIT_AX_372	
SSC_STSBIT_AX_373	
SSC_STSBIT_AX_374	
SSC_STSBIT_AX_375	
SSC_STSBIT_AX_376	

Bit number (constant)	Signal name
SSC_STSBIT_AX_377	Reserved
SSC_STSBIT_AX_378	
SSC_STSBIT_AX_379	
SSC_STSBIT_AX_380	
SSC_STSBIT_AX_381	
SSC_STSBIT_AX_382	
SSC_STSBIT_AX_383	
SSC_STSBIT_AX_384	

6.5 Station command bit

Bit number (constant)	Signal name
SSC_CMDBIT_UT_1	Reserved
SSC_CMDBIT_UT_2	
SSC_CMDBIT_UT_3	
SSC_CMDBIT_UT_4	
SSC_CMDBIT_UT_5	
SSC_CMDBIT_UT_RURST	RIO module alarm reset
SSC_CMDBIT_UT_7	Reserved
SSC_CMDBIT_UT_8	

Bit number (constant)	Signal name
SSC_CMDBIT_UT_9	Reserved
SSC_CMDBIT_UT_10	
SSC_CMDBIT_UT_11	
SSC_CMDBIT_UT_12	
SSC_CMDBIT_UT_13	
SSC_CMDBIT_UT_RCRST	RIO control alarm reset
SSC_CMDBIT_UT_15	Reserved
SSC_CMDBIT_UT_16	

Bit number (constant)	Signal name
SSC_CMDBIT_UT_17	Reserved
SSC_CMDBIT_UT_18	
SSC_CMDBIT_UT_19	
SSC_CMDBIT_UT_20	
SSC_CMDBIT_UT_21	
SSC_CMDBIT_UT_22	
SSC_CMDBIT_UT_23	
SSC_CMDBIT_UT_24	

Bit number (constant)	Signal name
SSC_CMDBIT_UT_25	Reserved
SSC_CMDBIT_UT_26	
SSC_CMDBIT_UT_27	
SSC_CMDBIT_UT_28	
SSC_CMDBIT_UT_29	
SSC_CMDBIT_UT_30	
SSC_CMDBIT_UT_31	
SSC_CMDBIT_UT_32	

Bit number (constant)	Signal name
SSC_CMDBIT_UT_MON	Monitor command
SSC_CMDBIT_UT_MONR	Monitor latch command
SSC_CMDBIT_UT_35	Reserved
SSC_CMDBIT_UT_36	
SSC_CMDBIT_UT_37	
SSC_CMDBIT_UT_38	
SSC_CMDBIT_UT_39	
SSC_CMDBIT_UT_40	

Bit number (constant)	Signal name
SSC_CMDBIT_UT_41	Reserved
SSC_CMDBIT_UT_42	
SSC_CMDBIT_UT_43	
SSC_CMDBIT_UT_44	
SSC_CMDBIT_UT_45	
SSC_CMDBIT_UT_46	
SSC_CMDBIT_UT_47	
SSC_CMDBIT_UT_48	

Bit number (constant)	Signal name
SSC_CMDBIT_UT_PWRT	Parameter write command
SSC_CMDBIT_UT_50	Reserved
SSC_CMDBIT_UT_51	
SSC_CMDBIT_UT_52	
SSC_CMDBIT_UT_53	
SSC_CMDBIT_UT_54	
SSC_CMDBIT_UT_55	
SSC_CMDBIT_UT_56	

Bit number (constant)	Signal name
SSC_CMDBIT_UT_PRD	Parameter read command
SSC_CMDBIT_UT_58	Reserved
SSC_CMDBIT_UT_59	
SSC_CMDBIT_UT_60	
SSC_CMDBIT_UT_61	
SSC_CMDBIT_UT_62	
SSC_CMDBIT_UT_63	
SSC_CMDBIT_UT_64	

Bit number (constant)	Signal name
SSC_CMDBIT_UT_65	Reserved
SSC_CMDBIT_UT_66	
SSC_CMDBIT_UT_67	
SSC_CMDBIT_UT_68	
SSC_CMDBIT_UT_69	
SSC_CMDBIT_UT_70	
SSC_CMDBIT_UT_71	
SSC_CMDBIT_UT_72	

Bit number (constant)	Signal name
SSC_CMDBIT_UT_73	Reserved
SSC_CMDBIT_UT_74	
SSC_CMDBIT_UT_75	
SSC_CMDBIT_UT_76	
SSC_CMDBIT_UT_77	
SSC_CMDBIT_UT_78	
SSC_CMDBIT_UT_79	
SSC_CMDBIT_UT_80	

Bit number (constant)	Signal name
SSC_CMDBIT_UT_81	Reserved
SSC_CMDBIT_UT_82	
SSC_CMDBIT_UT_83	
SSC_CMDBIT_UT_84	
SSC_CMDBIT_UT_85	
SSC_CMDBIT_UT_86	
SSC_CMDBIT_UT_87	
SSC_CMDBIT_UT_88	

Bit number (constant)	Signal name
SSC_CMDBIT_UT_89	Reserved
SSC_CMDBIT_UT_90	
SSC_CMDBIT_UT_91	
SSC_CMDBIT_UT_92	
SSC_CMDBIT_UT_93	
SSC_CMDBIT_UT_94	
SSC_CMDBIT_UT_95	
SSC_CMDBIT_UT_96	

Bit number (constant)	Signal name
SSC_CMDBIT_UT_97	Reserved
SSC_CMDBIT_UT_98	
SSC_CMDBIT_UT_99	
SSC_CMDBIT_UT_100	
SSC_CMDBIT_UT_101	
SSC_CMDBIT_UT_102	
SSC_CMDBIT_UT_103	
SSC_CMDBIT_UT_104	

Bit number (constant)	Signal name
SSC_CMDBIT_UT_105	Reserved
SSC_CMDBIT_UT_106	
SSC_CMDBIT_UT_107	
SSC_CMDBIT_UT_108	
SSC_CMDBIT_UT_109	
SSC_CMDBIT_UT_110	
SSC_CMDBIT_UT_111	
SSC_CMDBIT_UT_112	

Bit number (constant)	Signal name
SSC_CMDBIT_UT_113	Reserved
SSC_CMDBIT_UT_114	
SSC_CMDBIT_UT_115	
SSC_CMDBIT_UT_116	
SSC_CMDBIT_UT_117	
SSC_CMDBIT_UT_118	
SSC_CMDBIT_UT_119	
SSC_CMDBIT_UT_120	

Bit number (constant)	Signal name
SSC_CMDBIT_UT_121	Reserved
SSC_CMDBIT_UT_122	
SSC_CMDBIT_UT_123	
SSC_CMDBIT_UT_124	
SSC_CMDBIT_UT_125	
SSC_CMDBIT_UT_126	
SSC_CMDBIT_UT_127	
SSC_CMDBIT_UT_128	

6.6 Station status bit

Bit number (constant)	Signal name
SSC_STSBIT_UT_RURDY	Receiving controller ready on
SSC_STSBIT_UT_RUA	Outputting DO
SSC_STSBIT_UT_3	Reserved
SSC_STSBIT_UT_4	
SSC_STSBIT_UT_5	
SSC_STSBIT_UT_RUALM	RIO module alarm
SSC_STSBIT_UT_RUWRN	RIO module warning
SSC_STSBIT_UT_8	Reserved

Bit number (constant)	Signal name
SSC_STSBIT_UT_9	Reserved
SSC_STSBIT_UT_10	
SSC_STSBIT_UT_11	
SSC_STSBIT_UT_12	
SSC_STSBIT_UT_13	
SSC_STSBIT_UT_RCALM	RIO control alarm
SSC_STSBIT_UT_15	Reserved
SSC_STSBIT_UT_16	

Bit number (constant)	Signal name
SSC_STSBIT_UT_17	Reserved
SSC_STSBIT_UT_18	
SSC_STSBIT_UT_19	
SSC_STSBIT_UT_20	
SSC_STSBIT_UT_21	
SSC_STSBIT_UT_22	
SSC_STSBIT_UT_23	
SSC_STSBIT_UT_24	

Bit number (constant)	Signal name
SSC_STSBIT_UT_25	Reserved
SSC_STSBIT_UT_26	
SSC_STSBIT_UT_27	
SSC_STSBIT_UT_28	
SSC_STSBIT_UT_29	
SSC_STSBIT_UT_30	
SSC_STSBIT_UT_31	
SSC_STSBIT_UT_32	

Bit number (constant)	Signal name
SSC_STSBIT_UT_MOUT	Monitor output
SSC_STSBIT_UT_MRCH	Monitor latch
SSC_STSBIT_UT_MER1	Monitor number error 1
SSC_STSBIT_UT_MER2	Monitor number error 2
SSC_STSBIT_UT_MER3	Monitor number error 3
SSC_STSBIT_UT_MER4	Monitor number error 4
SSC_STSBIT_UT_MERIO	RIO module is not connected
SSC_STSBIT_UT_40	Reserved

Bit number (constant)	Signal name
SSC_STSBIT_UT_41	Reserved
SSC_STSBIT_UT_42	
SSC_STSBIT_UT_43	
SSC_STSBIT_UT_44	
SSC_STSBIT_UT_45	
SSC_STSBIT_UT_46	
SSC_STSBIT_UT_47	
SSC_STSBIT_UT_48	

Bit number (constant)	Signal name
SSC_STSBIT_UT_PWFIN1	Parameter write complete 1
SSC_STSBIT_UT_PWEN1	Parameter number error 1
SSC_STSBIT_UT_PWED1	Parameter data out of bounds 1
SSC_STSBIT_UT_52	Reserved
SSC_STSBIT_UT_PWFIN2	Parameter write complete 2
SSC_STSBIT_UT_PWEN2	Parameter number error 2
SSC_STSBIT_UT_PWED2	Parameter data out of bounds 2
SSC_STSBIT_UT_56	Reserved

Bit number (constant)	Signal name
SSC_STSBIT_UT_PRFIN1	Parameter read complete 1
SSC_STSBIT_UT_PREN1	Parameter number error 1
SSC_STSBIT_UT_PRFIN2	Parameter read complete 2
SSC_STSBIT_UT_PREN2	Parameter number error 2
SSC_STSBIT_UT_61	Reserved
SSC_STSBIT_UT_62	
SSC_STSBIT_UT_63	
SSC_STSBIT_UT_64	

Bit number (constant)	Signal name
SSC_STSBIT_UT_65	Reserved
SSC_STSBIT_UT_66	
SSC_STSBIT_UT_67	
SSC_STSBIT_UT_68	
SSC_STSBIT_UT_69	
SSC_STSBIT_UT_70	
SSC_STSBIT_UT_71	
SSC_STSBIT_UT_72	

Bit number (constant)	Signal name
SSC_STSBIT_UT_73	Reserved
SSC_STSBIT_UT_74	
SSC_STSBIT_UT_75	
SSC_STSBIT_UT_76	
SSC_STSBIT_UT_77	
SSC_STSBIT_UT_78	
SSC_STSBIT_UT_79	
SSC_STSBIT_UT_80	

Bit number (constant)	Signal name
SSC_STSBIT_UT_81	Reserved
SSC_STSBIT_UT_82	
SSC_STSBIT_UT_83	
SSC_STSBIT_UT_84	
SSC_STSBIT_UT_85	
SSC_STSBIT_UT_86	
SSC_STSBIT_UT_87	
SSC_STSBIT_UT_88	

Bit number (constant)	Signal name
SSC_STSBIT_UT_89	Reserved
SSC_STSBIT_UT_90	
SSC_STSBIT_UT_91	
SSC_STSBIT_UT_92	
SSC_STSBIT_UT_93	
SSC_STSBIT_UT_94	
SSC_STSBIT_UT_95	
SSC_STSBIT_UT_96	

Bit number (constant)	Signal name
SSC_STSBIT_UT_97	Reserved
SSC_STSBIT_UT_98	
SSC_STSBIT_UT_99	
SSC_STSBIT_UT_100	
SSC_STSBIT_UT_101	
SSC_STSBIT_UT_102	
SSC_STSBIT_UT_103	
SSC_STSBIT_UT_104	

Bit number (constant)	Signal name
SSC_STSBIT_UT_105	Reserved
SSC_STSBIT_UT_106	
SSC_STSBIT_UT_107	
SSC_STSBIT_UT_108	
SSC_STSBIT_UT_109	
SSC_STSBIT_UT_110	
SSC_STSBIT_UT_111	
SSC_STSBIT_UT_112	

Bit number (constant)	Signal name
SSC_STSBIT_UT_113	Reserved
SSC_STSBIT_UT_114	
SSC_STSBIT_UT_115	
SSC_STSBIT_UT_116	
SSC_STSBIT_UT_117	
SSC_STSBIT_UT_118	
SSC_STSBIT_UT_119	
SSC_STSBIT_UT_120	

Bit number (constant)	Signal name
SSC_STSBIT_UT_121	Reserved
SSC_STSBIT_UT_122	
SSC_STSBIT_UT_123	
SSC_STSBIT_UT_124	
SSC_STSBIT_UT_125	
SSC_STSBIT_UT_126	
SSC_STSBIT_UT_127	
SSC_STSBIT_UT_128	

7. INTERRUPT EVENT FACTOR LIST

The following lists the interrupt event factors.

(1) System interrupt

Event factor	Content
SSC_INT_SYS_SYSE	System error
SSC_INT_SYS_CALM	System alarm
SSC_INT_SYS_EMIO	During forced stop
SSC_INT_SYS_03	Reserved
SSC_INT_SYS_04	
SSC_INT_SYS_05	
SSC_INT_SYS_06	
SSC_INT_SYS_OCME	Operation cycle alarm
SSC_INT_SYS_OASF	Factor of other axes start interrupt is being sent
SSC_INT_SYS_PPI	Factor of pass position interrupt is being sent
SSC_INT_SYS_10	Reserved
SSC_INT_SYS_11	
SSC_INT_SYS_12	
SSC_INT_SYS_13	
SSC_INT_SYS_14	
SSC_INT_SYS_15	

(2) Axis interrupt

Event factor	Content
SSC_INT_AX_RDY	Servo ready
SSC_INT_AX_INP	In-position
SSC_INT_AX_ZSP	Zero speed
SSC_INT_AX_ZPAS	Passed Z-phase
SSC_INT_AX_TLC	Torque limit effective
SSC_INT_AX_SALM	Servo alarm
SSC_INT_AX_SWRN	Servo warning
SSC_INT_AX_ABSE	Absolute position disappearance
SSC_INT_AX_OP	During operation
SSC_INT_AX_CPO	Rough match
SSC_INT_AX_PF	Positioning finish
SSC_INT_AX_ZP	Home position return complete
SSC_INT_AX_SMZ	During smoothing of stopping
SSC_INT_AX_OALM	Operation alarm
SSC_INT_AX_OPF	Completion of operation
SSC_INT_AX_PSW	Position switch
SSC_INT_AX_GAINO	During gain switching
SSC_INT_AX_FCLSO	Fully closed loop control changing
SSC_INT_AX_TLSO	Selecting torque limit
SSC_INT_AX_SPC	During PID control
SSC_INT_AX_20	Reserved
SSC_INT_AX_MAK1	Mark detection 1
SSC_INT_AX_MAK2	Mark detection 2
SSC_INT_AX_PRSMO	During continuous operation to torque control
SSC_INT_AX_IWT	Interference check standby
SSC_INT_AX_SINP	Servo amplifier in-position
SSC_INT_AX_26	Reserved
SSC_INT_AX_27	
SSC_INT_AX_28	
SSC_INT_AX_29	
SSC_INT_AX_30	
SSC_INT_AX_31	

(3) Station interrupt

Event factor	Content
SSC_INT_UT_00	Reserved
SSC_INT_UT_01	
SSC_INT_UT_02	
SSC_INT_UT_03	
SSC_INT_UT_04	
SSC_INT_UT_RUALM	RIO module alarm
SSC_INT_UT_RUWRN	RIO module warning
SSC_INT_UT_07	Reserved
SSC_INT_UT_08	
SSC_INT_UT_09	
SSC_INT_UT_10	
SSC_INT_UT_11	
SSC_INT_UT_12	
SSC_INT_UT_RCALM	RIO control alarm
SSC_INT_UT_14	Reserved
SSC_INT_UT_15	

MEMO

8. LIST OF DETAILED ERROR CODES

The following shows the detailed error codes.

(1) Common error

Value	Constant definition	Cause/countermeasure
FFFFFFFFh	SSC_FUNC_ERR_UNKNOWN	No errors (-1) have occurred after using the API functions.
00000001h 00000002h 00000003h 00000004h 00000005h 00000006h 00000007h 00000008h 00000009h	SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
00000100h	SSC_FUNC_ERR_ARGUMENT_MISMATCH	The axis number and the command bit number, status bit, or alarm type do not correspond. Example: When "0" is set to the axis number and "SSC_CMDBIT_AX_SON" is set to the command bit number, etc. When "0" is set to the axis number and "SSC_SYSBIT_AX_RDY" is set to the status bit number, etc. When "0" is set to the axis number and "SSC_ALARM_OPERATION" is set to the alarm type, etc.
00010000h 00010100h 00010200h 00010300h 00010400h	SSC_FUNC_ERR_TIMEOUT_0□ □ = 1 to 5: Timeout location	A timeout occurred. Consider changing the set timeout value for API functions which have timeout parameters.

(2) Device functions error

Value	Constant definition	Cause/countermeasure
00020000h	SSC_FUNC_ERR_REOPEN	The sscOpen function is already called.
00020010h	SSC_FUNC_ERR_UNOPEN	The sscOpen function has not been called.
00020080h 00021010h	SSC_FUNC_ERR_NOT_FOUND_BOARD	• The position board which has the designated board ID could not be found. Confirm the board ID selection (dip switch) of the position board. • When connecting the position board from the position board test tool, check that "Confirm the hardware interrupts" is unchecked. MC-300
00021011h	SSC_FUNC_ERR_GET_CHANNEL_NUM	The mount channel information cannot be got. The operating system may not recognize the position board properly. Confirm that the position board is properly mounted using the device manager.
00021012h	SSC_FUNC_ERR_UNSUPPORT_DEVICE_DRIVER	The device driver is not a supported version. Use a API library that combines with the device driver contained in the utility software.

Value	Constant definition	Cause/countermeasure
00022000h	SSC_FUNC_ERR_CREATE_SEMAPHORE	An error occurred in the CreateSemaphore function (Win32API). Call the GetLastError function of Win32API and confirm the error details.
00022001h	SSC_FUNC_ERR_DELETE_SEMAPHORE	An error occurred in the CloseHandle function (Win32API). Call the GetLastError function of Win32API and confirm the error details.
00022002h	SSC_FUNC_ERR_WAIT_SEMAPHORE	An error occurred in the WaitForSingleObject function (Win32API). Call the GetLastError function of Win32API and confirm the error details.
00022003h	SSC_FUNC_ERR_RELEASE_SEMAPHORE	An error occurred in the ReleaseSemaphore function (Win32API). Call the GetLastError function of Win32API and confirm the error details.
00022010h	SSC_FUNC_ERR_CREATE_EVENT	An error occurred in the CreateEvent function (Win32API). Call the GetLastError function of Win32API and confirm the error details.
00022011h	SSC_FUNC_ERR_DELETE_EVENT	An error occurred in the CloseHandle function (Win32API). Call the GetLastError function of Win32API and confirm the error details.
00022012h	SSC_FUNC_ERR_RESET_EVENT	An error occurred in the ResetEvent function (Win32API). Call the GetLastError function of Win32API and confirm the error details.
00022013h	SSC_FUNC_ERR_SET_EVENT	An error occurred in the SetEvent function (Win32API). Call the GetLastError function of Win32API and confirm the error details.
00022014h	SSC_FUNC_ERR_WAIT_EVENT	An error occurred in the WaitForSingleObject function (Win32API). Call the GetLastError function of Win32API and confirm the error details.
00022015h	SSC_FUNC_ERR_WAIT_EVENT_MULTI	An error occurred in the WaitForMultipleObjects function (Win32API). Call the GetLastError function of Win32API and confirm the error details.
00022020h	SSC_FUNC_ERR_CREATE_THREAD	An error occurred in the CreateThread function (Win32API). Call the GetLastError function of Win32API and confirm the error details.
00022021h	SSC_FUNC_ERR_DELETE_THREAD	An error occurred in the CloseHandle function (Win32API). Call the GetLastError function of Win32API and confirm the error details.
00022022h	SSC_FUNC_ERR_THREAD_PRIORITY	An error occurred in the SetThreadPriority function (Win32API). Call the GetLastError function of Win32API and confirm the error details.
00022023h	SSC_FUNC_ERR_RESUME_THREAD	An error occurred in the ResumeThread function (Win32API). Call the GetLastError function of Win32API and confirm the error details.
00022024h	SSC_FUNC_ERR_GET_EXIT_CODE_THREAD	An error occurred in the GetExitCodeThread function (Win32API). Call the GetLastError function of Win32API and confirm the error details.
000230□□h	SSC_FUNC_ERR_DEVICE_DRIVER	An error occurred with a call of the device driver. Confirm that the device driver is installed.

(3) System functions error

Value	Constant definition	Cause/countermeasure
00030000h	SSC_FUNC_ERR_UNREADY_CHANNEL	The system is in the status other than system preparation completion. Reboot the system with the sscReboot function.
00030020h	SSC_FUNC_ERR_RUNNING_CHANNEL	The system is in the status of before system startup. Start the system with the sscSystemStart function.
00030030h	SSC_FUNC_ERR_NOW_ALARM_SYSTEM	A system error (E□□□h) occurred. Get the system status code with the sscGetSystemStatusCode function and remove the cause.
00030040h	SSC_FUNC_ERR_FLASHROM_PARAM_LOAD	The flash ROM parameter read error occurred.
00030041h	SSC_FUNC_ERR_STS_BIT_FRNG	The flash ROM transfer preparation error (FRNG) occurred.
00030042h	SSC_FUNC_ERR_STS_BIT_FSNG	The flash ROM transfer error (FSNG) occurred.
00030050h	SSC_FUNC_ERR_STS_BIT_RCE	The reconnection error (RCE) occurred.
00030051h	SSC_FUNC_ERR_STS_BIT_CCE	The disconnection error (CCE) occurred.
00030060h	SSC_FUNC_ERR_ALREADY_ENABLE_WDT	The user watchdog function has been already valid.
00030061h	SSC_FUNC_ERR_ALREADY_DISABLE_WDT	The user watchdog function has been already invalid.
00030062h	SSC_FUNC_ERR_STS_BIT_IFMO	Position board is in interface mode. When changing control mode in interface mode, use the sscIfmSetControlMode function.

(4) Parameter functions error

Value	Constant definition	Cause/countermeasure
00040000h 00040001h	SSC_FUNC_ERR_STS_BIT_PREN□ □ = 1 to 2: Array number of the parameter read numbers (for 2 numbers)	A parameter read error occurred. • A value outside the range is set in the parameter read number. • The axis number and the parameter read number do not correspond. (Example: When "System parameter" is set to the axis number and "Axis parameter" is set to the parameter read number, etc.)
00040002h 00040003h	SSC_FUNC_ERR_MISMATCH_PARAM_READ_NUM□ □ = 1 to 2: Array number of the parameter read numbers (for 2 numbers)	The command and the status of the parameter read number do not correspond.
00040020h 00040021h	SSC_FUNC_ERR_STS_BIT_PWEN□ □ = 1 to 2: Array number of the parameter write numbers (for 2 numbers)	A parameter write number error occurred. • A value outside the range is set in the parameter write number. • The axis number and the parameter write number do not correspond. (Example: When "System parameter" is set to the axis number and "Axis parameter" is set to the parameter write number, etc.)
00040022h 00040023h	SSC_FUNC_ERR_STS_BIT_PWED□ □ = 1 to 2: Array number of the parameter write data (for 2 data)	A value outside the range is set in the parameter write data.
00040024h 00040025h	SSC_FUNC_ERR_MISMATCH_PARAM_WRITE_NUM□ □ = 1 to 2: Array number of the parameter write numbers (for 2 numbers)	The command and the status of the parameter write number do not correspond.
00040026h 00040027h	SSC_FUNC_ERR_MISMATCH_PARAM_WRITE_DATA□ □ = 1 to 2: Array number of the parameter write data (for 2 data)	The command and the status of the parameter write data do not correspond.

(5) Monitor functions error

Value	Constant definition	Cause/countermeasure
00050000h	SSC_FUNC_ERR_STS_BIT_MER□ □ = 1 to 4: Array number of the monitor numbers (for 4 numbers)	A monitor number error occurred. • A value outside the range is set in the monitor number. • The axis number and the monitor number do not correspond. (Example: When "System monitor" is set to the axis number and "Axis monitor" is set to the monitor number, etc.)
00050001h		
00050002h		
00050003h		
00050004h	SSC_FUNC_ERR_STS_BIT_MESV	The servo information was set as a monitor number when a servo amplifier was not connected.
00050010h	SSC_FUNC_ERR_ALREADY_MONITOR_STOP	The monitor has already stopped.
00050011h	SSC_FUNC_ERR_NOT_START_MONITOR	The monitor has not been started.
00050012h	SSC_FUNC_ERR_MONITOR_2READ	The acquisition of the monitor data failed because of the occurrence of the deviation of the update timing of the lower and upper monitor data. Call the sscGetMonitor function again.

(6) Axis functions error

Value	Constant definition	Cause/countermeasure
00060010h	SSC_FUNC_ERR_NOW_DRIVING	During operation.
00060011h	SSC_FUNC_ERR_NOW_DRIVING_READY	During operation startup preparation (until the position board receives the signal after the start operation is requested).
00060020h	SSC_FUNC_ERR_NOW_ALARM_SERVO	A servo alarm is occurring. Get the alarm No. with the sscGetAlarm function and remove the cause.
00060030h	SSC_FUNC_ERR_NOW_ALARM_DRIVE	An operation alarm is occurring. Get the alarm No. with the sscGetAlarm function and remove the cause.
00060040h	SSC_FUNC_ERR_MISMATCH_DRIVE_MODE	The operation mode is other than the designated operation mode. (Example: sscChangeManualPosition is called during the automatic operation mode.)
00060041h	SSC_FUNC_ERR_CHG_POS_DIR	The movement direction differs between before and after the position change.
00060060h	SSC_FUNC_ERR_STS_BIT_PCE	The position change error signal (PCE) turned on.
00060070h	SSC_FUNC_ERR_STS_BIT_SCE	The speed change error signal (SCE) turned on.
00060080h	SSC_FUNC_ERR_STS_BIT_TACE	The acceleration time constant change error signal (TACE) turned on.
00060090h	SSC_FUNC_ERR_STS_BIT_TDCE	The deceleration time constant change error signal (TDCE) turned ON.
000600A0h	SSC_FUNC_ERR_POINT_NUMBER_OVER	Designated point number + point number offset value exceeded the point table range.
000600A1h	SSC_FUNC_ERR_STS_BIT_CTLMCE	The control mode switch error signal (CLTMCE) turned ON.
000600A2h	SSC_FUNC_ERR_STS_BIT_IPCH_ON	Changeable interpolation group setting is enabled. Set 0 to group number.
000600A3h	SSC_FUNC_ERR_STS_BIT_IPCH_OFF	Changeable interpolation group setting is disabled. Set 1 to 8 to group number.

Value	Constant definition	Cause/countermeasure
000600A4h	SSC_FUNC_ERR_SUB_AXIS_NUM	When changeable interpolation group is enabled, the interpolation axis No. set to the point table is an incorrect value. • The interpolation axis No. is outside the setting range • The interpolation axis No. is the same as the primary axis No. • The interpolation axis No. is the same as another interpolation axis No.
000600A5h	SSC_FUNC_ERR_NOT_LIP_DRIVING	Linear interpolation is not in operation.

(7) Sampling functions error

Value	Constant definition	Cause/countermeasure
00061010h	SSC_FUNC_ERR_ALREADY_START_SAMPLING	The sampling start signal (SMPS) is ON. Stop the sampling with the sscStopSampling function.
00061011h	SSC_FUNC_ERR_ALREADY_STOP_SAMPLING	The sampling has already stopped.
00061020h	SSC_FUNC_ERR_STS_BIT_SREN	A value outside the range is set in the sampling setting read number.
00061021h	SSC_FUNC_ERR_MISMATCH_SMP_PARAM_READ_NUM	The command and the status of the sampling setting read number do not correspond.
00061022h	SSC_FUNC_ERR_STS_BIT_SWEN	A value outside the range is set in the sampling setting write number.
00061023h	SSC_FUNC_ERR_STS_BIT_SWED	A value outside the range is set in the sampling setting write data.
00061024h	SSC_FUNC_ERR_MISMATCH_SMP_PARAM_WRITE_NUM	The command and the status of the sampling setting write number do not correspond.
00061025h	SSC_FUNC_ERR_MISMATCH_SMP_PARAM_WRITE_DATA	The command and the status of the sampling write data do not correspond.

(8) Log functions error

Value	Constant definition	Cause/countermeasure
00062000h	SSC_FUNC_ERR_NOW_LOGGING	The log operation being performed signal (LOGO) is ON. Stop the logging with the sscStopLog function.
00062001h	SSC_FUNC_ERR_STS_BIT_LOGRE	The reading of log data error signal (LOGRE) is turned ON.
00062002h	SSC_FUNC_ERR_STS_BIT_LOGIE	The log data initialization error signal (LOGIE) is turned ON.
00062003h	SSC_FUNC_ERR_ALREADY_START_LOG	The log command (LOGC) is ON. Stop the logging with the sscStopLog function.
00062004h	SSC_FUNC_ERR_ALREADY_STOP_LOG	Logging has already been stopped.
00062005h	SSC_FUNC_ERR_UNSUPPORT_ALH	Alarm history function is not supported.
00062006h	SSC_FUNC_ERR_STS_BIT_ALHRE	The alarm history read error signal (ALHRE) is turned ON.
00062007h	SSC_FUNC_ERR_STS_BIT_ALHIE	The alarm history initialization error signal (ALHIE) is turned ON.

(9) Interrupt functions error

Value	Constant definition	Cause/countermeasure
10000100h	SSC_FUNC_ERR_ALREADY_START_INT_DRIVER	The interrupt driver has already been started up.
10000101h	SSC_FUNC_ERR_ALREADY_END_INT_DRIVER	The interrupt driver has already been closed.
10000102h	SSC_FUNC_ERR_ALREADY_OTHER_PROCESS_INT	The interrupt driver has already been started up in other processing.
10000103h	SSC_FUNC_ERR_INT_DISABLE_MASK	The interrupt output mask selection (dip switch) is valid.
10000104h	SSC_FUNC_ERR_CLEAR_INT	The writing of the interrupt signal clear register is failed.
10000110h	SSC_FUNC_ERR_NOT_START_INT_DRIVER	The interrupt driver is closed. Call the sscIntStart function.
10000111h	SSC_FUNC_ERR_TERMINATE_INT_DRIVER	The sscIntEnd function was called while the interrupt for the designated event factor was being confirmed.
10000112h	SSC_FUNC_ERR_TERMINATE_NOTIFY_EVENT	An error occurred in the interrupt event notification thread while the interrupt for the designated event factor was being confirmed.
10000113h	SSC_FUNC_ERR_SET_HOST_APPLICATION_EVENT	A function which releases the standby status was called from the user program while the interrupt for the designated event factor was being confirmed.
10000200h	SSC_FUNC_ERR_ALREADY_REREGISTER_CALLBACK	The interrupt callback function has already been registered. To change the interrupt callback function, call the sscUnregisterIntCallback function
10000201h	SSC_FUNC_ERR_ALREADY_UNREREGISTER_CALLBACK	The interrupt callback function has already been unregistered.

(10) Mark detection functions error

Value	Constant definition	Cause/countermeasure
00067000h	SSC_FUNC_ERR_MARK_DETECT_UNUSABLE	Mark detection function is disabled. • Mark detection function is not supported. • Mark detection function has been disabled by the settings.
00067001h	SSC_FUNC_ERR_MARK_DETECT_UNDETECTED	There is no mark detection data that can be got. After checking that the mark detection count of the position board has been renewed, call the sscGetMarkDetectionData function.

(11) Interface mode functions error

Value	Constant definition	Cause/countermeasure
000D0000h	SSC_FUNC_ERR_IFM_INP_OFF	In-position signal is OFF. When setting "Does not wait until the in-position signal is ON" during in-position signal check mode, call the sscIfmSetHomePosition function when the in-position signal is ON.
000D0001h	SSC_FUNC_ERR_STS_BIT_ZSE	The home position set error (ZSE) occurred.
000D0002h	SSC_FUNC_ERR_IFM_CMD_BUF_FULL	There is no free space in the position command buffer. After checking that the position board transmit buffer number has been renewed, call the sscIfmRenewLatestBuffer function or sscIfmRenewLatestBufferEx function.
000D0003h	SSC_FUNC_ERR_DISABLE_EVENT_DETECT	Event detection function is not enabled.

(12) I/O device functions error

Value	Constant definition	Cause/countermeasure
000E0000h	SSC_FUNC_ERR_DVI_TABLE_RANGE_OVER	The "word_num" + "word_cnt" designated by the argument exceeds the size of the input device table.
000E0001h	SSC_FUNC_ERR_DVO_TABLE_RANGE_OVER	The "word_num" + "word_cnt" designated by the argument exceeds the size of the output device table.

(13) Transient transmit functions error

Value	Constant definition	Cause/countermeasure
000E1001h	SSC_FUNC_ERR_TRANSIENT_INVALID_DATA	Transient data is invalid.

MEMO

APPENDIX

App. 1 Differences between MR-MC2□□ and MR-MC3□□

App. 1.1 Differences between API libraries

Differences between the MR-MC2□□ and MR-MC3□□ API libraries are shown below.

Item	MR-MC2□□	MR-MC3□□
Maximum number of channels	1	1
Maximum axis number	32	64
Maximum station number	4	16
Maximum point number	320	2048
Maximum point number offset value	319	2047
Maximum other axes start table number	32	64
Maximum pass position condition number	64	128
Maximum interpolation group number	8	16
I/O bit device number	0000h to 0FFFh	0000h to 23FFh
I/O word device number	0000h to 0FFFh	0000h to 23FFh

App. 1.2 API functions added to the MR-MC3□□ API library

New API functions added to the MR-MC3□□ API library are shown below.

Function type	Function name	Function content	Reference section
Information functions	sscGetBoardSerialNumber	Gets position board serial No. information.	4.3.6
System functions	sscReconnectSSCNETEx	Reconnects the SSCNET communication.	4.5.5
	sscDisconnectSSCNETEx	Disconnects the SSCNET communication.	4.5.7
	sscGetControllingAxis	Gets the controlling axis information and controlling station information.	4.5.8
Operating functions	sscInterpolationStart	Starts interpolation operation.	4.9.8
Interface mode functions	sscIfmGetEventStatusBitsEx	Gets the status bit information of all axes for the designated status signal using the event detect function.	4.21.15

App. 1.3 MR-MC3□□ replacements for MR-MC2□□ API functions

(1) While it is possible to use MR-MC2□□ API functions with MR-MC3□□ models, we recommend using MR-MC3□□ API functions with these models.
MR-MC3□□ replacements for MR-MC2□□ API functions are shown in the table below.

Function type	Function name	
	MR-MC2□□ API functions	MR-MC3□□ API functions
System functions	sscReconnectSSCNET	sscReconnectSSCNETEx
	sscDisconnectSSCNET	sscDisconnectSSCNETEx
Operating functions	sscLinearStart	sscInterpolationStart

(2) It is not possible to use the MR-MC2□□ API functions shown in the table below with MR-MC3□□ models. Replace these functions with the appropriate MR-MC3□□ API functions.MR-MC3□□ replacements for MR-MC2□□ API functions are shown in the table below.

Function type	Function name	
	MR-MC2□□ API functions	MR-MC3□□ API functions
Interface mode functions	ssclfmGetMaximumBufferNumber	ssclfmGetMaximumBufferNumberEx
	ssclfmRenewLatestBuffer	ssclfmRenewLatestBufferEx
	ssclfmCheckLatestBuffer	ssclfmCheckLatestBufferEx
	ssclfmGetTransmitBuffer	ssclfmGetTransmitBufferEx
	ssclfmGetEventStatusBits	ssclfmGetEventStatusBitsEx

(3) The following API functions can be used as per usual, but be sure to reset their settings to "0" beforehand. This is because structures have been expanded to be compatible with MR-MC3□□ models.

Function type	Function name	Structure
Point table functions	sscSetPointDataEx	PNT_DATA_EX structure
	sscCheckPointDataEx	
Other axes start functions	sscSetOtherAxisStartData	OAS_DATA structure
	sscGetOtherAxisStartData	
Sampling functions	sscGetSamplingError	SMP_ERR structure
Interrupt functions	sscRegisterIntCallback	INT_CB_DATA structure

WARRANTY

Please confirm the following product warranty details before using this product.

1. Gratis Warranty Term and Gratis Warranty Range

We will repair any failure or defect hereinafter referred to as "failure" in our FA equipment hereinafter referred to as the "Product" arisen during warranty period at no charge due to causes for which we are responsible through the distributor from which you purchased the Product or our service provider. However, we will charge the actual cost of dispatching our engineer for an on-site repair work on request by customer in Japan or overseas countries. We are not responsible for any on-site readjustment and/or trial run that may be required after a defective unit is repaired or replaced.

[Gratis Warranty Term]

For terms of warranty, please contact your original place of purchase.

[Gratis Warranty Range]

(1) You are requested to conduct an initial failure diagnosis by yourself, as a general rule.
It can also be carried out by us or our service company upon your request and the actual cost will be charged. However, it will not be charged if we are responsible for the cause of the failure.

(2) This limited warranty applies only when the condition, method, environment, etc. of use are in compliance with the terms and conditions and instructions that are set forth in the instruction manual and user manual for the Product and the caution label affixed to the Product.

(3) Even during the term of warranty, the repair cost will be charged on you in the following cases;

1) A failure caused by your improper storing or handling, carelessness or negligence, etc., and a failure caused by your hardware or software problem
2) A failure caused by any alteration, etc. to the Product made on your side without our approval
3) A failure which may be regarded as avoidable, if your equipment in which the Product is incorporated is equipped with a safety device required by applicable laws and has any function or structure considered to be indispensable according to a common sense in the industry
4) A failure which may be regarded as avoidable if consumable parts designated in the instruction manual, etc. are duly maintained and replaced
5) Any replacement of consumable parts (battery, relay, fuse, etc.)
6) A failure caused by external factors such as inevitable accidents, including without limitation fire and abnormal fluctuation of voltage, and acts of God, including without limitation earthquake, lightning and natural disasters
7) A failure generated by an unforeseeable cause with a scientific technology that was not available at the time of the shipment of the Product from our company
8) Any other failures which we are not responsible for or which you acknowledge we are not responsible for

2. Onerous Repair Term after Discontinuation of Production

(1) We may accept the repair at charge for another seven (7) years after the production of the product is discontinued.
The announcement of the stop of production for each model can be seen in our Sales and Service, etc.

(2) Please note that the Product (including its spare parts) cannot be ordered after its stop of production.

3. Service in overseas countries

Our regional FA Center in overseas countries will accept the repair work of the Product; However, the terms and conditions of the repair work may differ depending on each FA Center. Please ask your local FA center for details.

4. Exclusion of Loss in Opportunity and Secondary Loss from Warranty Liability

Regardless of the gratis warranty term, Mitsubishi shall not be liable for compensation to:

(1) Damages caused by any cause found not to be the responsibility of Mitsubishi.
(2) Loss in opportunity, lost profits incurred to the user by Failures of Mitsubishi products.
(3) Special damages and secondary damages whether foreseeable or not, compensation for accidents, and compensation for damages to products other than Mitsubishi products.
(4) Replacement by the user, maintenance of on-site equipment, start-up test run and other tasks.

5. Change of Product specifications

Specifications listed in our catalogs, manuals or technical documents may be changed without notice.

6. Precautions for Choosing the Products

(1) For the use of our Position Board, its applications should be those that may not result in a serious damage even if any failure or malfunction occurs in Position Board, and a backup or fail-safe function should operate on an external system to Position Board when any failure or malfunction occurs.

(2) Our Position Board is designed and manufactured as a general purpose product for use at general industries. Therefore, applications substantially influential on the public interest for such as atomic power plants and other power plants of electric power companies, and also which require a special quality assurance system, including applications for railway companies and government or public offices are not recommended, and we assume no responsibility for any failure caused by these applications when used.
In addition, applications which may be substantially influential to human lives or properties for such as airlines, medical treatments, railway service, incineration and fuel systems, man-operated material handling equipment, entertainment machines, safety machines, etc. are not recommended, and we assume no responsibility for any failure caused by these applications when used.
We will review the acceptability of the abovementioned applications, if you agree not to require a specific quality for a specific application. Please contact us for consultation.

IB(NA)-0300225-H(2206)MEE
MODEL: MRMC2-U-API-E
MODEL CODE: 1XB970

MITSUBISHI ELECTRIC CORPORATION

HEAD OFFICE : TOKYO BUILDING, 2-7-3 MARUNOUCHI, CHIYODA-KU, TOKYO 100-8310, JAPAN
NAGOYA WORKS : 1-14 , YADA-MINAMI 5-CHOME , HIGASHI-KU, NAGOYA , JAPAN