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

(Read these precautions before using this product.)

Before using this product, please read this manual and the relevant manuals carefully and pay full attention to safety to handle the product correctly.

The precautions given in this manual are concerned with this product only.

In this manual, the safety precautions are classified into two levels: "⚠ WARNING" and "⚠ CAUTION".

⚠ **WARNING**	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 minor or moderate injury or property damage.

Under some circumstances, failure to observe the precautions given under "⚠ CAUTION" may lead to serious consequences.

Observe the precautions of both levels because they are important for personal and system safety.

Make sure that the end users read this manual and then keep the manual in a safe place for future reference.

[Design Precautions]

⚠ WARNING

- Configure safety circuits external to the personal computer to ensure that the entire system operates safely even when a fault occurs in the external power supply, the personal computer or the position board. Failure to do so may result in an accident due to an incorrect output or malfunction.
  (1) Emergency stop circuits, protection circuits, and protective interlock circuits for conflicting operations (such as forward/reverse rotations or upper/lower limit positioning) must be configured external to the position board.
- Configure a circuit so that the position board is turned on first and then the external power supply. If the external power supply is turned on first, an accident may occur due to an incorrect output or malfunction.
- For the operating status of each station after a communication failure, refer to manuals for the device used. For the manuals, please consult your local Mitsubishi representative. Incorrect output or malfunction due to a communication failure may result in an accident.
- When connecting an external device with a position board to modify data of a running position board, configure an interlock circuit in the program to ensure that the entire system will always operate safely. For other forms of control (such as program modification, parameter change, or operating status change) of a running position board, read the relevant manuals carefully and ensure that the operation is safe before proceeding. Improper operation may damage machines or cause accidents.
- Especially, when a remote position board is controlled by an external device, immediate action cannot be taken if a problem occurs in the position board due to a communication failure. To prevent this, configure an interlock circuit in the program, and determine corrective actions to be taken between the external device and position board in case of a communication failure.
- Do not write any data to the "area for manufacturer setting" of the dual port memory in the position board. Also, do not use any "use prohibited" signals as an output signal from the position board to each module. Doing so may cause malfunction of the position board system. For the "area for manufacturer setting" and the "use prohibited" signals, refer to the user's manual and the relevant manuals for the module used.

[Design Precautions]

⚠WARNING

- If a communication cable is disconnected, the network may be unstable, resulting in a communication failure of multiple stations. Configure an interlock circuit in the program to ensure that the entire system will always operate safely even if communications fail. Incorrect output or malfunction due to a communication failure may result in an accident.
- Configure safety circuits external to the personal computer to ensure that the entire system operates safely even when a fault occurs in the external power supply, the personal computer, or the position board. Failure to do so may result in an accident due to an incorrect output or malfunction.
  (1) OPR (Original Point Return) is controlled by two kinds of data: an OPR direction and an OPR speed. Deceleration starts when the near-point dog signal turns on. If an incorrect OPR direction is set, motion control may continue without deceleration. To prevent machine damage caused by this, configure an interlock circuit external to the position board.
  (2) When the position board detects an error, the motion slows down and stops or the motion suddenly stops. Set the parameter to meet the specifications of a positioning control system. In addition, set the OPR parameter and positioning data within the specified setting range.
  (3) Outputs may remain on or off, or become undefined due to a failure of a component such as an insulation element and transistor in an output circuit, where the position board cannot detect any error. In a system that the incorrect output could cause a serious accident, configure an external circuit for monitoring output signals.
    • If safety standards (ex., robot safety rules, etc.,) apply to the system using the position board, module, servo amplifier, and servo motor, make sure that the safety standards are satisfied.
    • Construct a safety circuit externally of the position board, module, or servo amplifier if the abnormal operation of the module or servo amplifier differs from the safety directive operation in the system.

[Design Precautions]

⚠CAUTION

- Do not install the control lines or communication cables together with the main circuit lines or power cables. Doing so may result in malfunction due to electromagnetic interference. Keep a distance of 100mm or more between those cables.
- During control of an inductive load such as a lamp, heater, or solenoid valve, a large current (approximately ten times greater than normal) may flow when the output is turned from off to on. Therefore, use a module that has a sufficient current rating.
- After the position board is powered on or the software is rebooted, the time taken to enter the system startup status varies depending on the system configuration and/or parameter settings. Design circuits so that the entire system will always operate safely, regardless of the time.
- Do not power off the position board or reboot the software while the settings are being written. Doing so will make the data in the flash ROM undefined. The values need to be set in the buffer memory and written to the flash ROM again. Doing so also may cause malfunction and failure of the position board.

[Security Precautions]

⚠WARNING

- To maintain the security (confidentiality, integrity, and availability) of the position board and the system against unauthorized access, denial-of-service (DoS) attacks, computer viruses, and other cyberattacks from external devices via the network, take appropriate measures such as firewalls, virtual private networks (VPNs), and antivirus solutions.

[Installation Precautions]

⚠WARNING

- Shut off the external power supply (all phases) used in the system before mounting or removing the position board. Failure to do so may result in electric shock or cause the position board to fail or malfunction.

[Installation Precautions]

⚠CAUTION

- Use the position board in an environment that meets the general specifications in the MR-MC200/MR-MC300 Series Position Board User's Manual (Details). Failure to do so may result in electric shock, fire, malfunction, or damage to or deterioration of the product.
- Use board fixing screws and securely tighten the position board. Tighten the screws within the specified torque range. Undertightening can cause drop of the component or wire, short circuit, or malfunction. Overtightening can damage the screw and/or module, resulting in drop, short circuit, or malfunction. For the specified torque range, refer to the attached manual for the personal computer.
- Beware that the position board and the heat sink could be very hot while power is on and immediately after power-off.
- Do not directly touch any conductive parts and electronic components of the position board or connector. Hold the front panel or edge of the print board. Doing so can cause malfunction or failure of the position board.
- Do not disassemble or modify the position board. Doing so may cause failure, malfunction, injury, or a fire.
- Before handling the position board, touch a conducting object such as a grounded metal to discharge the static electricity from the human body. Failure to do so may cause the position board to fail or malfunction.
- Install the position board to a personal computer which is compliant with PCI/CompactPCI®/PCI Express® standard. Failure to do so may cause a failure or malfunction.
- Securely insert the position board into the slot following the board installation instruction of the personal computer. Incorrect insertion of the position board may cause malfunction, failure, or drop of the board.
- When installing the position board, take care not to contact with other boards.
- When installing the position board, take care not to get injured by an implemented component or a surrounding member.
- Handle the position board in a place where static electricity will not be generated. Failure to do so may cause a failure or malfunction.
- The position board is included in an antistatic envelope. When storing or transporting it, be sure to put it in the antistatic envelope. Failure to do so may cause a failure or malfunction.
- Do not drop or apply a strong impact to the position board. Doing so may cause a failure or malfunction.

[Wiring Precautions]

⚠WARNING

- Shut off the external power supply (all phases) used in the system before installation and wiring. Failure to do so may result in electric shock or damage to the position board.
- After installation and wiring, attach the cover of the equipment the position board is installed to before turning it on for operation. Failure to do so may result in electric shock.

[Wiring Precautions]

⚠CAUTION

- Individually ground the FG terminals, the controllers, servo amplifiers and servo motors embedded with a position board with a ground resistance of 100 ohms or less. Failure to do so may result in electric shock or malfunction. Do not use a common grounding with other equipment.
- Use applicable solderless terminals and tighten them within the specified torque range. If any spade solderless terminal is used, it may be disconnected when the terminal screw comes loose, resulting in failure.
- Check the rated voltage and signal layout before wiring to the position board, and connect the cables correctly. Connecting a power supply with a different voltage rating or incorrect wiring may cause fire or failure.
- Connectors for external devices must be crimped or pressed with the tool specified by the manufacturer, or must be correctly soldered. Incomplete connections may cause short circuit, fire, or malfunction.
- Securely connect the connector to the position board. Poor contact may cause malfunction.
- Do not install the control lines or communication cables together with the main circuit lines or power cables. Doing so may result in malfunction due to noise. Keep a distance of 100 mm or more between those cables.
- Place the wires and the cables to connect the position board in a duct or clamp them. If not, dangling cables may swing or inadvertently be pulled, resulting in malfunction or damage to the position board or cables.
  In addition, the weight of the wires and the cables may put stress on the position board in an environment of strong vibrations and shocks.
- Check the interface type and correctly connect the cable. Incorrect wiring (connecting the cable to an incorrect interface) may cause failure of the position board and the external device.
- Tighten the terminal screws or connector screws within the specified torque range. Undertightening can cause drop of the screw, short circuit, fire, or malfunction. Overtightening can damage the screw and/or the position board, resulting in drop, short circuit, fire, or malfunction.
- When disconnecting the cable from the position board, do not pull the cable by the cable part. For the cable with the connector, hold the connector part of the cable. For the cable connected to the terminal block, loosen the terminal screw. Pulling the cable connected to the position board may result in malfunction or damage to the position board or the cable.
- Prevent foreign matter such as dust or wire chips from entering the personal computer embedded with a position board. Such foreign matter can cause a fire, failure, or malfunction.
- For cables to be used in the system, select the ones that meet the specifications in the MR-MC200/ MR-MC300 Series Position Board User's Manual (Details). If not, normal data transmission is not guaranteed.

[Startup and Maintenance Precautions]

⚠WARNING

- Do not touch any terminal while power is on. Doing so will cause electric shock or malfunction.
- Correctly connect the battery connector. Do not charge, disassemble, heat, short-circuit, solder, or throw the battery into the fire. Also, do not expose it to liquid or strong shock. Doing so will cause the battery to produce heat, explode, ignite, or leak, resulting in injury and fire.
- Shut off the external power supply (all phases) used in the system before cleaning the module or retightening the terminal screws, connector screws, or module fixing screws. Failure to do so may result in electric shock.

[Startup and Maintenance Precautions]

⚠CAUTION

- When connecting an external device with a position board or an intelligent function module to modify data of a running position board, configure an interlock in the program to ensure that the entire system will always operate safely. For other forms of control (such as program modification, parameter change, forced output, or operating status change) of a running position board, read the relevant manuals carefully and ensure that the operation is safe before proceeding. Improper operation may damage machines or cause accidents.
- Especially, when a remote position board is controlled by an external device, immediate action cannot be taken if a problem occurs in the position board due to a communication failure. To prevent this, configure an interlock in the program, and determine corrective actions to be taken between the external device and position board in case of a communication failure.
- Do not disassemble or modify the modules. Doing so may cause failure, malfunction, injury, or a fire.
- Use any radio communication device such as a cellular phone or PHS (Personal Handy-phone System) 25cm or more away in all directions from the position board. Failure to do so may cause malfunction.
- Shut off the external power supply (all phases) used in the system before mounting or removing the module. Failure to do so may cause the module to fail or malfunction.
- Tighten the screws within the specified torque range. Undertightening can cause drop of the component or wire, short circuit, or malfunction. Overtightening can damage the screw and/or module, resulting in drop, short circuit, or malfunction.
- After the first use of the product, do not mount/remove the position board more than 50 times. Exceeding the limit may cause malfunction.
- Startup and maintenance of a control panel must be performed by qualified maintenance personnel with knowledge of protection against electric shock. Lock the control panel so that only qualified maintenance personnel can operate it.
- Before handling the position board, touch a conducting object such as a grounded metal to discharge the static electricity from the human body. Wearing a grounded antistatic wrist strap is recommended. Failure to discharge the static electricity may cause the module to fail or malfunction.
- Use a clean and dry cloth to wipe off dirt on the position board.
- Before testing the operation, set a low speed value for the speed limit parameter so that the operation can be stopped immediately upon occurrence of a hazardous condition.
- Confirm and adjust the program and each parameter before operation. Unpredictable movements may occur depending on the machine.

[Startup and Maintenance Precautions]

⚠CAUTION

- 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 the operation, confirm the brake function.
- Do not perform a megger test (insulation resistance measurement) during inspection.
- After maintenance and inspections are completed, confirm that the position detection of the absolute position detection function is correct.
- Lock the control panel and prevent access to those who are not certified to handle or install electric equipment.
- The position board is included in an antistatic envelope. When storing or transporting it, be sure to put it in the antistatic envelope. Failure to do so may cause a failure or malfunction.
- The microprocessor built in the position board will reach a high temperature during operation. Do not touch the heat sink directly when replacing the position board. Doing so may result in a burn.

[Operating Precautions]

⚠CAUTION

- When changing data and operating status, and modifying program of the running position board from an external device such as a personal computer connected to an intelligent function module, read the relevant manuals carefully and ensure the safety before operation. Incorrect change or modification may cause system malfunction, damage to the machines, or accidents.
- Do not power off the position board or reboot the software while the setting values in the dual port memory are being written to the flash ROM in the position board. Doing so will make the data in the flash ROM undefined. The values need to be set in the dual port memory and written to the flash ROM again. Doing so also may cause malfunction or failure of the position board.
- Note that when the reference axis speed is specified for interpolation operation, the speed of the partner axis (2nd, 3rd, or 4th axis) may exceed the speed limit value.
- Do not go near the machine during test operations or during operations such as teaching. Doing so may lead to injuries.

[Computer Connection Precautions]

⚠CAUTION

- For Ethernet cables to be used in the system, select the ones that meet the specifications in the MR-MC200/MR-MC300 Series Position Board User's Manual (Details). If not, normal data transmission is not guaranteed.
- When connecting a personal computer to a module having a USB interface, observe the following precautions as well as the instructions described in the manual for the personal computer used. Failure to do so may cause the module to fail.
  (1) When the personal computer is AC-powered
  When the personal computer has a 3-pin AC plug or an AC plug with a grounding wire, connect the plug to a grounding receptacle or ground the grounding wire. Ground the personal computer and the module with a ground resistance of 100 ohms or less.
  When the personal computer has a 2-pin AC plug without a grounding wire, connect the computer to the module by following the procedure below. For power supplied to the personal computer and the module, using the same power source is recommended.
  1. Unplug the personal computer from the AC receptacle.
  2. Check that the personal computer is unplugged. Then, connect the personal computer to the module with a USB cable.
  3. Plug the personal computer into the AC receptacle.
  (2) When the personal computer is battery-powered
  The personal computer can be connected to the module without taking specific measures.

[Disposal Precautions]

⚠CAUTION

- When disposing of this product, treat it as industrial waste.

[Transportation Precautions]

⚠CAUTION

- The halogens (such as fluorine, chlorine, bromine, and iodine), which are contained in a fumigant used for disinfection and pest control of wood packaging materials, may cause failure of the product. Prevent the entry of fumigant residues into the product or consider other methods (such as heat treatment) instead of fumigation. The disinfection and pest control measures must be applied to unprocessed raw wood.
- The position board is a precision machine, so do not drop or apply strong impacts on it.

INTRODUCTION

Thank you for purchasing the Mitsubishi Electric position board.
This manual describes the API functions and the structures necessary for programming.
Before using this product, please read this manual and the relevant manuals carefully and develop familiarity with the functions and performance of the position board to handle the product correctly.
Please make sure that the end users read this manual.

Relevant products

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

Point

Symbols used in this manual are shown below.

- [MC200]: Symbols indicating that it corresponds to only MR-MC210/MR-MC211/MR-MC220U3/MR-MC220U6/MR-MC240/MR-MC241
- [MC300]: Symbols indicating that it corresponds to only MR-MC341

CONTENTS

CONTENTS

RELEVANT MANUALS

Manual name [manual number]	Description	Available form
MR-MC200/MR-MC300 Series Position Board User's Manual (API Library) [IB-0300225ENG] (This manual)	Library of functions and others that the host personal computer uses to control the position board.	Print book e-Manual PDF
MR-MC200/MR-MC300 Series Position Board User's Manual (Details) [IB-0300223ENG]	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.	Print book e-Manual PDF

Point

e-Manual refers to the Mitsubishi Electric FA electronic book manuals that can be browsed using a dedicated tool.

e-Manual has the following features:

- Required information can be cross-searched in multiple manuals.
- Other manuals can be accessed from the links in the manual.
- The hardware specifications of each part can be found from the product figures.
- Pages that users often browse can be bookmarked.
- Sample programs can be copied to an engineering tool.

TERMS

Unless otherwise specified, this manual uses the following terms.

Term	Description
API version	Software version of the API library for position board.
Buffer memory	Memory in a position board and an intelligent function module to store data such as setting values and monitor values.
Board version	System version of position board.
Control cycle	A cycle in which the SSCNET controller controls the operation such as command import, position control, status output, and communication with servo amplifier.
Intelligent function module	A module that has functions other than input and output, such as an A/D converter module and D/A converter module.
Link device	Internal devices (RX/RY/RWr/RWw) of the position board, sensing module, and SSCNETⅢ/H head module.
MR Configurator2	A product name of servo setup software.
MR-J3(W)-_B	Servo amplifier model MR-J3-_B_(-RJ)/MR-J3W-_B.
MR-J4(W_)-_B	Servo amplifier model MR-J4-_B_(-RJ)/MR-J4W_-_B.
MR-J5(W_)-_B	Servo amplifier model MR-J5-_B_(-RJ)/MR-J5W_-_B.
MR-MC2_ _	Another term 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_ _	Another term for PCI Express bus compatible position board MR-MC341.
Remote input (RX)	Information input from the sensing module, and SSCNETⅢ/H head module to the position board on a 1-bit basis.
Remote output (RY)	Information output from the position board to the sensing module, and SSCNETⅢ/H head module on a 1-bit basis.
Remote register (RWr)	Information for inputting to the position board from the sensing module, and SSCNETⅢ/H head module on a 16-bit (1-word) basis.
Remote register (RWw)	Information for outputting to the sensing module, and SSCNETⅢ/H head module from the position board on a 16-bit (1-word) basis.
SSCNETⅢ*1	High-speed synchronized network between the position board and the servo amplifier.
SSCNETⅢ/H*1	
Station No.	Station No. on the position board.
System program	Internal program that controls the position board.
Transient transmission	A function of data communication unperiodically among nodes (station) on network.
User program	Program created by the user that operates on the host personal computer.

*1 SSCNET: Servo System Controller NETwork

GENERIC TERMS AND ABBREVIATIONS

Unless otherwise specified, this manual uses the following generic terms and abbreviations.

Generic term/abbreviation	Description
API library	A generic term for the library of functions for positioning control that the host personal computer uses to control the position board.
Channel(CH)	An abbreviation for the SSCNET control channel.
Dual port memory	A generic term for a communication area in order to execute especially the motion control in the buffer memory.
Host personal computer	A generic term for computer equipped with position board and on which user program operates.
I/O device	A generic term for the link device on the dual port memory.
Position board	A generic term for MR-MC2_ _ and MR-MC3_ _.
Remote I/O module	A generic term for modules that connect I/O modules and intelligent function modules to SSCNETⅢ/H, including the sensing module and SSCNETⅢ/H head module.
Sensing module	A generic term for SSCNETⅢ/H compatible sensing module MR-MT2000 series.
Servo amplifier	An abbreviation for SSCNETⅢ/H compatible servo amplifier.
Test tool	An abbreviation for the position board test tool.
Utility software	A generic term for the Position Board Utility2 (MRZJW3-MC2-UTL) which includes the position board API library, the position board test tool, and the position board device driver.

1 SUMMARY

This API library is a collection of API functions for creating applications on the host personal computer 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 the API 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 the API library, up to 4 position boards can be used simultaneously.

Functional limitation depending on software version

Available functions are limited depending on the software version of the API library.
—: There is no restriction by the version.

Function/Item name	MR-MC2_ _		MR-MC3_ _		Change details
	API version	Board version	API version	Board version	
Digital input/output	1.02	A1	—	—	The function is added.
Digital output signal control for the other axes start	1.02	A1	—	—	The structure member is added.
Pass position interrupt	1.02	A1	—	—	The function is added.
Interrupt call back	1.02	—	—	—	The function is added.
Interrupt event notification	1.02	—	—	—	The function is added.
64-bit operating system compatible	1.10	—	—	—	The supported operating system is added.
Alarm history function	1.40	A3	—	—	The function is added.
Interface mode	1.40	A3	—	—	The function is added.
Speed-torque control compatible (interface mode only)	1.60	A4	—	—	The function is added.
Addition of position droop to high speed monitor (interface mode only)	1.60	A4	—	—	The function is added.
Mark detection function compatible	1.60	A5	—	—	The function is added.
Continuous operation to torque control compatible (automatic operation in standard mode only)	1.60	A5	—	—	The function is added.
Windows®8 compatible	1.60	—	—	—	The supported operating system is added.
Point table loop method compatible	1.80	A6	—	—	The function is added.
Remote I/O module compatible	1.80	A8	—	—	Extension of axis Nos.
I/O device compatible	1.80	A8	—	—	The function is added.
Transient transmit compatible	1.80	A8	—	—	The function is added.
Changeable interpolation group	1.90	A9	—	—	The function is added.
Sensing module compatible	1.90	B1	—	—	—
Windows®10 compatible	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	1.10	A1	The structure member is added.
MR-J5(W_)-_B compatible	Not supported	Not supported	1.30	A6	The function is added.

Point

- The API version can be checked by the file version in the property dialog of the files below.
  MR-MC2_ _: mc2xxstd.dll or mc2xxstd_x64.dll
  MR-MC3_ _: mc3xxstd.dll or mc3xxstd_x64.dll
  (Example) For version 1.00, the file version is shown as "1.0.0.0".
- For how to check the Board version, refer to "Confirming Serial No. and Operating System Software Version" in the following manual.
  MR-MC200/MR-MC300 Series Position Board User's Manual (Details)

2 CONDITIONS FOR USE

The following conditions when using the API functions apply.

- The 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

- The API functions provide the following library and header file.

Position board	32-bit operating system compatible library	64-bit operating system compatible library	Header file
MR-MC2_ _	• mc2xxstd.dll • mc2xxstd.lib (COFF format)	• mc2xxstd_x64.dll • mc2xxstd_x64.lib (COFF format)	• mc2xxstd.h (header file for C/C++) • mc2xxstd.cs (header file for C#) • mc2xxstd.vb (header file for 32-bit operating system compatible VB) • mc2xxstd_x64.vb (header file for 64-bit operating system compatible VB)
MR-MC3_ _	• mc3xxstd.dll • mc3xxstd.lib (COFF format)	• mc3xxstd_x64.dll • mc3xxstd_x64.lib (COFF format)	• mc3xxstd.h (header file for C/C++) • mc3xxstd.cs (header file for C#)

- The 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

- When using the API library by the Visual C# or Visual Basic project, execute the following.
  - Store the API library (.dll) in the same folder as the user program (.exe) that uses the API library. (Since the API library is not .NET Framework class library or a COM component, the dll references cannot be added to the project.)
  - Add the header file of the API library to the project.
  - In case of Visual C# projects, call the LoadLibraryDll function and load the library before use. Also, call the FreeLibraryDll function and unload the library after use. For details, refer to an API sample for C#.
- Only one user program can be opened simultaneously. Do not open the same position board by running two or more user programs.

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.

Visual C# project

The SscApi class of the mc2xxstd [MC200]/mc3xxstd [MC300] namespace defines this API.

When using the API library with a Visual C# project, perform the following.

- Read the argument type and the structure member type of each API as in the following table.

C++ argument type of API		C# argument type of API
unsigned char		byte
unsigned char*	in	ref byte
	out	out byte
unsigned char* (array head pointer)		byte[]
char		sbyte
char*	in	ref sbyte
	out	out sbyte
char* (array head pointer)		sbyte[]
unsigned short		ushort
unsigned short*	in	ref ushort
	out	out ushort
unsigned short* (array head pointer)		ushort[]
short		short
short*	in	ref short
	out	out short
short* (array head pointer)		short[]
unsigned int		uint
unsigned int*	in	ref uint
	out	out uint
unsigned int* (array head pointer)		uint[]
int		int
int*	in	ref int
	out	out int
int* (array head pointer)		int[]
unsigned long		uint
unsigned long*	in	ref uint
	out	out uint
unsigned long* (array head pointer)		uint[]
long		int
long*	in	ref int
	out	out int
long* (array head pointer)		int[]
unsigned long long		ulong
unsigned long long*	in	ref ulong
	out	out ulong
unsigned long long* (array head pointer)		ulong[]
long long		long
long long*	in	ref long
	out	out long
long long* (array head pointer)		long[]
wchar_t array		string

• When the argument of each API is structure pointer/structure array, read the argument type as in the following table.

C++ argument type of API		C# argument type of API
Structure pointer Example: PNT_DATA_EX*	in	ref PNT_DATA_EX Example: PNT_DATA_EX pnt_data_ex = new PNT_DATA_EX(); pnt_data_ex.Initialize(); SscApi.sscSetPointDataEx(board_id, channel, axnum, pntnum, ref pnt_data_ex);
	out	out PNT_DATA_EX Example: PNT_DATA_EX pnt_data_ex = new PNT_DATA_EX(); pnt_data_ex.Initialize(); SscApi.sscSetPointDataEx(board_id, channel, axnum, pntnum, out pnt_data_ex);

Structure array
Example: SMP_DATA* (array head pointer)
		SMP_DATA[]
		Example:
		SMP_DATA[] smp_data = new SMP_DATA[128];
		for (int i = 0; i < 128; i++)
		{
		  smp_data[i].Initialize();
		}
		SscApi.sscGetSamplingData(board_id, channel, page_num, out valid_num, smp_data);

3 LIST OF API FUNCTIONS

The list of the API functions is shown below.

○: Usable, △: Replacements recommend function, ×: Unusable

Function type	Function name	Function content	Usable/unusable		Reference section
			MR-MC2_ _	MR-MC3_ _	
Support functions	sscGetLastError	Gets the detailed error codes.	○	○	Page 28 sscGetLastError
	sscGetMountChannel	Gets the mount channel information.	○	○	Page 29 sscGetMountChannel
Device functions	sscOpen	Opens the memory access port.	○	○	Page 31 sscOpen
	sscClose	Closes the memory access port.	○	○	Page 32 sscClose
Information functions	sscGetControlCycle	Gets control cycle status.	○	○	Page 33 sscGetControlCycle
	sscGetBoardVersion	Gets position board system version information.	○	○	Page 34 sscGetBoardVersion
	sscGetDriverVersion	Gets the version information for the driver.	○	○	Page 35 sscGetDriverVersion
	sscGetOperationCycleMonitor	Gets operation cycle monitor data.	○	○	Page 36 sscGetOperationCycleMonitor
	sscClearOperationCycleMonitor	Clears (initializes) operation cycle monitor data.	○	○	Page 37 sscClearOperationCycleMonitor
	sscGetBoardSerialNumber	Gets position board serial No.	×	○	Page 38 sscGetBoardSerialNumber [MC300]
Parameter functions	sscResetAllParameter	Sets the initial values in all parameters before system startup.	○	○	Page 40 sscResetAllParameter
	sscChangeParameter	Writes the parameter. (Support for writing system parameter, servo parameter (MR-J4(W_)-_B/MR-J3(W)-_B, sensing module), control parameter, RIO module parameter, RIO control parameter)	○	○	Page 41 sscChangeParameter
	sscChange2Parameter	Writes the parameters (for 2 parameters). (Support for writing system parameter, servo parameter (MR-J4(W_)-_B/MR-J3(W)-_B, sensing module), control parameter, RIO module parameter, RIO control parameter)	○	○	Page 43 sscChange2Parameter
	sscCheckParameter	Reads the parameter set value. (Support for reading system parameter, servo parameter (MR-J4(W_)-_B/MR-J3(W)-_B, sensing module), control parameter, RIO module parameter, RIO control parameter)	○	○	Page 45 sscCheckParameter
	sscCheck2Parameter	Reads the parameter set values (for 2 parameters). (Support for reading system parameter, servo parameter (MR-J4(W_)-_B/MR-J3(W)-_B, sensing module), control parameter, RIO module parameter, RIO control parameter)	○	○	Page 47 sscCheck2Parameter
	sscLoadAllParameterFromFlashROM	Loads all the parameters from a flash ROM before system startup.	○	○	Page 49 sscLoadAllParameterFromFlashROM
	sscSaveAllParameterToFlashROM	Saves all the parameters into a flash ROM before system startup.	○	○	Page 50 sscSaveAllParameterToFlashROM
	sscCheckSvPrmChangeNumEx	Gets servo parameter change No.	○	○	Page 51 sscCheckSvPrmChangeNumEx
	sscChangeParameterEx	Writes the parameter. (Support for writing system parameter, servo parameter (MR-J5(W_)-_B/MR-J4(W_)-_B, sensing module), control parameter, RIO module parameter, RIO control parameter)	×	○	Page 52 sscChangeParameterEx [MC300]

Function type	Function name	Function content	Usable/unusable		Reference section
			MR-MC2_ _	MR-MC3_ _	
Parameter functions	sscChange2ParameterEx	Writes the parameters (for 2 parameters). (Support for writing system parameter, servo parameter (MR-J5(W_)-_B/MR-J4(W_)-_B, sensing module), control parameter, RIO module parameter, RIO control parameter)	×	○	Page 54 sscChange2ParameterEx [MC300]
	sscCheckParameterEx	Reads the parameter set value. (Support for reading system parameter, servo parameter (MR-J5(W_)-_B/MR-J4(W_)-_B, sensing module), control parameter, RIO module parameter, RIO control parameter)	×	○	Page 56 sscCheckParameterEx [MC300]
	sscCheck2ParameterEx	Reads the parameter set values (for 2 parameters). (Support for reading system parameter, servo parameter (MR-J5(W_)-_B/MR-J4(W_)-_B, sensing module), control parameter, RIO module parameter, RIO control parameter)	×	○	Page 58 sscCheck2ParameterEx [MC300]
System functions	sscReboot	Reboots the system.	○	○	Page 61 sscReboot
	sscSystemStart	Starts the system.	○	○	Page 62 sscSystemStart
	sscGetSystemStatusCode	Gets the system status code.	○	○	Page 63 sscGetSystemStatusCode
	sscReconnectSSCNET	Reconnects the SSCNET communication.	○	△	Page 64 sscReconnectSSCNET
	sscReconnectSSCNETEx		×	○	Page 65 sscReconnectSSCNETEx [MC300]
	sscDisconnectSSCNET	Disconnects the SSCNET communication.	○	△	Page 66 sscDisconnectSSCNET
	sscDisconnectSSCNETEx		×	○	Page 67 sscDisconnectSSCNETEx [MC300]
	sscGetControllingAxis	Gets the controlling axis information and controlling station information.	×	○	Page 68 sscGetControllingAxis [MC300]
Command/ status functions	sscSetCommandBitSignalEx	Arbitrarily sets the command bit.	○	○	Page 70 sscSetCommandBitSignalEx
	sscGetStatusBitSignalEx	Arbitrarily gets the status bit.	○	○	Page 71 sscGetStatusBitSignalEx
	sscWaitStatusBitSignalEx	Waits until the specified status bit turns ON/ OFF.	○	○	Page 72 sscWaitStatusBitSignalEx
Point table functions	sscSetPointDataEx	Sets the point data.	○	○	Page 74 sscSetPointDataEx
	sscCheckPointDataEx	Gets the point data.	○	○	Page 75 sscCheckPointDataEx
	sscSetPointOffset	Sets the point No. offset.	○	○	Page 76 sscSetPointOffset
	sscCheckPointOffset	Gets the point No. offset.	○	○	Page 77 sscCheckPointOffset
	sscGetDrivingPointNumber	Gets the operation point No.	○	○	Page 78 sscGetDrivingPointNumber
	sscSetLatestPointNumber	Sets the latest command point No.	○	○	Page 79 sscSetLatestPointNumber
Continuous operation to torque control data functions	sscSetPressData	Sets the continuous operation to torque control data.	○	○	Page 80 sscSetPressData
	sscGetPressData	Gets the continuous operation to torque control data.	○	○	Page 81 sscGetPressData

Function type	Function name	Function content	Usable/unusable		Reference section
			MR-MC2_ _	MR-MC3_ _	
Operating functions	sscJogStart	Starts JOG operation.	○	○	Page 82 sscJogStart
	sscJogStop	Stops JOG operation.	○	○	Page 84 sscJogStop
	sscJogStopNoWait	Stops JOG operation. (No wait function)	○	○	Page 85 sscJogStopNoWait
	sscIncStart	Starts incremental feed.	○	○	Page 86 sscIncStart
	sscAutoStart	Starts automatic operation.	○	○	Page 88 sscAutoStart
	sscHomeReturnStart	Starts home position return.	○	○	Page 89 sscHomeReturnStart
	sscLinearStart	Starts linear interpolation.	○	△	Page 90 sscLinearStart
	sscInterpolationStart	Starts interpolation operation.	×	○	Page 92 sscInterpolationStart [MC300]
	sscDataSetStart	Starts the home position reset (data set).	○	○	Page 94 sscDataSetStart
	sscDriveStop	Stops operation.	○	○	Page 95 sscDriveStop
	sscDriveStopNoWait	Stops operation. (No wait function)	○	○	Page 96 sscDriveStopNoWait
	sscDriveRapidStop	Stops operation rapidly.	○	○	Page 97 sscDriveRapidStop
	sscDriveRapidStopNoWait	Stops operation rapidly. (No wait function)	○	○	Page 98 sscDriveRapidStopNoWait
	sscSetDriveMode	Switches the operation mode.	○	○	Page 99 sscSetDriveMode
	sscGetDriveMode	Gets the operation mode.	○	○	Page 101 sscGetDriveMode
	sscGetDriveFinStatus	Gets the operation completion status.	○	○	Page 102 sscGetDriveFinStatus
	sscChangeControlMode	Switches the control mode of the servo amplifier.	○	○	Page 104 sscChangeControlMode
Change functions	sscChangeManualPosition	Changes position during incremental feed.	○	○	Page 105 sscChangeManualPosition
	sscChangeAutoPosition	Changes position during automatic operation.	○	○	Page 106 sscChangeAutoPosition
	sscChangeLinearPosition	Changes position during linear interpolation.	○	○	Page 107 sscChangeLinearPosition
	sscChangeManualSpeed	Changes speed of "JOG operation" or "incremental feed".	○	○	Page 109 sscChangeManualSpeed
	sscChangeAutoSpeed	Changes speed of "automatic operation" or "linear interpolation [MC200]/interpolation operation [MC300]".	○	○	Page 110 sscChangeAutoSpeed
	sscChangeManualAccTime	Changes acceleration time constant of "JOG operation" or "incremental feed".	○	○	Page 111 sscChangeManualAccTime
	sscChangeAutoAccTime	Changes acceleration time constant of "automatic operation" or "linear interpolation [MC200]/interpolation operation [MC300]".	○	○	Page 112 sscChangeAutoAccTime
	sscChangeManualDecTime	Changes deceleration time constant of "JOG operation" or "incremental feed".	○	○	Page 113 sscChangeManualDecTime
	sscChangeAutoDecTime	Changes deceleration time constant of "automatic operation" or "linear interpolation [MC200]/interpolation operation [MC300]".	○	○	Page 114 sscChangeAutoDecTime
Alarm functions	sscGetAlarm	Gets the alarm No. and the specific alarm No.	○	○	Page 115 sscGetAlarm
	sscResetAlarm	Resets the alarm.	○	○	Page 116 sscResetAlarm
General monitor functions	sscSetMonitor	Starts monitoring.	○	○	Page 119 sscSetMonitor
	sscStopMonitor	Stops monitoring.	○	○	Page 121 sscStopMonitor
	sscGetMonitor	Gets monitoring data.	○	○	Page 122 sscGetMonitor

Function type	Function name	Function content	Usable/unusable		Reference section
			MR-MC2_ _	MR-MC3_ _	
High speed monitor functions	sscGetCurrentCmdPositionFast	Gets the current command position.(High speed monitor function)	○	○	Page 123 sscGetCurrentCmdPositionFast
	sscGetCurrentFbPositionFast	Gets the current feedback position.(High speed monitor function)	○	○	Page 124 sscGetCurrentFbPositionFast
	sscGetIoStatusFast	Gets the external signal status.(High speed monitor function)	○	○	Page 125 sscGetIoStatusFast
	sscGetCmdSpeedFast	Gets the moving speed. (High speed monitor function)	○	○	Page 126 sscGetCmdSpeedFast
	sscGetFbSpeedFast	Gets the feedback moving speed. (High speed monitor function)	○	○	Page 127 sscGetFbSpeedFast
	sscGetCurrentFbFast	Gets the electrical current feedback. (High speed monitor function)	○	○	Page 128 sscGetCurrentFbFast
	sscGetPositionDroopFast	Gets the position droop. (High speed monitor function)	○	○	Page 129 sscGetPositionDroopFast
User watchdog functions	sscWdEnable	Enables the user watchdog function.	○	○	Page 130 sscWdEnable
	sscWdDisable	Disables the user watchdog function.	○	○	Page 131 sscWdDisable
	sscChangeWdCounter	Updates the watchdog check counter.	○	○	Page 132 sscChangeWdCounter
Other axes start functions	sscSetOtherAxisStartData	Sets the data for starting other axes.	○	○	Page 133 sscSetOtherAxisStartData
	sscGetOtherAxisStartData	Gets the data for starting other axes.	○	○	Page 134 sscGetOtherAxisStartData
	sscOtherAxisStartAbortOn	Turns the other axes start cancel signal (OSSTP□) to ON.	○	○	Page 135 sscOtherAxisStartAbortOn
	sscOtherAxisStartAbortOff	Turns the other axes start cancel signal (OSSTP□) to OFF.	○	○	Page 136 sscOtherAxisStartAbortOff
	sscGetOtherAxisStartStatus	Gets the other axes start status.	○	○	Page 137 sscGetOtherAxisStartStatus
Pass position interrupt functions	sscSetIntPassPositionData	Sets the pass position interrupt condition data.	○	○	Page 138 sscSetIntPassPositionData
	sscCheckIntPassPositionData	Gets the pass position interrupt condition data.	○	○	Page 139 sscCheckIntPassPositionData
	sscSetStartingPassNumber	Sets the pass position condition start and end Nos.	○	○	Page 140 sscSetStartingPassNumber
	sscGetExecutingPassNumber	Gets the running pass position condition No.	○	○	Page 141 sscGetExecutingPassNumber
Sampling functions	sscStartSampling	Starts sampling.	○	○	Page 142 sscStartSampling
	sscStopSampling	Stops sampling.	○	○	Page 143 sscStopSampling
	sscSetSamplingParameter	Writes the sampling parameters.	○	○	Page 144 sscSetSamplingParameter
	sscGetSamplingParameter	Reads the sampling parameters.	○	○	Page 145 sscGetSamplingParameter
	sscGetSamplingError	Gets the sampling error information.	○	○	Page 146 sscGetSamplingError
	sscGetSamplingStatus	Gets the sampling execution status.	○	○	Page 147 sscGetSamplingStatus
	sscGetSamplingData	Gets the sampling data.	○	○	Page 148 sscGetSamplingData
Log functions	sscStartLog	Starts the log.	○	○	Page 149 sscStartLog
	sscStopLog	Stops the log.	○	○	Page 150 sscStopLog
	sscCheckLogStatus	Gets the running status of the log.	○	○	Page 151 sscCheckLogStatus
	sscCheckLogEventNum	Gets the number of valid log data events.	○	○	Page 152 sscCheckLogEventNum
	sscReadLogData	Reads the log data.	○	○	Page 153 sscReadLogData
	sscClearLogData	Clears (initializes) the log data.	○	○	Page 154 sscClearLogData
	sscGetAlarmHistoryData	Gets alarm history data.	○	○	Page 155 sscGetAlarmHistoryData
	sscCheckAlarmHistoryEventNum	Gets the number of valid alarm history data events.	○	○	Page 156 sscCheckAlarmHistoryEventNum
	sscClearAlarmHistoryData	Clears (initializes) the alarm history data.	○	○	Page 157 sscClearAlarmHistoryData

Function type	Function name	Function content	Usable/unusable		Reference section
			MR-MC2_ _	MR-MC3_ _	
Digital input/output functions	sscGetDigitalInputDataBit	Gets the DI data of the designated digital input on a 1-point basis.	○	○	Page 158 sscGetDigitalInputDataBit
	sscGetDigitalInputDataWord	Gets the DI data of the designated digital input on a 16-point basis.	○	○	Page 159 sscGetDigitalInputDataWord
	sscSetDigitalOutputDataBit	Sets the DO data of the designated digital output on a 1-point basis.	○	○	Page 160 sscSetDigitalOutputDataBit
	sscSetDigitalOutputDataWord	Sets the DO data of the designated digital output on a 16-point basis.	○	○	Page 161 sscSetDigitalOutputDataWord
	sscGetDigitalOutputDataBit	Gets the DO data of the designated digital output on a 1-point basis.	○	○	Page 162 sscGetDigitalOutputDataBit
	sscGetDigitalOutputDataWord	Gets the DO data of the designated digital output on a 16-point basis.	○	○	Page 163 sscGetDigitalOutputDataWord
Mark detection functions	sscGetMarkDetectionData	Gets mark detection data.	○	○	Page 164 sscGetMarkDetectionData
	sscGetMarkDetectionCounter	Gets mark detection counter.	○	○	Page 166 sscGetMarkDetectionCounter
	sscClearMarkDetectionData	Clears (initializes) the mark detection data.	○	○	Page 167 sscClearMarkDetectionData
Interface mode functions	sscIfmGetReadErrorCount	Gets read error counter.	○	○	Page 168 sscIfmGetReadErrorCount
	sscIfmSetHomePosition	Performs home position set.	○	○	Page 169 sscIfmSetHomePosition
	sscIfmGetMaximumBufferNumber	Gets maximum buffer No. (Only position control mode)	○	△	Page 170 sscIfmGetMaximumBufferNumber
	sscIfmGetMaximumBufferNumberEx	Gets maximum buffer No. of the designated control mode.	○	○	Page 171 sscIfmGetMaximumBufferNumberEx
	sscIfmRenewLatestBuffer	Renews the latest command buffer No. and data. (Only position control mode)	○	△	Page 172 sscIfmRenewLatestBuffer
	sscIfmRenewLatestBufferEx	Renews the latest command buffer No. and data of the designated control mode.	○	○	Page 173 sscIfmRenewLatestBufferEx
	sscIfmCheckLatestBuffer	Gets the latest command buffer No. and data. (Only position control mode)	○	△	Page 174 sscIfmCheckLatestBuffer
	sscIfmCheckLatestBufferEx	Gets the latest command buffer No. and data of the designated control mode.	○	○	Page 175 sscIfmCheckLatestBufferEx
	sscIfmGetTransmitBuffer	Gets the transmit buffer No. and data. (Only position control mode)	○	△	Page 176 sscIfmGetTransmitBuffer
	sscIfmGetTransmitBufferEx	Gets the transmit buffer No. and data of the designated control mode.	○	○	Page 177 sscIfmGetTransmitBufferEx
	sscIfmTrqSetSpeedLimit	Sets the speed limit value for torque control.	○	○	Page 178 sscIfmTrqSetSpeedLimit
	sscIfmSetControlMode	Sets the control mode.	○	○	Page 179 sscIfmSetControlMode
	sscIfmGetControlMode	Gets the control mode.	○	○	Page 180 sscIfmGetControlMode
	sscIfmGetEventStatusBits	Gets the status bit information of all axes for the designated status signal using the event detect function.	○	△	Page 181 sscIfmGetEventStatusBits
	sscIfmGetEventStatusBitsEx		×	○	Page 182 sscIfmGetEventStatusBitsEx [MC300]

Function type	Function name	Function content	Usable/unusable		Reference section
			MR-MC2_ _	MR-MC3_ _	
Interrupt functions	sscIntStart	Starts up the interrupt driver.	○	○	Page 185 sscIntStart
	sscIntEnd	Closes the interrupt driver.	○	○	Page 186 sscIntEnd
	sscIntEnable	Enables interrupt output.	○	○	Page 187 sscIntEnable
	sscIntDisable	Disables interrupt output.	○	○	Page 188 sscIntDisable
	sscRegisterIntCallback	Registers the interrupt callback function.	○	○	Page 189 sscRegisterIntCallback
	sscUnregisterIntCallback	Unregisters the interrupt callback function.	○	○	Page 190 sscUnregisterIntCallback
	sscResetIntEvent	Sets the interrupt event signal status to nonsignaled.	○	○	Page 191 sscResetIntEvent
	sscSetIntEvent	Sets the interrupt event signal status to signaled.	○	○	Page 192 sscSetIntEvent
	sscWaitIntEvent	Waits until the interrupt event status becomes signaled.	○	○	Page 193 sscWaitIntEvent
	sscResetIntEventMulti	Sets the statuses of the multiple interrupt events to nonsignaled.	○	○	Page 195 sscResetIntEventMulti
	sscSetIntEventMulti	Sets the statuses of the multiple interrupt events to signaled.	○	○	Page 196 sscSetIntEventMulti
	sscWaitIntEventMulti	Waits until the statuses of the multiple interrupt events become signaled.	○	○	Page 197 sscWaitIntEventMulti
	sscResetIntOasEvent	Sets the status of the other axes start interrupt event to nonsignaled.	○	○	Page 199 sscResetIntOasEvent
	sscSetIntOasEvent	Sets the status of the other axes start interrupt event to signaled.	○	○	Page 200 sscSetIntOasEvent
	sscWaitIntOasEvent	Waits until the status of the other axes start interrupt event becomes signaled.	○	○	Page 201 sscWaitIntOasEvent
	sscResetIntPassPosition	Sets the status of the pass position interrupt event to nonsignaled.	○	○	Page 203 sscResetIntPassPosition
	sscSetIntPassPosition	Sets the status of the pass position interrupt event to signaled.	○	○	Page 204 sscSetIntPassPosition
	sscWaitIntPassPosition	Waits until the status of the pass position interrupt event becomes signaled.	○	○	Page 205 sscWaitIntPassPosition
	sscResetIntDriveFin	Sets the status of the operation completion interrupt event to nonsignaled.	○	○	Page 207 sscResetIntDriveFin
	sscSetIntDriveFin	Sets the status of the operation completion interrupt event to signaled.	○	○	Page 208 sscSetIntDriveFin
	sscWaitIntDriveFin	Waits until the status of the operation completion interrupt event becomes signaled.	○	○	Page 209 sscWaitIntDriveFin
I/O device functions	sscGetInputDeviceBit	Gets the designated input bit device on a 1-point basis.	○	○	Page 211 sscGetInputDeviceBit
	sscGetInputDeviceWord	Gets the designated input word device on a 1-word basis.	○	○	Page 212 sscGetInputDeviceWord
	sscSetOutputDeviceBit	Sets the designated output bit device on a 1-point basis with the dual port memory exclusive control function of the position board.	○	○	Page 213 sscSetOutputDeviceBit
	sscSetOutputDeviceWord	Sets the designated output word device on a 1-word basis without using the dual port memory exclusive control function of the position board.	○	○	Page 214 sscSetOutputDeviceWord
	sscGetOutputDeviceBit	Gets the designated output bit device on a 1-point basis.	○	○	Page 215 sscGetOutputDeviceBit
	sscGetOutputDeviceWord	Gets the designated output word device on a 1-word basis.	○	○	Page 216 sscGetOutputDeviceWord
Transient transmit functions	sscSendReceiveTransientData	Sends and receives the specified transient transmit data for axes or stations connected to SSCNET.	○	○	Page 217 sscSendReceiveTransientData

4 API FUNCTION DETAILS

4.1 Support Functions

sscGetLastError

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

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

Detailed description

■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 version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

None.

sscGetMountChannel

The mount channel information is got.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
mountch [out]	Pointer to 2-byte variable (2 bytes × 1) which stores the mount channel information

■Return value

Value	Description
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

Constant definition	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 version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

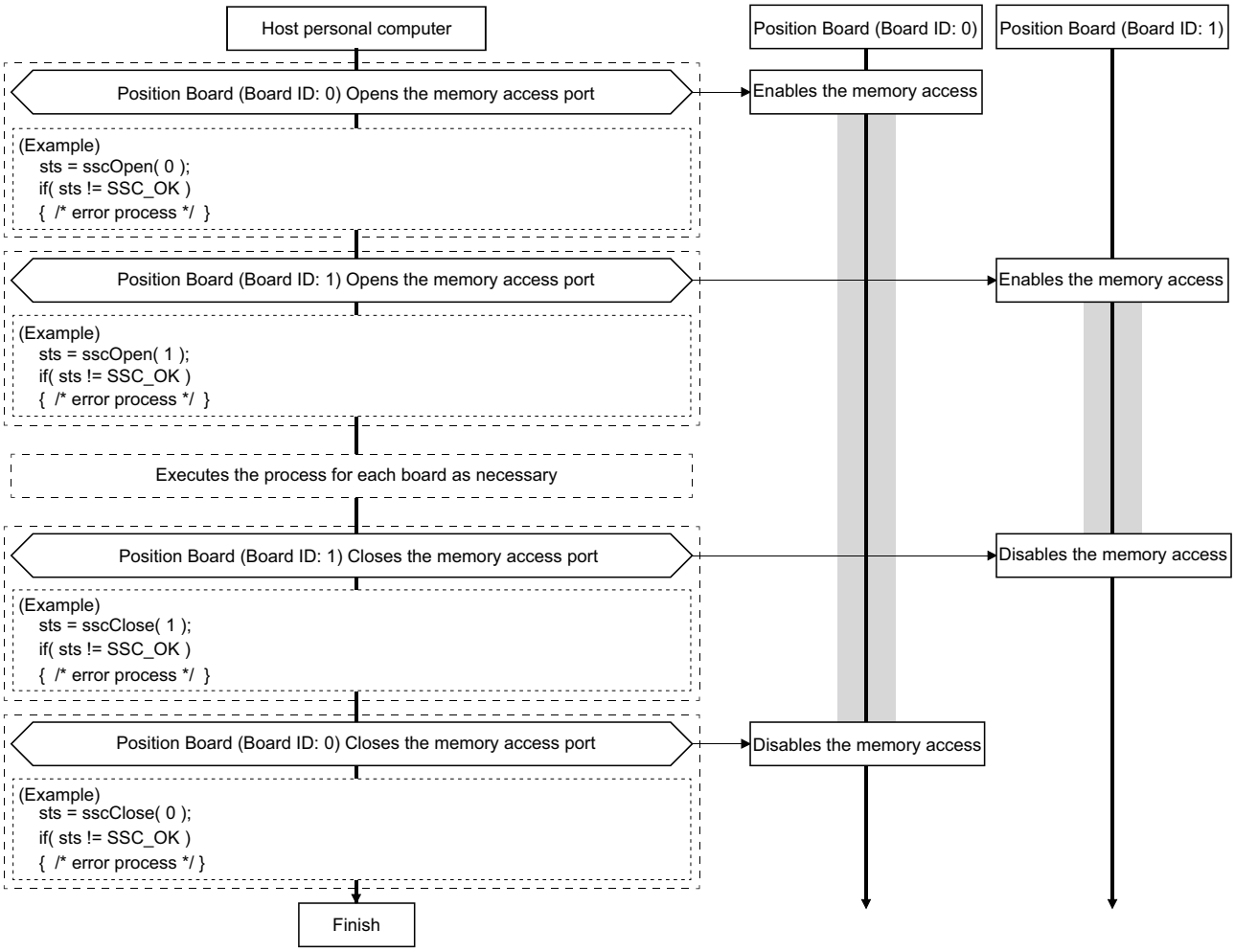
None.

4.2 Device Functions

Processing procedure

The device processing procedure for the memory access is shown below.

■An example of device processing procedure for memory access when 2 position boards (Board ID: 0 and Board ID: 1) are connected to the host personal computer is below.



Point

- When the same device (Board ID) is not used, the memory access ports can be opened at the same time.
- Do not call the sscOpen function/sscClose function sequentially.
- By organizing open/close of a memory access port at the beginning and the end of the user program process, an error with unopened memory access port when calling the API function can be prevented.
  (Example) User program process
  After turning ON the power, open the memory access ports in the user program initial process.
  ↓
  Execute an arbitrary process while energizing.
  ↓
  Before turning OFF the power, close the memory access ports in the user program end process.

sscOpen

The memory access port is opened.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)

■Return value

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

■Detailed error code

Constant definition	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 could not 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 (Windows API). Call the GetLastError function of Windows API and confirm the error details.
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. [MC300]
SSC_FUNC_ERR_UNSUPPORT_DEVICE_DRIVER	The device driver is not a supported version. Use the 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 version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscClose (☞ Page 32 sscClose)

sscClose

The memory access port is closed.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)

■Return value

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

■Detailed error code

Constant definition	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 (Windows API). Call the GetLastError function of Windows API and confirm the error details.

■Point

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

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscOpen (☞ Page 31 sscOpen)

4.3 Information Functions

sscGetControlCycle

The control cycle status is got.

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


Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
ctrl_cycle [out]	Pointer to 2-byte variable (2 bytes × 1) which stores the control cycle status ■Value • 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

Value	Description
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

Constant definition	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 version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

None.

sscGetBoardVersion

The system version information of the position board is got.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
version [out]	Pointer to 16-byte array (1 byte × 16) which stores the system version information

■Return value

Value	Description
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

Constant definition	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 version information.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscGetDriverVersion (☞ Page 35 sscGetDriverVersion)

sscGetDriverVersion

The driver version information is got.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (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

Value	Description
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

Constant definition	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 version information.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscGetBoardVersion (☞ Page 34 sscGetBoardVersion)

sscGetOperationCycleMonitor

The operation cycle monitor data is got.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (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 • SSC_BIT_OCME: Operation cycle alarm • SSC_BIT_OCMW: Operation cycle warning

■Return value

Value	Description
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

Constant definition	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 version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscClearOperationCycleMonitor (☞ Page 37 sscClearOperationCycleMonitor)

sscClearOperationCycleMonitor

The operation cycle monitor data is cleared (initialized).

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)

■Return value

Value	Description
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

Constant definition	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 version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscGetOperationCycleMonitor (☞ Page 36 sscGetOperationCycleMonitor)

sscGetBoardSerialNumber [MC300]

Gets the position board serial No.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
serialnum [out]	Pointer to 16-byte array (1 byte × 16) which stores the serial No.

■Return value

Value	Description
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

Constant definition	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 No.
- For the char* type argument of the API function for the Visual C# project, replace it with the sbyte[] type.

■Supported version

Position board	API version	Board version	Header file
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscGetBoardVersion (☞ Page 34 sscGetBoardVersion)

4.4 Parameter Functions

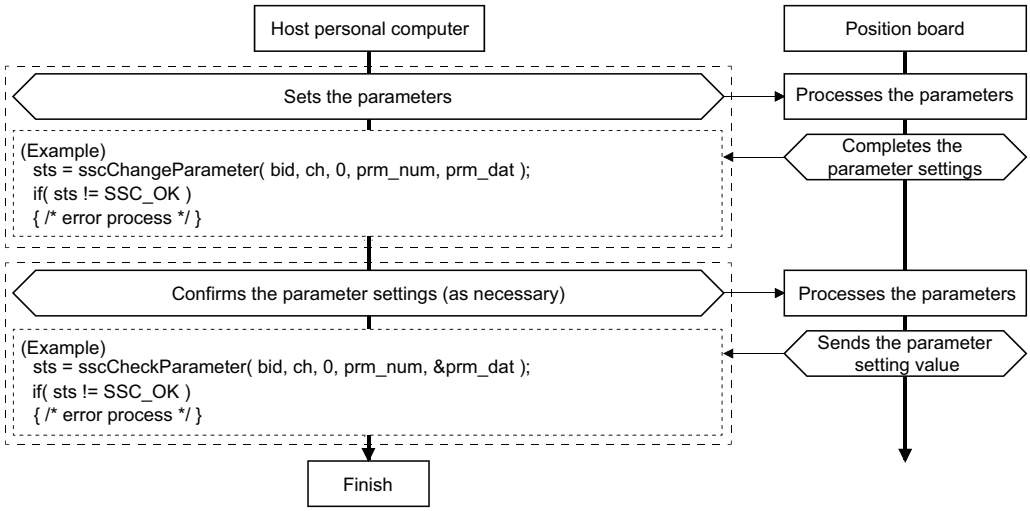
Processing procedure

An example of parameter processing procedure is shown below.

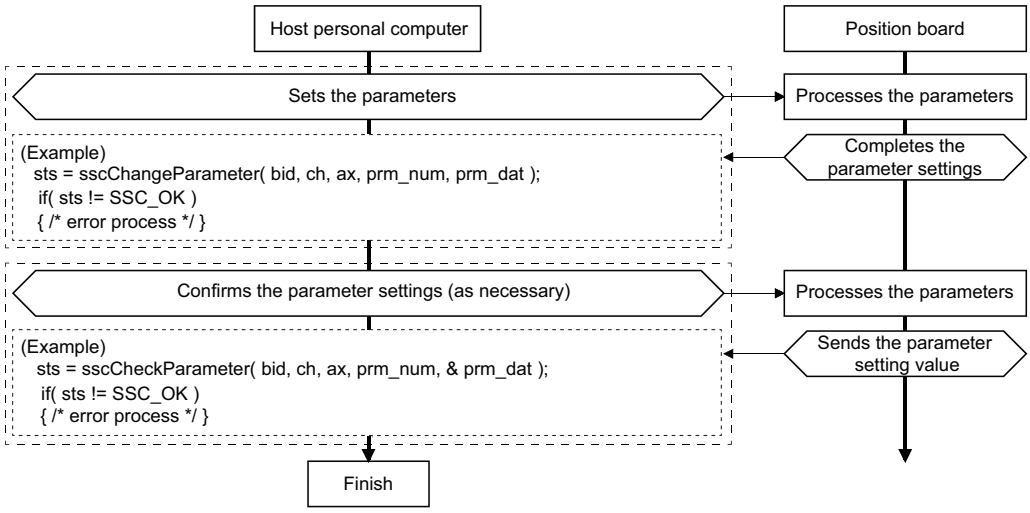
Point

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

■When setting the system parameters



■When setting the control/servo parameters



sscResetAllParameter

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

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
timeout [in]	Timeout time [ms] (0 to 65535)

■Return value

Value	Description
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

Constant definition	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 is 2 seconds (2000ms).

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

None.

sscChangeParameter

Each of the parameters is written.

This API function supports the writing of the system parameters, servo parameters (MR-J4(W_)-_B/MR-J3(W)-_B), sensing module, control parameters, RIO module parameters, and RIO control parameters. Due to unsupport of the servo parameters of MR-J5(W_)-_B, when writing the servo parameters of MR-J5(W_)-_B, use the sscChangeParameterEx function. [MC300]

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


Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (-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 No.
prmdata [in]	Parameter write data

■Return value

Value	Description
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

Constant definition	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 No. error occurred. • A value outside the range is set in the parameter write No.1. • The axis No. and the parameter write No. do not correspond. (Example: When "System parameter" is set to the axis No. and "Axis parameter" is set to the parameter write No., 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 No.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.
SSC_FUNC_ERR_PARAM_WRITE_EX_NUM1 [MC300]	The servo parameters of MR-J5(W_)-_B cannot be written. Use the sscChangeParameterEx function.

■Point

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

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.80	A8	mc2xxstd.h
MR-MC3_ _	1.30	A6	mc3xxstd.h

■Reference

- sscChange2Parameter (☞ Page 43 sscChange2Parameter)
- sscCheckParameter (☞ Page 45 sscCheckParameter)
- sscCheck2Parameter (☞ Page 47 sscCheck2Parameter)
- sscChangeParameterEx [MC300] (☞ Page 52 sscChangeParameterEx [MC300])
- sscChange2ParameterEx [MC300] (☞ Page 54 sscChange2ParameterEx [MC300])
- sscCheckParameterEx [MC300] (☞ Page 56 sscCheckParameterEx [MC300])
- sscCheck2ParameterEx [MC300] (☞ Page 58 sscCheck2ParameterEx [MC300])

sscChange2Parameter

Two of the parameters are written.
This API function supports the writing of the system parameters, servo parameters (MR-J4(W_)-_B/MR-J3(W)-_B), sensing module, control parameters, RIO module parameters, and RIO control parameters.
Due to unsupport of the servo parameters of MR-J5(W_)-_B, when writing the servo parameters of MR-J5(W_)-_B, use the sscChange2ParameterEx function. [MC300]

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


Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (-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 Nos. (for 2 Nos.)
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 • SSC_BIT_PWFIN: Parameter write completed • SSC_BIT_PWEN: Parameter No. error • SSC_BIT_PWED: Parameter data out of bounds

■Return value

Value	Description
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

Constant definition	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 No. of the parameter write Nos. (for 2 Nos.)	A parameter write No. error occurred. • A value outside the range is set in the parameter write No. • The axis No. and the parameter write No. do not correspond. (Example: When "System parameter" is set to the axis No. and "Axis parameter" is set to the parameter write No., etc.)
SSC_FUNC_ERR_STS_BIT_PWED☐ ☐ = 1 to 2: Array No. 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 No. of the parameter write Nos. (for 2 Nos.)	The command and the status of the parameter write No. do not correspond.
SSC_FUNC_ERR_MISMATCH_PARAM_WRITE_DATA☐ ☐ = 1 to 2: Array No. of the parameter write data (for 2 data)	The command and the status of the parameter write data do not correspond.
SSC_FUNC_ERR_PARAM_WRITE_EX_NUM☐ [MC300] ☐ = 1 to 2: Array No. of the parameter write data (for 2 data)	The servo parameters of MR-J5(W_)-_B cannot be written. Use the sscChange2ParameterEx function.

■Point

- This API 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 confirm the response of the position board.
- When a number of parameters are written to multiple axes sequentially, the write time can be shortened by calling this API function in each thread of "system" or "axis".

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.80	A8	mc2xxstd.h
MR-MC3_ _	1.30	A6	mc3xxstd.h

■Reference

- sscChangeParameter (☞ Page 41 sscChangeParameter)
- sscCheckParameter (☞ Page 45 sscCheckParameter)
- sscCheck2Parameter (☞ Page 47 sscCheck2Parameter)
- sscChangeParameterEx [MC300] (☞ Page 52 sscChangeParameterEx [MC300])
- sscChange2ParameterEx [MC300] (☞ Page 54 sscChange2ParameterEx [MC300])
- sscCheckParameterEx [MC300] (☞ Page 56 sscCheckParameterEx [MC300])
- sscCheck2ParameterEx [MC300] (☞ Page 58 sscCheck2ParameterEx [MC300])

sscCheckParameter

The set value of the designated parameter is read.
This API function supports the reading of the system parameters, servo parameters (MR-J4(W_)-_B/MR-J3(W)-_B), sensing module, control parameters, RIO module parameters, and RIO control parameters.
Due to unsupport of the servo parameters of MR-J5(W_)-_B, when reading the servo parameters of MR-J5(W_)-_B, use the sscCheckParameterEx function. [MC300]

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


Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (-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 No.
prmdata [out]	Pointer to 2-byte variable (2 bytes × 1) which stores the parameter read data

■Return value

Value	Description
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

Constant definition	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 No.1. • The axis No. and the parameter read No. do not correspond. (Example: When "System parameter" is set to the axis No. and "Axis parameter" is set to the parameter read No., etc.)
SSC_FUNC_ERR_MISMATCH_PARAM_READ_NUM1	The command and the status of the parameter read No.1 do not correspond.
SSC_FUNC_ERR_PARAM_READ_EX_NUM1 [MC300]	The servo parameters of MR-J5(W_)-_B cannot be read. Use the sscChange2ParameterEx function.

■Point

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

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.80	A8	mc2xxstd.h
MR-MC3_ _	1.30	A6	mc3xxstd.h

■Reference

- sscChangeParameter (☞ Page 41 sscChangeParameter)
- sscChange2Parameter (☞ Page 43 sscChange2Parameter)
- sscCheck2Parameter (☞ Page 47 sscCheck2Parameter)
- sscChangeParameterEx [MC300] (☞ Page 52 sscChangeParameterEx [MC300])
- sscChange2ParameterEx [MC300] (☞ Page 54 sscChange2ParameterEx [MC300])
- sscCheckParameterEx [MC300] (☞ Page 56 sscCheckParameterEx [MC300])
- sscCheck2ParameterEx [MC300] (☞ Page 58 sscCheck2ParameterEx [MC300])

sscCheck2Parameter

Two set values of the designated parameters are read.
This API function supports the reading of the system parameters, servo parameters (MR-J4(W_)-_B/MR-J3(W)-_B), sensing module, control parameters, RIO module parameters, and RIO control parameters.
Due to unsupport of the servo parameters of MR-J5(W_)-_B, when reading the servo parameters of MR-J5(W_)-_B, use the sscCheck2ParameterEx function. [MC300]

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


Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (-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 Nos. (for 2 Nos.)
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 • SSC_BIT_PRFIN: Parameter read completed • SSC_BIT_PREN: Parameter No. error

■Return value

Value	Description
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

Constant definition	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 No. of the parameter read Nos. (for 2 Nos.)	A parameter read error occurred. • A value outside the range is set in the parameter read No. • The axis No. and the parameter read No. do not correspond. (Example: When "System parameter" is set to the axis No. and "Axis parameter" is set to the parameter read No., etc.)
SSC_FUNC_ERR_MISMATCH_PARAM_READ_NUM☐ ☐ = 1 to 2: Array No. of the parameter read Nos. (for 2 Nos.)	The command and the status of the parameter read No. do not correspond.
SSC_FUNC_ERR_PARAM_READ_EX_NUM☐ [MC300] ☐ = 1 to 2: Array No. of the parameter read Nos. (for 2 Nos.)	The servo parameters of MR-J5(W_)-_B cannot be read. Use the sscCheck2ParameterEx function.

■Point

- This API 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 confirm the response of the position board.
- When a number of multiple parameters are read from multiple axes sequentially, the read time can be shortened by calling this API function in each thread of "system" or "axis".

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.80	A8	mc2xxstd.h
MR-MC3_ _	1.30	A6	mc3xxstd.h

■Reference

- sscChangeParameter (☞ Page 41 sscChangeParameter)
- sscChange2Parameter (☞ Page 43 sscChange2Parameter)
- sscCheck2Parameter (☞ Page 47 sscCheck2Parameter)
- sscChangeParameterEx [MC300] (☞ Page 52 sscChangeParameterEx [MC300])
- sscChange2ParameterEx [MC300] (☞ Page 54 sscChange2ParameterEx [MC300])
- sscCheckParameterEx [MC300] (☞ Page 56 sscCheckParameterEx [MC300])
- sscCheck2ParameterEx [MC300] (☞ Page 58 sscCheck2ParameterEx [MC300])

sscLoadAllParameterFromFlashROM

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

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
timeout [in]	Timeout time [ms] (0 to 65535)

■Return value

Value	Description
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

Constant definition	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 is 2 seconds (2000ms) [MC200]/5 seconds (5000ms) [MC300].

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscSaveAllParameterToFlashROM (☞ Page 50 sscSaveAllParameterToFlashROM)

sscSaveAllParameterToFlashROM

All parameters are saved to the flash ROM.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
timeout [in]	Timeout time [ms] (0 to 65535 [MC200]/0 to 600000 [MC300])

■Return value

Value	Description
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

Constant definition	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 signal (FRNG) is ON.
SSC_FUNC_ERR_STS_BIT_FSNG	The flash ROM transfer error signal (FSNG) is ON.

■Point

- When a value 10 seconds (10000ms) [MC200]/300 seconds (300000ms) [MC300] or less is designated as the timeout time, the timeout is 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 version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscLoadAllParameterFromFlashROM (☞ Page 49 sscLoadAllParameterFromFlashROM)

sscCheckSvPrmChangeNumEx

The servo parameter change No. is got.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (1 to 32 [MC200]/1 to 64 [MC300])
prmnum [out]	Pointer to 8-byte array (2 bytes × 4) which stores the servo parameter change No.

■Return value

Value	Description
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

Constant definition	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 No. [MC200]
- Use monitor for detailed information of the servo parameter change No.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.60	A0	mc2xxstd.h
MR-MC3_ _	1.30	A6	mc3xxstd.h

■Reference

None.

sscChangeParameterEx [MC300]

Each of the parameters is written.

This API function supports the writing of the system parameters, servo parameters (MR-J5(W_)-_B/MR-J4(W_)-_B), sensing module, control parameters, RIO module parameters, and RIO control parameters.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (-16 to 64) • 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 No.
prmdata [in]	Parameter write data

■Return value

Value	Description
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

Constant definition	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 No. error occurred. • A value outside the range is set in the parameter write No.1. • The axis No. and the parameter write No. do not correspond. (Example: When "System parameter" is set to the axis No. and "Axis parameter" is set to the parameter write No., 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 No.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.
SSC_FUNC_ERR_UNSUPPORT_PARAM_EX	The position board does not support the servo parameters of MR-J5(W_)-_B.

■Point

- This API function can write the parameters either before or while the system is running.
- It takes about 2 to 120 control cycles to confirm the response of the position board.
- When a number of parameters are written to multiple axes sequentially, the write time can be shortened by calling this API function in each thread of "system" or "axis".
- Parameter write data size differs depending on the parameter write No.

Axis No.	Parameter write No.	Parameter write data size
0	0000h to FFFFh	2 bytes (Set the data size to lower 2 bytes.)
1 to 64	2000h to 2FFFh	4 bytes
	Other than the above-mentioned	2 bytes (Set the data size to lower 2 bytes.)
-16 to -1	0000h to FFFFh	2 bytes (Set the data size to lower 2 bytes.)

■Supported version

Position board	API version	Board version	Header file
MR-MC3_ _	1.30	A6	mc3xxstd.h

■Reference

- sscChangeParameter (☞ Page 41 sscChangeParameter)
- sscChange2Parameter (☞ Page 43 sscChange2Parameter)
- sscCheckParameter (☞ Page 45 sscCheckParameter)
- sscCheck2Parameter (☞ Page 47 sscCheck2Parameter)
- sscChange2ParameterEx (☞ Page 54 sscChange2ParameterEx [MC300])
- sscCheckParameterEx (☞ Page 56 sscCheckParameterEx [MC300])
- sscCheck2ParameterEx (☞ Page 58 sscCheck2ParameterEx [MC300])

sscChange2ParameterEx [MC300]

Two of the parameters are written.

This API function supports the writing of the system parameters, servo parameters (MR-J5(W_)-_B/MR-J4(W_)-_B), sensing module, control parameters, RIO module parameters, and RIO control parameters.

```
int  sscChange2ParameterEx (
     int  board_id,
     int  channel,
     int  axnum,
     unsigned short  *prmnum,
     unsigned short  *prmdata,
     unsigned char  *status
);
```

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (-16 to 64) • 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 Nos. (for 2 Nos.)
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 • SSC_BIT_PWFIN: Parameter write completed • SSC_BIT_PWEN: Parameter No. error • SSC_BIT_PWED: Parameter data out of bounds

■Return value

Value	Description
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

Constant definition	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 No. of the parameter write Nos. (for 2 Nos.)	A parameter write No. error occurred. • A value outside the range is set in the parameter write No. • The axis No. and the parameter write No. do not correspond. (Example: When "System parameter" is set to the axis No. and "Axis parameter" is set to the parameter write No., etc.)
SSC_FUNC_ERR_STS_BIT_PWED□ □ = 1 to 2: Array No. 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 No. of the parameter write Nos. (for 2 Nos.)	The command and the status of the parameter write No. do not correspond.
SSC_FUNC_ERR_MISMATCH_PARAM_WRITE_DATA□ □ = 1 to 2: Array No. of the parameter write data (for 2 data)	The command and the status of the parameter write data do not correspond.
SSC_FUNC_ERR_PARAM_WRITE_EX	The position board does not support the servo parameters of MR-J5(W_)-_B.

■Point

- This API 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 120 control cycles to confirm the response of the position board.
- When a number of parameters are written to multiple axes sequentially, the write time can be shortened by calling this API function in each thread of "system" or "axis".
- Parameter write data size differs depending on the parameter write No.

Axis No.	Parameter write No.	Parameter write data size
0	0000h to FFFFh	2 bytes (Set the size to lower 2 bytes.)
1 to 64	2000h to 2FFFh	4 bytes
	Other than the above-mentioned	2 bytes (Set the size to lower 2 bytes.)
-16 to -1	0000h to FFFFh	2 bytes (Set the size to lower 2 bytes.)

■Supported version

Position board	API version	Board version	Header file
MR-MC3_ _	1.30	A6	mc3xxstd.h

■Reference

- sscChangeParameter (☞ Page 41 sscChangeParameter)
- sscChange2Parameter (☞ Page 43 sscChange2Parameter)
- sscCheckParameter (☞ Page 45 sscCheckParameter)
- sscCheck2Parameter (☞ Page 47 sscCheck2Parameter)
- sscChangeParameterEx (☞ Page 52 sscChangeParameterEx [MC300])
- sscCheckParameterEx (☞ Page 56 sscCheckParameterEx [MC300])
- sscCheck2ParameterEx (☞ Page 58 sscCheck2ParameterEx [MC300])

sscCheckParameterEx [MC300]

The set value of the designated parameter is read.

This API function supports the reading of the system parameters, servo parameters (MR-J5(W_)-_B/MR-J4(W_)-_B), sensing module, control parameters, RIO module parameters, and RIO control parameters.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (-16 to 64) • 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 No.
prmdata [out]	Pointer to 4-byte variable (4 bytes × 1) which stores the parameter read data

■Return value

Value	Description
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

Constant definition	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 No.1. • The axis No. and the parameter read No. do not correspond. (Example: When "System parameter" is set to the axis No. and "Axis parameter" is set to the parameter read No., etc.)
SSC_FUNC_ERR_MISMATCH_PARAM_READ_NUM1	The command and the status of the parameter read No.1 do not correspond.
SSC_FUNC_ERR_UNSUPPORT_PARAM_EX	The position board does not support the servo parameters of MR-J5(W_)-_B.

■Point

- This API function can read the parameters either before or while the system is running.
- It takes about 2 to 120 control cycles to confirm the response of the position board.
- When a number of multiple parameters are read from multiple axes sequentially, the reading time can be shortened by calling this API function in each thread of "system" or "axis".
- Parameter read data size differs depending on the parameter read No.

Axis No.	Parameter read No.	Parameter read data size
0	0000h to FFFFh	2 bytes (Store the size in lower 2 bytes.)
1 to 64	2000h to 2FFFh	4 bytes
	Other than the above-mentioned	2 bytes (Store the size in lower 2 bytes.)
-16 to -1	0000h to FFFFh	2 bytes (Store the size in lower 2 bytes.)

■Supported version

Position board	API version	Board version	Header file
MR-MC3_ _	1.30	A6	mc3xxstd.h

■Reference

- sscChangeParameter (☞ Page 41 sscChangeParameter)
- sscChange2Parameter (☞ Page 43 sscChange2Parameter)
- sscCheckParameter (☞ Page 45 sscCheckParameter)
- sscCheck2Parameter (☞ Page 47 sscCheck2Parameter)
- sscChangeParameterEx (☞ Page 52 sscChangeParameterEx [MC300])
- sscChange2ParameterEx (☞ Page 54 sscChange2ParameterEx [MC300])
- sscCheck2ParameterEx (☞ Page 58 sscCheck2ParameterEx [MC300])

sscCheck2ParameterEx [MC300]

Two set values of the designated parameters are read.

This API function supports the reading of the system parameters, servo parameters (MR-J5(W_)-_B/MR-J4(W_)-_B), sensing module, control parameters, RIO module parameters, and RIO control parameters.

```
int  sscCheck2ParameterEx (
     int  board_id,
     int  channel,
     int  axnum,
     unsigned short  *prmnum,
     unsigned short  *prmdata,
     unsigned char  *status
);
```

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (-16 to 64) • 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 Nos. (for 2 Nos.)
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 • SSC_BIT_PRFIN: Parameter read completed • SSC_BIT_PREN: Parameter No. error

■Return value

Value	Description
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

Constant definition	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 No. of the parameter read Nos. (for 2 Nos.)	A parameter read error occurred. • A value outside the range is set in the parameter read No. • The axis No. and the parameter read No. do not correspond. (Example: When "System parameter" is set to the axis No. and "Axis parameter" is set to the parameter read No., etc.)
SSC_FUNC_ERR_MISMATCH_PARAM_READ_NUM☐ ☐ = 1 to 2: Array No. of the parameter read Nos. (for 2 Nos.)	The command and the status of the parameter read No. do not correspond.
SSC_FUNC_ERR_UNSUPPORT_PARAM_EX	The position board does not support the servo parameters of MR-J5(W_)-_B.

■Point

- This API 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 120 control cycles to confirm the response of the position board.
- When a number of multiple parameters are read from multiple axes sequentially, the read time can be shortened by calling this API function in each thread of "system" or "axis".
- Parameter read data size differs depending on the parameter read No.

Axis No.	Parameter read No.	Parameter read data size
0	0000h to FFFFh	2 bytes (Store the size in lower 2 bytes.)
1 to 64	2000h to 2FFFh	4 bytes
	Other than the above-mentioned	2 bytes (Store the size in lower 2 bytes.)
-16 to -1	0000h to FFFFh	2 bytes (Store the size in lower 2 bytes.)

■Supported version

Position board	API version	Board version	Header file
MR-MC3_ _	1.30	A6	mc3xxstd.h

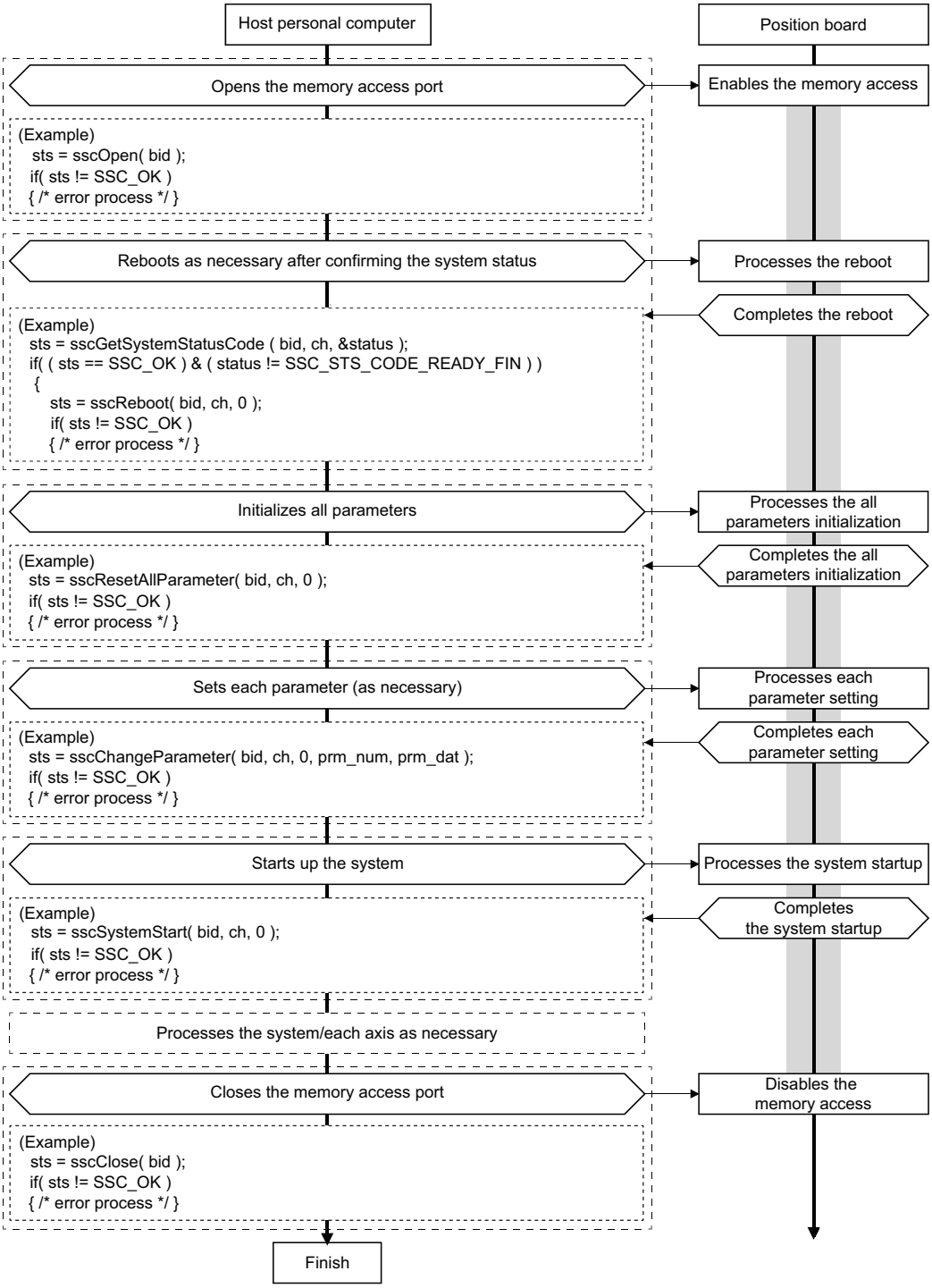
■Reference

- sscChangeParameter (☞ Page 41 sscChangeParameter)
- sscChange2Parameter (☞ Page 43 sscChange2Parameter)
- sscCheckParameter (☞ Page 45 sscCheckParameter)
- sscCheck2Parameter (☞ Page 47 sscCheck2Parameter)
- sscChangeParameterEx (☞ Page 52 sscChangeParameterEx [MC300])
- sscChange2ParameterEx (☞ Page 54 sscChange2ParameterEx [MC300])
- sscCheckParameterEx (☞ Page 56 sscCheckParameterEx [MC300])

4.5 System Functions

Processing procedure

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



sscReboot

The system is rebooted (system running → system preparation completion).

The function waits internally until the system preparation completion.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
timeout [in]	Timeout time [ms] (0 to 65535)

■Return value

Value	Description
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

Constant definition	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 is 10 seconds (10000ms) [MC200]/20 seconds (20000ms) [MC300].
- The system reboots even if the function is executed in the system preparation completion.
- Do not open the device from the position board test tool during reboot. It may not operate correctly. [MC300]

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscGetSystemStatusCode (☞ Page 63 sscGetSystemStatusCode)

sscSystemStart

The system is started after initializing the servo amplifier communication (system preparation completion → system running).

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
timeout [in]	Timeout time [ms] (0 to 65535)

■Return value

Value	Description
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

Constant definition	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 the 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 is 10 seconds (10000ms).
- Reboot when restarting the system again.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

None.

sscGetSystemStatusCode

The system status code is got.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
statuscode [out]	Pointer to 2-byte variable (2 bytes × 1) which stores the system status code ■Value • SSC_STS_CODE_READY_FIN: System preparation completion • SSC_STS_CODE_RUNNING: System running

■Return value

Value	Description
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

Constant definition	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 "System status" of "System Command/status" in the following manual.

📖MR-MC200/MR-MC300 Series Position Board User's Manual (Details)

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

None.

sscReconnectSSCNET

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

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (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

Value	Description
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

Constant definition	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 signal (RCE) is ON.

■Point

For the "Error code of reconnection/disconnection", refer to "Interface" of "Reconnect/disconnect function" in the following manual.

📖MR-MC200/MR-MC300 Series Position Board User's Manual (Details)

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscDisconnectSSCNET (☞ Page 66 sscDisconnectSSCNET)

sscReconnectSSCNETEx [MC300]

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

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
pSlaveInfo [out]	Pointer to 80-byte structure (80 bytes × 1) which stores the device information (controlling axis information and controlling station information) For the device information structure, refer to the following. ☞ Page 240 SLAVE_INFO Structure [MC300]
err_code [out]	Pointer to 2-byte variable (2 bytes × 1) which stores the error code of reconnection/disconnection

■Return value

Value	Description
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

Constant definition	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 signal (RCE) is ON.

■Point

For the "Error code of reconnection/disconnection", refer to "Interface" of "Reconnect/disconnect function" in the following manual.

📖MR-MC200/MR-MC300 Series Position Board User's Manual (Details)

■Supported version

Position board	API version	Board version	Header file
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscDisconnectSSCNETEx (☞ Page 67 sscDisconnectSSCNETEx [MC300])

sscDisconnectSSCNET

SSCNET communication which is connected to the devices in SSCNET communication (such as servo amplifiers) set by the disconnection axis No. or later is disconnected.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
com_num [in]	Disconnection axis No. (-4 to 32) • 1 to 32: Axis No. • -4 to -1: Station No. (-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

Value	Description
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

Constant definition	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 signal (CCE) is ON.

■Point

For the "Error code of reconnection/disconnection", refer to "Interface" of "Reconnect/disconnect function" in the following manual.

📖MR-MC200/MR-MC300 Series Position Board User's Manual (Details)

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscReconnectSSCNET (☞ Page 64 sscReconnectSSCNET)

sscDisconnectSSCNETEx [MC300]

SSCNET communication which is connected to the devices in SSCNET communication (such as servo amplifiers) set by the disconnection axis No. or later is disconnected.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
com_num [in]	Disconnection axis No. (-16 to 64) • 1 to 64: Axis No. • -16 to -1: Station No. (-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 device information (controlling axis information and controlling station information) For the device information structure, refer to the following. ☞ Page 240 SLAVE_INFO Structure [MC300]
err_code [out]	Pointer to 2-byte variable (2 bytes × 1) which stores the error code of reconnection/disconnection

■Return value

Value	Description
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

Constant definition	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 signal (CCE) is ON.

■Point

For the "Error code of reconnection/disconnection", refer to "Interface" of "Reconnect/disconnect function" in the following manual.

📖MR-MC200/MR-MC300 Series Position Board User's Manual (Details)

■Supported version

Position board	API version	Board version	Header file
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscReconnectSSCNETEx (☞ Page 65 sscReconnectSSCNETEx [MC300])

sscGetControllingAxis [MC300]

The controlling axis information and the controlling station information are got.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
pSlaveInfo [out]	Pointer to 80-byte structure (80 bytes × 1) which stores the device information (controlling axis information and controlling station information). For the device information structure, refer to the following. ☞ Page 240 SLAVE_INFO Structure [MC300]

■Return value

Value	Description
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

Constant definition	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 version	Board version	Header file
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

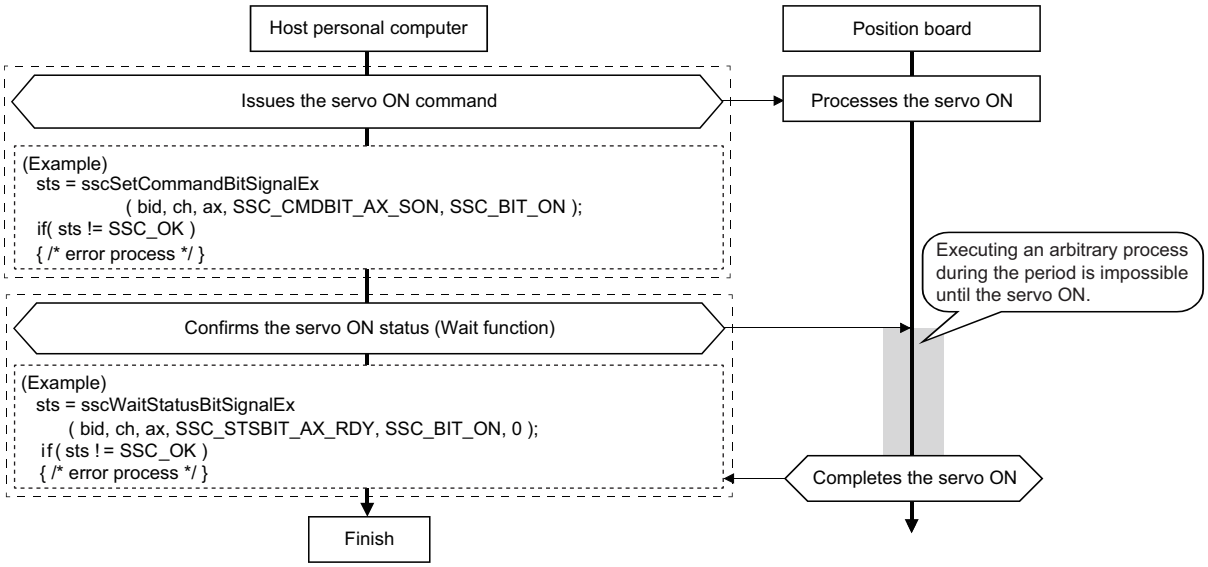
None.

4.6 Command/Status Functions

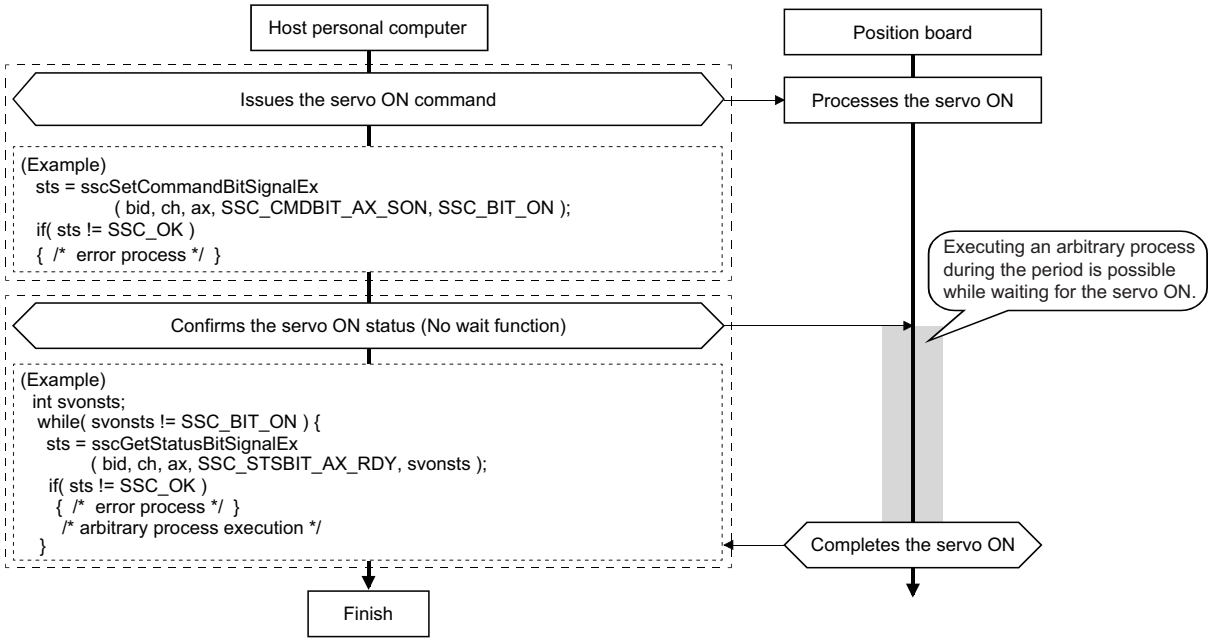
Processing procedure

An example of command/status processing procedure is shown below.

■When turning ON the servo with wait function



■When turning ON the servo with no wait function



sscSetCommandBitSignalEx

The designated command bit is turned ON or OFF.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (-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 No. For the command bit No., refer to the following. ☞ Page 241 BIT DEFINITION LIST
bitdata [in]	Command bit data ■Value • SSC_BIT_OFF: Bit OFF • SSC_BIT_ON: Bit ON

■Return value

Value	Description
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

Constant definition	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 No. and the command bit No. do not correspond. (Example: When "0" is set in the axis No. and "SSC_CMDBIT_AX_SON" is set in the command bit No., etc.)

■Point

None.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.80	A8	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

- sscGetStatusBitSignalEx (☞ Page 71 sscGetStatusBitSignalEx)
- sscWaitStatusBitSignalEx (☞ Page 72 sscWaitStatusBitSignalEx)

sscGetStatusBitSignalEx

The designated status bit is got.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (-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 No. For the status bit No., refer to the following. ☞ Page 241 BIT DEFINITION LIST
bitstatus [out]	Pointer to 4-byte variable (4 bytes × 1) which stores the status bit data ■Value • SSC_BIT_OFF: Bit OFF • SSC_BIT_ON: Bit ON

■Return value

Value	Description
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

Constant definition	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 No. and the status bit No. do not correspond. (Example: When "0" is set to the axis No. and "SSC_STSBIT_AX_RDY" is set to the status bit No., etc.)

■Point

None.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.80	A8	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

- sscSetCommandBitSignalEx (☞ Page 70 sscSetCommandBitSignalEx)
- sscWaitStatusBitSignalEx (☞ Page 72 sscWaitStatusBitSignalEx)

sscWaitStatusBitSignalEx

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

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (-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 No. For the status bit No., refer to the following. ☞ Page 241 BIT DEFINITION LIST
waitstatus [in]	Bit status to be waited ■Value • SSC_BIT_OFF: Bit OFF • SSC_BIT_ON: Bit ON
timeout [in]	Timeout time [ms] (1 to 65535, SSC_INFINITE)

■Return value

Value	Description
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

Constant definition	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 No. and the status bit No. do not correspond. (Example: When "0" is set to the axis No. and "SSC_STSBIT_AX_RDY" is set to the status bit No., etc.)

■Point

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

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.80	A8	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

- sscSetCommandBitSignalEx (☞ Page 70 sscSetCommandBitSignalEx)
- sscGetStatusBitSignalEx (☞ Page 71 sscGetStatusBitSignalEx)

4.7 Point Table Functions

sscSetPointDataEx

The point data is set.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (1 to 32 [MC200]/1 to 64 [MC300])
pntnum [in]	Point No. (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 For the point data structure, refer to the following. ☞ Page 218 PNT_DATA_EX Structure

■Return value

Value	Description
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

Constant definition	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	The designated "point No. + point No. offset value" exceeded the point table range.

■Point

A confirmation of the set point data contents is not performed.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscSetPointOffset (☞ Page 76 sscSetPointOffset)

sscCheckPointDataEx

The point data is got.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (1 to 32 [MC200]/1 to 64 [MC300])
pntnum [in]	Point No. (0 to 319 [MC200]/0 to 2047 [MC300])
pPntDataEx [out]	Pointer to 32-byte structure (32 bytes × 1) [MC200]/48-byte structure (48 bytes × 1) [MC300] which stores the point data For the point data structure, refer to the following. ☞ Page 218 PNT_DATA_EX Structure

■Return value

Value	Description
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

Constant definition	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	The designated "point No. + point No. offset value" exceeded the point table range.

■Point

None.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscCheckPointOffset (☞ Page 77 sscCheckPointOffset)

sscSetPointOffset

The point No. offset is set.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (1 to 32 [MC200]/1 to 64 [MC300])
offset [in]	Point No. offset value (0 to 319 [MC200]/0 to 2047 [MC300])

■Return value

Value	Description
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

Constant definition	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 version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscSetPointDataEx (☞ Page 74 sscSetPointDataEx)

sscCheckPointOffset

The point No. offset is got.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (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

Value	Description
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

Constant definition	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 version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscCheckPointDataEx (☞ Page 75 sscCheckPointDataEx)

sscGetDrivingPointNumber

The operation point No. is got.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (1 to 32 [MC200]/1 to 64 [MC300])
driving_pnt [out]	Pointer to 2-byte variable (2 bytes × 1) which stores the operation point No.

■Return value

Value	Description
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

Constant definition	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 "Automatic operation" or "linear interpolation [MC200]"/ "Interpolation operation [MC300]".

■Point

For the got operation point No., the value of "operation point No. + 1" is stored. When the operation is stopped, "0" is stored.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

None.

sscSetLatestPointNumber

The latest command point No. is set. Use this function when the point table loop method is used.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (1 to 32 [MC200]/1 to 64 [MC300])
latest_point [in]	Latest command point No. (1 to 320 [MC200]/1 to 2048 [MC300])

■Return value

Value	Description
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

Constant definition	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 No. to the value of "point No. + 1".

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.80	A6	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

- sscAutoStart (☞ Page 88 sscAutoStart)
- sscSetPointDataEx (☞ Page 74 sscSetPointDataEx)

4.8 Continuous Operation to Torque Control Data Functions

sscSetPressData

The continuous operation to torque control data is set.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (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 For the continuous operation to torque control data structure, refer to the following. ☞ Page 226 PRESS_DATA Structure

■Return value

Value	Description
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

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

■Point

A confirmation of the set continuous operation to torque control data contents is not performed.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.60	A5	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscGetPressData (☞ Page 81 sscGetPressData)

sscGetPressData

The continuous operation to torque control data is got.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (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 For the continuous operation to torque control data structure, refer to the following. ☞ Page 226 PRESS_DATA Structure

■Return value

Value	Description
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

Constant definition	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 version	Board version	Header file
MR-MC2_ _	1.60	A5	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscSetPressData (☞ Page 80 sscSetPressData)

4.9 Operating Functions

sscJogStart

The JOG operation is started.
After performing the necessary settings for the operation and changing to the JOG operation mode, the start operation signal (ST) is turned ON.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (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 • SSC_DIR_PLUS: + direction • SSC_DIR_MINUS: - direction

■Return value

Value	Description
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

Constant definition	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 the 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 version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

- sscJogStop (☞ Page 84 sscJogStop)
- sscJogStopNoWait (☞ Page 85 sscJogStopNoWait)
- sscDriveStop (☞ Page 95 sscDriveStop)
- sscDriveStopNoWait (☞ Page 96 sscDriveStopNoWait)
- sscDriveRapidStop (☞ Page 97 sscDriveRapidStop)
- sscDriveRapidStopNoWait (☞ Page 98 sscDriveRapidStopNoWait)

sscJogStop

The JOG operation is stopped.
The function waits internally from when the start operation signal (ST) is turned OFF until the operation processing signal (OP) is OFF (maximum 20 seconds).
If the operation processing signal (OP) is already OFF, the function is immediately terminated.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (1 to 32 [MC200]/1 to 64 [MC300])

■Return value

Value	Description
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

Constant definition	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 "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, use the sscJogStopNoWait function.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

- sscJogStart (☞ Page 82 sscJogStart)
- sscJogStopNoWait (☞ Page 85 sscJogStopNoWait)
- sscDriveStop (☞ Page 95 sscDriveStop)
- sscDriveStopNoWait (☞ Page 96 sscDriveStopNoWait)
- sscDriveRapidStop (☞ Page 97 sscDriveRapidStop)
- sscDriveRapidStopNoWait (☞ Page 98 sscDriveRapidStopNoWait)

sscJogStopNoWait

The JOG operation is stopped.

The start operation signal (ST) is turned OFF and the operation processing signal (OP) is returned.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (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 • SSC_DRIVING: During operation • SSC_DRIVE_FIN: Stop complete

■Return value

Value	Description
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

Constant definition	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 "JOG operation".

■Point

In order to make use of wait inside this function, use the sscJogStop function.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

- sscJogStart (☞ Page 82 sscJogStart)
- sscJogStop (☞ Page 84 sscJogStop)
- sscDriveStop (☞ Page 95 sscDriveStop)
- sscDriveStopNoWait (☞ Page 96 sscDriveStopNoWait)
- sscDriveRapidStop (☞ Page 97 sscDriveRapidStop)
- sscDriveRapidStopNoWait (☞ Page 98 sscDriveRapidStopNoWait)

sscIncStart

The incremental feed is started.

After performing the necessary settings for the operation and changing to the incremental feed mode, the fast start operation signal (FST) is turned ON.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (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

Value	Description
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

Constant definition	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 the 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, the movement direction is +, and if a negative value is designated, the movement direction is -.
- The response is not confirmed after the fast start operation signal (FST) is turned ON.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

- sscDriveStop (☞ Page 95 sscDriveStop)
- sscDriveStopNoWait (☞ Page 96 sscDriveStopNoWait)
- sscDriveRapidStop (☞ Page 97 sscDriveRapidStop)
- sscDriveRapidStopNoWait (☞ Page 98 sscDriveRapidStopNoWait)

sscAutoStart

The automatic operation is started.

After performing the necessary settings for the operation and changing to the automatic operation mode, the fast start operation signal (FST) is turned ON.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (1 to 32 [MC200]/1 to 64 [MC300])
point_s [in]	Start point No. (0 to 319 [MC200]/0 to 2047 [MC300])
point_e [in]	End point No. (0 to 319 [MC200]/0 to 2047 [MC300])

■Return value

Value	Description
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

Constant definition	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 the 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 version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

- sscDriveStop (☞ Page 95 sscDriveStop)
- sscDriveStopNoWait (☞ Page 96 sscDriveStopNoWait)
- sscDriveRapidStop (☞ Page 97 sscDriveRapidStop)
- sscDriveRapidStopNoWait (☞ Page 98 sscDriveRapidStopNoWait)

sscHomeReturnStart

The home position return is started.

After performing the necessary settings for the operation and changing to the home position return mode, the fast start operation signal (FST) is turned ON.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (1 to 32 [MC200]/1 to 64 [MC300])

■Return value

Value	Description
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

Constant definition	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 the 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 version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

- sscDriveStop (☞ Page 95 sscDriveStop)
- sscDriveStopNoWait (☞ Page 96 sscDriveStopNoWait)
- sscDriveRapidStop (☞ Page 97 sscDriveRapidStop)
- sscDriveRapidStopNoWait (☞ Page 98 sscDriveRapidStopNoWait)

sscLinearStart

The linear interpolation is started.
After performing the necessary settings for the operation and changing to the linear interpolation mode, the fast start operation signal (FST) is turned ON.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Primary axis No. (1 to 32 [MC200]/1 to 64 [MC300])
grpnum [in]	Interpolation group No. (0 to 8 [MC200]/0 to 16 [MC300]) • 0: Changeable interpolation group enabled • 1 to 8: Changeable interpolation group disabled [MC200] • 1 to 16: Changeable interpolation group disabled [MC300]
point_s [in]	Start point No. (0 to 319 [MC200]/0 to 2047 [MC300])
point_e [in]	End point No. (0 to 319 [MC200]/0 to 2047 [MC300])

■Return value

Value	Description
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

Constant definition	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	The designated "point No. + point No. offset value" exceeded the point table range.
SSC_FUNC_ERR_STS_BIT_IPCH_ON	The changeable interpolation group setting is enabled. Set "0" to the interpolation group No.
SSC_FUNC_ERR_STS_BIT_IPCH_OFF	The changeable interpolation group setting is disabled. Set "1 to 8 [MC200]/1 to 16 [MC300]" to the interpolation group No.
SSC_FUNC_ERR_SUB_AXIS_NUM	When the 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 set 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 the 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 version	Board version	Header file
MR-MC2_ _	1.90	A9	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

- sscDriveStop (☞ Page 95 sscDriveStop)
- sscDriveStopNoWait (☞ Page 96 sscDriveStopNoWait)
- sscDriveRapidStop (☞ Page 97 sscDriveRapidStop)
- sscDriveRapidStopNoWait (☞ Page 98 sscDriveRapidStopNoWait)

sscInterpolationStart [MC300]

The interpolation operation is started.

After performing the necessary settings for the operation and changing to the interpolation operation mode, the fast start operation signal (FST) is turned ON.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Primary axis No. (1 to 64)
grpnum [in]	Interpolation group No. (0 to 16) • 0: Changeable interpolation group enabled • 1 to 16: Changeable interpolation group disabled
point_s [in]	Start point No. (0 to 2047)
point_e [in]	End point No. (0 to 2047)

■Return value

Value	Description
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

Constant definition	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	The designated "point No. + point No. offset value" exceeded the point table range.
SSC_FUNC_ERR_STS_BIT_IPCH_ON	The changeable interpolation group setting is enabled. Set "0" to the interpolation group No.
SSC_FUNC_ERR_STS_BIT_IPCH_OFF	The changeable interpolation group setting is disabled. Set "1 to 16" to the interpolation group No.
SSC_FUNC_ERR_SUB_AXIS_NUM	When the 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 set 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 the 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 version	Board version	Header file
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

- sscDriveStop (☞ Page 95 sscDriveStop)
- sscDriveStopNoWait (☞ Page 96 sscDriveStopNoWait)
- sscDriveRapidStop (☞ Page 97 sscDriveRapidStop)
- sscDriveRapidStopNoWait (☞ Page 98 sscDriveRapidStopNoWait)

sscDataSetStart

The home position reset (data set) is started.

After performing the necessary settings for the operation and changing to the home position reset (data set) mode, the fast start operation signal (FST) is turned ON.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (1 to 32 [MC200]/1 to 64 [MC300])

■Return value

Value	Description
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

Constant definition	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 the 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 version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

- sscDriveStop (☞ Page 95 sscDriveStop)
- sscDriveStopNoWait (☞ Page 96 sscDriveStopNoWait)
- sscDriveRapidStop (☞ Page 97 sscDriveRapidStop)
- sscDriveRapidStopNoWait (☞ Page 98 sscDriveRapidStopNoWait)

sscDriveStop

The operations are stopped.
After the start operation signal (ST) is turned OFF and the stop operation signal (STP) is turned ON, the function waits internally until the operation processing signal (OP) is OFF.
After the operation processing signal (OP) is confirmed to be OFF, the stop operation signal (STP) is turned OFF.

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


Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (1 to 32 [MC200]/1 to 64 [MC300])
timeout [in]	Timeout time [ms] (0 to 65535)

■Return value

Value	Description
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

Constant definition	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 is 20 seconds (20000ms).

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

- sscDriveStopNoWait (☞ Page 96 sscDriveStopNoWait)
- sscDriveRapidStop (☞ Page 97 sscDriveRapidStop)
- sscDriveRapidStopNoWait (☞ Page 98 sscDriveRapidStopNoWait)

sscDriveStopNoWait

The operations are stopped.

After the start operation signal (ST) is turned OFF and the stop operation signal (STP) is turned ON, the stop complete status is returned.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (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 • SSC_DRIVING: During operation • SSC_DRIVE_FIN: Stop complete

■Return value

Value	Description
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

Constant definition	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, use the sscDriveStop function.
- If the stop complete status storage variable has not been confirmed to be "SSC_DRIVE_FIN", the stop operation signal (STP) remains ON.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

- sscDriveStop (☞ Page 95 sscDriveStop)
- sscDriveRapidStop (☞ Page 97 sscDriveRapidStop)
- sscDriveRapidStopNoWait (☞ Page 98 sscDriveRapidStopNoWait)

sscDriveRapidStop

The operations are stopped rapidly.
After the start operation signal (ST) is turned OFF and the rapid stop signal (RSTP) is turned ON, the function waits internally until the operation processing signal (OP) is turned OFF.
After the operation processing signal (OP) is confirmed to be OFF, the rapid stop signal (RSTP) is turned OFF.

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


Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (1 to 32 [MC200]/1 to 64 [MC300])
timeout [in]	Timeout time [ms] (0 to 65535)

■Return value

Value	Description
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

Constant definition	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 is 20 seconds (20000ms).
- In order to prevent this function from waiting internally, use the sscDriveRapidStopNoWait function.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

- sscDriveStop (☞ Page 95 sscDriveStop)
- sscDriveStopNoWait (☞ Page 96 sscDriveStopNoWait)
- sscDriveRapidStopNoWait (☞ Page 98 sscDriveRapidStopNoWait)

sscDriveRapidStopNoWait

The operations are stopped rapidly.
After the start operation signal (ST) is turned OFF and the rapid stop signal (RSTP) is turned ON, the stop complete status is returned.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (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 • SSC_DRIVING: During operation • SSC_DRIVE_FIN: Stop complete

■Return value

Value	Description
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

Constant definition	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, use the sscDriveRapidStop function.
- If the stop complete status storage variable has not been confirmed to be "SSC_DRIVE_FIN", the rapid stop signal (RSTP) remains ON.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

- sscDriveStop (☞ Page 95 sscDriveStop)
- sscDriveStopNoWait (☞ Page 96 sscDriveStopNoWait)
- sscDriveRapidStop (☞ Page 97 sscDriveRapidStop)

sscSetDriveMode

The operation mode is 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
);
```

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (1 to 32 [MC200]/1 to 64 [MC300])
drv_mode [in]	Operation mode ■Value • 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

Value	Description
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

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

■Point

- When the operation mode of an axis for which the other axes start is performed needs to be switched in advance, use this API function to change the operation mode.
- Since the functions which start operations (such as sscAutoStart) switch the operation mode inside the function, this API 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 API function is called.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

- sscGetDriveMode (☞ Page 101 sscGetDriveMode)
- sscSetOtherAxisStartData (☞ Page 133 sscSetOtherAxisStartData)

sscGetDriveMode

The operation mode is got.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (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 ■Value • 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: In Interpolation operation mode [MC300] • SSC_DRV_MODE_DST: In home position reset mode

■Return value

Value	Description
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

Constant definition	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 an axis for which the other axes start is performed needs to be switched in advance, use this API function to check the operation mode.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

- sscSetDriveMode (☞ Page 99 sscSetDriveMode)
- sscSetOtherAxisStartData (☞ Page 133 sscSetOtherAxisStartData)

sscGetDriveFinStatus

The completion of operation status is confirmed.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (1 to 32 [MC200]/1 to 64 [MC300])
fin_type [in]	Operation completion type ■Value • 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 • 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

Value	Description
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

Constant definition	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 "SSC_FIN_TYPE_CPO" in other than the automatic operation and the linear interpolation [MC200]/interpolation operation [MC300], the completion of operation confirmation type is judged by "SSC_FIN_TYPE_SMZ" unconditionally.
- When the operation completion type is set to "SSC_FIN_TYPE_INP", the operation mode is always "SSC_FIN_STS_MOV" when the in-position signal (INP) is OFF.
- For the interlock stop or interference check standby, the completion of operation status is "SSC_FIN_STS_MOV".
- When the deceleration check system is set to "SSC_SUBCMD_STOP_INP" in the automatic operation or the linear interpolation [MC200]/interpolation operation [MC300], the operation completion type is "SSC_FIN_TYPE_INP" even though "SSC_FIN_TYPE_SMZ" is set.
- The system alarm and the system error are not confirmed.
- Since the completion of operation status of this API function is judged by the operation completion type, it may differ from the operation processing signal (OP) and the operation completed signal (OPF) of the position board.
- For the JOG operation mode, the completion of operation status is not "SSC_FIN_STS_RDY".

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

None.

sscChangeControlMode

The control mode of the servo amplifier is switched.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (1 to 32 [MC200]/1 to 64 [MC300])
ctrl_mode [in]	Control mode ■Value • SSC_CTRL_MODE_POSITION: Position control mode • SSC_CTRL_MODE_PRESS: Continuous operation to torque control mode

■Return value

Value	Description
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

Constant definition	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 (CTLMCE) turned ON.

■Point

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

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.60	A5	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscSetPressData (☞ Page 80 sscSetPressData)

4.10 Change Functions

sscChangeManualPosition

The position is changed in the incremental feed.
The change position signal (PCHG) is turned ON, and the preparation for changing position completed signal (PCF) or the position change error signal (PCE) is confirmed to be ON.

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


Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (1 to 32 [MC200]/1 to 64 [MC300])
position [in]	Amount of incremental feed movement after change [command unit] (-2147483647 to 2147483647)

■Return value

Value	Description
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

Constant definition	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 "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 version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

- sscChangeManualSpeed (☞ Page 109 sscChangeManualSpeed)
- sscChangeManualAccTime (☞ Page 111 sscChangeManualAccTime)
- sscChangeManualDecTime (☞ Page 113 sscChangeManualDecTime)

sscChangeAutoPosition

The position is changed in the automatic operation.
The change position signal (PCHG) is turned ON, and the preparation for changing position completed signal (PCF) or the position change error signal (PCE) is confirmed to be ON.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (1 to 32 [MC200]/1 to 64 [MC300])
pntnum [in]	Point No. (0 to 319 [MC200]/0 to 2047 [MC300])
position [in]	Position data after change [command unit]

■Return value

Value	Description
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

Constant definition	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 "Automatic operation".
SSC_FUNC_ERR_POINT_NUMBER_OVER	The designated "point No. + point No. 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 version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

- sscGetDrivingPointNumber (☞ Page 78 sscGetDrivingPointNumber)
- sscChangeAutoSpeed (☞ Page 110 sscChangeAutoSpeed)
- sscChangeAutoAccTime (☞ Page 112 sscChangeAutoAccTime)
- sscChangeAutoDecTime (☞ Page 114 sscChangeAutoDecTime)

sscChangeLinearPosition

The position is changed in the linear interpolation.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (1 to 32 [MC200]/1 to 64 [MC300])
grpnum [in]	Interpolation group No. (0 to 8 [MC200]/0 to 16 [MC300]) • 0: Changeable interpolation group enabled • 1 to 8: Changeable interpolation group disabled [MC200] • 1 to 16: Changeable interpolation group disabled [MC300]
pntnum [in]	Point No. (0 to 319 [MC200]/0 to 2047 [MC300])
pPosition [in]	Pointer to 16-byte array (4 bytes × 4) which stores the position data [command unit] after change

■Return value

Value	Description
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

Constant definition	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 "Linear interpolation [MC200]/interpolation operation [MC300]".
SSC_FUNC_ERR_STS_BIT_IPCH_ON	The changeable interpolation group setting is enabled. Set "0" to the interpolation group No.
SSC_FUNC_ERR_STS_BIT_IPCH_OFF	The changeable interpolation group setting is disabled. Set "1 to 8 [MC200]/1 to 16 [MC300]" to the interpolation group No.
SSC_FUNC_ERR_NOT_LIP_DRIVING	The linear interpolation is not in operation.
SSC_FUNC_ERR_POINT_NUMBER_OVER	The designated "point No. + point No. 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 No. 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 version	Board version	Header file
MR-MC2_ _	1.90	A9	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

- sscGetDrivingPointNumber (☞ Page 78 sscGetDrivingPointNumber)
- sscChangeAutoSpeed (☞ Page 110 sscChangeAutoSpeed)
- sscChangeAutoAccTime (☞ Page 112 sscChangeAutoAccTime)
- sscChangeAutoDecTime (☞ Page 114 sscChangeAutoDecTime)

sscChangeManualSpeed

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

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


Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (1 to 32 [MC200]/1 to 64 [MC300])
speed [in]	Manual feed speed after change [speed unit] (0 to 2147483647)

■Return value

Value	Description
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

Constant definition	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 "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 version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

- sscChangeManualPosition (☞ Page 105 sscChangeManualPosition)
- sscChangeManualAccTime (☞ Page 111 sscChangeManualAccTime)
- sscChangeManualDecTime (☞ Page 113 sscChangeManualDecTime)

sscChangeAutoSpeed

The speed is changed in the automatic operation or the linear interpolation [MC200]/interpolation operation [MC300].
The change speed signal (SCHG) is turned ON, and the preparation for changing speed completed signal (SCF) or the speed change error signal (SCE) is confirmed to be ON.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (1 to 32 [MC200]/1 to 64 [MC300])
pntnum [in]	Point No. (0 to 319 [MC200]/0 to 2047 [MC300])
speed [in]	Feed speed after change [speed unit] (0 to 2147483647)

■Return value

Value	Description
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

Constant definition	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 "Automatic operation" or "Linear interpolation [MC200]/ Interpolation operation [MC300]".
SSC_FUNC_ERR_POINT_NUMBER_OVER	The designated "point No. + point No. 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 version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

- sscGetDrivingPointNumber (☞ Page 78 sscGetDrivingPointNumber)
- sscChangeAutoPosition (☞ Page 106 sscChangeAutoPosition)
- sscChangeAutoAccTime (☞ Page 112 sscChangeAutoAccTime)
- sscChangeAutoDecTime (☞ Page 114 sscChangeAutoDecTime)

sscChangeManualAccTime

The acceleration time constant is changed in the JOG operation or the incremental feed.
The change acceleration time constant signal (TACHG) is turned ON, and the preparation for changing acceleration time constant completed signal (TACF) or the acceleration time constant change error signal (TACE) is confirmed to be ON.

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


Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (1 to 32 [MC200]/1 to 64 [MC300])
acctime [in]	Manual feed speed acceleration time constant after change [ms] (0 to 20000)

■Return value

Value	Description
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

Constant definition	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 "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 version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

- sscChangeManualPosition (☞ Page 105 sscChangeManualPosition)
- sscChangeManualSpeed (☞ Page 109 sscChangeManualSpeed)
- sscChangeManualDecTime (☞ Page 113 sscChangeManualDecTime)

sscChangeAutoAccTime

The acceleration time constant is changed in the automatic operation or the linear interpolation [MC200]/interpolation operation [MC300].
The change acceleration time constant signal (TACHG) is turned ON, and the preparation for changing acceleration time constant completed signal (TACF) or the acceleration time constant change error signal (TACE) is confirmed to be ON.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (1 to 32 [MC200]/1 to 64 [MC300])
pntnum [in]	Point No. (0 to 319 [MC200]/0 to 2047 [MC300])
acctime [in]	Acceleration time constant after change [ms] (0 to 20000)

■Return value

Value	Description
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

Constant definition	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 "Automatic operation" or "Linear interpolation [MC200]/ Interpolation operation [MC300]".
SSC_FUNC_ERR_POINT_NUMBER_OVER	The designated "point No. + point No. 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 version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

- sscGetDrivingPointNumber (☞ Page 78 sscGetDrivingPointNumber)
- sscChangeAutoPosition (☞ Page 106 sscChangeAutoPosition)
- sscChangeAutoSpeed (☞ Page 110 sscChangeAutoSpeed)
- sscChangeAutoDecTime (☞ Page 114 sscChangeAutoDecTime)

sscChangeManualDecTime

The deceleration time constant is changed in the JOG operation or the incremental feed.
The change deceleration time constant signal (TDCHG) is turned ON, and the preparation for changing deceleration time constant completed signal (TDCF) or the deceleration time constant change error signal (TDCE) is confirmed to be ON.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (1 to 32 [MC200]/1 to 64 [MC300])
dectime [in]	Manual feed speed deceleration time constant after change [ms] (0 to 20000)

■Return value

Value	Description
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

Constant definition	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 "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 version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

- sscChangeManualPosition (☞ Page 105 sscChangeManualPosition)
- sscChangeManualSpeed (☞ Page 109 sscChangeManualSpeed)
- sscChangeManualAccTime (☞ Page 111 sscChangeManualAccTime)

sscChangeAutoDecTime

The deceleration time constant is changed in the automatic operation or the linear interpolation [MC200]/interpolation operation [MC300].
The change deceleration time constant signal (TDCHG) is turned ON, and the preparation for changing deceleration time constant completed signal (TDCF) or the deceleration time constant change error signal (TDCE) is confirmed to be ON.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (1 to 32 [MC200]/1 to 64 [MC300])
pntnum [in]	Point No. (0 to 319 [MC200]/0 to 2047 [MC300])
dectime [in]	Deceleration time constant after change [ms] (0 to 20000)

■Return value

Value	Description
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

Constant definition	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 "Automatic operation" or "Linear interpolation [MC200]/ Interpolation operation [MC300]".
SSC_FUNC_ERR_POINT_NUMBER_OVER	The designated "point No. + point No. 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 version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

- sscGetDrivingPointNumber (☞ Page 78 sscGetDrivingPointNumber)
- sscChangeAutoPosition (☞ Page 106 sscChangeAutoPosition)
- sscChangeAutoSpeed (☞ Page 110 sscChangeAutoSpeed)
- sscChangeAutoAccTime (☞ Page 112 sscChangeAutoAccTime)

4.11 Alarm Functions

sscGetAlarm

The alarm No. and the specific alarm No. are got.

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


Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (-4 to 32 [MC200]/-16 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 • 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 No. variable
detail_code [out]	Pointer to 2-byte variable (2 bytes × 1) which stores the specific alarm No. variable

■Return value

Value	Description
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

Constant definition	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 No. and the alarm type do not correspond. (Example: When "0" is set to the axis No. and "SSC_ALARM_OPERATION" is set to the alarm type, etc.)

■Point

If an alarm has not occurred, "0" is got as the alarm No. and the specific alarm No.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.80	A8	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscResetAlarm (☞ Page 116 sscResetAlarm)

sscResetAlarm

The alarm is reset.

After the alarm reset signal is turned ON and the alarm signal is confirmed to be OFF, the alarm reset signal is turned OFF.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (-4 to 32 [MC200]/-16 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 • 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

Value	Description
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

Constant definition	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 No. and the alarm type do not correspond. (Example: When "0" is set to the axis No. and "SSC_ALARM_OPERATION" is set to the alarm type, etc.)

■Point

None.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.80	A8	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

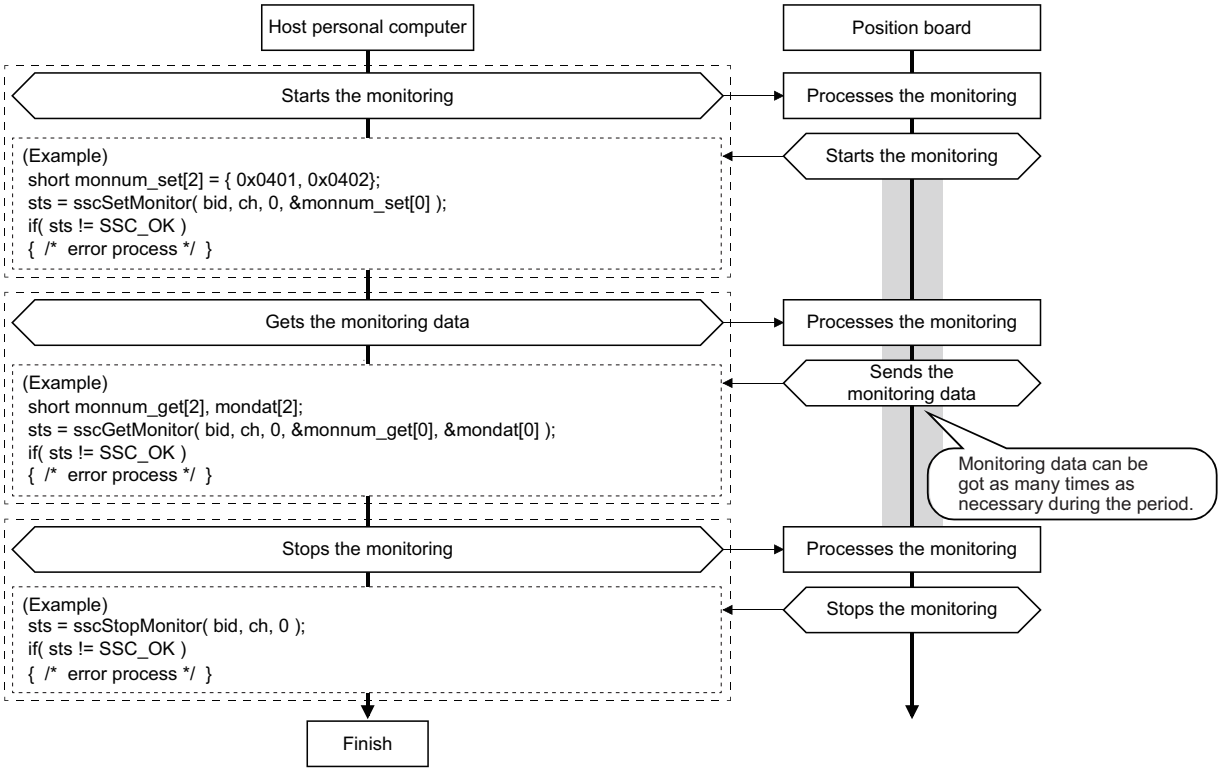
sscGetAlarm (☞ Page 115 sscGetAlarm)

4.12 General Monitor Functions

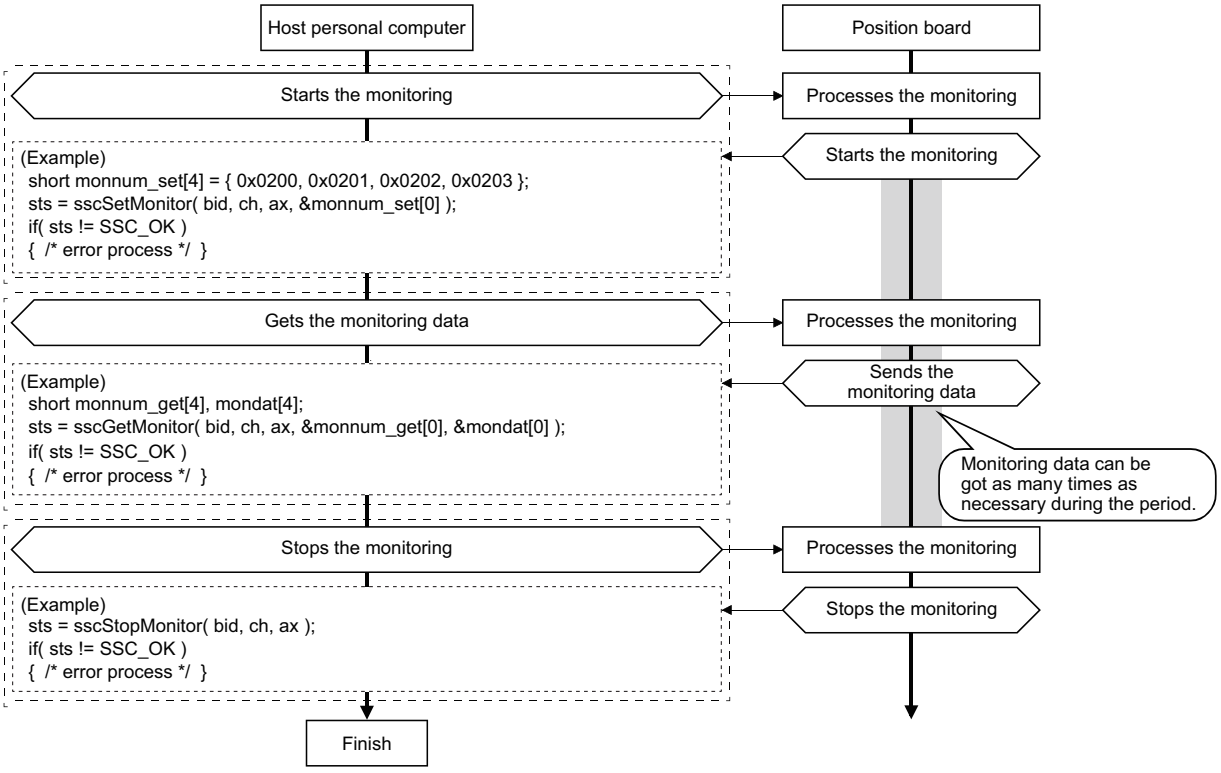
Processing procedure

An example of general monitor processing procedure is shown below.

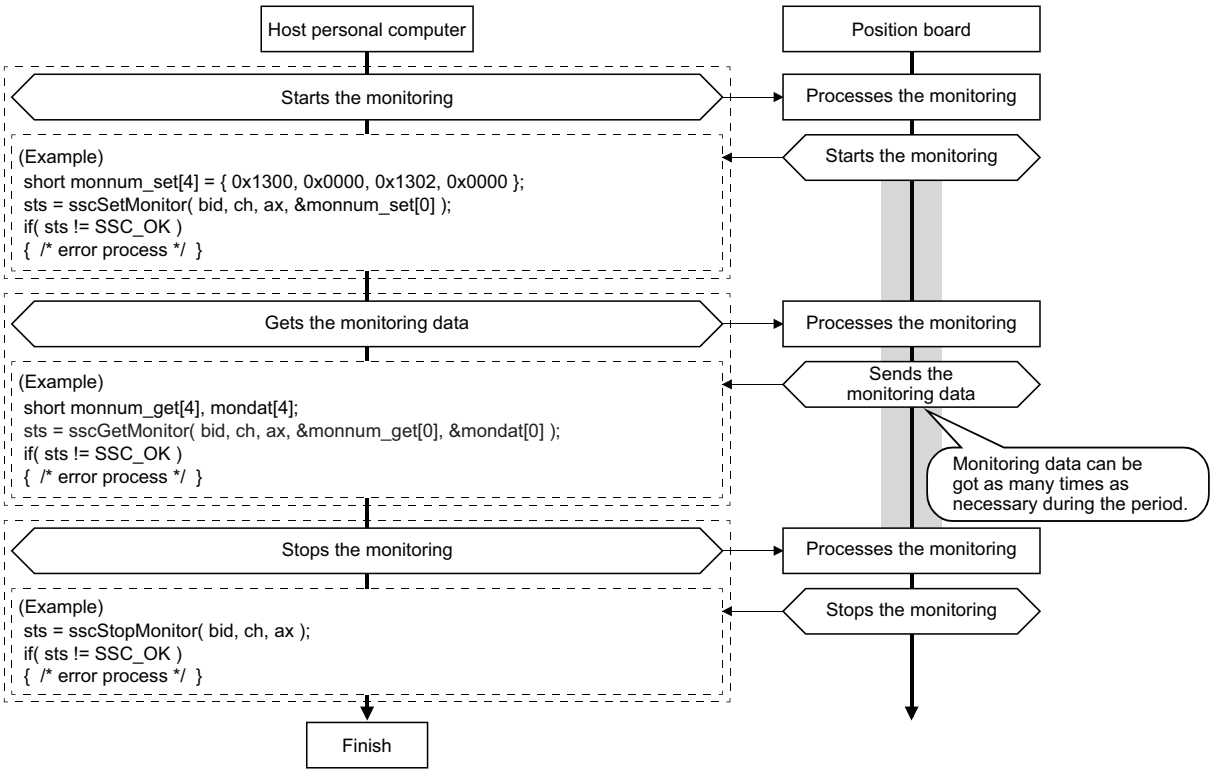
■When getting system information



■When getting servo information/operation information



■When getting operation information (2-word)



sscSetMonitor

The monitoring is started.
After setting 2 monitor Nos. during the system monitoring and 4 monitor Nos. during the axis monitoring, the monitor command signal (MON) is turned ON, and the monitor output signal (MOUT) is confirmed to be ON.
If the monitor output signal (MOUT) is already ON, the monitoring restarts after the monitor command signal (MON) is turned OFF.

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


Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (-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 Nos. (for 4 Nos.)

■Return value

Value	Description
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

Constant definition	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 No. of the monitor Nos. (for 4 Nos.)	A monitor No. error occurred. • A value outside the range is set in the monitor No. • The axis No. and the monitor No. do not correspond. (Example: When "System monitor" is set to the axis No. and "Axis monitor" is set to the monitor No., etc.)
SSC_FUNC_ERR_STS_BIT_MESV	The servo information was set as a monitor No. 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 Nos. 1 and 3 should be used, and monitor Nos. 2 and 4 should be set to "0".
- For the system monitor, since the monitor No.3 and 4 are not used, set "0".
- For the monitor No. which is not used, set "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 version	Board version	Header file
MR-MC2_ _	1.80	A8	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

- sscStopMonitor (☞ Page 119 sscSetMonitor)
- sscGetMonitor (☞ Page 122 sscGetMonitor)

sscStopMonitor

The monitoring is stopped.

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

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (-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

Value	Description
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

Constant definition	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 version	Board version	Header file
MR-MC2_ _	1.80	A8	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscSetMonitor (☞ Page 119 sscSetMonitor)

sscGetMonitor

The monitoring data is got.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (-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 Nos.
mondata [out]	Pointer to 8-byte array (2 bytes × 4) which stores the monitor data.

■Return value

Value	Description
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

Constant definition	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 value is not stored in 3 and 4 of both the monitor No. and the monitor data.
- To get a current command position, a current feedback position, an external signal status, a moving speed, a feedback moving speed, or an electrical current feedback, use high speed monitor functions to get the data quickly.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.80	A8	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscSetMonitor (☞ Page 119 sscSetMonitor)

4.13 High Speed Monitor Functions

sscGetCurrentCmdPositionFast

The current command position is got.

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


Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (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 unit]

■Return value

Value	Description
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

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

■Point

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

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscGetCurrentFbPositionFast (☞ Page 124 sscGetCurrentFbPositionFast)

sscGetCurrentFbPositionFast

The current feedback position is got.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (1 to 32 [MC200]/1 to 64 [MC300])
position [out]	Pointer to 4-byte variable (4 bytes × 1) which stores the current feedback position variable [command unit]

■Return value

Value	Description
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

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

■Point

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

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscGetCurrentCmdPositionFast (☞ Page 124 sscGetCurrentFbPositionFast)

sscGetIoStatusFast

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

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (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 • 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

Value	Description
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

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

■Point

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

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

None.

sscGetCmdSpeedFast

The moving speed is got.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (1 to 32 [MC200]/1 to 64 [MC300])
speed [out]	Pointer to 4-byte variable (4 bytes × 1) which stores the moving speed [speed unit]

■Return value

Value	Description
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

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

■Point

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

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscGetFbSpeedFast (☞ Page 127 sscGetFbSpeedFast)

sscGetFbSpeedFast

The feedback moving speed is got.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (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 unit]

■Return value

Value	Description
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

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

■Point

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

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscGetCmdSpeedFast (☞ Page 126 sscGetCmdSpeedFast)

sscGetCurrentFbFast

The electrical current feedback is got.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (1 to 32 [MC200]/1 to 64 [MC300])
currentFb [out]	Pointer to 2-byte variable (2 bytes × 1) which stores the electrical current feedback [0.1%]

■Return value

Value	Description
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

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

■Point

"High speed monitor function" of the position board is used to get the electrical current feedback.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

None.

sscGetPositionDroopFast

The position droop is got.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (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

Value	Description
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

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

■Point

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

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.60	A4	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

None.

4.14 User Watchdog Functions

sscWdEnable

The user watchdog function is enabled.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
wdcnt [in]	Watchdog timer start counter

■Return value

Value	Description
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

Constant definition	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 version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscWdDisable (☞ Page 131 sscWdDisable)

sscWdDisable

The user watchdog function is disabled.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)

■Return value

Value	Description
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

Constant definition	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 version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscWdEnable (☞ Page 130 sscWdEnable)

sscChangeWdCounter

The watchdog check counter is updated (+1).

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)

■Return value

Value	Description
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

Constant definition	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 version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscWdEnable (☞ Page 130 sscWdEnable)

4.15 Other Axes Start Functions

sscSetOtherAxisStartData

The other axes start data is set.

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


Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
oas_num [in]	Other axes start table No. (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 For the other axes start data structure, refer to the following. ☞ Page 222 OAS_DATA Structure

■Return value

Value	Description
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

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

■Point

A confirmation of the set other axes start data contents is not performed.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscGetOtherAxisStartData (☞ Page 134 sscGetOtherAxisStartData)

sscGetOtherAxisStartData

The other axes start table data is got.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
oas_num [in]	Other axes start table No. (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 For the other axes start data structure, refer to the following. ☞ Page 222 OAS_DATA Structure

■Return value

Value	Description
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

Constant definition	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 version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscSetOtherAxisStartData (☞ Page 134 sscGetOtherAxisStartData)

sscOtherAxisStartAbortOn

The other axes start cancel signal (OSSTP☐) is turned ON and the other axes start is canceled.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
oas_num [in]	Other axes start table No. (1 to 32 [MC200]/1 to 64 [MC300])

■Return value

Value	Description
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

Constant definition	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 version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

- sscOtherAxisStartAbortOff (☞ Page 136 sscOtherAxisStartAbortOff)
- sscGetOtherAxisStartStatus (☞ Page 137 sscGetOtherAxisStartStatus)

sscOtherAxisStartAbortOff

The other axes start cancel signal (OSSTP□) is turned OFF to cancel the abortion of the other axes start.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
oas_num [in]	Other axes start table No. (1 to 32 [MC200]/1 to 64 [MC300])

■Return value

Value	Description
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

Constant definition	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 version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

- sscOtherAxisStartAbortOn (☞ Page 135 sscOtherAxisStartAbortOn)
- sscGetOtherAxisStartStatus (☞ Page 137 sscGetOtherAxisStartStatus)

sscGetOtherAxisStartStatus

The other axes start status (other axes start status bit) is got.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
oas_num [in]	Other axes start table No. (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 • SSC_BIT_OSOP: Other axes start notice • SSC_BIT_OSFIN: Other axes start completed • SSC_BIT_OSERR: Other axes start incompleted

■Return value

Value	Description
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

Constant definition	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 version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

None.

4.16 Pass Position Interrupt Functions

sscSetIntPassPositionData

The pass position interrupt condition data is set.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
pass_num [in]	Pass position condition No. (1 to 64 [MC200]/1 to 128 [MC300])
pass_option [in]	Pass position option Set the data in the logical sum of each value. ■Value [Pass direction] • SSC_PASS_DIR_PLUS: + direction pass position interrupt output • SSC_PASS_DIR_MINUS: - direction pass position interrupt output [Judgment condition] • SSC_PASS_JUDGE_CMD_POS: Current position • SSC_PASS_JUDGE_FB_POS: Feedback position [Cancel condition] [MC300] • SSC_PASS_CANCEL_INP: In-position signal (INP) is ON. • SSC_PASS_CANCEL_NONE: No cancel
pass_data [in]	Pass position data [command unit] (-2147483648 to 2147483647)

■Return value

Value	Description
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

Constant definition	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 No. is valid for the pass position option.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.02	A1	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscCheckIntPassPositionData (☞ Page 139 sscCheckIntPassPositionData)

sscCheckIntPassPositionData

The pass position interrupt condition data is got.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
pass_num [in]	Pass position condition No. (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 unit]

■Return value

Value	Description
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

Constant definition	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 version	Board version	Header file
MR-MC2_ _	1.02	A1	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscSetIntPassPositionData (☞ Page 138 sscSetIntPassPositionData)

sscSetStartingPassNumber

The pass position condition start and end Nos. are set.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (1 to 32 [MC200]/1 to 64 [MC300])
pass_start [in]	Pass position condition start No. (1 to 64 [MC200]/1 to 128 [MC300])
pass_end [in]	Pass position condition end No. (1 to 64 [MC200]/1 to 128 [MC300])

■Return value

Value	Description
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

Constant definition	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 version	Board version	Header file
MR-MC2_ _	1.02	A1	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

None.

sscGetExecutingPassNumber

The running pass position condition No. is got.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (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 No.

■Return value

Value	Description
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

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

■Point

- The end No. of the pass position condition is stored in the running pass position condition No. after the pass position condition completion.
- The canceled pass position condition No. is stored in the running pass position condition No. after the pass position condition incompletion.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.02	A1	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

None.

4.17 Sampling Functions

sscStartSampling

The sampling is started.

The sampling start signal (SMPS) is turned ON.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)

■Return value

Value	Description
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

Constant definition	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 version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

- sscStopSampling (☞ Page 143 sscStopSampling)
- sscSetSamplingParameter (☞ Page 144 sscSetSamplingParameter)
- sscGetSamplingStatus (☞ Page 147 sscGetSamplingStatus)

sscStopSampling

The sampling is stopped.

The sampling start signal (SMPS) is turned OFF, and the function waits until all sampling status signals (waiting for sampling trigger (SMPW), sampling is being performed (SMPO), sampling completed (SMPF), and sampling error (SMPE)) are turned OFF.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)

■Return value

Value	Description
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

Constant definition	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 API function after the sampling completed signal (SMPF) or the sampling error signal (SMPE) turns ON.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscStartSampling (☞ Page 142 sscStartSampling)

sscSetSamplingParameter

The sampling parameters are written.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
prm_num [in]	Sampling setting write No.
prm_data [in]	Sampling setting write data

■Return value

Value	Description
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

Constant definition	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 No.
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 No. 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 version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscGetSamplingParameter (☞ Page 145 sscGetSamplingParameter)

sscGetSamplingParameter

The sampling parameters are read.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
prm_num [in]	Sampling setting read No.
prm_data [out]	Pointer to 4-byte variable (4 bytes × 1) which stores the sampling setting read data

■Return value

Value	Description
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

Constant definition	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 No.
SSC_FUNC_ERR_MISMATCH_SMP_PARAM_READ_NUM	The command and the status of the sampling setting read No. do not correspond.

■Point

None.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscSetSamplingParameter (☞ Page 144 sscSetSamplingParameter)

sscGetSamplingError

The sampling error information is got.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (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 For the sampling error information structure, refer to the following. ☞ Page 227 SMP_ERR Structure

■Return value

Value	Description
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

Constant definition	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 version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscGetSamplingStatus (☞ Page 147 sscGetSamplingStatus)

sscGetSamplingStatus

The sampling execution status (sampling status bit and sampling complete page No.) is got.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
status [out]	Pointer to 1-byte variable (1 byte × 1) which stores the sampling status bit ■Value • SSC_BIT_SMPW: Waiting for sampling trigger • SSC_BIT_SMPO: Sampling is being performed • SSC_BIT_SMPF: Sampling completed • SSC_BIT_SMPE: Sampling error
fin_page [out]	Pointer to 2-byte variable (2 bytes × 1) which stores the sampling completion page No.

■Return value

Value	Description
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

Constant definition	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 version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

- sscGetSamplingError (☞ Page 146 sscGetSamplingError)
- sscGetSamplingData (☞ Page 148 sscGetSamplingData)

sscGetSamplingData

The sampling read enabled points and the sampling data (for 32 points [MC200]/128 points [MC300]) are 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
);
```

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
page_num [in]	Sampling read page No. (0 to 256 [MC200]/0 to 512 [MC300])
valid_num [out]	Pointer to the valid points variable of the sampling read
pSmpData [out]	Pointer to 4224-byte structure (132 bytes × 32) [MC200]/16896-byte structure (132 bytes × 128) [MC300] which stores the sampling data For the sampling data information structure, refer to the following. ☞ Page 229 SMP_DATA Structure

■Return value

Value	Description
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

Constant definition	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 No. to "0" using this API function before starting the sampling.
- For MR-MC3_ _ API version 1.00, the sampling data from 32769 points (page No. 257) and after cannot be gotten.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

- sscStartSampling (☞ Page 142 sscStartSampling)
- sscGetSamplingStatus (☞ Page 147 sscGetSamplingStatus)

4.18 Log Functions

sscStartLog

The logging is started.

The logging is started by turning the log command signal (LOGC) ON, and the log operation being performed signal (LOGO) is confirmed to be ON.

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


Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)

■Return value

Value	Description
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

Constant definition	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 signal (LOGC) is ON. Stop the logging with the sscStopLog function.

■Point

None.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscStopLog (☞ Page 150 sscStopLog)

sscStopLog

The logging is stopped.

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

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)

■Return value

Value	Description
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

Constant definition	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 version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscStartLog (☞ Page 149 sscStartLog)

sscCheckLogStatus

The log operation status is got.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
status [out]	Pointer to 4-byte variable (4 bytes × 1) which stores the log operation status ■Value • 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

Value	Description
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

Constant definition	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 version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscStartLog (☞ Page 149 sscStartLog)

sscCheckLogEventNum

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

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
eventnum [out]	Pointer to 4-byte variable (4 bytes × 1) which stores the number of valid log data events

■Return value

Value	Description
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

Constant definition	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 version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscStartLog (☞ Page 149 sscStartLog)

sscReadLogData

The 16 events of log data are read.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
page_num [in]	Log data read page No. (1 to 256)
pLogData [out]	Pointer to 256-byte structure (16 bytes × 16) which stores the log data For the log data information structure, refer to the following. ☞ Page 231 LOG_DATA Structure

■Return value

Value	Description
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

Constant definition	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. Stop the logging with the sscStopLog function.
SSC_FUNC_ERR_STS_BIT_LOGRE	The reading of log data error signal (LOGRE) is turned ON.

■Point

None.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscClearLogData (☞ Page 154 sscClearLogData)

sscClearLogData

The log data is cleared (initialized).

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)

■Return value

Value	Description
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

Constant definition	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0☐ ☐ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_TIMEOUT_0☐	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. Stop the logging with the sscStopLog function.
SSC_FUNC_ERR_STS_BIT_LOGIE	The log data initialization error signal (LOGIE) is turned ON.

■Point

None.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscReadLogData (☞ Page 153 sscReadLogData)

sscGetAlarmHistoryData

The 4 events of alarm history data are read.

```
int  sscGetAlarmHistoryData (
     int  board_id,
     int  channel,
     int  page_num,
     ALH_DATA  *pAlhData
);
```

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
page_num [in]	Alarm history data read page No. (1 to 512)
pAlhData [out]	Pointer to 256-byte structure (64 bytes × 4) which stores the alarm history data For the alarm history data information structure, refer to the following. ☞ Page 232 ALH_DATA Structure

■Return value

Value	Description
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

Constant definition	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_UNSUPPORT_ALH	Alarm history function is not supported.
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 version	Board version	Header file
MR-MC2_ _	1.40	A3	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscClearAlarmHistoryData (☞ Page 157 sscClearAlarmHistoryData)

sscCheckAlarmHistoryEventNum

The number of valid alarm history data events recorded in the alarm history data is got.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
eventnum [out]	Pointer to 4-byte variable (4 bytes × 1) which stores the number of valid alarm history data events

■Return value

Value	Description
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

Constant definition	Cause/countermeasure
SSC_FUNC_ERR_ARGUMENT_0□ □ = 1 to 9: Argument location	The argument is outside the set range.
SSC_FUNC_ERR_UNSUPPORT_ALH	Alarm history function is not supported.

■Point

None.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.40	A3	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

None.

sscClearAlarmHistoryData

Clears (initializes) alarm history data.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)

■Return value

Value	Description
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

Constant definition	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 (2 seconds [MC200]/10 seconds [MC300]) has elapsed.
SSC_FUNC_ERR_UNSUPPORT_ALH	Alarm history function is not supported.
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 version	Board version	Header file
MR-MC2_ _	1.40	A3	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscGetAlarmHistoryData (☞ Page 155 sscGetAlarmHistoryData)

4.19 Digital Input/Output Functions

sscGetDigitalInputDataBit

The DI data of the designated digital input is got on a 1-point basis.

```
int  sscGetDigitalInputDataBit (
     int  board_id,
     int  channel,
     int  din_num,
     int  *din
);
```

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
din_num [in]	Digital input No. (0 to 1023)
din [out]	Pointer to 4-byte variable (4 bytes × 1) which stores the DI data status of the digital input ■Value • SSC_BIT_OFF: Input signal OFF • SSC_BIT_ON: Input signal ON

■Return value

Value	Description
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

Constant definition	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 version	Board version	Header file
MR-MC2_ _	1.02	A1	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscGetDigitalInputDataWord (☞ Page 159 sscGetDigitalInputDataWord)

sscGetDigitalInputDataWord

The DI data of the designated digital input is got on a 16-point basis.

```
int  sscGetDigitalInputDataWord (
     int  board_id,
     int  channel,
     int  din_word_num,
     unsigned short  *din
);
```

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
din_word_num [in]	Digital input word No. (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_8□□□

■Return value

Value	Description
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

Constant definition	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 version	Board version	Header file
MR-MC2_ _	1.02	A1	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscGetDigitalInputDataBit (☞ Page 158 sscGetDigitalInputDataBit)

sscSetDigitalOutputDataBit

The DO data of the designated digital output is set on a 1-point basis.

```
int  sscSetDigitalOutputDataBit (
     int  board_id,
     int  channel,
     int  dout_num,
     int  dout
);
```

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
dout_num [in]	Digital output No. (0 to 1023)
dout [in] (0 to 1)	DO data of the digital output ■Value • SSC_BIT_OFF: Output signal OFF • SSC_BIT_ON: Output signal ON

■Return value

Value	Description
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

Constant definition	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 position board, the timeout time (1 second) has elapsed.

■Point

This API function sets the digital output with the exclusive control function of the position board.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.02	A1	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

- sscSetDigitalOutputDataWord (☞ Page 161 sscSetDigitalOutputDataWord)
- sscGetDigitalOutputDataBit (☞ Page 162 sscGetDigitalOutputDataBit)

sscSetDigitalOutputDataWord

The DO data of the designated digital output is got on a 16-point basis.

```
int  sscSetDigitalOutputDataWord (
     int  board_id,
     int  channel,
     int  dout_word_num,
     unsigned short  dout
);
```

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
dout_word_num [in]	Digital output word No. (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_8□□□

■Return value

Value	Description
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

Constant definition	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 position board, the timeout time (1 second) has elapsed.

■Point

This API function sets the digital output with the exclusive control function of the position board.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.02	A1	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

- sscSetDigitalOutputDataBit (☞ Page 162 sscGetDigitalOutputDataBit)
- sscGetDigitalOutputDataWord (☞ Page 163 sscGetDigitalOutputDataWord)

sscGetDigitalOutputDataBit

The DO data of the designated digital output is got on a 1-point basis.

```
int  sscGetDigitalOutputDataBit (
     int  board_id,
     int  channel,
     int  dout_num,
     int  *dout
);
```

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
dout_num [in]	Digital output No. (0 to 1023)
dout [out]	Pointer to 4-byte variable (4 bytes × 1) which stores the DO data status of the digital output ■Value • SSC_BIT_OFF: Output signal OFF • SSC_BIT_ON: Output signal ON

■Return value

Value	Description
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

Constant definition	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 version	Board version	Header file
MR-MC2_ _	1.02	A1	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

- sscSetDigitalOutputDataBit (☞ Page 160 sscSetDigitalOutputDataBit)
- sscGetDigitalOutputDataWord (☞ Page 163 sscGetDigitalOutputDataWord)

sscGetDigitalOutputDataWord

The DO data of the designated digital output is got on a 16-point basis.

```
int  sscGetDigitalOutputDataWord (
     int  board_id,
     int  channel,
     int  dout_word_num,
     unsigned short  *dout
);
```

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
dout_word_num [in]	Digital output word No. (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_8□□□

■Return value

Value	Description
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

Constant definition	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 version	Board version	Header file
MR-MC2_ _	1.02	A1	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

- sscSetDigitalOutputDataWord (☞ Page 161 sscSetDigitalOutputDataWord)
- sscGetDigitalOutputDataBit (☞ Page 162 sscGetDigitalOutputDataBit)

4.20 Mark Detection Functions

sscGetMarkDetectionData

The mark detection data is got.

```
int  sscGetMarkDetectionData (
       int  board_id,
       int  channel,
       int  axnum,
       int  mark_num,
       int  *read_fin_num,
       int  *edge,
       int  *position
);
```

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (1 to 32 [MC200]/1 to 64 [MC300])
mark_num [in]	Mark detection setting No. (1 to 2)
read_fin_num [out]	Pointer to 4-byte variable (4 bytes × 1) which stores the read complete buffer No. (1 to 255)
edge [out]	Pointer to 1-byte structure (1 byte × 1) which stores the mark detection edge data ■Value • SSC_MARK_EDGE_NONE: Not detected • 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

Value	Description
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

Constant definition	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 No. is renewed (+1). (For continuous detection mode, the read complete buffer No. is set to "1" when the read complete buffer No. exceeds "255". For ring buffer mode, the read complete buffer No. is set to "1" when the No. 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 No., mark detection edge data, mark detection positioning data).

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.60	A5	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscGetMarkDetectionCounter (☞ Page 166 sscGetMarkDetectionCounter)

sscGetMarkDetectionCounter

The mark detection counter is got.

```
int  sscGetMarkDetectionCounter (
     int  board_id,
     int  channel,
     int  axnum,
     int  mark_num,
     int  *detected_counter
);
```

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (1 to 32 [MC200]/1 to 64 [MC300])
mark_num [in]	Mark detection setting No. (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

Value	Description
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

Constant definition	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 version	Board version	Header file
MR-MC2_ _	1.60	A5	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscGetMarkDetectionData (☞ Page 164 sscGetMarkDetectionData)

sscClearMarkDetectionData

The mark detection data is cleared (initialized).

```
int  sscClearMarkDetectionData (
     int  board_id,
     int  channel,
     int  axnum,
     int  mark_num
);
```

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (1 to 32 [MC200]/1 to 64 [MC300])
mark_num [in]	Mark detection setting No. (1 to 2)

■Return value

Value	Description
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

Constant definition	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.

■Point

The read complete buffer No. is set to "0" after mark detection data is cleared.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.60	A5	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscGetMarkDetectionData (☞ Page 164 sscGetMarkDetectionData)

4.21 Interface Mode Functions

sscIfmGetReadErrorCount

The read error counter is got.

```
int  sscIfmGetReadErrorCount (
      int  board_id,
      int  channel,
      short  *err_cnt
);
```

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
err_cnt [out]	Pointer to 2-byte variable (2 bytes × 1) which stores the read error counter

■Return value

Value	Description
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

Constant definition	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 version	Board version	Header file
MR-MC2_ _	1.40	A3	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

None.

sscIfmSetHomePosition

Home position is set, and after completion, the home position data ("home position multiple revolution data (parameter No.024D)" and "home position within 1 revolution position (parameter No.024E, parameter No.024F)") is got.

```
int  sscIfmSetHomePosition (
     int  board_id,
     int  channel,
     int  axnum,
     int  mode,
     short  *param
);
```

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (1 to 32 [MC200]/1 to 64 [MC300])
mode [in]	In-position signal (INP) check mode ■Value • SSC_IFM_CHK_INP_WAIT: Waits until the in-position signal (INP) is ON. • SSC_IFM_CHK_INP_NOWAIT: Does not wait until the in-position signal (INP) is ON.
param [out]]	Pointer to 6-byte array (2 bytes × 3) which stores the home position data ■Array No. • 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

Value	Description
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

Constant definition	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 (INP) is OFF. When setting "Does not wait until the in-position signal (INP) is ON" during in-position signal (INP) check mode, call the sscIfmSetHomePosition function when the in-position signal (INP) is ON.
SSC_FUNC_ERR_STS_BIT_ZSE	The home position set error signal (ZSE) is ON.

■Point

None.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.40	A3	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

None.

sscIfmGetMaximumBufferNumber

The maximum buffer No. for position control mode is got.

```
int  sscIfmGetMaximumBufferNumber (
     int  board_id,
     int  channel,
     int  axnum,
     short  *bufnum
);
```

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (1 to 32)
bufnum [out]	Pointer to 2-byte variable (2 bytes × 1) which stores the maximum buffer No.

■Return value

Value	Description
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

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

■Point

- For API version 1.60 or later, use the sscIfmGetMaximumBufferNumberEx function. [MC200]
- Use the sscIfmGetMaximumBufferNumberEx function. [MC300]

☞ Page 171 sscIfmGetMaximumBufferNumberEx

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.40	A3	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

None.

sscIfmGetMaximumBufferNumberEx

The maximum buffer No. for the designated control mode is got.

```
int  sscIfmGetMaximumBufferNumberEx (
     int  board_id,
     int  channel,
     int  axnum,
     unsigned short  ctrl_mode,
     short  *bufnum
);
```

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (1 to 32 [MC200]/1 to 64 [MC300])
ctrl_mode [in]	Control mode ■Operation mode • 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 No.

■Return value

Value	Description
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

Constant definition	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 version	Board version	Header file
MR-MC2_ _	1.60	A4	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

None.

sscIfmRenewLatestBuffer

The latest command buffer No. and data for position control mode is renewed, and the renewed latest command buffer No. is got.

```
int  sscIfmRenewLatestBuffer (
     int  board_id,
     int  channel,
     int  axnum,
     long  bufdata,
     short  *bufnum
);
```

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (1 to 32)
bufdata [in]	Renew data
bufnum [out]	Pointer to 2-byte variable (2 bytes × 1) which stores the latest command buffer No.

■Return value

Value	Description
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

Constant definition	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 No. has been renewed, call the sscIfmRenewLatestBuffer function or sscIfmRenewLatestBufferEx function.

■Point

[MC200]

- For API version 1.60 or later, use the sscIfmRenewLatestBufferEx function.
  ☞ Page 173 sscIfmRenewLatestBufferEx
- For API version 1.80 or later, when a NULL pointer is designated as the latest command buffer No., data will not be got.
- For API version 1.60 or earlier, when a NULL pointer is designated as the latest command buffer No., an access violation occurs, and therefore do not designate a NULL pointer.

[MC300]

- Use the sscIfmRenewLatestBufferEx function.
  ☞ Page 173 sscIfmRenewLatestBufferEx
- When a NULL pointer is designated as the latest command buffer No., data will not be got.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.40	A3	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

- sscIfmCheckLatestBuffer (☞ Page 174 sscIfmCheckLatestBuffer)
- sscIfmGetTransmitBuffer (☞ Page 176 sscIfmGetTransmitBuffer)

sscIfmRenewLatestBufferEx

The latest command buffer No. and data for the designated control mode is renewed, and the renewed latest command buffer No. is got.

```
int  sscIfmRenewLatestBufferEx (
     int  board_id,
     int  channel,
     int  axnum,
     unsigned short  ctrl_mode,
     long  bufdata,
     short  *bufnum
);
```


Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (1 to 32 [MC200]/1 to 64 [MC300])
ctrl_mode [in]	Control mode ■Operation mode • 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]	Renew data
bufnum [out]	Pointer to 2-byte variable (2 bytes × 1) which stores the latest command buffer No.

■Return value

Value	Description
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

Constant definition	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 No. has been renewed, call the sscIfmRenewLatestBuffer function or sscIfmRenewLatestBufferEx function.

■Point

[MC200]

- For API version 1.80 or later, when a NULL pointer is designated as the latest command buffer No., data will not be got.
- For API version 1.60 or earlier, when a NULL pointer is designated as the latest command buffer No., an access violation occurs, and therefore do not designate a NULL pointer.

[MC300]

- When a NULL pointer is designated as the latest command buffer No., data will not be got.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.60	A4	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

- sscIfmCheckLatestBufferEx (☞ Page 175 sscIfmCheckLatestBufferEx)
- sscIfmGetTransmitBufferEx (☞ Page 177 sscIfmGetTransmitBufferEx)

sscIfmCheckLatestBuffer

The latest command buffer No. and latest command buffer data for the position control mode is got.

```
int  sscIfmCheckLatestBuffer (
     int  board_id,
     int  channel,
     int  axnum,
     short  *bufnum,
     long  *bufdata
);
```

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (1 to 32)
bufnum [out]	Pointer to 2-byte variable (2 bytes × 1) which stores the latest command buffer No.
bufdata [out]	Pointer to 2-byte variable (2 bytes × 1) which stores the latest command buffer data

■Return value

Value	Description
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

Constant definition	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 No. is got.
- For API version 1.60 or later, use the sscIfmCheckLatestBufferEx function. [MC200]
- Use the sscIfmCheckLatestBufferEx function. [MC300]

☞ Page 175 sscIfmCheckLatestBufferEx

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.40	A3	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscIfmGetTransmitBuffer (☞ Page 176 sscIfmGetTransmitBuffer)

sscIfmCheckLatestBufferEx

The latest command buffer No. and latest command buffer data for the designated control mode is got.

```
int  sscIfmCheckLatestBufferEx (
     int  board_id,
     int  channel,
     int  axnum,
     unsigned short  ctrl_mode,
     short  *bufnum,
     long  *bufdata
);
```

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (1 to 32 [MC200]/1 to 64 [MC300])
ctrl_mode [in]	Control mode ■Operation mode • 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 No.
bufdata [out]	Pointer to 2-byte variable (2 bytes × 1) which stores the latest command buffer data

■Return value

Value	Description
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

Constant definition	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 No. is got.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.60	A4	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscIfmGetTransmitBufferEx (☞ Page 177 sscIfmGetTransmitBufferEx)

sscIfmGetTransmitBuffer

The transmit buffer No. and transmit buffer data for position control mode is got.

```
int  sscIfmGetTransmitBuffer (
     int  board_id,
     int  channel,
     int  axnum,
     short  *bufnum,
     long  *bufdata
);
```

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (1 to 32)
bufnum [out]	Pointer to 2-byte variable (2 bytes × 1) which stores the transmit buffer No.
bufdata [out]	Pointer to 4-byte variable (4 bytes × 1) which stores the transmit buffer data

■Return value

Value	Description
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

Constant definition	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 No. is got.
- For API version 1.60 or later, use the sscIfmGetTransmitBufferEx function. [MC200]
- Use the sscIfmGetTransmitBufferEx function. [MC300]

☞ Page 177 sscIfmGetTransmitBufferEx

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.40	A3	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscIfmCheckLatestBuffer (☞ Page 174 sscIfmCheckLatestBuffer)

sscIfmGetTransmitBufferEx

The transmit buffer No. and transmit buffer data for the designated control mode is got.

```
int  sscIfmGetTransmitBufferEx (
     int  board_id,
     int  channel,
     int  axnum,
     unsigned short  ctrl_mode,
     short  *bufnum,
     long  *bufdata
);
```

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (1 to 32 [MC200]/1 to 64 [MC300])
ctrl_mode [in]	Control mode ■Operation mode • 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 No.
bufdata [out]	Pointer to 4-byte variable (4 bytes × 1) which stores the transmit buffer data

■Return value

Value	Description
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

Constant definition	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 No. is got.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.60	A4	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscIfmCheckLatestBufferEx (☞ Page 175 sscIfmCheckLatestBufferEx)

sscIfmTrqSetSpeedLimit

The speed limit value for torque control is set.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (1 to 32 [MC200]/1 to 64 [MC300])
speed [in]	Speed limit value for torque control [0.01r/min] (0 to 1000000000)

■Return value

Value	Description
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

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

■Point

Call this API function before changing to torque control mode.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.60	A4	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscIfmSetControlMode (☞ Page 179 sscIfmSetControlMode)

sscIfmSetControlMode

The control mode is set.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (1 to 32 [MC200]/1 to 64 [MC300])
ctrl_mode [in]	Control mode ■Operation mode • 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

Value	Description
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

Constant definition	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 version	Board version	Header file
MR-MC2_ _	1.60	A4	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscIfmGetControlMode (☞ Page 180 sscIfmGetControlMode)

sscIfmGetControlMode

The control mode is got.

```
int  sscIfmGetControlMode (
     int  board_id,
     int  channel,
     int  axnum,
     unsigned short  *ctrl_mode,
     char  *status
);
```

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (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 • 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 • 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

Value	Description
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

Constant definition	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 version	Board version	Header file
MR-MC2_ _	1.60	A4	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscIfmSetControlMode (☞ Page 179 sscIfmSetControlMode)

sscIfmGetEventStatusBits

The status bit information of all axes for the designated status signal using the event detect function is got.

```
int  sscIfmGetEventStatusBits (
     int  board_id,
     int  channel,
     int  bitnum,
     unsigned long  *status_bits
);
```

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
bitnum [in]	Status bit No. ■Value • 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 erased • 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

Value	Description
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

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

■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 version	Board version	Header file
MR-MC2_ _	1.80	A6	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscGetStatusBitSignalEx (☞ Page 71 sscGetStatusBitSignalEx)

sscIfmGetEventStatusBitsEx [MC300]

The status bit information of all axes for the designated status signal using the event detect function is got.

```
int  sscIfmGetEventStatusBitsEx (
     int  board_id,
     int  channel,
     int  bitnum,
     SLAVE_INFO  *pSlaveInfo
);
```

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
bitnum [in]	Status bit No. ■Value • 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 erased • 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 device information (status bit data of all axes) For the device information structure, refer to the following. (For non-controlled axes and non-controlled stations information, the status bit is permanently OFF) ☞ Page 240 SLAVE_INFO Structure [MC300]

■Return value

Value	Description
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

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

■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 version	Board version	Header file
MR-MC3_ _	1.00	A0	mc3xxstd.h

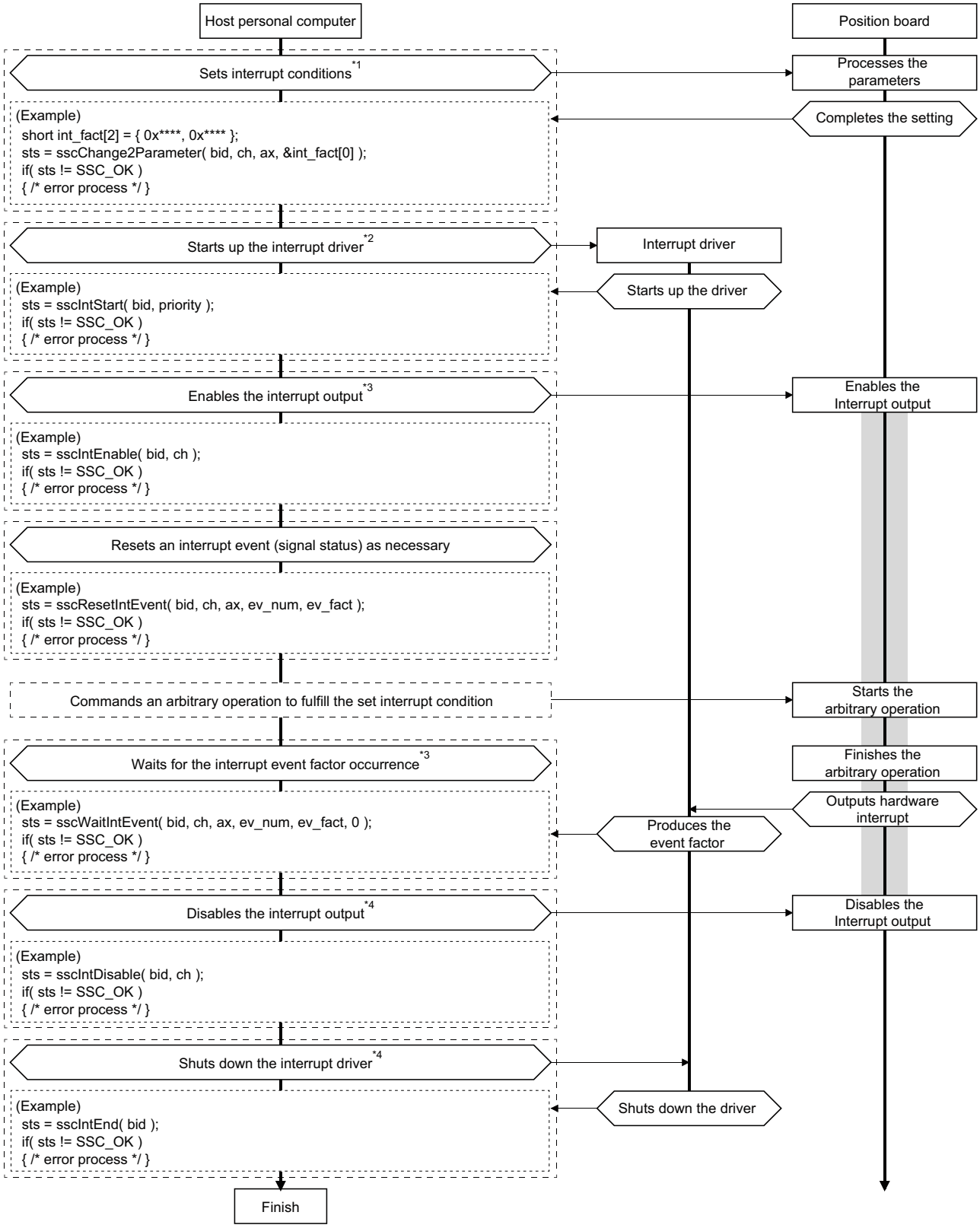
■Reference

None.

4.22 Interrupt Functions

Processing procedure

An example of using interrupt functions processing procedure is shown below.



*1 Only when changing the current set interrupt condition parameters ("interrupt condition 1 (parameter No.0204)", "interrupt condition 2 (parameter No.0205)"), call the function.
*2 Always enable the interrupt output after starting up the interrupt driver. (When the hardware interrupt is outputted while the interrupt driver is not operating properly, the host personal computer may hang-up because the hardware interrupt cannot be canceled.)
*3 After starting the interrupt, an interrupt event wait can be executed with the interrupt event wait functions. Also, an operation completion wait can be executed with the sscWaitIntDriveFin function.
*4 Always shut down the interrupt driver after disabling the interrupt output. (For the same reason of the *2 above)

sscIntStart

The interrupt driver is started up.

This function is used when performing the interrupt monitoring by using the functions related to the interrupt event wait.

```
int  sscIntStart (
     int  board_id,
     int  priority
);
```

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
priority [in]	Priority No.

■Return value

Value	Description
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

Constant definition	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 [MC200]	The interrupt output mask selection (dip switch) is valid.
SSC_FUNC_ERR_CREATE_EVENT	An error occurred in the CreateEvent function (Windows API). Call the GetLastError function of Windows API and confirm the error details.
SSC_FUNC_ERR_CREATE_THREAD	An error occurred in the CreateThread function (Windows API). Call the GetLastError function of Windows API and confirm the error details.
SSC_FUNC_ERR_THREAD_PRIORITY	An error occurred in the SetThreadPriority function (Windows API). Call the GetLastError function of Windows API and confirm the error details.
SSC_FUNC_ERR_RESUME_THREAD	An error occurred in the ResumeThread function (Windows API). Call the GetLastError function of Windows API and confirm the error details.
SSC_FUNC_ERR_CLEAR_INT [MC200]	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. Confirm that the device driver is installed.

■Point

- The interrupt driver priority No. is set using the SetThreadPriority function (Windows API).
- For details about the priority set values, refer to the reference manual for the operating system.
- After calling this API function, the interrupt should be enabled using the sscIntEnable function.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

- sscIntEnd (☞ Page 186 sscIntEnd)
- sscIntEnable (☞ Page 187 sscIntEnable)

sscIntEnd

The interrupt driver is closed.

```
int  sscIntEnd (
     int  board_id
);
```

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)

■Return value

Value	Description
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

Constant definition	Description
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. Confirm that the device driver is installed.
SSC_FUNC_ERR_SET_EVENT	An error occurred in the SetEvent function (Windows API). Call the GetLastError function of Windows API 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 (Windows API). Call the GetLastError function of Windows API and confirm the error details.
SSC_FUNC_ERR_DELETE_EVENT	An error occurred in the CloseHandle function (Windows API). Call the GetLastError function of Windows API and confirm the error details.
SSC_FUNC_ERR_GET_EXIT_CODE_THREAD	An error occurred in the GetExitCodeThread function (Windows API). Call the GetLastError function of Windows API and confirm the error details.

■Point

None.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscIntStart (☞ Page 185 sscIntStart)

sscIntEnable

The interrupt output start signal (ITS) is turned ON and the interrupt output is enabled.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)

■Return value

Value	Description
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

Constant definition	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 (Windows API). Call the GetLastError function of Windows API and confirm the error details.

■Point

None.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

- sscIntStart (☞ Page 185 sscIntStart)
- sscIntDisable (☞ Page 188 sscIntDisable)

sscIntDisable

The interrupt output start signal (ITS) is turned OFF and the interrupt output is disabled.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)

■Return value

Value	Description
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

Constant definition	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 (Windows API). Call the GetLastError function of Windows API and confirm the error details.

■Point

None.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscIntEnable (☞ Page 187 sscIntEnable)

sscRegisterIntCallback

The interrupt callback function is registered.

The registered function is called back from the interrupt driver started by the sscIntStart function when an interrupt occurs.

```
int  sscRegisterIntCallback (
     int  board_id,
     int  channel,
     void  *cbfunc
);
```

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
cbfunc [in]	Callback function pointer For the callback structure, refer to the following. ☞ Page 236 INT_CB_DATA Structure

■Return value

Value	Description
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

Constant definition	Description
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 processing stop signal (ITE) and the outputting 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 version	Board version	Header file
MR-MC2_ _	1.02	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

- sscIntStart (☞ Page 185 sscIntStart)
- sscIntEnable (☞ Page 187 sscIntEnable)
- sscUnregisterIntCallback(☞ Page 190 sscUnregisterIntCallback)

sscUnregisterIntCallback

The interrupt call back function is unregistered.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)

■Return value

Value	Description
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

Constant definition	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 version	Board version	Header file
MR-MC2_ _	1.02	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscRegisterIntCallback (☞ Page 189 sscRegisterIntCallback)

sscResetIntEvent

The interrupt event signal status is nonsignaled.

This function is used if interrupt events occurring prior to calling the sscWaitIntEvent function are to be disabled.

```
int  sscResetIntEvent (
     int  board_id,
     int  channel,
     int  axnum,
     int  eventnum,
     int  eventfactor
);
```

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (-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 No. (0 to 1)
eventfactor [in]	Event factor For the event factors, refer to the following. ☞ Page 289 INTERRUPT EVENT FACTOR LIST

■Return value

Value	Description
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

Constant definition	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 stopped. Call the sscIntStart function.
SSC_FUNC_ERR_RESET_EVENT	An error occurred in the ResetEvent function (Windows API). Call the GetLastError function of Windows API and confirm the error details.

■Point

None.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.80	A8	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscWaitIntEvent (☞ Page 193 sscWaitIntEvent)

sscSetIntEvent

The interrupt event signal status is 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
);
```

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (-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 No. (0 to 1)
eventfactor [in]	Event factor For the event factors, refer to the following. ☞ Page 289 INTERRUPT EVENT FACTOR LIST

■Return value

Value	Description
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

Constant definition	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 stopped. Call the sscIntStart function.
SSC_FUNC_ERR_SET_EVENT	An error occurred in the SetEvent function (Windows API). Call the GetLastError function of Windows API and confirm the error details.

■Point

When the interrupt standby status is released by calling this API function, an error occurs in the sscWaitIntEvent function.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.80	A8	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscWaitIntEvent (☞ Page 193 sscWaitIntEvent)

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 up to 2 interrupt events to occur from the same factor by changing the event wait No.

```
int  sscWaitIntEvent (
     int  board_id,
     int  channel,
     int  axnum,
     int  eventnum,
     int  eventfactor,
     int  timeout
);
```


Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (-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 No. (0 to 1)
eventfactor [in]	Event factor For the event factors, refer to the following. ☞ Page 289 INTERRUPT EVENT FACTOR LIST
timeout [in]	Timeout time [ms] (1 to 65535, SSC_INFINITE)

■Return value

Value	Cause/countermeasure
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

Constant definition	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 stopped. Call the sscIntStart function.
SSC_FUNC_ERR_WAIT_EVENT	An error occurred in the WaitForSingleObject function (Windows API). Call the GetLastError function of Windows API 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 has elapsed.
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 waited.

■Point

When "SSC_INFINITE" is designated as the timeout time, the timeout is not confirmed. Instead, this function infinitely waits until the interrupt event occurs.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.80	A8	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

- sscResetIntEvent (☞ Page 191 sscResetIntEvent)
- sscSetIntEvent (☞ Page 192 sscSetIntEvent)
- sscWaitIntEventMulti (☞ Page 197 sscWaitIntEventMulti)

sscResetIntEventMulti

The multiple interrupt event signal status is 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
);
```

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (-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

Value	Cause/countermeasure
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

Constant definition	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 stopped. Call the sscIntStart function.
SSC_FUNC_ERR_RESET_EVENT	An error occurred in the ResetEvent function (Windows API). Call the GetLastError function of Windows API and confirm the error details.

■Point

None.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.80	A8	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscWaitIntEventMulti (☞ Page 197 sscWaitIntEventMulti)

sscSetIntEventMulti

The multiple interrupt event signal status is 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
);
```

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (-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

Value	Cause/countermeasure
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

Constant definition	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 stopped. Call the sscIntStart function.
SSC_FUNC_ERR_SET_EVENT	An error occurred in the SetEvent function (Windows API). Call the GetLastError function of Windows API and confirm the error details.

■Point

When the interrupt standby status is released by calling this API function, an error occurs in the sscWaitIntEventMulti function.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.80	A8	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscWaitIntEventMulti (☞ Page 197 sscWaitIntEventMulti)

sscWaitIntEventMulti

This function waits until the multiple interrupt event status is signaled.

This function is used to wait for the interrupt from the position board for any multiple event factors.

```
int  sscWaitIntEventMulti (
     int  board_id,
     int  channel,
     int  axnum,
     int  timeout,
     unsigned long  *eventcode
);
```

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (-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] (1 to 65535, SSC_INFINITE)
eventcode [out]	Pointer to 4-byte variable (4 bytes × 1) which stores the multiple event factors For the event factors, refer to the following. ☞ Page 289 INTERRUPT EVENT FACTOR LIST

■Return value

Value	Cause/countermeasure
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

Constant definition	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 stopped. Call the sscIntStart function.
SSC_FUNC_ERR_WAIT_EVENT_MULTI	An error occurred in the WaitForMultipleObjects function (Windows API). Call the GetLastError function of Windows API 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	A function which releases the standby status was called from the user program while the interrupt for the designated event factor was being waited.

■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 is stored.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.80	A8	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

- sscWaitIntEvent (☞ Page 193 sscWaitIntEvent)
- sscResetIntEventMulti (☞ Page 195 sscResetIntEventMulti)
- sscSetIntEventMulti (☞ Page 196 sscSetIntEventMulti)

sscResetIntOasEvent

The other axes start interrupt event signal status is nonsignaled.
This function is used if the other axes start interrupt event occurring prior to calling the sscWaitIntOasEvent function is to be disabled.

```
int  sscResetIntOasEvent (
      int  board_id,
      int  channel,
      int  axnum,
      int  oas_num
);
```

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (1 to 32 [MC200]/1 to 64 [MC300])
oas_num [in]	Other axes start table No. (1 to 32 [MC200]/1 to 64 [MC300])

■Return value

Value	Description
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

Constant definition	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 stopped. Call the sscIntStart function.
SSC_FUNC_ERR_RESET_EVENT	An error occurred in the ResetEvent function (Windows API). Call the GetLastError function of Windows API and confirm the error details.

■Point

None.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscWaitIntOasEvent (☞ Page 201 sscWaitIntOasEvent)

sscSetIntOasEvent

The other axes start interrupt event signal status is 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
);
```

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. of other axes start table (1 to 32 [MC200]/1 to 64 [MC300])
oas_num [in]	Other axes start table No. (1 to 32 [MC200]/1 to 64 [MC300])

■Return value

Value	Description
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

Constant definition	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 stopped. Call the sscIntStart function.
SSC_FUNC_ERR_SET_EVENT	An error occurred in the SetEvent function (Windows API). Call the GetLastError function of Windows API and confirm the error details.

■Point

When the standby status is released by calling this API function, an error occurs in the sscWaitIntOasEvent function.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.02	A1	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscWaitIntOasEvent (☞ Page 201 sscWaitIntOasEvent)

sscWaitIntOasEvent

This function waits until the other axes start interrupt event status is signaled.
This function is used to wait for the interrupt from the position board for the designated other axes start wait type.
The confirmed status is stored in the variable which is designated by the pointer.

```
int  sscWaitIntOasEvent (
     int  board_id,
     int  channel,
     int  axnum,
     int  oas_num,
     int  oas_type,
     int  *oas_status,
     int  timeout
);
```


Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. of other axes start table (1 to 32 [MC200]/1 to 64 [MC300])
oas_num [in]	Other axes start table No. (1 to 32 [MC200]/1 to 64 [MC300])
oas_type [in]	Other axes start wait type ■Value • 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 • 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] (1 to 65535, SSC_INFINITE)

■Return value

Value	Description
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

Constant definition	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 stopped. Call the sscIntStart function.
SSC_FUNC_ERR_WAIT_EVENT	An error occurred in the WaitForSingleObject function (Windows API). Call the GetLastError function of Windows API 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 has elapsed.
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 waited.

■Point

- When "SSC_OAS_STS_FIN" or "SSC_OAS_STS_ERR" during other axes start notice wait, this function returns from the standby status.
- When an alarm occurs in the axis of the other axes start before the other axes start notice, "SSC_OAS_STS_OP_ERR" occurs.
- The following interrupt conditions are used for this API function. When using this API function, make sure to set an applicable interrupt condition by sscChange2Parameter function.
  OASF (Factor of other axes start interrupt is being sent)
  OPF (Operation completed)
- When "SSC_INFINITE" is designated as the timeout time, the timeout is not confirmed. Instead, this function infinitely waits until the event occurs.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

- sscResetIntOasEvent (☞ Page 199 sscResetIntOasEvent)
- sscSetIntOasEvent (☞ Page 200 sscSetIntOasEvent)

sscResetIntPassPosition

The pass position interrupt event status is nonsignaled.

This function is used to disable the pass position interrupt event occurring prior to calling the sscWaitIntPassPosition function.

```
int  sscResetIntPassPosition (
     int  board_id,
     int  channel,
     int  pass_start,
     int  pass_end
);
```

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
pass_start [in]	Pass position condition start No. (1 to 64 [MC200]/1 to 128 [MC300])
pass_end [in]	Pass position condition end No. (1 to 64 [MC200]/1 to 128 [MC300])

■Return value

Value	Description
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

Constant definition	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 stopped. Call the sscIntStart function.
SSC_FUNC_ERR_RESET_EVENT	An error occurred in the ResetEvent function (Windows API). Call the GetLastError function of Windows API and confirm the error details.

■Point

None.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.02	A1	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscWaitIntPassPosition (☞ Page 205 sscWaitIntPassPosition)

sscSetIntPassPosition

The pass position interrupt event status is 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
);
```

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
pass_start [in]	Pass position condition start No. (1 to 64 [MC200]/1 to 128 [MC300])
pass_end [in]	Pass position condition end No. (1 to 64 [MC200]/1 to 128 [MC300])

■Return value

Value	Description
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

Constant definition	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 stopped. Call the sscIntStart function.
SSC_FUNC_ERR_SET_EVENT	An error occurred in the SetEvent function (Windows API). Call the GetLastError function of Windows API and confirm the error details.

■Point

When the standby status is released by calling this API function, an error occurs in the sscWaitIntPassPosition function.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.02	A1	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscWaitIntPassPosition (☞ Page 205 sscWaitIntPassPosition)

sscWaitIntPassPosition

This function waits until the pass position interrupt event status is signaled.
This function is used to wait for the interrupt from the position board for the designated pass position condition No.
The confirmed status is stored in the variable which is designated by the pointer.

```
int  sscWaitIntPassPosition (
     int  board_id,
     int  channel,
     int  pass_num,
     int  *pass_status,
     int  timeout
);
```


Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
pass_num [in]	Pass position condition No. (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 • SSC_PASS_STS_FIN: Pass position interrupt complete • SSC_PASS_STS_ERR: Pass position interrupt incomplete
timeout [in]	Timeout time [ms] (1 to 65535, SSC_INFINITE)

■Return value

Value	Description
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

Constant definition	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 stopped. Call the sscIntStart function.
SSC_FUNC_ERR_WAIT_EVENT	An error occurred in the WaitForSingleObject function (Windows API). Call the GetLastError function of Windows API 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 has elapsed.
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 waited.

■Point

When "SSC_INFINITE" is designated as the timeout time, the timeout is not confirmed. Instead, this function infinitely waits until the interrupt event turns ON or OFF.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.02	A1	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

- sscResetIntPassPosition (☞ Page 203 sscResetIntPassPosition)
- sscSetIntPassPosition (☞ Page 204 sscSetIntPassPosition)

sscResetIntDriveFin

The operation completion interrupt event status is nonsignaled.
This function is used to disable the operation completion interrupt event occurring prior to calling the sscWaitIntDriveFin function.

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

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (1 to 32 [MC200]/1 to 64 [MC300])

■Return value

Value	Description
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

Constant definition	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 stopped. Call the sscIntStart function.
SSC_FUNC_ERR_RESET_EVENT	An error occurred in the ResetEvent function (Windows API). Call the GetLastError function of Windows API and confirm the error details.

■Point

None.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscWaitIntDriveFin (☞ Page 209 sscWaitIntDriveFin)

sscSetIntDriveFin

The operation completion interrupt event status is 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
);
```

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (1 to 32 [MC200]/1 to 64 [MC300])

■Return value

Value	Description
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

Constant definition	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 stopped. Call the sscIntStart function.
SSC_FUNC_ERR_SET_EVENT	An error occurred in the SetEvent function (Windows API). Call the GetLastError function of Windows API and confirm the error details.

■Point

When the standby status is released by calling this API function, an error occurs in the sscWaitIntDriveFin function.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.02	A1	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscWaitIntDriveFin (☞ Page 209 sscWaitIntDriveFin)

sscWaitIntDriveFin

This function waits until the operation completion interrupt event status is signaled.
This function is used to wait for the interrupt from the position board for the designated operation completion type.
The confirmed status is stored in the variable which is designated by the pointer.

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


Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (1 to 32 [MC200]/1 to 64 [MC300])
fin_type [in]	Operation completion type ■Value • 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 • 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] (1 to 65535, SSC_INFINITE)

■Return value

Value	Description
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

Constant definition	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 stopped. Call the sscIntStart function.
SSC_FUNC_ERR_WAIT_EVENT	An error occurred in the WaitForSingleObject function (Windows API). Call the GetLastError function of Windows API 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 has elapsed.
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 waited.

■Point

- The completion of operation check condition depends on the operation completion type designated by the argument.

Operation completion type	Completion of operation check condition
In case of "SSC_FIN_TYPE_SMZ", "SSC_FIN_TYPE_CPO", or "SSC_FIN_TYPE_INP"	The shutdown occurs after waiting for the designated conditions to be met.
In case of "SSC_FIN_TYPE_CPO"	The status is "SSC_FIN_STS_STP" when the rough match signal (CPO) is ON in the automatic operation mode or the linear interpolation [MC200]/interpolation operation [MC300] mode.
If a stop operation signal (STP) is input during operation	The status is "SSC_FIN_STS_STP" after the stop is completed.

- The rough match signal (CPO) is only output during the automatic operation mode and the linear interpolation [MC200]/interpolation operation [MC300] mode. Therefore, if the completion of operation check is performed in an operation mode other than the automatic operation mode or the linear interpolation [MC200]/interpolation operation [MC300] mode, the operation completion type other than "SSC_FIN_TYPE_CPO" should be used.
- The following interrupt conditions are used for this API function. When using this API function, make sure to set an applicable interrupt condition by the sscChange2Parameter function.

Interrupt condition	Remarks
INP (In-position)	Unnecessary when the operation completion type "SSC_FIN_TYPE_INP" is not used.
SALM (During servo alarm)	—
CPO (Rough match)	Unnecessary when the operation completion type "SSC_FIN_TYPE_CPO" is not used.
OALM (During operation alarm)	—
OPF (Operation completed)	—
SYSE (During system error)	Unnecessary when the operation completion type "SSC_FIN_TYPE_INP" 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 confirmed. 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] mode, the Operation completion type is "SSC_FIN_TYPE_INP" even though "SSC_FIN_TYPE_SMZ" is set.
- The completion of operation status when timeout occurs is indefinite.
- Since the completion of operation status of this API function is judged by the operation completion type, it may differ from the operation processing signal (OP) and the operation completed signal (OPF) of the position board.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

- sscResetIntDriveFin (☞ Page 207 sscResetIntDriveFin)
- sscSetIntDriveFin (☞ Page 208 sscSetIntDriveFin)
- sscGetDriveFinStatus (☞ Page 102 sscGetDriveFinStatus)

4.23 I/O Device Functions

sscGetInputDeviceBit

The designated input bit device is got on a 1-point basis.

```
int  sscGetInputDeviceBit (
     int  board_id,
     int  channel,
     int  bit_num,
     int  *dev_in
);
```


Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
bit_num [in]	Input bit device No. (0000h to 0FFFh [MC200]/0000h to 23FFh [MC300])
dev_in [out]	Pointer to 4-byte variable (4 bytes × 1) which stores the status of the input bit device ■Value • SSC_BIT_OFF: Input signal OFF • SSC_BIT_ON: Input signal ON

■Return value

Value	Description
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

Constant definition	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 version	Board version	Header file
MR-MC2_ _	1.80	A8	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscGetInputDeviceWord (☞ Page 212 sscGetInputDeviceWord)

sscGetInputDeviceWord

The designated input word device is got on a 1-word basis.

```
int  sscGetInputDeviceWord (
     int  board_id,
     int  channel,
     int  word_num,
     int  word_cnt,
     unsigned short  *dev_in
);
```

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
word_num [in]	Input word device No. (0000h to 00FFh [MC200]/0000h to 023Fh [MC300])
word_cnt [in]	Word points from the input word device No. (0001h to 0100h [MC200]/0001h to 0240h [MC300])
dev_in [out]	Pointer to the array (2 bytes × word_cnt) which stores the status of the input word device

■Return value

Value	Description
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

Constant definition	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 version	Board version	Header file
MR-MC2_ _	1.80	A8	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscGetInputDeviceBit (☞ Page 211 sscGetInputDeviceBit)

sscSetOutputDeviceBit

The designated output bit device is set on a 1-point basis with the dual port memory exclusive control function of the position board.

```
int  sscSetOutputDeviceBit (
     int  board_id,
     int  channel,
     int  bit_num,
     int  dev_out
);
```

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
bit_num [in]	Output bit device No. (0000h to 0FFFh [MC200]/0000h to 23FFh [MC300])
dev_out [in] (0 to 1)	Setting data ■Value • SSC_BIT_OFF: Output signal OFF • SSC_BIT_ON: Output signal ON

■Return value

Value	Description
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

Constant definition	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 4: Timeout location	During the confirmation of response after executing the command to the position board, the timeout time (10 second) has elapsed.

■Point

This API 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 version	Board version	Header file
MR-MC2_ _	1.80	A8	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

- sscSetOutputDeviceWord (☞ Page 214 sscSetOutputDeviceWord)
- sscGetOutputDeviceBit (☞ Page 215 sscGetOutputDeviceBit)

sscSetOutputDeviceWord

The designated output word device is set on a 1-word basis without using the dual port memory exclusive control function of the position board.

```
int  sscSetOutputDeviceWord (
     int  board_id,
     int  channel,
     int  word_num,
     int  word_cnt,
     unsigned short  *dev_out
);
```

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
word_num [in]	Output word device No. (0000h to 00FFh [MC200]/0000h to 023Fh [MC300])
word_cnt [in]	Word points from the output word device No. (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

Value	Description
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

Constant definition	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 output word devices are set for the applicable output signals during controlling output signals by the other axes start function, the data may become inconsistent.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.80	A8	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

- sscSetOutputDeviceBit (☞ Page 213 sscSetOutputDeviceBit)
- sscGetOutputDeviceWord (☞ Page 216 sscGetOutputDeviceWord)

sscGetOutputDeviceBit

The designated output bit device is got on a 1-point basis.

```
int  sscGetOutputDeviceBit (
     int  board_id,
     int  channel,
     int  bit_num,
     int  *dev_out
);
```

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
bit_num [in]	Output bit device No. (0000h to 0FFFh [MC200]/0000h to 23FFh [MC300])
dev_out [out]	Pointer to 4-byte variable (4 bytes × 1) which stores the status of the output bit device ■Value • SSC_BIT_OFF: Output signal OFF • SSC_BIT_ON: Output signal ON

■Return value

Value	Description
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

Constant definition	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 version	Board version	Header file
MR-MC2_ _	1.80	A8	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

- sscGetOutputDeviceWord (☞ Page 216 sscGetOutputDeviceWord)
- sscSetOutputDeviceBit (☞ Page 213 sscSetOutputDeviceBit)

sscGetOutputDeviceWord

The designated output word device is got on a 1-word basis.

```
int  sscGetOutputDeviceWord (
     int  board_id,
     int  channel,
     int  word_num,
     int  word_cnt,
     unsigned short  *dev_out
);
```

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
word_num [in]	Output word device No. (0000h to 00FFh [MC200]/0000h to 023Fh [MC300])
word_cnt [in]	Word points from the output word device No. (0001h to 0100h [MC200]/0001h to 0240h [MC300])
dev_out [out]	Pointer to the array (2 bytes × word_cnt) which stores the status of the output word device

■Return value

Value	Description
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

Constant definition	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 version	Board version	Header file
MR-MC2_ _	1.80	A8	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

- sscGetOutputDeviceBit (☞ Page 215 sscGetOutputDeviceBit)
- sscSetOutputDeviceWord (☞ Page 214 sscSetOutputDeviceWord)

4.24 Transient Transmit Functions

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
);
```


Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (-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 For the transient transmit command data structure, refer to the following. ☞ Page 234 TRANSIENT_CMD Structure
pTransientSts [out]	Pointer to 16-byte structure (16 bytes × 1) which stores the transient transmit status data For the transient transmit status data structure, refer to the following. ☞ Page 235 TRANSIENT_STS Structure
timeout [in]	Timeout time [ms] (0 to 65535)

■Return value

Value	Description
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

Constant definition	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 has 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).
- For API Version 2.20 or earlier [MC200]/API Version 1.10 or earlier [MC300], the function name is "sscSendRecieveTransientData".

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.80	A8	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

None.

5 STRUCTURE LIST

5.1 PNT_DATA_EX Structure

This point data structure is used for the automatic operation and the linear interpolation [MC200]/interpolation operation [MC300].

PNT_DATA_EX structure [MC200]

```
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  reserve1[2];
    unsigned char  s_curve;
    unsigned char  reserve2[3];
    unsigned char  sub_axnum[3];
    unsigned char  reserve3[5];
/* 0020h */
} PNT_DATA_EX;
```

Detailed description

■Member

Member	Description
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 the data in the logical sum of each value. ■Value [Position command system] • SSC_SUBCMD_POS_ABS: Absolute position command • SSC_SUBCMD_POS_INC: Relative position command [Deceleration check system] • SSC_SUBCMD_STOP_INP: In-position stop • SSC_SUBCMD_STOP_SMZ: Smoothing stop • SSC_SUBCMD_STOP_CONTINUE: Continue operation [Speed switching point specification] • SSC_SUBCMD_PNT_SWITCH_AFTER: After point switching • SSC_SUBCMD_PNT_SWITCH_BEFORE: Before point switching [Dwell specification] • SSC_SUBCMD_DWELL: Dwell • SSC_SUBCMD_PREDWELL: Predwell [Pass position interrupt specification] • SSC_SUBCMD_PASS_POS_DISABLE: Disabled • SSC_SUBCMD_PASS_POS_ENABLE: Enabled [Continuous operation to torque control specification] • SSC_SUBCMD_PRESS_DISABLE: Disabled • SSC_SUBCMD_PRESS_ENABLE: Enabled [Loop specification] • SSC_SUBCMD_PNT_LOOP_DISABLE: Disabled • 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)

Member	Description
s_curve	S-curve ratio [%] (0 to 100)
sub_axnum[3]	Interpolation axis No. (0 to 32) Set the axis No. 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 version	Board version	Header file
MR-MC2_ _	1.90	A9	mc2xxstd.h

■Reference

- sscSetPointDataEx (☞ Page 74 sscSetPointDataEx)
- sscCheckPointDataEx (☞ Page 75 sscCheckPointDataEx)

PNT_DATA_EX structure [MC300]

```
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;
    char  reserve1[3];
    unsigned char  sub_axnum[4];
    long  arc_coord;
/* 0020h */
    unsigned short  ac_dc_data[4];
    unsigned short  subcmd2;
    char  reserve2[6];
/* 0030h */
} PNT_DATA_EX;
```

Detailed description

■Member

Member	Description
position	Position data [command unit]
speed	Feed speed [speed unit] (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 the data in the logical sum of each value. ■Value [Position command system] • SSC_SUBCMD_POS_ABS: Absolute position command • SSC_SUBCMD_POS_INC: Relative position command [Vibration suppression command filter 1 command] • SSC_SUBCMD_DAMPING_DISABLE: Vibration suppression command filter 1 disabled • SSC_SUBCMD_DAMPING_ENABLE: Vibration suppression command filter 1 enabled [Deceleration check system] • SSC_SUBCMD_STOP_INP: In-position stop • SSC_SUBCMD_STOP_SMZ: Smoothing stop • SSC_SUBCMD_STOP_CONTINUE: Continue operation [Speed switching point specification] • SSC_SUBCMD_PNT_SWITCH_AFTER: After point switching • SSC_SUBCMD_PNT_SWITCH_BEFORE: Before point switching [Dwell specification] • SSC_SUBCMD_DWELL: Dwell • SSC_SUBCMD_PREDWELL: Predwell [Pass position interrupt specification] • SSC_SUBCMD_PASS_POS_DISABLE: Disabled • SSC_SUBCMD_PASS_POS_ENABLE: Enabled [Continuous operation to torque control specification] • SSC_SUBCMD_PRESS_DISABLE: Disabled • SSC_SUBCMD_PRESS_ENABLE: Enabled [Loop specification] • SSC_SUBCMD_PNT_LOOP_DISABLE: Disabled • SSC_SUBCMD_PNT_LOOP_START: Loop start point • SSC_SUBCMD_PNT_LOOP_END: Loop end point [Interpolation system] • SSC_SUBCMD_INTERP_LINEAR: Linear interpolation • SSC_SUBCMD_INTERP_ARC: Auxiliary 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)

Member	Description
s_curve	S-curve ratio [%] (0 to 100)
sub_axnum[4]	Interpolation axis No. (0 to 64) Set the axis No. of the auxiliary axes to be in the same interpolation group. When not using auxiliary axes, set "0".
arc_coord	Arc coordinates [command unit] Set the auxiliary 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 [Acceleration/deceleration system] • SSC_SUBCMD2_ACCDEC_LINE_S: Linear acceleration/deceleration and S-curve acceleration/deceleration • SSC_SUBCMD2_ACCDEC_JERK: Jerk ratio acceleration/deceleration

■Point

None.

■Supported version

Position board	API version	Board version	Header file
MR-MC3_ _	1.10	A1	mc3xxstd.h

■Reference

- sscSetPointDataEx (☞ Page 74 sscSetPointDataEx)
- sscCheckPointDataEx (☞ Page 75 sscCheckPointDataEx)

5.2 OAS_DATA Structure

This other axis start data structure is used for the other axes start.

OAS_DATA structure [MC200]

```
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;
```

Detailed description

■Member

Member	Description
opt_own	Axis option Set the data in the logical sum of each value. ■Value [Axis judgment condition] • SSC_OAS_OWN_REMAINING_DISTANCE: Remaining distance specification • SSC_OAS_OWN_POSITION_PASS: Specified position pass specification [Axis judgment coordinate] • SSC_OAS_OWN_JUDGE_COORD_FB: Current feedback position • SSC_OAS_OWN_JUDGE_COORD_CMD: Command position
opt_observ	Observed axis option Set the data in the logical sum of each value. ■Value [Observed axis specification] • SSC_OAS_OBSERV_DISABLE: Disabled • SSC_OAS_OBSERV_ENABLE: Enabled [Observed axis judgment condition] • SSC_OAS_OBSERV_POSITION_PASS: Observed axis specified position pass specification [Observed axis judgment coordinate] • SSC_OAS_OBSERV_JUDGE_COORD_FB: Current feedback position • SSC_OAS_OBSERV_JUDGE_COORD_CMD: Command position [Observed axis specified position pass judgment condition] • SSC_OAS_OBSERV_DATA_LESS: The condition is satisfied when the observed axis position is less than or equal to the observed axis specified position data. • SSC_OAS_OBSERV_DATA_MORE: The condition is satisfied when the observed axis position is more than or equal to the observed axis specified position data.
data_own	Axis remaining distance data (or axis pass position data) [command unit]
data_observ	Observed axis remaining distance data (or observed axis pass position data) [command unit]
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)

Member	Description
dout_ctrl	Digital output signal control/Output device signal control ■Value [Digital output signal control] • SSC_OAS_DO_DISABLE: Disabled • SSC_OAS_DO_ENABLE: Enabled
dout_num	Digital output signal No./Output device signal No. (00h 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 version	Board version	Header file
MR-MC2_ _	1.02	A1	mc2xxstd.h

■Reference

- sscSetOtherAxisStartData (☞ Page 133 sscSetOtherAxisStartData)
- sscGetOtherAxisStartData (☞ Page 134 sscGetOtherAxisStartData)

OAS_DATA structure [MC300]

```
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;
```

Detailed description

■Member

Member	Description
opt_own	Axis option Set the data in the logical sum of each value. ■Value [Axis judgment condition] • SSC_OAS_OWN_REMAINING_DISTANCE: Remaining distance specification • SSC_OAS_OWN_POSITION_PASS: Specified position pass specification [Axis judgment coordinate] • SSC_OAS_OWN_JUDGE_COORD_FB: Current feedback position • SSC_OAS_OWN_JUDGE_COORD_CMD: Command position
opt_observ	Observed axis option Set the data in the logical sum of each value. ■Value [Observed axis specification] • SSC_OAS_OBSERV_DISABLE: Disabled • SSC_OAS_OBSERV_ENABLE: Enabled [Observed axis judgment condition] • SSC_OAS_OBSERV_POSITION_PASS: Observed axis specified position pass specification [Observed axis judgment coordinate] • SSC_OAS_OBSERV_JUDGE_COORD_FB: Current feedback position • SSC_OAS_OBSERV_JUDGE_COORD_CMD: Command position [Observed axis specified position pass judgment condition] • SSC_OAS_OBSERV_DATA_LESS: The condition is satisfied when the observed axis position is less than or equal to the observed axis specified position data. • SSC_OAS_OBSERV_DATA_MORE: The condition is satisfied when the observed axis position is more than or equal to the observed axis specified position data.
data_own	Axis remaining distance data (or axis pass position data) [command unit]
data_observ	Observed axis remaining distance data (or observed axis pass position data) [command unit]
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 [Digital output signal control] • SSC_OAS_DO_DISABLE: Disabled • SSC_OAS_DO_ENABLE: Enabled
dout_num	Digital output signal No./Output device signal No. (00h to 3Fh)

Member	Description
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 version	Board version	Header file
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

- sscSetOtherAxisStartData (☞ Page 133 sscSetOtherAxisStartData)
- sscGetOtherAxisStartData (☞ Page 134 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;
```

Detailed description

■Member

Member	Description
switch_position	Continuous operation to torque control switching position [command position]
position_limit	Press limit position [command position]
speed_limit	Continuous operation to torque control speed limit value [speed unit] (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 [Start switch to continuous operation to torque control condition] • SSC_PRESS_START_AUTO_CMD: Automatic switch (command position) • SSC_PRESS_START_AUTO_FB: Automatic switch (current feedback position) • SSC_PRESS_START_MANUAL: Manual switch [End switch to continuous operation to torque control condition] • SSC_PRESS_END_AUTO: Automatic switch • SSC_PRESS_END_MANUAL: Manual switch

■Point

None.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.60	A5	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

- sscSetPressData (☞ Page 80 sscSetPressData)
- sscGetPressData (☞ Page 81 sscGetPressData)

5.4 SMP_ERR Structure

This sampling error structure is used for the sampling.

SMP_ERR structure [MC200]

```
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;
```


Detailed description

■Member

Member	Description
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 version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h

■Reference

sscGetSamplingError (☞ Page 146 sscGetSamplingError)

SMP_ERR structure [MC300]

```
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;
```

Detailed description

■Member

Member	Description
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 version	Board version	Header file
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscGetSamplingError (☞ Page 146 sscGetSamplingError)

5.5 SMP_DATA Structure

This sampling data structure is used for the sampling.

SMP_DATA structure [MC200]

```
typedef struct {
/* 0000h */
    long  smpdata[32];
/* 0080h */
    unsigned short  smpbit[1];
    unsigned char  reserve1[2];
/* 0084h */
} SMP_DATA;
```

Detailed description

■Member

Member	Description
smpdata[32]	Data 1 to 32
smpbit[1]	Bit information

■Point

None.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h

■Reference

sscGetSamplingData (☞ Page 148 sscGetSamplingData)

SMP_DATA structure [MC300]

```
typedef struct {
/* 0000h */
    long  smpdata[32];
/* 0080h */
    unsigned long  smpbit[1];
/* 0084h */
} SMP_DATA;
```

Detailed description

■Member

Member	Description
smpdata[32]	Data 1 to 32
smpbit[1]	Bit information

■Point

None.

■Supported version

Position board	API version	Board version	Header file
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscGetSamplingData (☞ Page 148 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;
```

Detailed description

■Member

Member	Description
axnum	Axis No.
eventcode	Event code
eventtime	Time stamp
eventdata[4]	Information for each event

■Point

None.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.00	A0	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscReadLogData (☞ Page 153 sscReadLogData)

5.7 ALH_DATA Structure

This alarm history data structure is used for the alarm history function.

ALH_DATA structure [MC200]

```
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;
```

Detailed description

■Member

Member	Description
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 No.
alarm_code	Alarm No.
drive_mode	Operation mode
cmd_pos	Current position [command unit]
fb_pos	Feedback position [command unit]
check_sum	Checksum

■Point

None.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.40	A3	mc2xxstd.h

■Reference

sscGetAlarmHistoryData (☞ Page 155 sscGetAlarmHistoryData)

ALH_DATA structure [MC300]

```
typedef struct {
/* 0000h */
    long long  system_time;
    unsigned long  free_run_cnt;
    unsigned char  ctrl_cycle;
    unsigned char  event_code;
    unsigned char  ctrl_cycle_low_speed;
    char  reserve1;
/* 0010h */
    char  reserve2[4];
    unsigned short  axnum;
    unsigned short  alarm_code;
    unsigned char  drive_mode;
    char  reserve3[3];
    long  cmd_pos;
/* 0020h */
    long  fb_pos;
    unsigned short  alarm_code_j5;
    unsigned short  alarm_detail_code_j5;
    char  reserve4[23];
    unsigned char  check_sum;
/* 0040h */
} ALH_DATA;
```


Detailed description

■Member

Member	Description
system_time	System start-up time
free_run_cnt	Free-run counter
ctrl_cycle	Control cycle
event_code	Event code
ctrl_cycle_low_speed	Low speed control cycle
axnum	Error axis No.
alarm_code	Alarm No.
drive_mode	Operation mode
cmd_pos	Current position [command unit]
fb_pos	Feedback position [command unit]
alarm_code_j5	Alarm No. (MR-J5(W_)-_B)
alarm_detail_code_j5	Specific alarm No. (MR-J5(W_)-_B)
check_sum	Checksum

■Point

None.

■Supported version

Position board	API version	Board version	Header file
MR-MC3_ _	1.30	A6	mc3xxstd.h

■Reference

sscGetAlarmHistoryData (☞ Page 155 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;
```

Detailed description

■Member

Member	Description
cmd_req	Command send request ■Value SSC_TRANSIENT_CMD_SINGLE: Transient request
command	Transient command
req_data[4]	Request data

■Point

None.

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.80	A8	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscSendReceiveTransientData (☞ Page 217 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;
```

Detailed description

■Member

Member	Description
status	Transient status Obtained data is set with the logical sum of each value. ■Value • 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 version	Board version	Header file
MR-MC2_ _	1.80	A8	mc2xxstd.h
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscSendReceiveTransientData (☞ Page 217 sscSendReceiveTransientData)

5.10 INT_CB_DATA Structure

This interrupt data structure is used for the interrupt callback function.

INT_CB_DATA structure [MC200]

```
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;
```

Detailed description

■Member

Member	Description
board_id	Board ID No.
channel	Channel No.
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.
  For description of the data stored, refer to "Factor of Event" in the following.
  📖MR-MC200/MR-MC300 Series Position Board User's Manual (Details)

■Supported version

Position board	API version	Board version	Header file
MR-MC2_ _	1.80	A8	mc2xxstd.h

■Reference

sscRegisterIntCallback (☞ Page 189 sscRegisterIntCallback)

INT_CB_DATA structure [MC300]

```
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 */
    char  reserve7[0x290];
/* 0800h */
} INT_CB_DATA;
```

Detailed description

■Member

Member	Description
board_id	Board ID No.
channel	Channel No.
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

■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.
  For description of the data stored, refer to "Factor of Event" in the following.
  📖MR-MC200/MR-MC300 Series Position Board User's Manual (Details)

■Supported version

Position board	API version	Board version	Header file
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

sscRegisterIntCallback (☞ Page 189 sscRegisterIntCallback)

5.11 SLAVE_INFO Structure [MC300]

This device 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;
```

Detailed description

■Member

Member	Description
axis[64]	Axis information. ■Value • SSC_SLAVE_ON: Controlled axis or status bit ON • SSC_SLAVE_OFF: Non-controlled axis or status bit OFF
unit[16]	Station information. ■Value • 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 version	Board version	Header file
MR-MC3_ _	1.00	A0	mc3xxstd.h

■Reference

- sscReconnectSSCNETEx (☞ Page 65 sscReconnectSSCNETEx [MC300])
- sscDisconnectSSCNETEx (☞ Page 67 sscDisconnectSSCNETEx [MC300])
- sscIfmGetEventStatusBitsEx (☞ Page 182 sscIfmGetEventStatusBitsEx [MC300])

6 BIT DEFINITION LIST

The following tables list the bit definitions to be specified in the sscSetCommandBitSignalEx function, the sscGetStatusBitSignalEx function, and the sscWaitStatusBitSignalEx function.

6.1 System Command Bit

Bit No. (constant)	Signal name
SSC_CMDBIT_SYS_ITE	Interrupt processing stop
SSC_CMDBIT_SYS_ITS	Interrupt output start
SSC_CMDBIT_SYS_03	For manufacturer setting
SSC_CMDBIT_SYS_04	
SSC_CMDBIT_SYS_HMA	During user program memory access
SSC_CMDBIT_SYS_06	For manufacturer setting
SSC_CMDBIT_SYS_07	
SSC_CMDBIT_SYS_08	

Bit No. (constant)	Signal name
SSC_CMDBIT_SYS_SMPS	Sampling start
SSC_CMDBIT_SYS_10	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_SYS_SEMI	Software forced stop
SSC_CMDBIT_SYS_18	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_SYS_25	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_SYS_ITFE	Interrupt processing fast stop
SSC_CMDBIT_SYS_34	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_SYS_41	For manufacturer setting
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 No. (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 No. (constant)	Signal name
SSC_CMDBIT_SYS_57	For manufacturer setting
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 No. (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	For manufacturer setting
SSC_CMDBIT_SYS_SMON	System monitor command
SSC_CMDBIT_SYS_SMONR	System monitor latch command
SSC_CMDBIT_SYS_71	For manufacturer setting
SSC_CMDBIT_SYS_72	

Bit No. (constant)	Signal name
SSC_CMDBIT_SYS_73	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_SYS_LOGC	Log command
SSC_CMDBIT_SYS_LOGR	Reading of log data command
SSC_CMDBIT_SYS_83	For manufacturer setting
SSC_CMDBIT_SYS_LOGI	Log data initialization command
SSC_CMDBIT_SYS_85	For manufacturer setting
SSC_CMDBIT_SYS_OCMC	Operation cycle monitor clear
SSC_CMDBIT_SYS_87	For manufacturer setting
SSC_CMDBIT_SYS_88	

Bit No. (constant)	Signal name
SSC_CMDBIT_SYS_RCC	Reconnection command
SSC_CMDBIT_SYS_90	For manufacturer setting
SSC_CMDBIT_SYS_91	
SSC_CMDBIT_SYS_CCC	Disconnection command
SSC_CMDBIT_SYS_93	For manufacturer setting
SSC_CMDBIT_SYS_94	
SSC_CMDBIT_SYS_95	
SSC_CMDBIT_SYS_96	

Bit No. (constant)	Signal name
SSC_CMDBIT_SYS_97	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_SYS_105	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_SYS_113	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_SYS_121	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_SYS_SPWRT	Parameter write command
SSC_CMDBIT_SYS_130	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_SYS_SPRD	Parameter read command
SSC_CMDBIT_SYS_138	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_SYS_SMPSW	Sampling setting write command
SSC_CMDBIT_SYS_146	For manufacturer setting
SSC_CMDBIT_SYS_147	
SSC_CMDBIT_SYS_148	
SSC_CMDBIT_SYS_SMPSR	Sampling setting read command
SSC_CMDBIT_SYS_150	For manufacturer setting
SSC_CMDBIT_SYS_151	
SSC_CMDBIT_SYS_152	

Bit No. (constant)	Signal name
SSC_CMDBIT_SYS_153	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_SYS_161	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_SYS_169	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_SYS_FTR	Flash ROM transfer preparation
SSC_CMDBIT_SYS_FTS	Flash ROM transfer execution
SSC_CMDBIT_SYS_179	For manufacturer setting
SSC_CMDBIT_SYS_180	
SSC_CMDBIT_SYS_FIR	Flash ROM initialization preparation
SSC_CMDBIT_SYS_FIS	Flash ROM initialization execution
SSC_CMDBIT_SYS_183	For manufacturer setting
SSC_CMDBIT_SYS_184	

Bit No. (constant)	Signal name
SSC_CMDBIT_SYS_ALHR	Alarm history read command
SSC_CMDBIT_SYS_186	For manufacturer setting
SSC_CMDBIT_SYS_ALHI	Alarm history initialization command
SSC_CMDBIT_SYS_188	For manufacturer setting
SSC_CMDBIT_SYS_189	
SSC_CMDBIT_SYS_190	
SSC_CMDBIT_SYS_191	
SSC_CMDBIT_SYS_192	

Bit No. (constant)	Signal name
SSC_CMDBIT_SYS_193	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_SYS_201	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_SYS_209	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_SYS_217	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_SYS_225	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_SYS_233	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_SYS_241	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_SYS_249	For manufacturer setting
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	

System command bit [MC300]

Bit No. (constant)	Signal name
SSC_CMDBIT_SYS_257	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_SYS_265	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_SYS_273	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_SYS_281	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_SYS_289	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_SYS_297	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_SYS_305	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_SYS_313	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_SYS_321	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_SYS_329	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_SYS_337	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_SYS_345	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_SYS_353	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_SYS_361	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_SYS_369	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_SYS_377	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_SYS_ITO	Outputting factor of interrupt
SSC_STSBIT_SYS_IITO	During interface mode interrupt valid
SSC_STSBIT_SYS_EVDO	Event detect enabled
SSC_STSBIT_SYS_HRIF	Highly response I/F enabled
SSC_STSBIT_SYS_BMA	During system program memory access
SSC_STSBIT_SYS_PRINF	Continuous operation to torque control compatible information
SSC_STSBIT_SYS_07	For manufacturer setting
SSC_STSBIT_SYS_IFMO	In interface mode

Bit No. (constant)	Signal name
SSC_STSBIT_SYS_SMPW	Waiting for sampling trigger
SSC_STSBIT_SYS_SMPO	Sampling is being performed
SSC_STSBIT_SYS_SMPF	Sampling completed
SSC_STSBIT_SYS_SMPE	Sampling error
SSC_STSBIT_SYS_13	For manufacturer setting
SSC_STSBIT_SYS_AHINF	Alarm history information
SSC_STSBIT_SYS_15	For manufacturer setting
SSC_STSBIT_SYS_16	

Bit No. (constant)	Signal name
SSC_STSBIT_SYS_EMIO	Being executed forced stop
SSC_STSBIT_SYS_18	For manufacturer setting
SSC_STSBIT_SYS_TSTO	In test mode
SSC_STSBIT_SYS_20	For manufacturer setting
SSC_STSBIT_SYS_21	
SSC_STSBIT_SYS_22	
SSC_STSBIT_SYS_EMID	External forced stop disabled
SSC_STSBIT_SYS_24	For manufacturer setting

Bit No. (constant)	Signal name
SSC_STSBIT_SYS_25	For manufacturer setting
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	For manufacturer setting

Bit No. (constant)	Signal name
SSC_STSBIT_SYS_33	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_SYS_41	For manufacturer setting
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 No. (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 No. (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 No. (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	For manufacturer setting
SSC_STSBIT_SYS_SMOUT	Monitor output
SSC_STSBIT_SYS_SMRCH	Monitor latch
SSC_STSBIT_SYS_SMER1	Monitor No. error 1
SSC_STSBIT_SYS_SMER2	Monitor No. error 2

Bit No. (constant)	Signal name
SSC_STSBIT_SYS_73	For manufacturer setting
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 No. (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 No. (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	For manufacturer setting
SSC_STSBIT_SYS_96	

Bit No. (constant)	Signal name
SSC_STSBIT_SYS_97	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_SYS_105	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_SYS_113	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_SYS_121	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_SYS_SPWFIN1	Parameter write completed 1
SSC_STSBIT_SYS_SPWEN1	Parameter No. error 1
SSC_STSBIT_SYS_SPWED1	Parameter data out of bounds 1
SSC_STSBIT_SYS_132	For manufacturer setting
SSC_STSBIT_SYS_SPWFIN2	Parameter write completed 2
SSC_STSBIT_SYS_SPWEN2	Parameter No. error 2
SSC_STSBIT_SYS_SPWED2	Parameter data out of bounds 2
SSC_STSBIT_SYS_136	For manufacturer setting

Bit No. (constant)	Signal name
SSC_STSBIT_SYS_SPRFIN1	Parameter read completed 1
SSC_STSBIT_SYS_SPREN1	Parameter No. error 1
SSC_STSBIT_SYS_SPRFIN2	Parameter read completed 2
SSC_STSBIT_SYS_SPREN2	Parameter No. error 2
SSC_STSBIT_SYS_141	For manufacturer setting
SSC_STSBIT_SYS_142	
SSC_STSBIT_SYS_143	
SSC_STSBIT_SYS_144	

Bit No. (constant)	Signal name
SSC_STSBIT_SYS_SWFIN	Sampling setting write completed
SSC_STSBIT_SYS_SWEN	Sampling setting No. error
SSC_STSBIT_SYS_SWED	Sampling setting data out of bounds
SSC_STSBIT_SYS_148	For manufacturer setting
SSC_STSBIT_SYS_SRFIN	Sampling setting read completed
SSC_STSBIT_SYS_SREN	Sampling setting No. error
SSC_STSBIT_SYS_151	For manufacturer setting
SSC_STSBIT_SYS_152	

Bit No. (constant)	Signal name
SSC_STSBIT_SYS_153	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_SYS_161	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_SYS_169	For manufacturer setting
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 No. (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 No. (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	For manufacturer setting
SSC_STSBIT_SYS_190	
SSC_STSBIT_SYS_191	
SSC_STSBIT_SYS_192	

Bit No. (constant)	Signal name
SSC_STSBIT_SYS_193	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_SYS_201	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_SYS_209	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_SYS_217	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_SYS_225	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_SYS_233	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_SYS_241	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_SYS_249	For manufacturer setting
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	

System status bit [MC300]

Bit No. (constant)	Signal name
SSC_STSBIT_SYS_257	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_SYS_265	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_SYS_273	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_SYS_281	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_SYS_289	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_SYS_297	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_SYS_305	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_SYS_313	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_SYS_321	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_SYS_329	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_SYS_337	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_SYS_345	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_SYS_353	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_SYS_361	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_SYS_369	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_SYS_377	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_AX_SON	Servo on
SSC_CMDBIT_AX_2	For manufacturer setting
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	For manufacturer setting
SSC_CMDBIT_AX_8	

Bit No. (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	For manufacturer setting
SSC_CMDBIT_AX_ORST	Operation alarm reset
SSC_CMDBIT_AX_15	For manufacturer setting
SSC_CMDBIT_AX_16	

Bit No. (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	For manufacturer setting
SSC_CMDBIT_AX_LIP	Linear interpolation mode [MC200]/Interpolation operation mode [MC300]
SSC_CMDBIT_AX_DST	Home position reset mode
SSC_CMDBIT_AX_24	For manufacturer setting

Bit No. (constant)	Signal name
SSC_CMDBIT_AX_25	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_AX_ITL	Interlock
SSC_CMDBIT_AX_RMONR	High speed monitor latch command
SSC_CMDBIT_AX_35	For manufacturer setting
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	For manufacturer setting

Bit No. (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	Change position
SSC_CMDBIT_AX_45	For manufacturer setting
SSC_CMDBIT_AX_46	
SSC_CMDBIT_AX_47	
SSC_CMDBIT_AX_48	

Bit No. (constant)	Signal name
SSC_CMDBIT_AX_FST	Fast start operation
SSC_CMDBIT_AX_50	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_AX_PPISTP	Pass position interrupt cancel
SSC_CMDBIT_AX_58	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_AX_GAIN	Gain switching command
SSC_CMDBIT_AX_FCLS	Fully closed loop control change command
SSC_CMDBIT_AX_67	For manufacturer setting
SSC_CMDBIT_AX_CPC	PID control command
SSC_CMDBIT_AX_69	For manufacturer setting
SSC_CMDBIT_AX_70	
SSC_CMDBIT_AX_71	
SSC_CMDBIT_AX_72	

Bit No. (constant)	Signal name
SSC_CMDBIT_AX_73	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_AX_81	For manufacturer setting
SSC_CMDBIT_AX_82	
SSC_CMDBIT_AX_83	
SSC_CMDBIT_AX_84	
SSC_CMDBIT_AX_ZSC	Home position set command
SSC_CMDBIT_AX_86	For manufacturer setting
SSC_CMDBIT_AX_87	
SSC_CMDBIT_AX_88	

Bit No. (constant)	Signal name
SSC_CMDBIT_AX_89	For manufacturer setting
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	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_AX_97	For manufacturer setting
SSC_CMDBIT_AX_98	
SSC_CMDBIT_AX_99	
SSC_CMDBIT_AX_100	
SSC_CMDBIT_AX_CTLMC	Control mode switch command
SSC_CMDBIT_AX_102	For manufacturer setting
SSC_CMDBIT_AX_103	
SSC_CMDBIT_AX_104	

Bit No. (constant)	Signal name
SSC_CMDBIT_AX_105	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_AX_113	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_AX_GAIN2	Gain switching command 2 [MC300]
SSC_CMDBIT_AX_122	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_AX_MON	Monitor command
SSC_CMDBIT_AX_MONR	Monitor latch command
SSC_CMDBIT_AX_131	For manufacturer setting
SSC_CMDBIT_AX_132	
SSC_CMDBIT_AX_133	
SSC_CMDBIT_AX_134	
SSC_CMDBIT_AX_135	
SSC_CMDBIT_AX_136	

Bit No. (constant)	Signal name
SSC_CMDBIT_AX_137	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_AX_145	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_AX_153	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_AX_PWRT	Parameter write command
SSC_CMDBIT_AX_162	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_AX_PRD	Parameter read command
SSC_CMDBIT_AX_170	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_AX_177	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_AX_185	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_AX_193	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_AX_201	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_AX_209	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_AX_217	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_AX_225	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_AX_233	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_AX_241	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_AX_249	For manufacturer setting
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	

Axis command bit [MC300]

Bit No. (constant)	Signal name
SSC_CMDBIT_AX_257	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_AX_265	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_AX_273	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_AX_281	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_AX_289	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_AX_297	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_AX_305	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_AX_313	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_AX_321	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_AX_329	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_AX_337	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_AX_345	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_AX_353	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_AX_361	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_AX_369	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_AX_377	For manufacturer setting
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 No. (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 erased

Bit No. (constant)	Signal name
SSC_STSBIT_AX_OP	Operation processing
SSC_STSBIT_AX_CPO	Rough match
SSC_STSBIT_AX_PF	Positioning completed
SSC_STSBIT_AX_ZP	Home position return completed
SSC_STSBIT_AX_SMZ	Smoothing stop
SSC_STSBIT_AX_OALM	Operation alarm
SSC_STSBIT_AX_OPF	Operation completed
SSC_STSBIT_AX_PSW	Position switch

Bit No. (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	For manufacturer setting
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	For manufacturer setting

Bit No. (constant)	Signal name
SSC_STSBIT_AX_25	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_AX_ISTP	Interlock stop
SSC_STSBIT_AX_RMRCH	High speed monitor being latched
SSC_STSBIT_AX_POV	Exceeded stop position
SSC_STSBIT_AX_STO	Start up acceptance completed
SSC_STSBIT_AX_37	For manufacturer setting
SSC_STSBIT_AX_38	
SSC_STSBIT_AX_ZREQ	Home position return request
SSC_STSBIT_AX_40	For manufacturer setting

Bit No. (constant)	Signal name
SSC_STSBIT_AX_SCF	Preparation for changing speed completed
SSC_STSBIT_AX_TACF	Preparation for changing acceleration time constant completed
SSC_STSBIT_AX_TDCF*1	Preparation for changing deceleration time constant completed
SSC_STSBIT_AX_PCF	Preparation for changing position completed
SSC_STSBIT_AX_SCE	Speed change error
SSC_STSBIT_AX_TACE*2	Acceleration time constant change error
SSC_STSBIT_AX_TDCE	Deceleration time constant change error
SSC_STSBIT_AX_PCE	Position change error

*1 For API Version 2.00 or earlier, the bit No. is "SSC_STSBIT_AX_TACE". [MC200]
*2 For API Version 2.00 or earlier, the bit No. is "SSC_STSBIT_AX_TDCF". [MC200]

Bit No. (constant)	Signal name
SSC_STSBIT_AX_49	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_AX_PPIOP	Operating pass position interrupt
SSC_STSBIT_AX_PPIFIN	Pass position interrupt completed
SSC_STSBIT_AX_PPIERR	Pass position interrupt incompleted
SSC_STSBIT_AX_60	For manufacturer setting
SSC_STSBIT_AX_61	
SSC_STSBIT_AX_62	
SSC_STSBIT_AX_63	
SSC_STSBIT_AX_AUTLO	In point table loop

Bit No. (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	For manufacturer setting
SSC_STSBIT_AX_70	
SSC_STSBIT_AX_71	
SSC_STSBIT_AX_PRSMO	During continuous operation to torque control

Bit No. (constant)	Signal name
SSC_STSBIT_AX_IWT	Interference check standby
SSC_STSBIT_AX_SINP	Servo amplifier in-position
SSC_STSBIT_AX_75	For manufacturer setting
SSC_STSBIT_AX_76	
SSC_STSBIT_AX_77	
SSC_STSBIT_AX_78	
SSC_STSBIT_AX_79	
SSC_STSBIT_AX_80	

Bit No. (constant)	Signal name
SSC_STSBIT_AX_81	For manufacturer setting
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	For manufacturer setting
SSC_STSBIT_AX_87	
SSC_STSBIT_AX_88	

Bit No. (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 No. (constant)	Signal name
SSC_STSBIT_AX_97	For manufacturer setting
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	For manufacturer setting
SSC_STSBIT_AX_103	
SSC_STSBIT_AX_104	

Bit No. (constant)	Signal name
SSC_STSBIT_AX_105	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_AX_113	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_AX_GAIN2O	Selecting gain switching 2 [MC300]
SSC_STSBIT_AX_122	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_AX_MOUT	Monitor output
SSC_STSBIT_AX_MRCH	Monitor latch
SSC_STSBIT_AX_MER1	Monitor No. error 1
SSC_STSBIT_AX_MER2	Monitor No. error 2
SSC_STSBIT_AX_MER3	Monitor No. error 3
SSC_STSBIT_AX_MER4	Monitor No. error 4
SSC_STSBIT_AX_MESV	Servo amplifier is not connected
SSC_STSBIT_AX_136	For manufacturer setting

Bit No. (constant)	Signal name
SSC_STSBIT_AX_137	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_AX_145	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_AX_153	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_AX_PWFIN1	Parameter write completed 1
SSC_STSBIT_AX_PWEN1	Parameter No. error 1
SSC_STSBIT_AX_PWED1	Parameter data out of bounds 1
SSC_STSBIT_AX_164	For manufacturer setting
SSC_STSBIT_AX_PWFIN2	Parameter write completed 2
SSC_STSBIT_AX_PWEN2	Parameter No. error 2
SSC_STSBIT_AX_PWED2	Parameter data out of bounds 2
SSC_STSBIT_AX_PSCHG	Changes to servo parameters exist

Bit No. (constant)	Signal name
SSC_STSBIT_AX_PRFIN1	Parameter read completed 1
SSC_STSBIT_AX_PREN1	Parameter No. error 1
SSC_STSBIT_AX_PRFIN2	Parameter read completed 2
SSC_STSBIT_AX_PREN2	Parameter No. error 2
SSC_STSBIT_AX_173	For manufacturer setting
SSC_STSBIT_AX_174	
SSC_STSBIT_AX_175	
SSC_STSBIT_AX_176	

Bit No. (constant)	Signal name
SSC_STSBIT_AX_177	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_AX_185	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_AX_193	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_AX_201	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_AX_209	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_AX_217	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_AX_225	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_AX_233	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_AX_241	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_AX_249	For manufacturer setting
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	

Axis status bit [MC300]

Bit No. (constant)	Signal name
SSC_STSBIT_AX_257	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_AX_265	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_AX_273	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_AX_281	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_AX_289	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_AX_297	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_AX_305	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_AX_313	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_AX_321	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_AX_329	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_AX_337	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_AX_345	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_AX_353	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_AX_361	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_AX_369	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_AX_377	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_UT_1	For manufacturer setting
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	For manufacturer setting
SSC_CMDBIT_UT_8	

Bit No. (constant)	Signal name
SSC_CMDBIT_UT_9	For manufacturer setting
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	For manufacturer setting
SSC_CMDBIT_UT_16	

Bit No. (constant)	Signal name
SSC_CMDBIT_UT_17	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_UT_25	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_UT_MON	Monitor command
SSC_CMDBIT_UT_MONR	Monitor latch command
SSC_CMDBIT_UT_35	For manufacturer setting
SSC_CMDBIT_UT_36	
SSC_CMDBIT_UT_37	
SSC_CMDBIT_UT_38	
SSC_CMDBIT_UT_39	
SSC_CMDBIT_UT_40	

Bit No. (constant)	Signal name
SSC_CMDBIT_UT_41	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_UT_PWRT	Parameter write command
SSC_CMDBIT_UT_50	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_UT_PRD	Parameter read command
SSC_CMDBIT_UT_58	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_UT_65	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_UT_73	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_UT_81	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_UT_89	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_UT_97	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_UT_105	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_UT_113	For manufacturer setting
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 No. (constant)	Signal name
SSC_CMDBIT_UT_121	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_UT_RURDY	Receiving controller ready on
SSC_STSBIT_UT_RUA	Outputting DO
SSC_STSBIT_UT_3	For manufacturer setting
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	For manufacturer setting

Bit No. (constant)	Signal name
SSC_STSBIT_UT_9	For manufacturer setting
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	For manufacturer setting
SSC_STSBIT_UT_16	

Bit No. (constant)	Signal name
SSC_STSBIT_UT_17	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_UT_25	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_UT_MOUT	Monitor output
SSC_STSBIT_UT_MRCH	Monitor latch
SSC_STSBIT_UT_MER1	Monitor No. error 1
SSC_STSBIT_UT_MER2	Monitor No. error 2
SSC_STSBIT_UT_MER3	Monitor No. error 3
SSC_STSBIT_UT_MER4	Monitor No. error 4
SSC_STSBIT_UT_MERIO	RIO module is not connected
SSC_STSBIT_UT_40	For manufacturer setting

Bit No. (constant)	Signal name
SSC_STSBIT_UT_41	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_UT_PWFIN1	Parameter write completed 1
SSC_STSBIT_UT_PWEN1	Parameter No. error 1
SSC_STSBIT_UT_PWED1	Parameter data out of bounds 1
SSC_STSBIT_UT_52	For manufacturer setting
SSC_STSBIT_UT_PWFIN2	Parameter write completed 2
SSC_STSBIT_UT_PWEN2	Parameter No. error 2
SSC_STSBIT_UT_PWED2	Parameter data out of bounds 2
SSC_STSBIT_UT_56	For manufacturer setting

Bit No. (constant)	Signal name
SSC_STSBIT_UT_PRFIN1	Parameter read completed 1
SSC_STSBIT_UT_PREN1	Parameter No. error 1
SSC_STSBIT_UT_PRFIN2	Parameter read completed 2
SSC_STSBIT_UT_PREN2	Parameter No. error 2
SSC_STSBIT_UT_61	For manufacturer setting
SSC_STSBIT_UT_62	
SSC_STSBIT_UT_63	
SSC_STSBIT_UT_64	

Bit No. (constant)	Signal name
SSC_STSBIT_UT_65	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_UT_73	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_UT_81	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_UT_89	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_UT_97	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_UT_105	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_UT_113	For manufacturer setting
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 No. (constant)	Signal name
SSC_STSBIT_UT_121	For manufacturer setting
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.

System interrupt

Event factor	Content
SSC_INT_SYS_SYSE	During system error
SSC_INT_SYS_CALM	During system alarm
SSC_INT_SYS_EMIO	During forced stop
SSC_INT_SYS_03	For manufacturer setting
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	For manufacturer setting
SSC_INT_SYS_11	
SSC_INT_SYS_12	
SSC_INT_SYS_13	
SSC_INT_SYS_14	
SSC_INT_SYS_15	

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 erased
SSC_INT_AX_OP	Operation processing
SSC_INT_AX_CPO	Rough match
SSC_INT_AX_PF	Positioning completed
SSC_INT_AX_ZP	Home position return completed
SSC_INT_AX_SMZ	Smoothing stop
SSC_INT_AX_OALM	Operation alarm
SSC_INT_AX_OPF	Operation completed
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	For manufacturer setting
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

Event factor	Content
SSC_INT_AX_26	For manufacturer setting
SSC_INT_AX_27	
SSC_INT_AX_28	
SSC_INT_AX_29	
SSC_INT_AX_30	
SSC_INT_AX_31	

Station interrupt

Event factor	Content
SSC_INT_UT_00	For manufacturer setting
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	For manufacturer setting
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	For manufacturer setting
SSC_INT_UT_15	

8 LIST OF DETAILED ERROR CODES

The following shows the detailed error codes.

Common error

Value	Constant definition	Cause/countermeasure
FFFFFFFFH	SSC_FUNC_ERR_UNKNOWN	No errors (-1) have occurred after using the API functions.
FFFFFFFCH	SSC_FUNC_ERR_UNSUPPORT_BOARD_VERSION	The position board does not support this function.
00000001H	SSC_FUNC_ERR_ARGUMENT_0☐ ☐ = 1 to 9: Argument location	The argument is outside the set range.
00000002H		
00000003H		
00000004H		
00000005H		
00000006H		
00000007H		
00000008H		
00000009H		
00000100H	SSC_FUNC_ERR_ARGUMENT_MISMATCH	The axis No. and the command bit No./status bit No./alarm type do not correspond. <Example> • When "0" is set to the axis No. and "SSC_CMDBIT_AX_SON" is set to the command bit No., etc. • When "0" is set to the axis No. and "SSC_SYSBIT_AX_RDY" is set to the status bit No., etc. • When "0" is set to the axis No. and "SSC_ALARM_OPERATION" is set to the alarm type, etc.
00010000H	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.
00010100H		
00010200H		
00010300H		
00010400H		
00010500H		
00010600H		
00010700H		
00010800H		
00010900H		
00010A00H		
00010B00H		
00010C00H		

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.
00021010H 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. [MC300]
00021011H	SSC_FUNC_ERR_GET_CHANNEL_NUM	The mount channel information could not 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 the API library that combines with the device driver contained in the utility software.
00022000H	SSC_FUNC_ERR_CREATE_SEMAPHORE	An error occurred in the CreateSemaphore function (Windows API). Call the GetLastError function of Windows API and confirm the error details.
00022001H	SSC_FUNC_ERR_DELETE_SEMAPHORE	An error occurred in the CloseHandle function (Windows API). Call the GetLastError function of Windows API and confirm the error details.
00022002H	SSC_FUNC_ERR_WAIT_SEMAPHORE	An error occurred in the WaitForSingleObject function (Windows API). Call the GetLastError function of Windows API and confirm the error details.
00022003H	SSC_FUNC_ERR_RELEASE_SEMAPHORE	An error occurred in the ReleaseSemaphore function (Windows API). Call the GetLastError function of Windows API and confirm the error details.
00022010H	SSC_FUNC_ERR_CREATE_EVENT	An error occurred in the CreateEvent function (Windows API). Call the GetLastError function of Windows API and confirm the error details.
00022011H	SSC_FUNC_ERR_DELETE_EVENT	An error occurred in the CloseHandle function (Windows API). Call the GetLastError function of Windows API and confirm the error details.
00022012H	SSC_FUNC_ERR_RESET_EVENT	An error occurred in the ResetEvent function (Windows API). Call the GetLastError function of Windows API and confirm the error details.
00022013H	SSC_FUNC_ERR_SET_EVENT	An error occurred in the SetEvent function (Windows API). Call the GetLastError function of Windows API and confirm the error details.
00022014H	SSC_FUNC_ERR_WAIT_EVENT	An error occurred in the WaitForSingleObject function (Windows API). Call the GetLastError function of Windows API and confirm the error details.
00022015H	SSC_FUNC_ERR_WAIT_EVENT_MULTI	An error occurred in the WaitForMultipleObjects function (Windows API). Call the GetLastError function of Windows API and confirm the error details.
00022020H	SSC_FUNC_ERR_CREATE_THREAD	An error occurred in the CreateThread function (Windows API). Call the GetLastError function of Windows API and confirm the error details.
00022021H	SSC_FUNC_ERR_DELETE_THREAD	An error occurred in the CloseHandle function (Windows API). Call the GetLastError function of Windows API and confirm the error details.
00022022H	SSC_FUNC_ERR_THREAD_PRIORITY	An error occurred in the SetThreadPriority function (Windows API). Call the GetLastError function of Windows API and confirm the error details.
00022023H	SSC_FUNC_ERR_RESUME_THREAD	An error occurred in the ResumeThread function (Windows API). Call the GetLastError function of Windows API and confirm the error details.
00022024H	SSC_FUNC_ERR_GET_EXIT_CODE_THREAD	An error occurred in the GetExitCodeThread function (Windows API). Call the GetLastError function of Windows API 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.

System functions error

Value	Constant definition	Cause/countermeasure
00030000H	SSC_FUNC_ERR_UNREADY_CHANNEL	The system is in the status other than the 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 signal (FRNG) turned ON.
00030042H	SSC_FUNC_ERR_STS_BIT_FSNG	The flash ROM transfer error signal (FSNG) turned ON.
00030050H	SSC_FUNC_ERR_STS_BIT_RCE	The reconnection error signal (RCE) turned ON.
00030051H	SSC_FUNC_ERR_STS_BIT_CCE	The disconnection error signal (CCE) turned ON.
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.

Parameter functions error

Value	Constant definition	Cause/countermeasure
00040000H 00040001H	SSC_FUNC_ERR_STS_BIT_PREN□ □ = 1 to 2: Array No. of the parameter read Nos. (for 2 Nos.)	A parameter read error occurred. • A value outside the range is set in the parameter read No. • The axis No. and the parameter read No. do not correspond. (Example: When "System parameter" is set to the axis No. and "Axis parameter" is set to the parameter read No., etc.)
00040002H 00040003H	SSC_FUNC_ERR_MISMATCH_PARAM_READ_NUM□ □ = 1 to 2: Array No. of the parameter read Nos. (for 2 Nos.)	The command and the status of the parameter read No. do not correspond.
00040004H 00040005H	SSC_FUNC_ERR_PARAM_READ_EX_NUM□ [MC300] □ = 1 to 2: Array No. of the parameter read Nos. (for 2 Nos.)	The servo parameters of MR-J5(W_)-_B cannot be read.
00040020H 00040021H	SSC_FUNC_ERR_STS_BIT_PWEN□ □ = 1 to 2: Array No. of the parameter write Nos. (for 2 Nos.)	A parameter write No. error occurred. • A value outside the range is set in the parameter write No. • The axis No. and the parameter write No. do not correspond. (Example: When "System parameter" is set to the axis No. and "Axis parameter" is set to the parameter write No., etc.)
00040022H 00040023H	SSC_FUNC_ERR_STS_BIT_PWED□ □ = 1 to 2: Array No. 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 No. of the parameter write Nos. (for 2 Nos.)	The command and the status of the parameter write No. do not correspond.
00040026H 00040027H	SSC_FUNC_ERR_MISMATCH_PARAM_WRITE_DATA□ □ = 1 to 2: Array No. of the parameter write data (for 2 data)	The command and the status of the parameter write data do not correspond.
00040028H 00040029H	SSC_FUNC_ERR_PARAM_WRITE_EX_NUM□ [MC300] □ = 1 to 2: Array No. of the parameter write data (for 2 Nos.)	The servo parameters of MR-J5(W_)-_B cannot be written.
0004002AH	SSC_FUNC_ERR_UNSUPPORT_PARAM_EX [MC300]	The position board does not support the servo parameters of MR-J5(W_)-_B.

Monitor functions error

Value	Constant definition	Cause/countermeasure
00050000H	SSC_FUNC_ERR_STS_BIT_MER□ □ = 1 to 4: Array No. of the monitor Nos. (for 4 Nos.)	A monitor No. error occurred. • A value outside the range is set in the monitor No. • The axis No. and the monitor No. do not correspond. (Example: When "System monitor" is set to the axis No. and "Axis monitor" is set to the monitor No., etc.)
00050001H		
00050002H		
00050003H		
00050004H	SSC_FUNC_ERR_STS_BIT_MESV	The servo information was set as a monitor No. 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.

Axis functions error

Value	Constant definition	Cause/countermeasure
00060010H	SSC_FUNC_ERR_NOW_DRIVING	During operation.
00060011H	SSC_FUNC_ERR_NOW_DRIVING_READY	During the 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	The designated "point No. + point No. offset value" exceeded the point table range.
000600A1H	SSC_FUNC_ERR_STS_BIT_CTLMCE	The control mode switch error signal (CTLMCE) turned ON.
000600A2H	SSC_FUNC_ERR_STS_BIT_IPCH_ON	The changeable interpolation group setting is enabled. Set "0" to the interpolation group No.
000600A3H	SSC_FUNC_ERR_STS_BIT_IPCH_OFF	The changeable interpolation group setting is disabled. Set "1 to 8 [MC200]/1 to 16 [MC300]" to the interpolation group No.
000600A4H	SSC_FUNC_ERR_SUB_AXIS_NUM	When the 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 set 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	The linear interpolation is not in operation.

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 No.
00061021H	SSC_FUNC_ERR_MISMATCH_SMP_PARAM_READ_NUM	The command and the status of the sampling setting read No. do not correspond.
00061022H	SSC_FUNC_ERR_STS_BIT_SWEN	A value outside the range is set in the sampling setting write No.
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 No. 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.

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 signal (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.

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 [MC200]	The interrupt output mask selection (dip switch) is valid.
10000104H	SSC_FUNC_ERR_CLEAR_INT [MC200]	The writing of the interrupt signal clear register is failed.
10000110H	SSC_FUNC_ERR_NOT_START_INT_DRIVER	The interrupt driver is stopped. 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 waited.
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.

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.

Interface mode functions error

Value	Constant definition	Cause/countermeasure
000D0000H	SSC_FUNC_ERR_IFM_INP_OFF	In-position signal (INP) is OFF. When setting "Does not wait until the in-position signal (INP) is ON" during in-position signal (INP) check mode, call the sscIfmSetHomePosition function when the in-position signal (INP) is ON.
000D0001H	SSC_FUNC_ERR_STS_BIT_ZSE	The home position set error signal (ZSE) turned ON.
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 No. has been renewed, call the sscIfmRenewLatestBuffer function or sscIfmRenewLatestBufferEx function.

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.

Transient transmit functions error

Value	Constant definition	Cause/countermeasure
000E1001H	SSC_FUNC_ERR_TRANSIENT_INVALID_DATA	Transient data is invalid.

APPENDIX

Appendix 1 Differences between MR-MC2_ _ and MR-MC3_ _

Differences between API libraries

Differences between the MR-MC2_ _ and MR-MC3_ _ API libraries are shown below.

Item	MR-MC2_ _	MR-MC3_ _
Maximum No. of channels	1	1
Maximum axis No.	32	64
Maximum station No.	4	16
Maximum point No.	320	2048
Maximum point No. offset value	319	2047
Maximum other axes start table No.	32	64
Maximum pass position condition No.	64	128
Maximum interpolation group No.	8	16
I/O bit device No.	0000h to 0FFFh	0000h to 23FFh
I/O word device No.	0000h to 0FFFh	0000h to 23FFh

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.	Page 38 sscGetBoardSerialNumber [MC300]
System functions	sscReconnectSSCNETEx	Reconnects the SSCNET communication.	Page 65 sscReconnectSSCNETEx [MC300]
	sscDisconnectSSCNETEx	Disconnects the SSCNET communication.	Page 67 sscDisconnectSSCNETEx [MC300]
	sscGetControllingAxis	Gets the controlling axis information and controlling station information.	Page 68 sscGetControllingAxis [MC300]
Operating functions	sscInterpolationStart	Starts interpolation operation.	Page 92 sscInterpolationStart [MC300]
Interface mode functions	sscIfmGetEventStatusBitsEx	Gets the status bit information of all axes for the designated status signal using the event detect function.	Page 182 sscIfmGetEventStatusBitsEx [MC300]

A

MR-MC3_ _ replacements for MR-MC2_ _ API functions

- While it is possible to use MR-MC2_ _ API functions shown in the table below with MR-MC3_ _ models, we recommend using MR-MC3_ _ API functions with these models.

Function type	Function name	
	MR-MC2_ _ API functions	MR-MC3_ _ API functions
System functions	sscReconnectSSCNET	sscReconnectSSCNETEx
	sscDisconnectSSCNET	sscDisconnectSSCNETEx
Operating functions	sscLinearStart	sscInterpolationStart

- The following MR-MC2_ _ API functions shown in the table below cannot be used with MR-MC3_ _. Replace these functions with the appropriate MR-MC3_ _ API functions.

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

- The following API functions can be used as per usual, but be sure to reset their settings to "0" before using them since those 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

MEMO

INDEX

T

U

I

REVISIONS

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

Revision date	*Manual number	Description
December 2013	IB(NA)-0300225ENG-A	First edition
December 2014	IB(NA)-0300225ENG-B	■Added models MR-MC240, MR-MC241 ■Added functions 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 (sscIfmGetMaximumBufferNumberEx, sscIfmRenewLatestBufferEx, sscIfmCheckLatestBufferEx, sscIfmGetTransmitBufferEx, sscIfmTrqSetSpeedLimit, sscIfmSetControlMode, sscIfmGetControlMode), PRESS_DATA structure ■Added or modified parts Chapter 2, Section 6.2, 6.3, 6.4, Chapter 7, Chapter 8
July 2015	IB(NA)-0300225ENG-C	■Added functions Point table functions (sscSetLatestPointNumber), Interface mode functions (sscIfmGetEventStatusBits), I/O device functions (sscGetInputDeviceBit, sscGetInputDeviceWord, sscSetOutputDeviceBit, sscSetOutputDeviceWord, sscGetOutputDeviceBit, sscGetOutputDeviceWord), Transient transmit functions (sscSendReceiveTransientData), TRANSIENT_CMD structure, TRANSIENT_STS structure ■Added or modified parts Section 4.4, 4.5, 4.6, 4.11, 4.12, 4.21, 4.22, 5.1, 5.10, 6.2, 6.4, 6.5, 6.6, Chapter 7, Chapter 8
February 2017	IB(NA)-0300225ENG-D	■Added models MR-MC220U3, MR-MC220U6 ■Added or modified parts SAFETY PRECAUTIONS, Section 4.9, 4.10, 4.21, 4.23, 4.24, 5.1, 5.2, 6.2, Chapter 8, Warranty
September 2017	IB(NA)-0300225ENG-E	■Added or modified parts Chapter 2
March 2018	IB(NA)-0300225ENG-F	■Added models MR-MC341 ■Added functions Information functions (sscGetBoardSerialNumber), System functions (sscReconnectSSCNETEx, sscDisconnectSSCNETEx, sscGetControllingAxis), Operating functions (sscInterpolationStart), Interface mode functions (sscIfmGetEventStatusBitsEx), SLAVE_INFO structure ■Added or modified parts INTRODUCTION, Chapter 2, Section 4.2, 4.4, 4.5, 4.7, 4.9, 4.10, 4.22, 5.1, 5.2, 5.4, 5.5, 5.10, 6.1, 6.2, 6.3, 6.4, Chapter 8
September 2018	IB(NA)-0300225ENG-G	■Added or modified parts Section 4.9, 5.1, Appendix 1
June 2022	IB(NA)-0300225ENG-H	■Added or modified parts Chapter 2
June 2024	IB(NA)-0300225ENG-J	■Added functions Parameter Functions (sscChangeParameterEx, sscChange2ParameterEx, sscCheckParameterEx, sscCheck2ParameterEx) ■Added or modified parts SAFETY PRECAUTIONS, TERMS, GENERIC TERMS AND ABBREVIATIONS, Chapter 1, Chapter 2, Chapter 3, Section 4.3, 4.4, 4.5, 4.6, 4.9, 4.10, 4.16, 4.18, 4.19, 4.20, 4.21, 4.22, 4.23, 4.24, 5.1, 5.7, 5.10, 6.1, 6.2, 6.3, 6.4, Chapter 8, INDEX, INFORMATION AND SERVICES, TRADEMARKS

Japanese manual number: IB-0300224-K

WARRANTY

Warranty

1. Warranty period and coverage

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.

[Term]

For terms of warranty, please contact your original place of purchase.

[Limitations]

(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, fan, smoothing capacitor, 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. Term of warranty after the stop 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. Application and use of the Product

(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 the position board, and a backup or fail-safe function should operate on an external system to the 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.

(3) Mitsubishi shall have no responsibility or liability for any problems involving programmable controller trouble and system trouble caused by DoS attacks, unauthorized access, computer viruses, and other cyberattacks.

INFORMATION AND SERVICES

For further information and services, please contact your local Mitsubishi Electric sales office or representative.
Visit our website to find our locations worldwide.

MITSUBISHI ELECTRIC Factory Automation Global Website
Locations Worldwide
www.MitsubishiElectric.com/fa/about-us/overseas/

TRADEMARKS

IB(NA)-0300225ENG-J(2406)MEE
MODEL: MRMC2-U-API-E
MODEL CODE: 1XB970

MITSUBISHI ELECTRIC CORPORATION

HEAD OFFICE: TOKYO BLDG., 2-7-3, MARUNOUCHI, CHIYODA-KU, TOKYO 100-8310, JAPAN
NAGOYA WORKS: 1-14, YADA-MINAMI 5-CHOME, HIGASHI-KU, NAGOYA 461-8670, JAPAN