

Personal Computer Embedded Type Servo System Controller

Motion Control Board User's Manual (Motion API)

-MR-EM441G



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	<ul style="list-style-type: none">● 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 Motion control board. Failure to do so may result in an accident due to an incorrect output or malfunction.<ol style="list-style-type: none">(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 Motion control board.(2) When the Motion control board detects an abnormal condition, it stops the operation and all outputs are:<ul style="list-style-type: none">• Held the parameter setting if the self-diagnostic function of the Motion control board detects an error such as a watchdog timer error.(3) All outputs may be turned on if an error occurs in a part, such as an I/O control part, where the Motion control board cannot detect any error. To ensure safety operation in such a case, provide a safety mechanism or a fail-safe circuit external to the Motion control board.(4) Outputs may remain on or off, or the output status may become undefined due to a failure of a component such as a relay and transistor in an output circuit. Configure an external circuit for monitoring output signals that could cause a serious accident.● In an output circuit, when a load current exceeding the rated current or an overcurrent caused by a load short-circuit flows for a long time, it may cause smoke and fire. To prevent this, configure an external safety circuit, such as a fuse.● Configure a circuit so that the Motion control 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.● Configure a circuit so that the external power supply is turned off first and then turned off the Motion control board or reboot the software. If the Motion control board is turned off first, an accident may occur due to an incorrect output or malfunction.
--------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

[Design Precautions]

WARNING

- For the operating status of each station after a communication failure, refer to manuals for the network 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 Motion control board to modify data of a running Motion control 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, forced output, or operating status change) of a running Motion control 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 Motion control board is controlled by an external device, immediate action cannot be taken if a problem occurs in the Motion control 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 Motion control board in case of a communication failure.
 - Do not write any data to the "system area" or "area for manufacturer setting" of the buffer memory in the Motion control board. Also, do not use any "use prohibited" signals as an output signal from the Motion control board to each module. Doing so may cause malfunction of the Motion control board system. For the "system area", "area for manufacturer setting", and the "use prohibited" signals, refer to the user's manual, this manual, and the relevant manuals for the module used.
 - 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 Motion control 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 Motion control board.
 - (2) When the Motion control 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 Motion control 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 Motion control board, module, and drive unit, and servo motor, make sure that the safety standards are satisfied.
 - Construct a safety circuit externally of the Motion control board, module, or drive unit if the abnormal operation of the module or drive unit 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 Motion control board is powered on or the software is rebooted, the time taken to enter the system startup status varies depending on the system configuration, parameter settings, and/or program size. Design circuits so that the entire system will always operate safely, regardless of the time.
 - Do not power off the Motion control 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 Motion control board.
-

[Security Precautions]

WARNING

- To maintain the security (confidentiality, integrity, and availability) of the Motion control 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 Motion control board. Failure to do so may result in electric shock or cause the Motion control board to fail or malfunction.
-

[Installation Precautions]

CAUTION

- Use the Motion control board in an environment that meets the general specifications in the Motion Control Board User's Manual (Motion Control). 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 Motion control 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 Motion control 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 Motion control board or connector. Hold the front panel or edge of the print board. Doing so can cause malfunction or failure of the Motion control board.
 - Do not disassemble or modify the Motion control board. Doing so may cause failure, malfunction, injury, or a fire.
 - Before handling the Motion control 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 Motion control board to fail or malfunction.
 - Install the Motion control board to a personal computer which is compliant with PCI Express® standard. Failure to do so may cause a failure or malfunction.
 - Securely insert the Motion control board into the slot following the board installation instruction of the personal computer. Incorrect insertion of the Motion control board may cause malfunction, failure, or drop of the board.
 - When installing the Motion control board, take care not to contact with other boards.
 - When installing the Motion control board, take care not to get injured by an implemented component or a surrounding member.
 - Handle the Motion control board in a place where static electricity will not be generated. Failure to do so may cause a failure or malfunction.
 - The Motion control 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 Motion control 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 Motion control board.
 - After installation and wiring, attach the cover of the equipment the Motion control 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 Motion control 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 Motion control 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 Motion control 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 Motion control board in a duct or clamp them. If not, dangling cables may swing or inadvertently be pulled, resulting in malfunction or damage to the Motion control board or cables.

In addition, the weight of the wires and the cables may put stress on the Motion control 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 Motion control 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 Motion control board, resulting in drop, short circuit, fire, or malfunction.
 - When disconnecting the cable from the Motion control 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 Motion control board may result in malfunction or damage to the Motion control board or the cable.
 - Prevent foreign matter such as dust or wire chips from entering the personal computer embedded with a Motion control board. Such foreign matter can cause a fire, failure, or malfunction.
 - For Ethernet cables to be used in the system, select the ones that meet the specifications in the Motion Control Board User's Manual (Motion Control). 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 Motion control board or an intelligent function module to modify data of a running Motion control 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 Motion control 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 Motion control board is controlled by an external device, immediate action cannot be taken if a problem occurs in the Motion control 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 Motion control 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 Motion control 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 Motion control 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 Motion control 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 Motion control 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 Motion control 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 Motion control 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 Motion control board will reach a high temperature during operation. Do not touch the heat sink directly when replacing the Motion control board. Doing so may result in a burn.
-

[Operating Precautions]

CAUTION

- When changing data and operating status, and modifying program of the running Motion control 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 Motion control board or reboot the software while the setting values in the buffer memory are being written to the flash ROM in the Motion control board. 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 or failure of the Motion control 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 Motion Control Board User's Manual (Motion Control). If not, normal data transmission is not guaranteed.
-

[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 Motion control board is a precision machine, so do not drop or apply strong impacts on it.
-

INTRODUCTION

Thank you for purchasing the personal computer embedded type servo system controllers.

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 personal computer embedded type servo system controller to handle the product correctly.

Please make sure that the end users read this manual.

Relevant product

MR-EM441G

CONTENTS

SAFETY PRECAUTIONS	1
INTRODUCTION	9
RELEVANT MANUALS	16
TERMS	17
GENERIC TERMS AND ABBREVIATIONS	18
CHAPTER 1 OVERVIEW	19
1.1 CONDITIONS FOR USE	20
CHAPTER 2 LIST OF API FUNCTIONS	23
CHAPTER 3 API FUNCTION DETAILS	28
3.1 Support Functions	28
sscGetLastError	28
sscGetMountChannel	29
3.2 Device Functions	30
sscOpen	31
sscClose	32
3.3 Information Functions	33
sscGetControlCycle	33
sscGetBoardVersionEx	34
sscGetOperationCycleMonitorEx	35
sscGetBoardSerialNumber	36
sscGetBoardSerialNumberW	37
sscGetMotionApiVersion	38
3.4 Parameter Functions	39
sscChangeParameter	40
sscChange2Parameter	42
sscCheckParameter	44
sscCheck2Parameter	45
3.5 System Functions	47
sscReboot	48
sscSystemStart	49
sscGetSystemStatusCode	50
sscGetSystemStatusCodeEx	51
sscGetControllingAxis	52
3.6 Command/Status Functions	53
sscSetCommandBitSignalEx	54
sscGetStatusBitSignalEx	55
sscWaitStatusBitSignalEx	56
sscGetStatusBits	57
sscCheckBitStatus	58
3.7 Point Table Functions	59
sscSetPointDataEx	59
sscCheckPointDataEx	60
sscSetPointOffset	61
sscCheckPointOffset	62
sscGetDrivingPointNumber	63

	sscSetLatestPointNumber	64
3.8	Continuous Operation to Torque Control Data Functions	65
	sscSetPressData	65
	sscGetPressData	66
	sscSetPressTargetTorque	67
	sscGetPressTargetTorque	68
3.9	Operating Functions	69
	sscJogStart	69
	sscJogStop	71
	sscJogStopNoWait	72
	sscIncStart	73
	sscAutoStart	75
	sscHomeReturnStart	76
	sscLinearStart	77
	sscInterpolationStart	79
	sscDataSetStart	81
	sscDriveStop	82
	sscDriveStopNoWait	83
	sscDriveRapidStop	84
	sscDriveRapidStopNoWait	85
	sscSetDriveMode	86
	sscGetDriveMode	88
	sscGetDriveFinStatus	89
	sscChangeControlMode	91
3.10	Change Functions	92
	sscChangeManualPosition	92
	sscChangeAutoPosition	93
	sscChangeLinearPosition	94
	sscChangeManualSpeed	96
	sscChangeAutoSpeed	97
	sscChangeManualAccTime	98
	sscChangeAutoAccTime	99
	sscChangeManualDecTime	100
	sscChangeAutoDecTime	101
3.11	Alarm Functions	102
	sscGetAlarm	102
	sscResetAlarm	103
	sscGetSystemErrorCode	104
	sscResetAllError	105
3.12	General Monitor Functions	106
	sscSetMonitorEm	108
	sscStopMonitor	109
	sscGetMonitorEm	110
3.13	High Speed Monitor Functions	111
	sscGetCurrentCmdPositionFast	111
	sscGetCurrentFbPositionFast	112
	sscGetIoStatusFast	113
	sscGetCmdSpeedFast	114
	sscGetFbSpeedFast	115
	sscGetCurrentFbFast	116
3.14	User Watchdog Functions	117

sscWdEnable	117
sscWdDisable	118
sscChangeWdCounter	119
sscCheckPCleBusConnection	120
3.15 Other Axes Start Functions	121
sscSetOtherAxisStartData	121
sscGetOtherAxisStartData	122
sscOtherAxisStartAbortOn	123
sscOtherAxisStartAbortOff	124
sscGetOtherAxisStartStatus	125
3.16 Pass Position Interrupt Functions	126
sscSetIntPassPositionData	126
sscCheckIntPassPositionData	127
sscSetStartingPassNumber	128
sscGetExecutingPassNumber	129
3.17 Sampling Functions	130
sscStartSampling	130
sscStopSampling	131
sscSetSamplingParameter	132
sscGetSamplingParameter	133
sscGetSamplingError	134
sscGetSamplingStatus	135
sscGetSamplingDataEm	136
3.18 Interrupt Functions	137
sscIntStart	139
sscIntEnd	140
sscIntEnable	141
sscIntDisable	142
sscRegisterIntCallback	143
sscUnregisterIntCallback	144
sscResetIntEvent	145
sscSetIntEvent	146
sscWaitIntEvent	147
sscResetIntOasEvent	149
sscSetIntOasEvent	150
sscWaitIntOasEvent	151
sscResetIntPassPosition	153
sscSetIntPassPosition	154
sscWaitIntPassPosition	155
sscResetIntDriveFin	157
sscSetIntDriveFin	158
sscWaitIntDriveFin	159
3.19 I/O Device Functions	161
sscGetInputDeviceBit	161
sscGetInputDeviceWord	162
sscGetInputDeviceDword	163
sscSetOutputDeviceBit	164
sscSetOutputDeviceBitNonExclusively	165
sscSetOutputDeviceWord	166
sscSetOutputDeviceDword	167
sscChangeOutputDeviceWord	168

sscChangeOutputDeviceDword	169
sscGetOutputDeviceBit	170
sscGetOutputDeviceWord	171
sscGetOutputDeviceDword	172
3.20 Initialization/Finalization Functions	173
sscInitializeLibrary	174
sscFinalizeLibrary	175
3.21 Link Device Access Functions	176
sscGetBitLinkDevice	176
sscGetWordLinkDevice	178
sscSetBitLinkDevice	179
sscSetWordLinkDevice	180
3.22 File Operation Functions	181
sscWriteFile	182
sscReadFile	184
sscDeleteFile	186
sscCreateDirectory	187
sscDeleteDirectory	189
sscGetFileList	190
sscGetFileListEx	192
sscFreeFileList	194
sscFreeFileListEx	195
3.23 SLMP Functions	196
sscSmpReadSlaveObject	196
sscSmpWriteSlaveObject	198
sscSmpSendCommand	200
3.24 General Input/Output Functions	203
sscGetGeneralInputDataBit	203
sscGetGeneralInputDataWord	204
sscSetGeneralOutputDataBit	205
sscSetGeneralOutputDataBitExclusively	206
sscSetGeneralOutputDataWord	207
sscSetGeneralOutputDataWordExclusively	208
sscGetGeneralOutputDataBit	209
sscGetGeneralOutputDataWord	210
3.25 Arbitrary User Area Access Functions	211
sscSetUserArea	211
sscGetUserArea	212
3.26 Event History Functions	213
sscReadEventHistoryData	213
sscClearEventHistoryData	214
3.27 Pressure Control Functions	215
sscSetPrsProfile	215
sscGetPrsProfile	216
sscGetPrsStatus	217
sscGetExecutingPrsNumber	218
sscGetPrsForwardSidePosition	219
sscGetPrsSpeedLimit	220
sscGetPrsPressureCommand	221
sscGetPrsLoadCellPressure	222
sscGetPrsError	223

sscSetPrsProfileLimitPosition	224
sscGetPrsProfileLimitPosition	225
sscSetPrsProfileSpeedLimit	226
sscGetPrsProfileSpeedLimit	227
sscSetPrsProfilePressureCommand	228
sscGetPrsProfilePressureCommand	229

CHAPTER 4 STRUCTURE LIST 230

4.1 PNT_DATA_EX Structure	230
4.2 OAS_DATA Structure	232
4.3 PRESS_DATA Structure	234
4.4 SMP_ERR Structure	235
4.5 SMP_DATA_EM Structure	236
4.6 INT_CB_DATA Structure	237
4.7 SLAVE_INFO Structure	238
4.8 FILE_INFO Structure	239
4.9 FILE_INFO_EX Structure	240
4.10 FILE_TIME Structure	241
4.11 SLMP_SLV_OBJ_CMD Structure	242
4.12 SLMP_SLV_OBJ_STS Structure	243
4.13 SLMP_CONTROL Structure	244
4.14 PRS_PROFILE Structure	246
PRS_SWITCH_TO_PRESSURE structure	247
PRS_CMN_SETTING structure	248
PRS_FEED_DATA structure	249
PRS_FEED_STEP structure	250
PRS_SWITCH_TO_DWELL structure	251
PRS_DWELL_DATA structure	252
PRS_DWELL_STEP structure	253
PRS_RELEASE_DATA structure	254
PRS_SWITCH structure	255

CHAPTER 5 BIT DEFINITION LIST 257

5.1 System Command Bit	257
5.2 System Status Bit	266
5.3 Axis Command Bit	274
5.4 Axis Status Bit	287

CHAPTER 6 INTERRUPT EVENT FACTOR LIST 300

CHAPTER 7 LIST OF DETAILED ERROR CODES 302

APPENDIX 308

Appendix 1 Differences between MR-MC341 and MR-EM441G	308
Differences between Motion API/MR-EM441G	308
API functions added to the MR-EM441G Motion API	308
MR-EM441G replacements for MR-MC341 API functions	311

INDEX 313

REVISIONS	316
-----------------	-----

WARRANTY317

INFORMATION AND SERVICES318

TRADEMARKS318

RELEVANT MANUALS

Manual name [manual number]	Description	Available form
Motion Control Board User's Manual (Motion API) [IB-0300601ENG] (This manual)	API functions to control the Motion control board from the host personal computer	Print book e-Manual PDF
Motion Control Board User's Manual (Motion Control) [IB-0300599ENG]	Specifications, procedures before operation, system configuration, wiring, functions, table maps, parameters, and troubleshooting of the Motion control board	Print book e-Manual PDF
Motion Control Board User's Manual (Network) [IB-0300600ENG]	Functions, parameter settings, troubleshooting, and buffer memory of CC-Link IE TSN	Print book e-Manual PDF



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 Motion API for the Motion control board.
Axis	A motion control target, referring to a station with axis No. allocation. The Motion control board exchanges the data of cyclic transmission.
Board version	System version of the Motion control board.
Buffer memory	Memory to store data such as setting values and monitor values. The Motion control board also uses it for data communication with the user program. The buffer memory No. (indicating the buffer memory address) given in this manual is prefixed with "G". The buffer memory No. is given on a 1-word basis.
Cyclic data	Data that are sent by cyclic transmission.
Cyclic transmission	A function by which data is periodically exchanged among stations on the same network.
Device station	A station (local or remote station) that is connected on CC-Link IE TSN excluding the master station.
Event history	Event data groups which are saved in a Motion control board.
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	An internal device (RX/Ry/RWw/LB/LW) of the Motion control board and the device station. The station set to the motion control station uses the I/O device (RX/Ry/RWw/RWw) on the dual port memory as a link device.
Local station	A station that performs cyclic transmission and transient transmission with the master station and other local stations. The local station can receive cyclic data of other stations in the multicast mode.
Master station	A station that controls the entire network. This station can perform cyclic transmission and transient transmission with all stations. Only one master station can be used in a network.
Motion control board	Another term for MR-EM441G.
Motion control station	A device station that exchanges cyclic data by an I/O device and motion control.
MR Configurator2	A product name of servo setup software.
MR-EM441G	Another term for the Motion control board connectable to CC-Link IE TSN.
MR-MC200	Another term for the following position boards: Compatible with PCI bus MR-MC210/MR-MC211/ Compatible with CompactPCI bus MR-MC220U3/MR-MC220U6/ Compatible with PCI Express bus MR-MC240/MR-MC241.
MR-MC300	Another term for the position board compatible with PCI Express bus MR-MC341.
Multicast mode	A communication mode used to send cyclic data to multiple stations.
Object	Various data of a device station compatible with CANopen.
Remote I/O	An I/O device or a link device control target, referring to a station without axis No. allocation (a station not used as an axis). The Motion control board exchanges the data of cyclic transmission.
Remote station	A station that exchanges I/O signals (bit data) and I/O data (word data) with another station by cyclic transmission. This station can perform transient transmission.
SLMP	A SeamLess Message Protocol. This protocol is used to access an SLMP-compatible device or a programmable controller connected to an SLMP-compatible device from an external device.
Standard station	A device station other than the motion control station.
Transient transmission	A function of data communication unperiodically among nodes (station) on network.
User program	A user created program (by using the Motion API) which is operated by the host personal computer.

GENERIC TERMS AND ABBREVIATIONS

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

Generic term/abbreviation	Description
Data link	A generic term for cyclic transmission and transient transmission.
Device	A generic term for a device which is connected with a Motion control board via CC-Link IE TSN.
Drive unit	A generic term for motor drive devices such as a servo amplifier.
Dual port memory	A generic term for a communication area in order to execute especially the motion control in the buffer memory. The dual port memory address is the offset value from the start address of the dual port memory (buffer memory No.G12000000). Also, the dual port memory address is noted in byte unit.
EM Motion SDK	An abbreviation for the MELSOFT EM Motion SDK which is the software development kit for motion control.
Engineering tool	A generic term for the motion test tool and MR Configurator2.
Host personal computer	A generic term for the personal computer mounted with the Motion control board on which the user program for controlling the Motion control board operates.
I/O device	A generic term for the link device on the dual port memory. It is used at the station set as the motion control station.
LB	An abbreviation for a link relay of a link device. Bit data sent from each station of the network.
LW	An abbreviation for a link register of a link device. 16-bit (1-word) data sent from each station of the network.
Module	A generic term for devices of the master station (Motion control board) and device stations.
Motion API	A generic term for library functions to control the Motion control board from the host personal computer.
Motion test tool	An abbreviation for the start-up and examination tool for Motion control board.
RW _r	An abbreviation for a remote register of the link device. 16-bit (1-word) data input from a device station to the master station. (Not applicable to some local stations.)
RW _w	An abbreviation for a remote register of the link device. 16-bit (1-word) data output from the master station to a device station. (Not applicable to some local stations.)
RX	An abbreviation for a remote input of the link device. Bit data input from a device station to the master station. (Not applicable to some local stations.)
RY	An abbreviation for a remote output of the link device. Bit data output from the master station to a device station. (Not applicable to some local stations.)
SB	An abbreviation for a link special relay. Bit data that indicates the operating status and data link status of a module on CC-Link IE.
Servo amplifier	A generic term for servo amplifiers connectable to CC-Link IE TSN.
SW	An abbreviation for a link special register. 16-bit (1-word) data that indicates the operating status and data link status of a module on CC-Link IE.

1 OVERVIEW

The Motion API is a collection of API functions for creating applications on the host personal computer which control the Motion control board (MR-EM441G).

By using the Motion API, it is possible to open and close communication with the Motion control board, initialize a communication with the servo amplifier, change parameters, start operations in each operating mode, and perform monitoring.


When using the Motion API, up to 4 Motion control boards can be used simultaneously.

Functional limitation depending on software version

Available functions are limited depending on the software version of this motion API.

Function/Item name	API version	Board version	Change details
User watchdog	02	1.10	The function is added.
Interrupt callback	01	1.10	The function is added.
Confirmation of PCIe bus connection status	01	1.20	The function is added.
Creation/deletion of directory	01	1.20	The function is added.
SLMP message send/receive	01	1.20	The function is added.
Arbitrary user area access functions	01	1.20	The function is added.
Event history functions	03	1.20	The function is added.
Other axes start DO output compatible	05	1.30	The function is added.
Local station compatible	05	1.30	The function is added.
Windows11 compatible	05	1.30	The supported operating system is added.
Continuous operation to torque control compatible	05	1.30	The function is added.
Pressure control compatible	05	1.30	The function is added.

Point

- The API version can be checked by the file version in the property dialog of the files below.
em4xxstd.dll or em4xxstd_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.
 Motion Control Board User's Manual (Motion Control)

1.1 CONDITIONS FOR USE

The following conditions when using the Motion API apply.

- The Motion API is assumed to be used with a compiler that runs on the following operating systems.

Item	Description
OS	<ul style="list-style-type: none">• Windows® 10 Enterprise (32-bit/64-bit)• Windows® 10 Pro (32-bit/64-bit)• Windows® 10 IoT Enterprise (64-bit)*1• Windows® 11 Enterprise (64-bit)*1• Windows® 11 Pro (64-bit)*1• Windows® 11 IoT Enterprise (64-bit)*1
Compiler	<ul style="list-style-type: none">• Microsoft® Visual C++® 2022/2019/2017/2015/2013/2012• Microsoft® Visual C#® 2022/2019/2017/2015/2013/2012

*1 The 32-bit version is not supported.

- The Motion API provides the following library and header file.

Library		Header file
32-bit OS compatible	64-bit OS compatible	
<ul style="list-style-type: none">• em4xxstd.dll• em4xxstd.lib (COFF format)	<ul style="list-style-type: none">• em4xxstd_x64.dll• em4xxstd_x64.lib (COFF format)	<ul style="list-style-type: none">• em4xxstd.h (header file for C/C++)• em4xxstd.cs (header file for C#)

- The Motion API supports the following.

Bus specification	Model
PCI Express bus compatible Motion control board	MR-EM441G

- When using the Motion API by the Visual C# project, execute the following.
 - Store the Motion API (.dll) in the same folder as the user program (.exe) that uses the Motion API. (Since the Motion API 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 Motion API 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 Motion control board by running two or more user programs.

Point

- Use `__stdcall` as the calling convention of the Motion API.
- The following operating system and library combinations are not supported.
 - 32-bit operating system + 64-bit operating system compatible library (em4xxstd_x64.dll)
 - 64-bit operating system + 32-bit operating system compatible library (em4xxstd.dll)
- Be sure to test the user program thoroughly when incorporating with the user equipment.
- The MR-EM441G library cannot be used together on one user program.

Visual C# project

The SscApi class of the em4xxstd namespace defines the API.

When using the Motion API 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_EM* (array head pointer)		SMP_DATA_EM[] Example: SMP_DATA_EM[] smp_data = new SMP_DATA_EM[128]; for (int i = 0; i < 128; i++) { smp_data[i].Initialize(); } SscApi.sscGetSamplingDataEm(board_id, channel, page_num, out valid_num, smp_data);

2 LIST OF API FUNCTIONS

The list of the API functions of MR-EM441G is shown below.

Function type	Function name	Function content	Reference section
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 the operation cycle monitor (setting cycle).	Page 33 sscGetControlCycle
	sscGetBoardVersionEx	Gets the Motion control board version information.	Page 34 sscGetBoardVersionEx
	sscGetOperationCycleMonitorEx	Gets the operation cycle monitor data.	Page 35 sscGetOperationCycleMonitorEx
	sscGetBoardSerialNumber	Gets the Motion control board serial No.	Page 36 sscGetBoardSerialNumber
	sscGetBoardSerialNumberW	Gets the Motion control board serial No. (Multi byte characters)	Page 37 sscGetBoardSerialNumberW
	sscGetMotionApiVersion	Gets the Motion API version.	Page 38 sscGetMotionApiVersion
Parameter functions	sscChangeParameter	Writes the parameter.	Page 40 sscChangeParameter
	sscChange2Parameter	Writes the parameters (for 2 parameters).	Page 42 sscChange2Parameter
	sscCheckParameter	Reads the parameter set value.	Page 44 sscCheckParameter
	sscCheck2Parameter	Reads the parameter set values (for 2 parameters).	Page 45 sscCheck2Parameter
System functions	sscReboot	Reboots the system.	Page 48 sscReboot
	sscSystemStart	Starts the system.	Page 49 sscSystemStart
	sscGetSystemStatusCode	Gets the system status code.	Page 50 sscGetSystemStatusCode
	sscGetSystemStatusCodeEx		Page 51 sscGetSystemStatusCodeEx
	sscGetControllingAxis	Gets the controlling axis information and the controlling station information.	Page 52 sscGetControllingAxis
Command/ status functions	sscSetCommandBitSignalEx	Arbitrarily sets the command bit.	Page 54 sscSetCommandBitSignalEx
	sscGetStatusBitSignalEx	Arbitrarily gets the status bit.	Page 55 sscGetStatusBitSignalEx
	sscWaitStatusBitSignalEx	Waits until the specified status bit turns ON/OFF.	Page 56 sscWaitStatusBitSignalEx
	sscGetStatusBits	Gets the status bit all at once.	Page 57 sscGetStatusBits
	sscCheckBitStatus	Gets the status bit specified from the status bit array.	Page 58 sscCheckBitStatus
Point table functions	sscSetPointDataEx	Sets the point data.	Page 59 sscSetPointDataEx
	sscCheckPointDataEx	Gets the point data.	Page 60 sscCheckPointDataEx
	sscSetPointOffset	Sets the point No. offset.	Page 61 sscSetPointOffset
	sscCheckPointOffset	Gets the point No. offset.	Page 62 sscCheckPointOffset
	sscGetDrivingPointNumber	Gets the operation point No.	Page 63 sscGetDrivingPointNumber
	sscSetLatestPointNumber	Sets the latest command point No.	Page 64 sscSetLatestPointNumber
Continuous operation to torque control data functions	sscSetPressData	Sets the continuous operation to torque control data.	Page 65 sscSetPressData
	sscGetPressData	Gets the continuous operation to torque control data.	Page 66 sscGetPressData
	sscSetPressTargetTorque	Sets the target torque of continuous operation to torque control data.	Page 67 sscSetPressTargetTorque
	sscGetPressTargetTorque	Gets the target torque of continuous operation to torque control data.	Page 68 sscGetPressTargetTorque

Function type	Function name	Function content	Reference section
Operating functions	sscJogStart	Starts the JOG operation.	Page 69 sscJogStart
	sscJogStop	Stops the JOG operation.	Page 71 sscJogStop
	sscJogStopNoWait	Stops the JOG operation. (No wait function)	Page 72 sscJogStopNoWait
	sscIncStart	Starts the incremental feed.	Page 73 sscIncStart
	sscAutoStart	Starts the automatic operation.	Page 75 sscAutoStart
	sscHomeReturnStart	Starts the home position return.	Page 76 sscHomeReturnStart
	sscLinearStart	Starts the linear interpolation.	Page 77 sscLinearStart
	sscInterpolationStart	Starts the interpolation operation.	Page 79 sscInterpolationStart
	sscDataSetStart	Starts the home position reset (data set).	Page 81 sscDataSetStart
	sscDriveStop	Stops the operations.	Page 82 sscDriveStop
	sscDriveStopNoWait	Stops the operations. (No wait function)	Page 83 sscDriveStopNoWait
	sscDriveRapidStop	Stops the operations rapidly.	Page 84 sscDriveRapidStop
	sscDriveRapidStopNoWait	Stops the operations rapidly. (No wait function)	Page 85 sscDriveRapidStopNoWait
	sscSetDriveMode	Switches the operation mode.	Page 86 sscSetDriveMode
	sscGetDriveMode	Gets the operation mode.	Page 88 sscGetDriveMode
	sscGetDriveFinStatus	Gets the operation completion status.	Page 89 sscGetDriveFinStatus
Change functions	sscChangeControlMode	Switches the control mode of the servo amplifier.	Page 91 sscChangeControlMode
	sscChangeManualPosition	Changes the position during the incremental feed.	Page 92 sscChangeManualPosition
	sscChangeAutoPosition	Changes the position during the automatic operation.	Page 93 sscChangeAutoPosition
	sscChangeLinearPosition	Changes the position during the linear interpolation.	Page 94 sscChangeLinearPosition
	sscChangeManualSpeed	Changes the speed in the JOG operation or the incremental feed.	Page 96 sscChangeManualSpeed
	sscChangeAutoSpeed	Changes the speed in the automatic operation or the linear interpolation.	Page 97 sscChangeAutoSpeed
	sscChangeManualAccTime	Changes the acceleration time constant in the JOG operation or the incremental feed.	Page 98 sscChangeManualAccTime
	sscChangeAutoAccTime	Changes the acceleration time constant in the automatic operation or the linear interpolation.	Page 99 sscChangeAutoAccTime
	sscChangeManualDecTime	Changes the deceleration time constant in the JOG operation or the incremental feed.	Page 100 sscChangeManualDecTime
Alarm functions	sscChangeAutoDecTime	Changes the deceleration time constant in the automatic operation or the linear interpolation.	Page 101 sscChangeAutoDecTime
	sscGetAlarm	Gets the alarm No. and the specific alarm No.	Page 102 sscGetAlarm
	sscResetAlarm	Resets the alarm.	Page 103 sscResetAlarm
	sscGetSystemErrorCode	Gets the system error code.	Page 104 sscGetSystemErrorCode
General monitor functions	sscResetAllError	Executes the all error reset.	Page 105 sscResetAllError
	sscSetMonitorEm	Starts the monitoring.	Page 108 sscSetMonitorEm
	sscStopMonitor	Stops the monitoring.	Page 109 sscStopMonitor
High speed monitor functions	sscGetMonitorEm	Gets the monitoring data.	Page 110 sscGetMonitorEm
	sscGetCurrentCmdPositionFast	Gets the current command position. (High speed monitor function)	Page 111 sscGetCurrentCmdPositionFast
	sscGetCurrentFbPositionFast	Gets the current feedback position. (High speed monitor function)	Page 112 sscGetCurrentFbPositionFast
	sscGetIoStatusFast	Gets the external signal status. (High speed monitor function)	Page 113 sscGetIoStatusFast
	sscGetCmdSpeedFast	Gets the moving speed. (High speed monitor function)	Page 114 sscGetCmdSpeedFast
	sscGetFbSpeedFast	Gets the feedback moving speed. (High speed monitor function)	Page 115 sscGetFbSpeedFast
User watchdog functions	sscGetCurrentFbFast	Gets the electrical current feedback. (High speed monitor function)	Page 116 sscGetCurrentFbFast
	sscWdEnable	Enables the user watchdog function.	Page 117 sscWdEnable
	sscWdDisable	Disables the user watchdog function.	Page 118 sscWdDisable
	sscChangeWdCounter	Updates the watchdog check counter.	Page 119 sscChangeWdCounter
	sscCheckPCleBusConnection	Confirms the PCIe bus connection status.	Page 120 sscCheckPCleBusConnection

Function type	Function name	Function content	Reference section
Other axes start functions	sscSetOtherAxisStartData	Sets the data for starting other axes.	Page 121 sscSetOtherAxisStartData
	sscGetOtherAxisStartData	Gets the data for starting other axes.	Page 122 sscGetOtherAxisStartData
	sscOtherAxisStartAbortOn	Turns the other axes start cancel signal (OSSTP□) to ON.	Page 123 sscOtherAxisStartAbortOn
	sscOtherAxisStartAbortOff	Turns the other axes start cancel signal (OSSTP□) to OFF	Page 124 sscOtherAxisStartAbortOff
	sscGetOtherAxisStartStatus	Gets the other axes start status.	Page 125 sscGetOtherAxisStartStatus
Pass position interrupt functions	sscSetIntPassPositionData	Sets the pass position interrupt condition data.	Page 126 sscSetIntPassPositionData
	sscCheckIntPassPositionData	Gets the pass position interrupt condition data.	Page 127 sscCheckIntPassPositionData
	sscSetStartingPassNumber	Sets the pass position condition start and end Nos.	Page 128 sscSetStartingPassNumber
	sscGetExecutingPassNumber	Gets the running pass position condition No.	Page 129 sscGetExecutingPassNumber
Sampling functions	sscStartSampling	Starts the sampling.	Page 130 sscStartSampling
	sscStopSampling	Stops the sampling.	Page 131 sscStopSampling
	sscSetSamplingParameter	Writes the sampling parameters.	Page 132 sscSetSamplingParameter
	sscGetSamplingParameter	Reads the sampling parameters.	Page 133 sscGetSamplingParameter
	sscGetSamplingError	Gets the sampling error information.	Page 134 sscGetSamplingError
	sscGetSamplingStatus	Gets the sampling execution information.	Page 135 sscGetSamplingStatus
	sscGetSamplingDataEm	Gets the sampling data.	Page 136 sscGetSamplingDataEm
Interrupt functions	sscIntStart	Starts up the interrupt driver.	Page 139 sscIntStart
	sscIntEnd	Closes the interrupt driver.	Page 140 sscIntEnd
	sscIntEnable	Enables the interrupt output.	Page 141 sscIntEnable
	sscIntDisable	Disables the interrupt output.	Page 142 sscIntDisable
	sscRegisterIntCallback	Registers the interrupt callback function.	Page 143 sscRegisterIntCallback
	sscUnregisterIntCallback	Unregisters the interrupt callback function.	Page 144 sscUnregisterIntCallback
	sscResetIntEvent	Sets the interrupt event signal status to nonsignaled.	Page 145 sscResetIntEvent
	sscSetIntEvent	Sets the interrupt event signal status to signaled.	Page 146 sscSetIntEvent
	sscWaitIntEvent	Waits until the interrupt event status becomes signaled.	Page 147 sscWaitIntEvent
	sscResetIntOasEvent	Sets the status of the other axes start interrupt event to nonsignaled.	Page 149 sscResetIntOasEvent
	sscSetIntOasEvent	Sets the status of the other axes start interrupt event to signaled.	Page 150 sscSetIntOasEvent
	sscWaitIntOasEvent	Waits until the status of the other axes start interrupt event becomes signaled.	Page 151 sscWaitIntOasEvent
	sscResetIntPassPosition	Sets the status of the pass position interrupt event to nonsignaled.	Page 153 sscResetIntPassPosition
	sscSetIntPassPosition	Sets the status of the pass position interrupt event to signaled.	Page 154 sscSetIntPassPosition
	sscWaitIntPassPosition	Waits until the status of the pass position interrupt event becomes signaled.	Page 155 sscWaitIntPassPosition
	sscResetIntDriveFin	Sets the status of the operation completion interrupt event to nonsignaled.	Page 157 sscResetIntDriveFin
	sscSetIntDriveFin	Sets the status of the operation completion interrupt event to signaled.	Page 158 sscSetIntDriveFin
	sscWaitIntDriveFin	Waits until the status of the operation completion interrupt event becomes signaled.	Page 159 sscWaitIntDriveFin

Function type	Function name	Function content	Reference section
I/O device functions	sscGetInputDeviceBit	Gets the input bit device of the motion control station on a 1-point basis.	Page 161 sscGetInputDeviceBit
	sscGetInputDeviceWord	Gets the input word device of the motion control station on a 1-word basis.	Page 162 sscGetInputDeviceWord
	sscGetInputDeviceDword	Gets the input word device of the motion control station on a 2-word basis.	Page 163 sscGetInputDeviceDword
	sscSetOutputDeviceBit	Sets the output bit device of the motion control station on a 1-point basis with the dual port memory exclusive control function of the Motion control board.	Page 164 sscSetOutputDeviceBit
	sscSetOutputDeviceBitNonExclusively	Sets the output bit device of the motion control station on a 1-point basis without using the dual port memory exclusive control function of the Motion control board.	Page 165 sscSetOutputDeviceBitNonExclusively
	sscSetOutputDeviceWord	Sets the output word device of the motion control station on a 1-word basis without using the dual port memory exclusive control function of the Motion control board.	Page 166 sscSetOutputDeviceWord
	sscSetOutputDeviceDword	Sets the output word device of the motion control station on a 2-word basis without using the dual port memory exclusive control function of the Motion control board.	Page 167 sscSetOutputDeviceDword
	sscChangeOutputDeviceWord	Sets the value of a specific bit of the device designated on a 1-word basis among the output word devices of the motion control station with the dual port memory exclusive control function of the Motion control board.	Page 168 sscChangeOutputDeviceWord
	sscChangeOutputDeviceDword	Sets the value of a specific bit of the device designated on a 2-word basis among the output word devices of the motion control station with the dual port memory exclusive control function of the Motion control board.	Page 169 sscChangeOutputDeviceDword
	sscGetOutputDeviceBit	Gets the output bit device of the motion control station on a 1-point basis.	Page 170 sscGetOutputDeviceBit
	sscGetOutputDeviceWord	Gets the output word device of the motion control station on a 1-word basis.	Page 171 sscGetOutputDeviceWord
	sscGetOutputDeviceDword	Gets the output word device of the motion control station on a 2-word basis.	Page 172 sscGetOutputDeviceDword
Initialize/finalize functions	sscInitializeLibrary	Executes the process of initialization of the Motion API.	Page 174 sscInitializeLibrary
	sscFinalizeLibrary	Executes the process of finalization of the Motion API.	Page 175 sscFinalizeLibrary
Link device access functions	sscGetBitLinkDevice	Gets more than one link device of the general station on a 1-bit basis.	Page 176 sscGetBitLinkDevice
	sscGetWordLinkDevice	Gets more than one link device of the general station on a 1-word basis.	Page 178 sscGetWordLinkDevice
	sscSetBitLinkDevice	Sets more than one link device of the general station on a 1-bit basis.	Page 179 sscSetBitLinkDevice
	sscSetWordLinkDevice	Sets more than one link device of the general station on a 1-word basis.	Page 180 sscSetWordLinkDevice
File operation functions	sscWriteFile	Writes the file.	Page 182 sscWriteFile
	sscReadFile	Reads the file.	Page 184 sscReadFile
	sscDeleteFile	Deletes the file.	Page 186 sscDeleteFile
	sscCreateDirectory	Creates the directory.	Page 187 sscCreateDirectory
	sscDeleteDirectory	Deletes the directory.	Page 189 sscDeleteDirectory
	sscGetFileList	Gets the file list.	Page 190 sscGetFileList
	sscGetFileListEx		Page 192 sscGetFileListEx
	sscFreeFileList	Release the file list.	Page 194 sscFreeFileList
	sscFreeFileListEx		Page 195 sscFreeFileListEx
SLMP functions	sscSlmpReadSlaveObject	Reads the object of the device which is connected to the network.	Page 196 sscSlmpReadSlaveObject
	sscSlmpWriteSlaveObject	Writes the object of the device which is connected to the network.	Page 198 sscSlmpWriteSlaveObject
	sscSlmpSendCommand	Sends/receives the SLMP message to the device which is connected to the network.	Page 200 sscSlmpSendCommand

Function type	Function name	Function content	Reference section
General input/output functions	sscGetGeneralInputDataBit	Gets the designated general input/output DI data on a 1-point basis.	Page 203 sscGetGeneralInputDataBit
	sscGetGeneralInputDataWord	Gets the designated general input/output DI data on a 4-point basis.	Page 204 sscGetGeneralInputDataWord
	sscSetGeneralOutputDataBit	Sets the designated general input/output DO data on a 1-point basis.	Page 205 sscSetGeneralOutputDataBit
	sscSetGeneralOutputDataBitExclusively	Sets the designated general input/output DO data on a 1-point basis with the general output exclusive control function of the Motion control board.	Page 206 sscSetGeneralOutputDataBitExclusively
	sscSetGeneralOutputDataWord	Sets the designated general input/output DO data on a 4-point basis.	Page 207 sscSetGeneralOutputDataWord
	sscSetGeneralOutputDataWordExclusively	Sets the designated general input/output DO data on a 4-point basis with the general output exclusive control function of the Motion control board.	Page 208 sscSetGeneralOutputDataWordExclusively
	sscGetGeneralOutputDataBit	Gets the designated general input/output DO data on a 1-point basis.	Page 209 sscGetGeneralOutputDataBit
	sscGetGeneralOutputDataWord	Gets the designated general input/output DO data on a 4-point basis.	Page 210 sscGetGeneralOutputDataWord
Arbitrary user area access functions	sscSetUserArea	Writes the data in the arbitrary user area.	Page 211 sscSetUserArea
	sscGetUserArea	Reads the data in the arbitrary user area.	Page 212 sscGetUserArea
Event history functions	sscReadEventHistoryData	Reads the event history.	Page 213 sscReadEventHistoryData
	sscClearEventHistoryData	Clear the event history.	Page 214 sscClearEventHistoryData
Pressure control functions	sscSetPrsProfile	Sets the pressure control profile.	Page 215 sscSetPrsProfile
	sscGetPrsProfile	Gets the pressure control profile.	Page 216 sscGetPrsProfile
	sscGetPrsStatus	Gets the pressure control status.	Page 217 sscGetPrsStatus
	sscGetExecutingPrsNumber	Gets the profile No. of pressure control in progress.	Page 218 sscGetExecutingPrsNumber
	sscGetPrsForwardSidePosition	Gets the tip position in pressure control.	Page 219 sscGetPrsForwardSidePosition
	sscGetPrsSpeedLimit	Gets the speed limit value.	Page 220 sscGetPrsSpeedLimit
	sscGetPrsPressureCommand	Gets the pressure command value.	Page 221 sscGetPrsPressureCommand
	sscGetPrsLoadCellPressure	Gets the load cell pressure.	Page 222 sscGetPrsLoadCellPressure
	sscGetPrsError	Gets the error information during pressure control.	Page 223 sscGetPrsError
	sscSetPrsProfileLimitPosition	Sets the limit position of the designated pressure control profile.	Page 224 sscSetPrsProfileLimitPosition
	sscGetPrsProfileLimitPosition	Gets the limit position of the designated pressure control profile.	Page 225 sscGetPrsProfileLimitPosition
	sscSetPrsProfileSpeedLimit	Sets the speed limit value of the designated pressure control profile.	Page 226 sscSetPrsProfileSpeedLimit
	sscGetPrsProfileSpeedLimit	Gets the speed limit value of the designated pressure control profile.	Page 227 sscGetPrsProfileSpeedLimit
	sscSetPrsProfilePressureCommand	Sets the pressure command of the designated pressure control profile.	Page 228 sscSetPrsProfilePressureCommand
	sscGetPrsProfilePressureCommand	Gets the pressure command of the designated pressure control profile.	Page 229 sscGetPrsProfilePressureCommand

3 API FUNCTION DETAILS

3.1 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

API version	Board version	Header file
1.00	01	em4xxstd.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

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

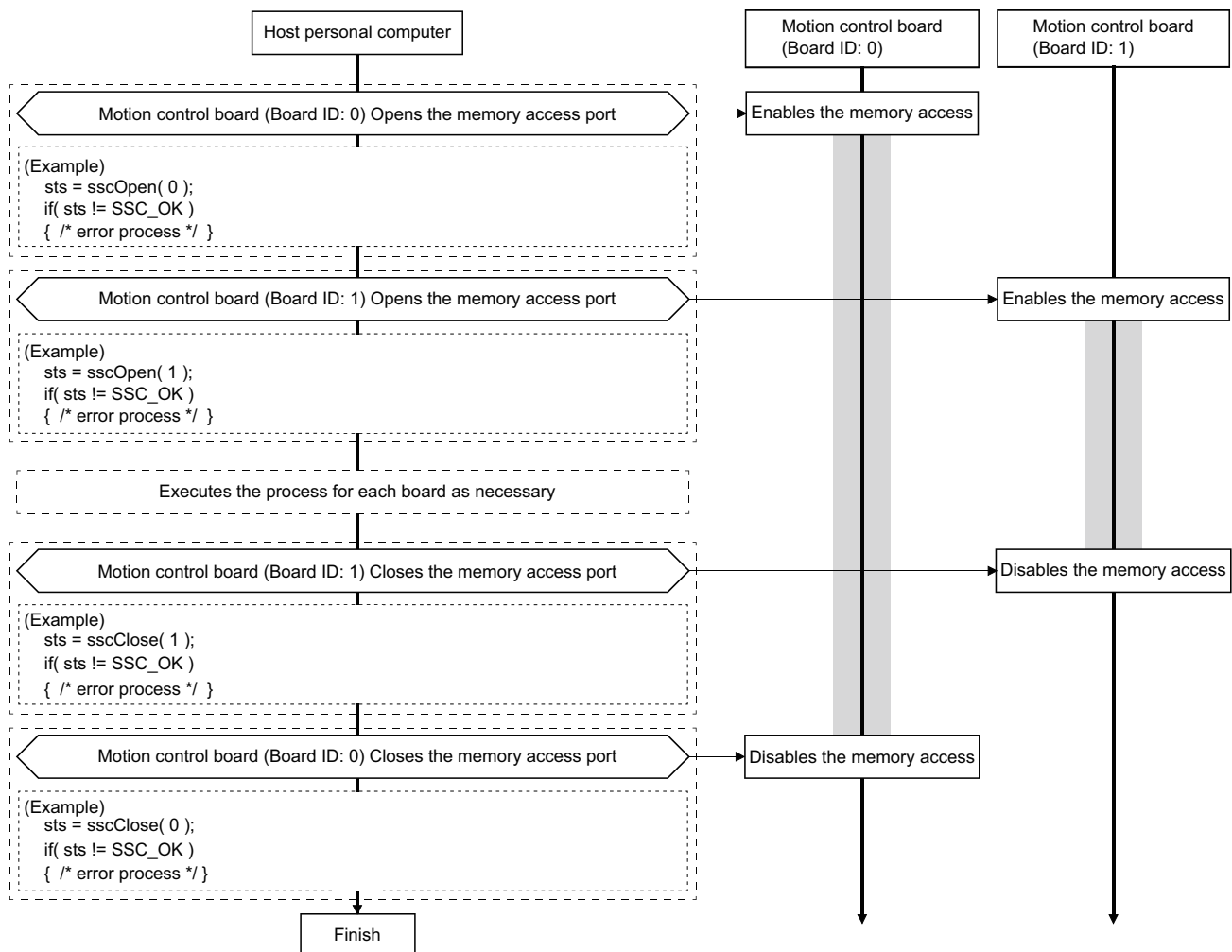
None.

3.2 Device Functions

Processing procedure

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

■When 2 Motion control boards (Board ID: 0 and Board ID: 1) are connected to the host personal computer



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 Motion control board properly. Confirm that the Motion control 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 Motion control board cannot be found. Confirm that the device driver is installed, and restart the host personal computer.
SSC_FUNC_ERR_UNSUPPORT_DEVICE_DRIVER	The device driver is not a supported version. Use the Motion API that combines with the device driver contained in the EM Motion SDK.
SSC_FUNC_ERR_NOT_INITIALIZED	The initialization of the Motion API (sscInitializeLibrary function) is not executed.
SSC_FUNC_ERR_MEMORY_ACCESS_DISABLE	It is under condition that the Motion control board cannot be accessed. Wait a while and try the execution again.
SSC_FUNC_ERR_PCIE_NOT_INITIALIZED	The initialization of the Motion control board is not completed. Wait a while and try the execution again.
SSC_FUNC_ERR_TIME_IS_NOT_SET	The time setting of the Motion control board is not completed. Restart the host personal computer.
SSC_FUNC_ERR_NOT_EXIST_BOARD_ID	The Motion control board of the designated board ID could not be found. Confirm the board ID selection (dip switch) of the Motion control board.

Point

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

Supported version

API version	Board version	Header file
1.00	01	em4xxstd.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

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

sscOpen ( Page 31 sscOpen)

3.3 Information Functions

sscGetControlCycle

The operation cycle monitor (setting cycle) 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 operation cycle monitor (setting cycle) ■Value <ul style="list-style-type: none">• SSC_CTRL_CYCLE_ERROR: Not set• SSC_CTRL_CYCLE_62: 62.50μs• SSC_CTRL_CYCLE_125: 125.00μs• SSC_CTRL_CYCLE_250: 250.00μs• SSC_CTRL_CYCLE_500: 500.00μs• SSC_CTRL_CYCLE_1000: 1000.00μs• SSC_CTRL_CYCLE_2000: 2000.00μs

■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

API version	Board version	Header file
1.00	03	em4xxstd.h

■Reference

None.

sscGetBoardVersionEx

The version information of the Motion control board is got.

```
int sscGetBoardVersionEx (  
    int board_id,  
    int channel,  
    int version_type,  
    wchar_t *version  
);
```

Detailed description

Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
version_type [in]	Version type to get ■Value <ul style="list-style-type: none">SSC_VERSION_SYSTEM: Basic system software versionSSC_VERSION_ADDON: Add-on versionSSC_VERSION_BOOT: Boot software versionSSC_VERSION_NETWORK_BOOT: Network boot software version
version [out]	Pointer to 32-byte array (2 bytes × 16) which stores the version information Use the Unicode (UTF-16LE) for a character code.

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

An example of the API function for the Visual C# project is shown below.

```
StringBuilder version = new StringBuilder(16);  
SscApi.sscGetBoardVersionEx(board_id, channel, version_type, version);  
string version_str =version.ToString();
```

Supported version

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

None.

sscGetOperationCycleMonitorEx

The operation cycle monitor data is got.

```
int sscGetOperationCycleMonitorEx (
    int board_id,
    int channel,
    unsigned long *now,
    unsigned long *max,
    unsigned long *over
);
```

Detailed description

Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
now [out]	Pointer to 4-byte variable which stores the operation cycle current time [ns]
max [out]	Pointer to 4-byte variable which stores the operation cycle maximum time [ns]
over [out]	Pointer to 4-byte variable which stores the operation cycle over status <ul style="list-style-type: none"> • 0: OFF • 1: ON (Cycle over is occurred)

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 cycle over status of the argument over is ON, "Cycle Over (error code: 320CH)" is occurred.

Supported version

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

None.

sscGetBoardSerialNumber

The Motion control board serial No. is got.

```
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

API version	Board version	Header file
1.00	01	em4xxstd.h

■Reference

sscGetBoardVersionEx ( Page 34 sscGetBoardVersionEx)

sscGetBoardSerialNumberW

The Motion control board serial No. is got. (Multibyte characters)

```
int sscGetBoardSerialNumberW (
    int board_id,
    wchar_t *serialnum
);
```

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
serialnum [out]	Pointer to 34-byte array (2 bytes × 17) which stores the serial No. Use the Unicode (UTF-16LE) for a character code.

■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

An example of the API function for the Visual C# project is shown below.

```
StringBuilder serial_numW = new StringBuilder(17);
SscApi.sscGetBoardSerialNumberW(board_id, serial_numW);
string serial_numW_str = serial_numW.ToString();
```

■Supported version

API version	Board version	Header file
1.00	01	em4xxstd.h

■Reference

sscGetBoardVersionEx ( Page 34 sscGetBoardVersionEx)

sscGetMotionApiVersion

The Motion API version is got.

```
int sscGetMotionApiVersion (  
    wchar_t *version  
);
```

Detailed description

■Argument

Argument	Description
version [out]	Pointer to 18-byte or more array (2 bytes × 9) which stores the version information

■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

None.

■Point

- For the argument (lpRsvParam) which stores the version information, ensure 9 characters including NULL codes.
- An example of the API function for the Visual C# project is shown below.

```
StringBuilder version = new StringBuilder(9);  
SscApi.sscGetMotionApiVersion(version);  
string version_str =version.ToString();
```

■Supported version

API version	Board version	Header file
1.00	01	em4xxstd.h

■Reference

None.

3.4 Parameter Functions

Processing procedure

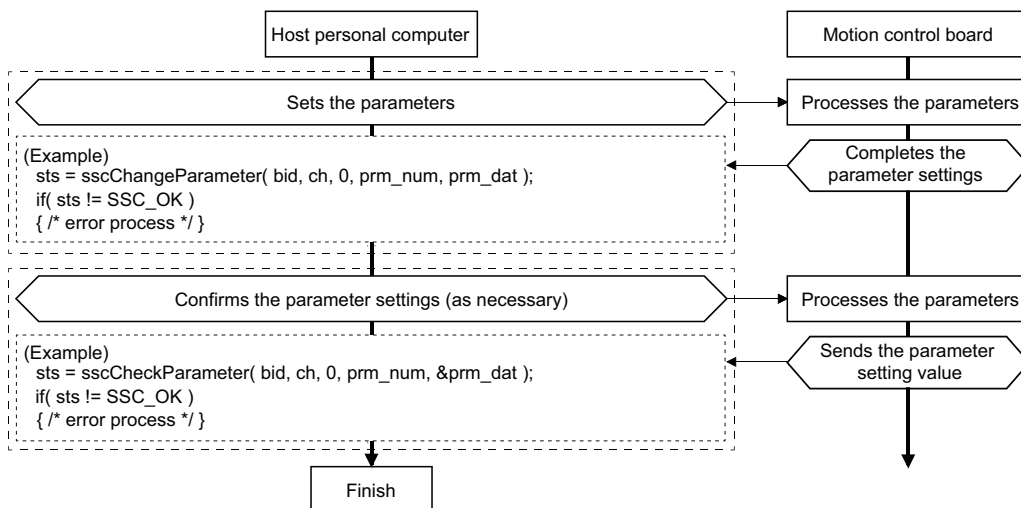
An example of parameter processing procedure is shown below.



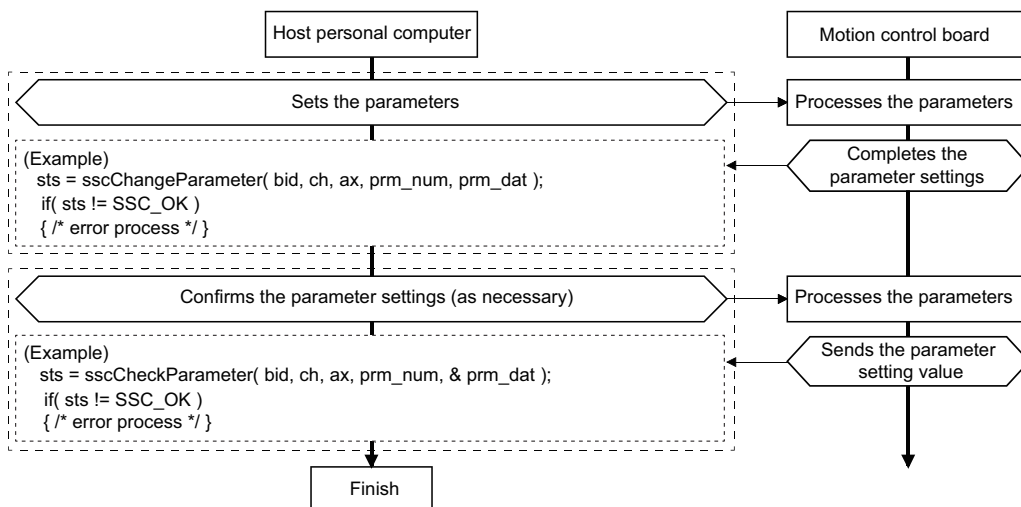
Change the parameters different from initial values only.

3

■When setting the system parameters



■When setting the control/axis management parameters



sscChangeParameter

Each of the parameters is written.

```
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. (0 to 64) <ul style="list-style-type: none">• 0: System parameter• 1 to 64: Axis parameter
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 Motion control board, the timeout time (1 second) has elapsed.
SSC_FUNC_ERR_STS_BIT_PWEN1	A parameter write No. error occurred. <ul style="list-style-type: none">• 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.




Point

- This API function can write the parameters either before or while the system is running.
- It takes about 2 to 10 operation cycles to confirm the response of the Motion control 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

API version	Board version	Header file
1.00	01	em4xxstd.h

■Reference

- [sscChange2Parameter](#) ( Page 42 [sscChange2Parameter](#))
- [sscCheckParameter](#) ( Page 44 [sscCheckParameter](#))
- [sscCheck2Parameter](#) ( Page 45 [sscCheck2Parameter](#))

sscChange2Parameter

Two of the parameters are written.

```
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. (0 to 64) <ul style="list-style-type: none">• 0: System parameter• 1 to 64: Axis parameter
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 <ul style="list-style-type: none">• 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 Motion control 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. <ul style="list-style-type: none">• 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.




■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 operation cycles to confirm the response of the Motion control 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

API version	Board version	Header file
1.00	01	em4xxstd.h

■Reference

- sscChangeParameter ( Page 40 sscChangeParameter)
- sscCheckParameter ( Page 44 sscCheckParameter)
- sscCheck2Parameter ( Page 45 sscCheck2Parameter)

sscCheckParameter

The set value of the designated parameter is read.

```
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. (0 to 64) <ul style="list-style-type: none">• 0: System parameter• 1 to 64: Axis parameter
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 Motion control 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. <ul style="list-style-type: none">• 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.




Point

- This API function can read the parameters either before or while the system is running.
- It takes about 2 to 10 operation cycles to confirm the response of the Motion control 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

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

- sscChangeParameter ( Page 40 sscChangeParameter)
- sscChange2Parameter ( Page 42 sscChange2Parameter)
- sscCheck2Parameter ( Page 45 sscCheck2Parameter)

sscCheck2Parameter

Two set values of the designated parameters are read.

```
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. (0 to 64) <ul style="list-style-type: none"> • 0: System parameter • 1 to 64: Axis parameter
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. <ul style="list-style-type: none"> ■Value <ul style="list-style-type: none"> • SSC_BIT_PRFIN: Parameter read complete • 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 Motion control 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. <ul style="list-style-type: none"> • 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.




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

■Supported version

API version	Board version	Header file
1.00	01	em4xxstd.h

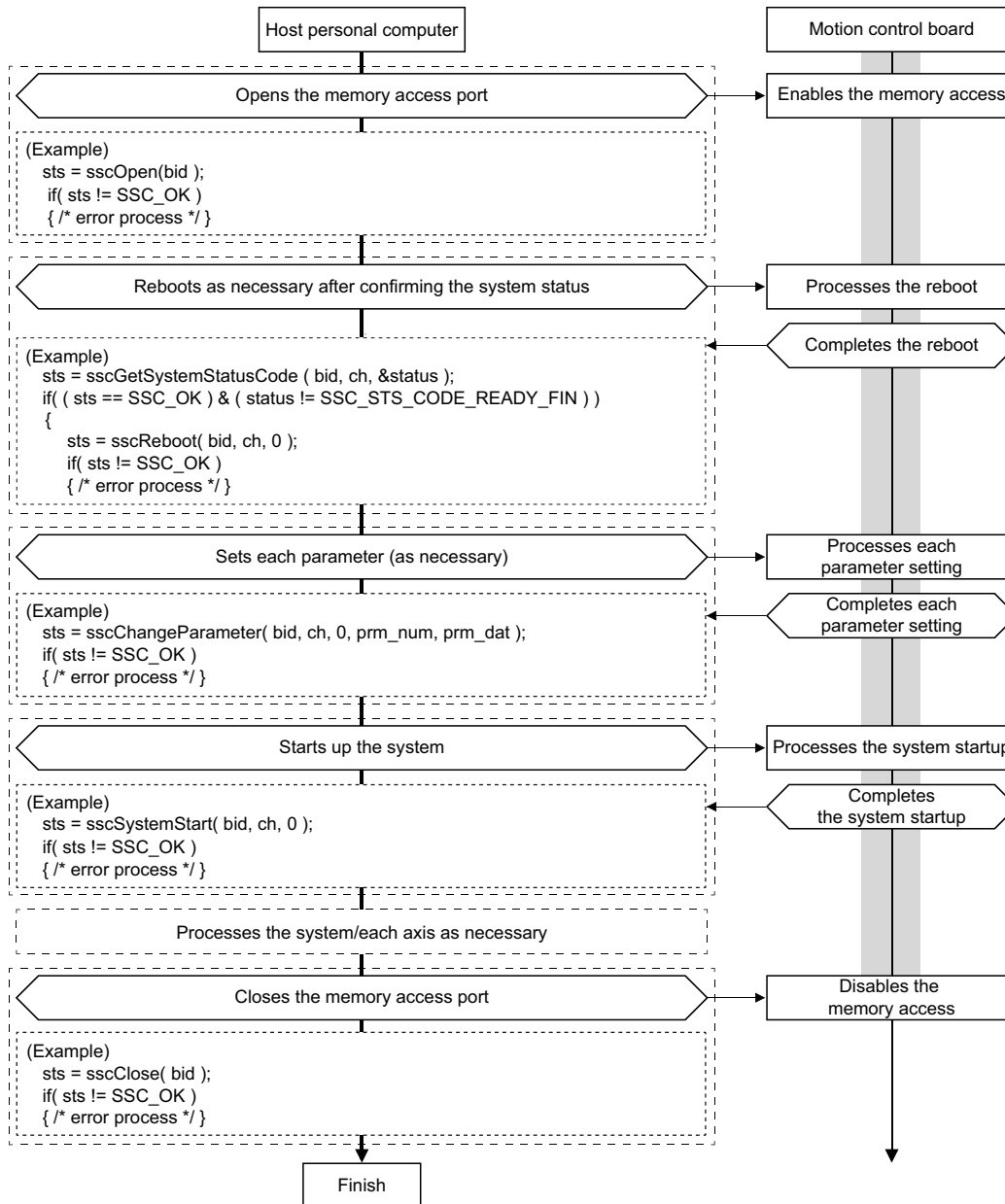
■Reference

- sscChangeParameter ( Page 40 sscChangeParameter)
- sscChange2Parameter ( Page 42 sscChange2Parameter)
- sscCheckParameter ( Page 44 sscCheckParameter)

3.5 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 Motion control board, the designated timeout time has elapsed.

Point

- When a value 20 seconds (20000ms) or less is designated as the timeout time, the timeout is 20 seconds (20000ms).
- The system reboots even if the function is executed in the system preparation completion.
- Do not open the device from the motion test tool during reboot. It may not operate correctly.

Supported version

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

sscGetSystemStatusCode ( Page 50 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 Motion control board, the designated timeout time has elapsed. Confirm that the CC-Link IE TSN connection is 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


API version	Board version	Header file
1.00	01	em4xxstd.h

■Reference

None.

sscGetSystemStatusCode

The system status code is got.

This API function is compatible with the function of MR-MC200/MR-MC300 series. Use the sscGetSystemStatusCodeEx function which does not require the type change of the acquired system status code. ( Page 51

sscGetSystemStatusCodeEx)

```
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 <ul style="list-style-type: none">SSC_STS_CODE_READY_FIN: System preparation completionSSC_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 Startup Processing" in the following manual.

 Motion Control Board User's Manual (Motion Control)

Supported version

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

None.

sscGetSystemStatusCodeEx

The system status code is got.

```
int sscGetSystemStatusCodeEx (
    int board_id,
    int channel,
    unsigned 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 <div> <div>Value</div> <ul style="list-style-type: none"> SSC_STS_CODE_READY_FIN: System preparation completion SSC_STS_CODE_RUNNING: System running </div>

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 Startup Processing" in the following manual.

 Motion Control Board User's Manual (Motion Control)

Supported version

API version	Board version	Header file
1.10	01	em4xxstd.h

Reference

None.


sscGetControllingAxis

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 238 SLAVE_INFO 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

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

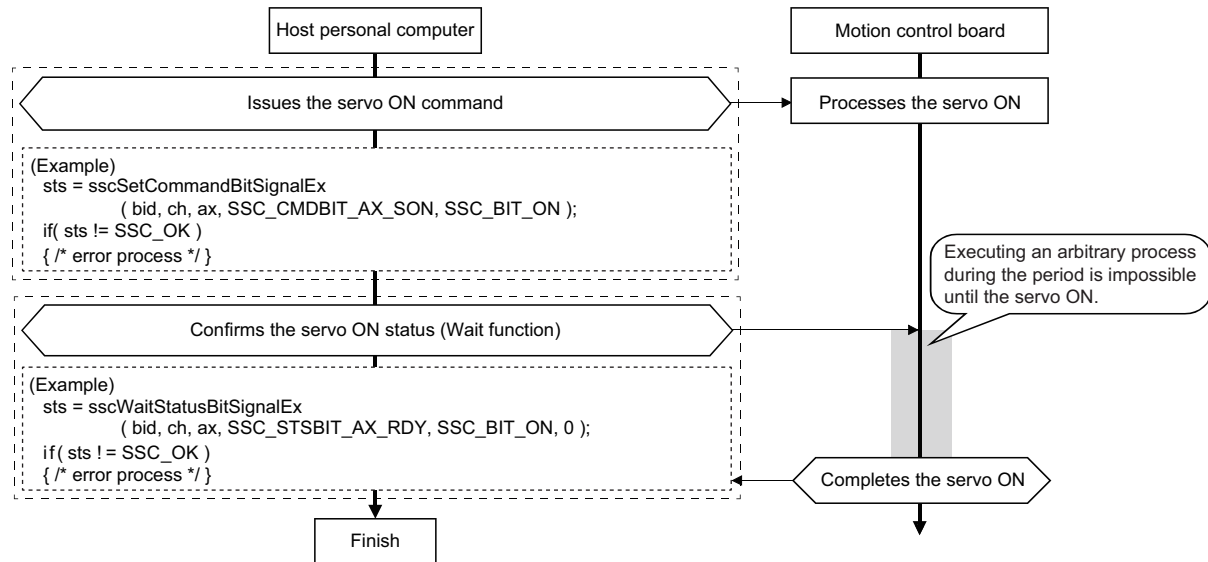
None.

3.6 Command/Status Functions

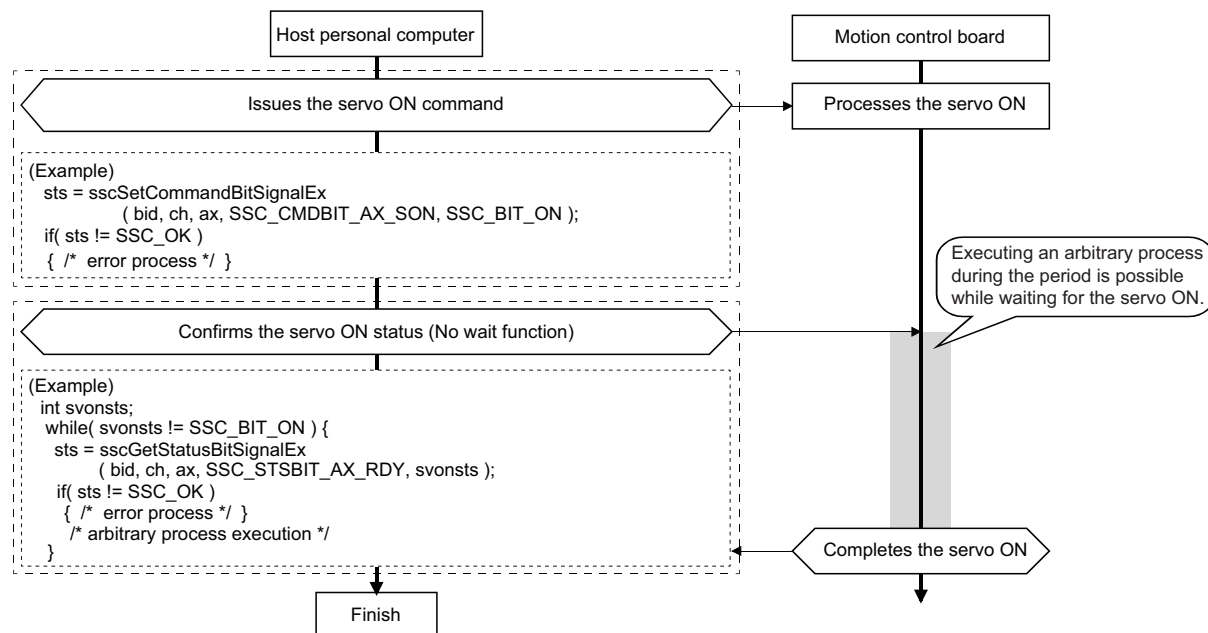
Processing procedure

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. (0 to 64) <ul style="list-style-type: none">• 0: System command bit• 1 to 64: Axis command bit
bitnum [in]	Command bit No. For the command bit No., refer to the following.  Page 257 BIT DEFINITION LIST
bitdata [in]	Command bit data ■Value <ul style="list-style-type: none">• 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 Motion control 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

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

- sscGetStatusBitSignalEx ( Page 55 sscGetStatusBitSignalEx)
- sscWaitStatusBitSignalEx ( Page 56 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. (0 to 64) <ul style="list-style-type: none"> • 0: System status bit • 1 to 64: Axis status bit
bitnum [in]	Status bit No. For the status bit No., refer to the following.  Page 257 BIT DEFINITION LIST
bitstatus [out]	Pointer to 4-byte variable (4 bytes × 1) which stores the status bit data <ul style="list-style-type: none"> ■ Value <ul style="list-style-type: none"> • 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

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

- sscSetCommandBitSignalEx ( Page 54 sscSetCommandBitSignalEx)
- sscWaitStatusBitSignalEx ( Page 56 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. (0 to 64) <ul style="list-style-type: none">• 0: System status bit• 1 to 64: Axis status bit
bitnum [in]	Status bit No. For the status bit No., refer to the following.  Page 257 BIT DEFINITION LIST
waitstatus [in]	Bit status to be waited ■Value <ul style="list-style-type: none">• 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 Motion control 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

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

- sscSetCommandBitSignalEx ( Page 54 sscSetCommandBitSignalEx)
- sscGetStatusBitSignalEx ( Page 55 sscGetStatusBitSignalEx)

sscGetStatusBits

The status bits are got all at once.

```
int sscGetStatusBits (
    int board_id,
    int channel,
    int axnum,
    int type,
    unsigned char *statusbits
);
```

Detailed description

Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (0 to 64) <ul style="list-style-type: none"> • 0: System status bit • 1 to 64: Each axis status bit
type [in]	Status bit type <ul style="list-style-type: none"> ■ Value <ul style="list-style-type: none"> • SSC_BITS_HIGH_SPEED: High-speed area • SSC_BITS_LOW_SPEED: Low-speed area
statusbits [out]	Pointer to 32-byte array (1 byte × 32) which stores the status bits

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

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

- sscSetCommandBitSignalEx ( Page 54 sscSetCommandBitSignalEx)
- sscGetStatusBitSignalEx ( Page 55 sscGetStatusBitSignalEx)
- sscCheckBitStatus ( Page 58 sscCheckBitStatus)


sscCheckBitStatus

The status bit specified from the status bit array is got.

```
int sscCheckBitStatus (  
    int axnum,  
    int type,  
    int bit_num,  
    unsigned char *statusbits,  
    int *status  
);
```

Detailed description

Argument

Argument	Description
axnum [in]	Axis No. (0 to 64) <ul style="list-style-type: none">• 0: System status bit• 1 to 64: Each axis status bit
type [in]	Status bit type <ul style="list-style-type: none">■Value<ul style="list-style-type: none">• SSC_RAPID_BITS: High-speed area• SSC_SLOW_BITS: Low-speed area
bit_num [in]	Status bit No. For the status bit No., refer to the following.  Page 257 BIT DEFINITION LIST
statusbits [in]	Pointer to 32-byte array (1 byte × 32) which stores the status bit Use the argument (statusbits) which is got by the sscGetStatusBits function.
status [out]	Pointer to 4-byte variable (4 bytes × 1) which stores the status bit data <ul style="list-style-type: none">■Value<ul style="list-style-type: none">• 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.)

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

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

sscGetStatusBits ( Page 57 sscGetStatusBits)

3.7 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 64)
pntnum [in]	Point No. (0 to 2047)
pPntDataEx [in]	Pointer to 48-byte structure (48 bytes × 1) which stores the point data For the point data structure, refer to the following.  Page 230 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

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

sscSetPointOffset ( Page 61 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 64)
pntnum [in]	Point No. (0 to 2047)
pPntDataEx [out]	Pointer to 48-byte structure (48 bytes × 1) which stores the point data For the point data structure, refer to the following.  Page 230 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

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

sscCheckPointOffset ( Page 62 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 64)
offset [in]	Point No. offset value (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.

■Point

None.

■Supported version

API version	Board version	Header file
1.00	01	em4xxstd.h

■Reference

sscSetPointDataEx ([📖](#) Page 59 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 64)
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

API version	Board version	Header file
1.00	01	em4xxstd.h

■Reference

sscCheckPointDataEx ([📖](#) Page 60 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 64)
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 "Interpolation operation".

■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

API version	Board version	Header file
1.00	01	em4xxstd.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 64)
latest_point [in]	Latest command point No. (1 to 2048)

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

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

- sscAutoStart ( Page 75 sscAutoStart)
- sscSetPointDataEx ( Page 59 sscSetPointDataEx)

3.8 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 64)
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 234 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.
SSC_FUNC_ERR_UNSUPPORTED_BOARD_VERSION	The Motion control board does not support this function.

Point

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

Supported version

API version	Board version	Header file
1.30	05	em4xxstd.h

Reference

sscGetPressData ( Page 66 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 64)
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 234 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.
SSC_FUNC_ERR_UNUPPORT_BOARD_VERSION	The Motion control board does not support this function.

Point

None.

Supported version

API version	Board version	Header file
1.30	05	em4xxstd.h

Reference

sscSetPressData ( Page 65 sscSetPressData)

sscSetPressTargetTorque

The target torque of continuous operation to torque control data is set.

```
int sscSetPressTargetTorque (
    int board_id,
    int channel,
    int axnum,
    short target_torque
);
```

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 64)
target_torque [in]	Target torque [0.1%] (-32768 to 32767)

■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_UNUPPORT_BOARD_VERSION	The Motion control board does not support this function.

■Point

None.

■Supported version

API version	Board version	Header file
1.30	05	em4xxstd.h

■Reference

sscGetPressTargetTorque ( Page 68 sscGetPressTargetTorque)

sscGetPressTargetTorque

The target torque of continuous operation to torque control data is got.

```
int sscGetPressTargetTorque (  
    int board_id,  
    int channel,  
    int axnum,  
    short target_torque  
);
```

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 64)
target_torque [out]	Pointer to 2-byte variable (2 bytes × 1) which stores the target torque [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.
SSC_FUNC_ERR_UNUPPORT_BOARD_VERSION	The Motion control board does not support this function.

■Point

None.

■Supported version

API version	Board version	Header file
1.30	05	em4xxstd.h

■Reference

sscSetPressTargetTorque ( Page 67 sscSetPressTargetTorque)

3.9 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 64)
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 Motion control 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 drive unit 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 Motion control 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

API version	Board version	Header file
1.00	01	em4xxstd.h

■Reference

- sscJogStop ( Page 71 sscJogStop)
- sscJogStopNoWait ( Page 72 sscJogStopNoWait)
- sscDriveStop ( Page 82 sscDriveStop)
- sscDriveStopNoWait ( Page 83 sscDriveStopNoWait)
- sscDriveRapidStop ( Page 84 sscDriveRapidStop)
- sscDriveRapidStopNoWait ( Page 85 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 64)

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 Motion control 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

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

- sscJogStart ( Page 69 sscJogStart)
- sscJogStopNoWait ( Page 72 sscJogStopNoWait)
- sscDriveStop ( Page 82 sscDriveStop)
- sscDriveStopNoWait ( Page 83 sscDriveStopNoWait)
- sscDriveRapidStop ( Page 84 sscDriveRapidStop)
- sscDriveRapidStopNoWait ( Page 85 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 64)
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

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

- sscJogStart ( Page 69 sscJogStart)
- sscJogStop ( Page 71 sscJogStop)
- sscDriveStop ( Page 82 sscDriveStop)
- sscDriveStopNoWait ( Page 83 sscDriveStopNoWait)
- sscDriveRapidStop ( Page 84 sscDriveRapidStop)
- sscDriveRapidStopNoWait ( Page 85 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 64)
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 Motion control 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 drive unit 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 Motion control 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

API version	Board version	Header file
1.00	01	em4xxstd.h

■Reference

- sscDriveStop ( Page 82 sscDriveStop)
- sscDriveStopNoWait ( Page 83 sscDriveStopNoWait)
- sscDriveRapidStop ( Page 84 sscDriveRapidStop)
- sscDriveRapidStopNoWait ( Page 85 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 64)
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_NOW_DRIVING_READY	During the operation startup preparation (until the Motion control 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 drive unit 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 Motion control 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

API version	Board version	Header file
1.00	01	em4xxstd.h

■Reference

- sscDriveStop ( Page 82 sscDriveStop)
- sscDriveStopNoWait ( Page 83 sscDriveStopNoWait)
- sscDriveRapidStop ( Page 84 sscDriveRapidStop)
- sscDriveRapidStopNoWait ( Page 85 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 64)

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 Motion control 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 drive unit 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 Motion control 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

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

- sscDriveStop ( Page 82 sscDriveStop)
- sscDriveStopNoWait ( Page 83 sscDriveStopNoWait)
- sscDriveRapidStop ( Page 84 sscDriveRapidStop)
- sscDriveRapidStopNoWait ( Page 85 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 64)
grpnum [in]	Interpolation group No. (0 to 16) <ul style="list-style-type: none"> • 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. <ul style="list-style-type: none"> • 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 Motion control 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 drive unit 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 Motion control 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

API version	Board version	Header file
1.00	01	em4xxstd.h

■Reference

- sscDriveStop ( Page 82 sscDriveStop)
- sscDriveStopNoWait ( Page 83 sscDriveStopNoWait)
- sscDriveRapidStop ( Page 84 sscDriveRapidStop)
- sscDriveRapidStopNoWait ( Page 85 sscDriveRapidStopNoWait)

sscInterpolationStart

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) <ul style="list-style-type: none"> • 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. <ul style="list-style-type: none"> • 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 Motion control 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 drive unit 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 Motion control 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

API version	Board version	Header file
1.00	01	em4xxstd.h

■Reference

- sscDriveStop ( Page 82 sscDriveStop)
- sscDriveStopNoWait ( Page 83 sscDriveStopNoWait)
- sscDriveRapidStop ( Page 84 sscDriveRapidStop)
- sscDriveRapidStopNoWait ( Page 85 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 64)

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 Motion control 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 drive unit 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 Motion control 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

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

- sscDriveStop ( Page 82 sscDriveStop)
- sscDriveStopNoWait ( Page 83 sscDriveStopNoWait)
- sscDriveRapidStop ( Page 84 sscDriveRapidStop)
- sscDriveRapidStopNoWait ( Page 85 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 64)
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 Motion control 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

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

- sscDriveStopNoWait ( Page 83 sscDriveStopNoWait)
- sscDriveRapidStop ( Page 84 sscDriveRapidStop)
- sscDriveRapidStopNoWait ( Page 85 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 64)
stpsts [out]	Pointer to 2-byte variable (2 bytes × 1) which stores the stop complete status <div> <div>Value</div> <ul style="list-style-type: none"> SSC_DRIVING: During operation SSC_DRIVE_FIN: Stop complete </div>

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 Motion control 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

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

- sscDriveStop ( Page 82 sscDriveStop)
- sscDriveRapidStop ( Page 84 sscDriveRapidStop)
- sscDriveRapidStopNoWait ( Page 85 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 64)
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 Motion control 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

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

- sscDriveStop ( Page 82 sscDriveStop)
- sscDriveStopNoWait ( Page 83 sscDriveStopNoWait)
- sscDriveRapidStopNoWait ( Page 85 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 64)
stpsts [out]	Pointer to 2-byte variable (2 bytes × 1) which stores the stop complete status <div> <div>Value</div> <ul style="list-style-type: none"> SSC_DRIVING: During operation SSC_DRIVE_FIN: Stop complete </div>

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 Motion control 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

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

- sscDriveStop ( Page 82 sscDriveStop)
- sscDriveStopNoWait ( Page 83 sscDriveStopNoWait)
- sscDriveRapidStop ( Page 84 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 64)
drv_mode [in]	Operation mode ■Value <ul style="list-style-type: none">SSC_DRV_MODE_AUTO: Automatic operation modeSSC_DRV_MODE_HOME: Home position return modeSSC_DRV_MODE_JOG: JOG operation modeSSC_DRV_MODE_INC: Incremental feed modeSSC_DRV_MODE_INTERP: Interpolation operation modeSSC_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 drive unit 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 Motion control 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

API version	Board version	Header file
1.00	01	em4xxstd.h

■Reference

- sscGetDriveMode ( Page 88 sscGetDriveMode)
- sscSetOtherAxisStartData ( Page 121 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 64)
drv_mode [out]	Pointer to 4-byte variable (4 bytes × 1) which stores the operation mode status ■Value <ul style="list-style-type: none">SSC_DRV_MODE_NONE: Operation mode not setSSC_DRV_MODE_AUTO: In automatic operation modeSSC_DRV_MODE_HOME: In home position return modeSSC_DRV_MODE_JOG: In JOG operation modeSSC_DRV_MODE_INC: In incremental feed modeSSC_DRV_MODE_INTERP: Interpolation operation modeSSC_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

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

- sscSetDriveMode ( Page 86 sscSetDriveMode)
- sscSetOtherAxisStartData ( Page 121 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 64)
fin_type [in]	Operation completion type Value <ul style="list-style-type: none"> 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 <ul style="list-style-type: none"> 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 mode or the interpolation operation mode, 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 interlock stop, 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 mode or the interpolation operation mode, 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 Motion control board.
- For the JOG operation mode, the completion of operation status is not "SSC_FIN_STS_RDY".

■Supported version

API version	Board version	Header file
1.00	01	em4xxstd.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 64)
ctrl_mode [in]	Control mode ■Value <ul style="list-style-type: none"> 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_UNUPPORT_BOARD_VERSION	The Motion control board does not support this 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 drive unit 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 Motion control board, the timeout time (1 second) has elapsed.
SSC_FUNC_ERR_STS_BIT_CTLMCE	The control mode switching error signal (CTLMCE) turned ON.

Point

It takes about 6 operation cycles + 4 to 6 ms to confirm the response of the Motion control board.

Supported version

API version	Board version	Header file
1.30	05	em4xxstd.h

Reference

sscSetPressData ( Page 65 sscSetPressData)

3.10 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 64)
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 Motion control 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

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

- sscChangeManualSpeed ( Page 96 sscChangeManualSpeed)
- sscChangeManualAccTime ( Page 98 sscChangeManualAccTime)
- sscChangeManualDecTime ( Page 100 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 64)
pntnum [in]	Point No. (0 to 2047)
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 Motion control 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

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

- sscGetDrivingPointNumber ( Page 63 sscGetDrivingPointNumber)
- sscChangeAutoSpeed ( Page 97 sscChangeAutoSpeed)
- sscChangeAutoAccTime ( Page 99 sscChangeAutoAccTime)
- sscChangeAutoDecTime ( Page 101 sscChangeAutoDecTime)

sscChangeLinearPosition

The position is changed in the linear interpolation.

```
int sscChangeLinearPosition (  
    int board_id,  
    int channel,  
    int axnum,  
    int grpnnum,  
    int pntrnum,  
    long *pPosition  
);
```

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)
grpnnum [in]	Interpolation group No. (0 to 16) <ul style="list-style-type: none">• 0: Changeable interpolation group enabled• 1 to 16: Changeable interpolation group disabled
pntrnum [in]	Point No. (0 to 2047)
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 "Interpolation operation".
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_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 Motion control 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

API version	Board version	Header file
1.00	01	em4xxstd.h

■Reference

- [sscGetDrivingPointNumber](#) ( Page 63 [sscGetDrivingPointNumber](#))
- [sscChangeAutoSpeed](#) ( Page 97 [sscChangeAutoSpeed](#))
- [sscChangeAutoAccTime](#) ( Page 99 [sscChangeAutoAccTime](#))
- [sscChangeAutoDecTime](#) ( Page 101 [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 64)
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 Motion control 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

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

- sscChangeManualPosition ( Page 92 sscChangeManualPosition)
- sscChangeManualAccTime ( Page 98 sscChangeManualAccTime)
- sscChangeManualDecTime ( Page 100 sscChangeManualDecTime)

sscChangeAutoSpeed

The speed is changed in the automatic operation or the interpolation operation.

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 64)
pntnum [in]	Point No. (0 to 2047)
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 "Interpolation 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 Motion control 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

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

- sscGetDrivingPointNumber ( Page 63 sscGetDrivingPointNumber)
- sscChangeAutoPosition ( Page 93 sscChangeAutoPosition)
- sscChangeAutoAccTime ( Page 99 sscChangeAutoAccTime)
- sscChangeAutoDecTime ( Page 101 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 64)
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 Motion control 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

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

- sscChangeManualPosition ( Page 92 sscChangeManualPosition)
- sscChangeManualSpeed ( Page 96 sscChangeManualSpeed)
- sscChangeManualDecTime ( Page 100 sscChangeManualDecTime)

sscChangeAutoAccTime

The acceleration time constant is changed in the automatic operation or the interpolation operation.

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 64)
pntnum [in]	Point No. (0 to 2047)
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 "Interpolation 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 Motion control 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

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

- sscGetDrivingPointNumber ( Page 63 sscGetDrivingPointNumber)
- sscChangeAutoPosition ( Page 93 sscChangeAutoPosition)
- sscChangeAutoSpeed ( Page 97 sscChangeAutoSpeed)
- sscChangeAutoDecTime ( Page 101 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 64)
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 Motion control 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

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

- sscChangeManualPosition ( Page 92 sscChangeManualPosition)
- sscChangeManualSpeed ( Page 96 sscChangeManualSpeed)
- sscChangeManualAccTime ( Page 98 sscChangeManualAccTime)

sscChangeAutoDecTime

The deceleration time constant is changed in the automatic operation or the interpolation operation.

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 64)
pntnum [in]	Point No. (0 to 2047)
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 "Interpolation 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 Motion control 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

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

- sscGetDrivingPointNumber ( Page 63 sscGetDrivingPointNumber)
- sscChangeAutoPosition ( Page 93 sscChangeAutoPosition)
- sscChangeAutoSpeed ( Page 97 sscChangeAutoSpeed)
- sscChangeAutoAccTime ( Page 99 sscChangeAutoAccTime)

3.11 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. (0 to 64) <ul style="list-style-type: none">• 0: System alarm• 1 to 64: Axis alarm
alarm_type [in]	Alarm type ■Value <ul style="list-style-type: none">• SSC_ALARM_SYSTEM: System alarm• SSC_ALARM_SERVO: Drive unit alarm• SSC_ALARM_OPERATION: Operation 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

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

sscResetAlarm ( Page 103 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. (0 to 64) <ul style="list-style-type: none"> • 0: System alarm • 1 to 64: Axis alarm
alarm_type [in]	Alarm type <ul style="list-style-type: none"> ■Value <ul style="list-style-type: none"> • SSC_ALARM_SYSTEM: System alarm • SSC_ALARM_SERVO: Drive unit alarm • SSC_ALARM_OPERATION: Operation 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

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

sscGetAlarm ( Page 102 sscGetAlarm)

sscGetSystemErrorCode

The system error is got.

```
int sscGetSystemErrorCode (  
    int board_id,  
    int channel,  
    unsigned short *code  
);
```

Detailed description

■Argument

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

■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 error codes, refer to "ALARM No." in the following manual.

 Motion Control Board User's Manual (Motion Control)

Or, refer to "List of Error Codes" in the following manual.

 Motion Control Board User's Manual (Network)

■Supported version

API version	Board version	Header file
1.00	01	em4xxstd.h

■Reference

None.

sscResetAllError

The all error reset is executed, and the system error, the motion control error, the network error, and the base system error that are occurring are reset.

After the all error reset signal (AERST) is turned ON and the all error reset completed signal (AERF) or the all error reset error signal (AERE) is confirmed to be ON, the all error reset signal (AERST) is turned OFF.

```
int sscResetAllError (
    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 Motion control board, the timeout time (10 seconds) has elapsed.
SSC_FUNC_ERR_STS_BIT_AERE	The all error reset error signal (AERE) turned ON.

Point

None.

Supported version

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

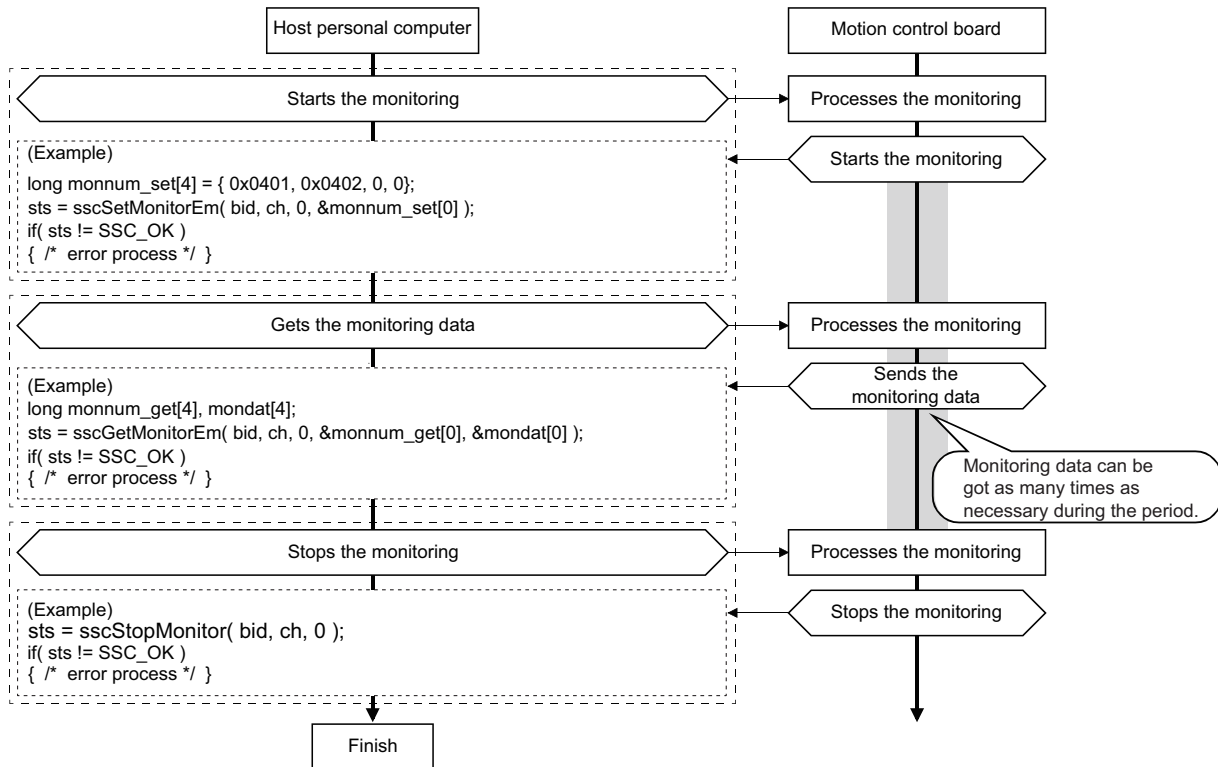
None.

3.12 General Monitor Functions

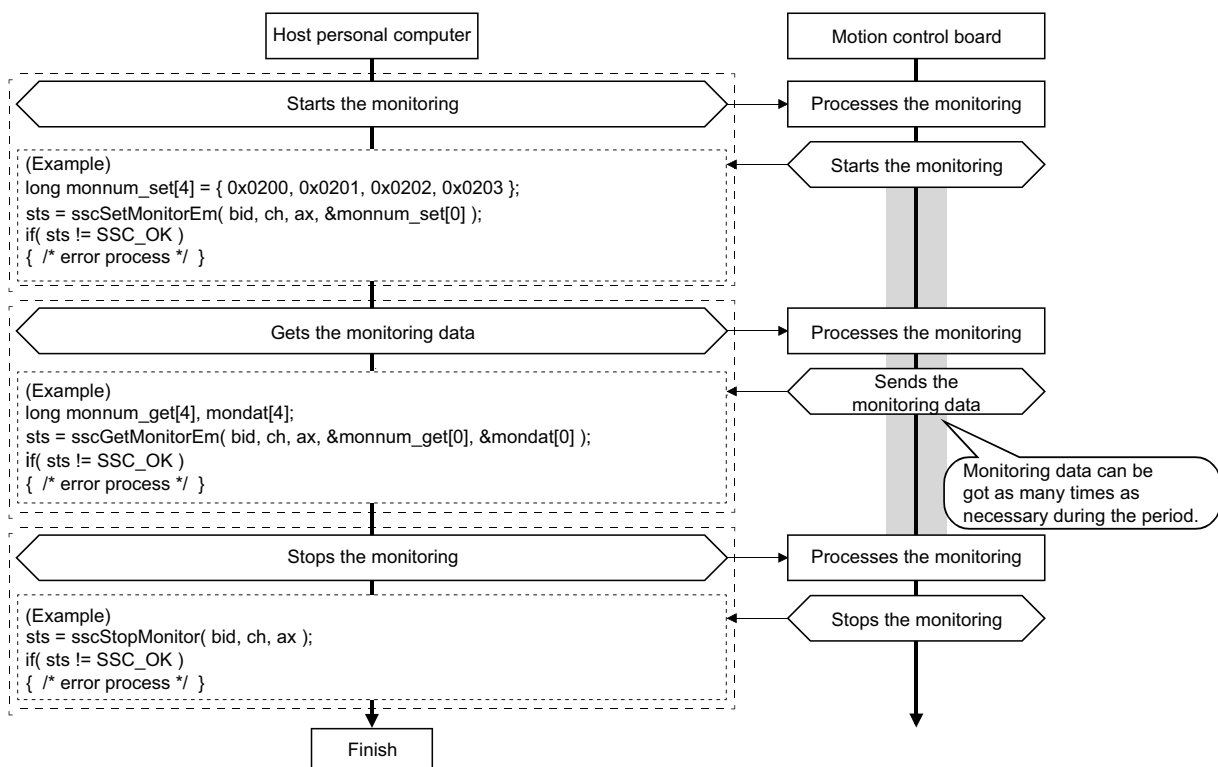
Processing procedure

An example of general monitor processing procedure is shown below.

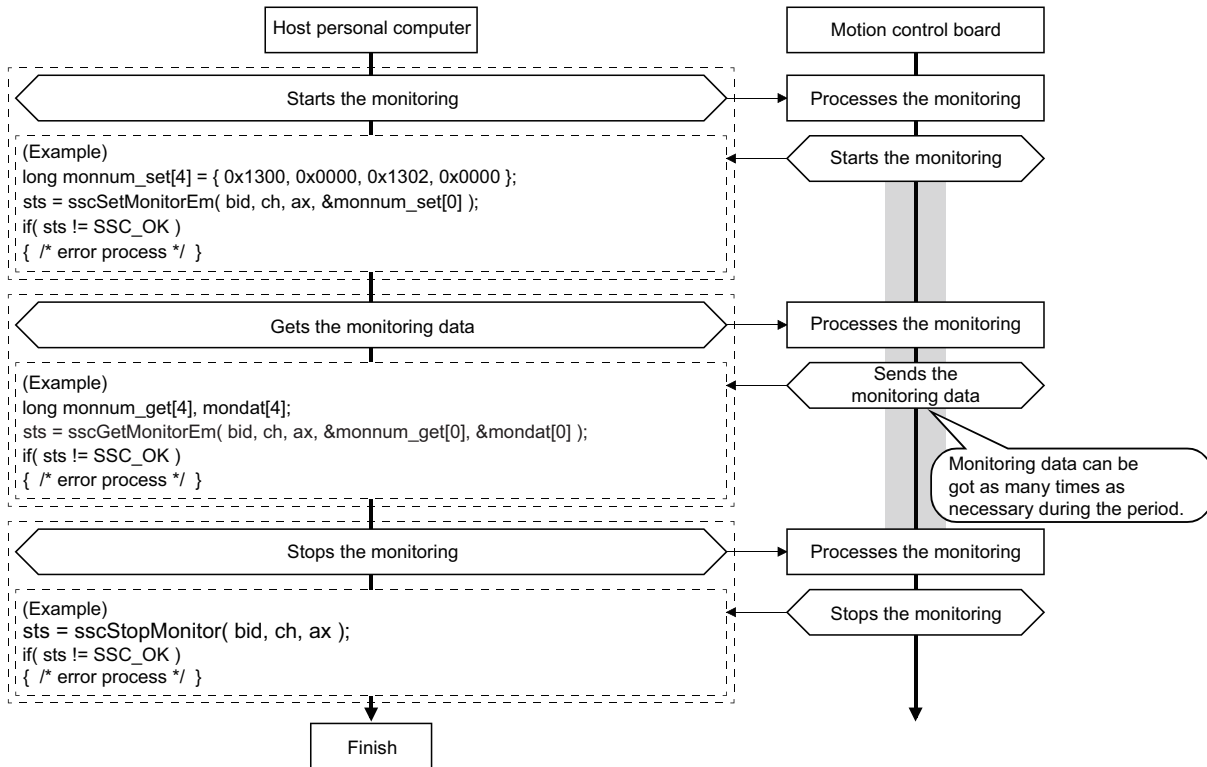
■When getting system information



■When getting servo information/operation information



■When getting operation information (2-word)



sscSetMonitorEm

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 sscSetMonitorEm (  
    int board_id,  
    int channel,  
    int axnum,  
    long *monnum  
);
```

Detailed description

Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (0 to 64) <ul style="list-style-type: none">• 0: System monitor• 1 to 64: Axis monitor
monnum [in]	Pointer to 16-byte array (4 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. <ul style="list-style-type: none">• 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 Motion control board, the timeout time (1 second) has elapsed.

Point

- 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 Motion control board, it takes an operation cycle to several ms to set the monitoring because of the time it takes to confirm the response of the Motion control board.

Supported version

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

- sscStopMonitor (🔗 Page 109 sscStopMonitor)
- sscGetMonitorEm (🔗 Page 110 sscGetMonitorEm)

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. (0 to 64) <ul style="list-style-type: none"> • 0: System alarm • 1 to 64: Axis 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_ALREADY_MONITOR_STOP	The monitor has already stopped.
SSC_FUNC_ERR_TIMEOUT_01	During the confirmation of response after executing the command to the Motion control board, the timeout time (1 second) has elapsed.

■Point

Depending on the control status of the Motion control board, it takes an operation cycle to several ms to stop the monitoring because of the time it takes to confirm the response of the Motion control board.

■Supported version

API version	Board version	Header file
1.00	01	em4xxstd.h

■Reference

sscSetMonitorEm ( Page 108 sscSetMonitorEm)

sscGetMonitorEm

The monitoring data is got.

```
int sscGetMonitorEm (  
    int board_id,  
    int channel,  
    int axnum,  
    long *monnum,  
    long *mondata  
);
```

Detailed description

Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
axnum [in]	Axis No. (0 to 64) <ul style="list-style-type: none">• 0: System monitor• 1 to 64: Axis monitor
monnum [out]	Pointer to 16-byte array (4 bytes × 4) which stores the monitor Nos.
mondata [out]	Pointer to 16-byte array (4 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

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

sscSetMonitorEm ([📄](#) Page 108 sscSetMonitorEm)

3.13 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 64)
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 Motion control board is used to get the current command position.

■Supported version

API version	Board version	Header file
1.00	01	em4xxstd.h

■Reference

sscGetCurrentFbPositionFast ( Page 112 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 64)
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 Motion control board is used to get the current feedback position.

■Supported version

API version	Board version	Header file
1.00	01	em4xxstd.h

■Reference

sscGetCurrentCmdPositionFast ( Page 111 sscGetCurrentCmdPositionFast)

sscGetloStatusFast

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

```
int sscGetloStatusFast (
    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 64)
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 <ul style="list-style-type: none"> • 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 Motion control board is used to get the external signal status.

■Supported version

API version	Board version	Header file
1.00	01	em4xxstd.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 64)
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 Motion control board is used to get the moving speed.

■Supported version

API version	Board version	Header file
1.00	01	em4xxstd.h

■Reference

sscGetFbSpeedFast ( Page 115 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 64)
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 Motion control board is used to get the feedback moving speed.

■Supported version

API version	Board version	Header file
1.00	01	em4xxstd.h

■Reference

sscGetCmdSpeedFast ( Page 114 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 64)
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 Motion control board is used to get the electrical current feedback.

■Supported version

API version	Board version	Header file
1.00	01	em4xxstd.h

■Reference

None.

3.14 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 monitoring time [ms]

■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_UNUPPORT_BOARD_VERSION	The Motion control board does not support this function.
SSC_FUNC_ERR_ALREADY_ENABLE_WDT	The user watchdog function has been already valid.

■Point

None.

■Supported version

API version	Board version	Header file
1.10	02	em4xxstd.h

■Reference

sscWdDisable ( Page 118 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_UNUPPORT_BOARD_VERSION	The Motion control board does not support this function.
SSC_FUNC_ERR_ALREADY_DISABLE_WDT	The user watchdog function has been already invalid.

Point

None.

Supported version

API version	Board version	Header file
1.10	02	em4xxstd.h

Reference

sscWdEnable ([↩](#) Page 117 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.
SSC_FUNC_ERR_UNUPPORT_BOARD_VERSION	The Motion control board does not support this function.

■Point

None.

■Supported version

API version	Board version	Header file
1.10	02	em4xxstd.h

■Reference

sscWdEnable ( Page 117 sscWdEnable)

sscCheckPCleBusConnection

The PCIe bus connection status is confirmed.

```
int sscCheckPCleBusConnection (  
    int board_id,  
    int *connect_status  
);
```

Detailed description

Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
connect_status [out]	Pointer to 4-byte variable (4 bytes × 1) which stores the PCIe bus connection status ■Value • SSC_PCIE_CONNECT: PCIe bus is connecting • SSC_PCIE_DISCONNECT: PCIe bus is disconnected

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

In order to minimize the effects against other processing, calling this API function in a low priority thread or a timer is recommended.

Supported version

API version	Board version	Header file
1.20	01	em4xxstd.h

Reference

sscWdEnable ( Page 117 sscWdEnable)

3.15 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 64)
pOasData [in]	Pointer to 128-byte structure (128 bytes × 1) which stores the other axes start data For the other axes start data structure, refer to the following.  Page 232 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

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

sscGetOtherAxisStartData ( Page 122 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 64)
pOasData [out]	Pointer to 128-byte structure (128 bytes × 1) which stores the other axes start data For the other axes start data structure, refer to the following.  Page 232 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

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

sscSetOtherAxisStartData ( Page 121 sscSetOtherAxisStartData)

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 64)

■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 Motion control 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

API version	Board version	Header file
1.00	01	em4xxstd.h

■Reference

- sscOtherAxisStartAbortOff ( Page 124 sscOtherAxisStartAbortOff)
- sscGetOtherAxisStartStatus ( Page 125 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 64)

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 Motion control board, the timeout time (1 second) has elapsed.



Point

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

Supported version

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

- sscOtherAxisStartAbortOn ( Page 123 sscOtherAxisStartAbortOn)
- sscGetOtherAxisStartStatus ( Page 125 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 64)
status [out]	<p>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.</p> <p>■Value</p> <ul style="list-style-type: none"> • SSC_BIT_OSOP: Other axes start notice • SSC_BIT_OSFIN: Other axes start complete • SSC_BIT_OSERR: Other axes start incomplete • SSC_BIT OSDTO: Other axes start output signal timer control • SSC_BIT OSDCE: Other axes start output signal control error • SSC_BIT OSDDL: Other axes start output signal control delay

■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

API version	Board version	Header file
1.30	05	em4xxstd.h

■Reference

None.

3.16 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 128)
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] • 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

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

sscCheckIntPassPositionData ( Page 127 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 128)
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

API version	Board version	Header file
1.00	01	em4xxstd.h

■Reference

sscSetIntPassPositionData ( Page 126 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 64)
pass_start [in]	Pass position condition start No. (1 to 128)
pass_end [in]	Pass position condition end No. (1 to 128)

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

API version	Board version	Header file
1.00	01	em4xxstd.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 64)
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 incompleteness.

■Supported version

API version	Board version	Header file
1.00	01	em4xxstd.h

■Reference

None.

3.17 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

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

- sscStopSampling ( Page 131 sscStopSampling)
- sscSetSamplingParameter ( Page 132 sscSetSamplingParameter)
- sscGetSamplingStatus ( Page 135 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 Motion control 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

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

sscStartSampling ( Page 130 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 Motion control 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

API version	Board version	Header file
1.00	01	em4xxstd.h

■Reference

sscGetSamplingParameter ( Page 133 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 Motion control 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

API version	Board version	header file
1.00	01	em4xxstd.h

Reference

sscSetSamplingParameter ( Page 132 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 48-byte structure (48 bytes × 1) which stores the sampling error information For the sampling error information structure, refer to the following.  Page 235 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

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

sscGetSamplingStatus ( Page 135 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 <ul style="list-style-type: none"> 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

API version	Board version	Header folder
1.00	01	em4xxstd.h

Reference

- sscGetSamplingError ( Page 134 sscGetSamplingError)
- sscGetSamplingDataEm ( Page 136 sscGetSamplingDataEm)


sscGetSamplingDataEm

The sampling read enabled points and the sampling data (for 128 points) are got.
The sampling read enabled points are not confirmed.

```
int sscGetSamplingDataEm (  
    int board_id,  
    int channel,  
    int page_num,  
    short *valid_num,  
    SMP_DATA_EM *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 512)
valid_num [out]	Pointer to the valid points variable of the sampling read
pSmpData [out]	Pointer to 18944-byte structure (148 bytes × 128) which stores the sampling data For the sampling data information structure, refer to the following.  Page 236 SMP_DATA_EM 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 Motion control 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.

Supported version

API version	Board version	Header file
1.00	01	em4xxstd.h

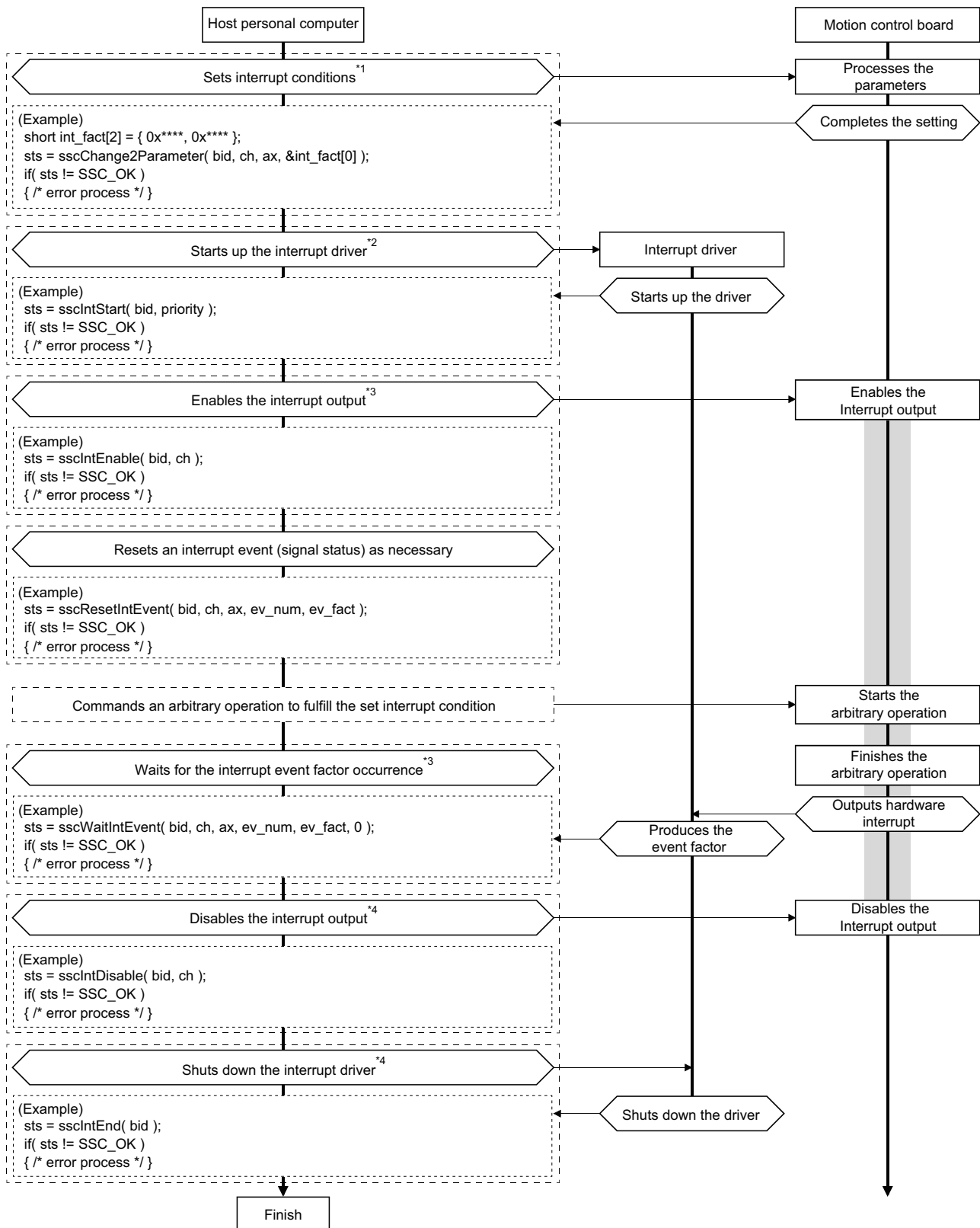
Reference

- sscStartSampling ( Page 130 sscStartSampling)
- sscGetSamplingStatus ( Page 135 sscGetSamplingStatus)

3.18 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_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_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

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

- sscIntEnd ( Page 140 sscIntEnd)
- sscIntEnable ( Page 141 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

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

sscIntStart ( Page 139 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

API version	Board version	Header file
1.00	01	em4xxstd.h

■Reference

- sscIntStart ( Page 139 sscIntStart)
- sscIntDisable ( Page 142 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

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

sscIntEnable ([↩](#) Page 141 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 237 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

API version	Board version	Header file
1.10	01	em4xxstd.h

Reference

- sscIntStart ( Page 139 sscIntStart)
- sscIntEnable ( Page 141 sscIntEnable)
- sscUnregisterIntCallback( Page 144 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_UNREGISTER_CALLBACK	The interrupt callback function has already been unregistered.

Point

None.

Supported version

API version	Board version	Header file
1.10	01	em4xxstd.h

Reference

sscRegisterIntCallback ( Page 143 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. (0 to 64) <ul style="list-style-type: none"> • 0: System interrupt event • 1 to 64: Axis interrupt event
eventnum [in]	Event wait No. (0 to 1)
eventfactor [in]	Event factor For the event factors, refer to the following.  Page 300 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

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

sscWaitIntEvent ( Page 147 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 Motion control 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. (0 to 64) <ul style="list-style-type: none">• 0: System interrupt event• 1 to 64: Axis interrupt event
eventnum [in]	Event wait No. (0 to 1)
eventfactor [in]	Event factor For the event factors, refer to the following.  Page 300 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

API version	Board version	Header file
1.00	01	em4xstd.h

Reference

sscWaitIntEvent ( Page 147 sscWaitIntEvent)

sscWaitIntEvent

This function waits until the interrupt event status becomes signaled.


This function is used to wait for the interrupt from the Motion control 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. (0 to 64) <ul style="list-style-type: none"> • 0: System interrupt event • 1 to 64: Axis interrupt event
eventnum [in]	Event wait No. (0 to 1)
eventfactor [in]	Event factor For the event factors, refer to the following.  Page 300 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

API version	Board version	Header file
1.00	01	em4xstd.h

■Reference

- [sscResetIntEvent](#) ( Page 145 [sscResetIntEvent](#))
- [sscSetIntEvent](#) ( Page 146 [sscSetIntEvent](#))

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. of other axes start table (1 to 64)
oas_num [in]	Other axes start table No. (1 to 64)

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

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

sscWaitIntOasEvent ( Page 151 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 Motion control 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 64)
oas_num [in]	Other axes start table No. (1 to 64)

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

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

sscWaitIntOasEvent ( Page 151 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 Motion control 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 64)
oas_num [in]	Other axes start table No. (1 to 64)
oas_type [in]	Other axes start wait type ■Value <ul style="list-style-type: none"> 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 <ul style="list-style-type: none"> 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" occurs during the 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 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

API version	Board version	Header file
1.00	01	em4xxstd.h

■Reference

- sscResetIntOasEvent ( Page 149 sscResetIntOasEvent)
- sscSetIntOasEvent ( Page 150 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 128)
pass_end [in]	Pass position condition end No. (1 to 128)

■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

API version	Board version	Header file
1.00	01	em4xxstd.h

■Reference

sscWaitIntPassPosition ( Page 155 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 Motion control 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 128)
pass_end [in]	Pass position condition end No. (1 to 128)

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

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

sscWaitIntPassPosition ( Page 155 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 Motion control 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 128)
pass_status [out]	Pointer to 4-byte variable (4 bytes × 1) which stores the pass position status Value <ul style="list-style-type: none"> 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

API version	Board version	Header file
1.00	01	em4xxstd.h

■Reference

- [sscResetIntPassPosition](#) ( Page 153 [sscResetIntPassPosition](#))
- [sscSetIntPassPosition](#) ( Page 154 [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 64)

■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

API version	Board version	Header file
1.00	01	em4xxstd.h

■Reference

sscWaitIntDriveFin ( Page 159 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 Motion control 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 64)

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

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

sscWaitIntDriveFin ( Page 159 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 Motion control 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 64)
fin_type [in]	Operation completion type ■Value <ul style="list-style-type: none"> 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 <ul style="list-style-type: none"> 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 interpolation operation 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 interpolation operation mode. Therefore, if the completion of operation check is performed in an operation mode other than the automatic operation mode or the interpolation operation 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 (Drive unit alarm)	—
CPO (Rough match)	Unnecessary when the operation completion type "SSC_FIN_TYPE_CPO" is not used.
OALM (Operation alarm)	—
OPF (Operation completed)	—
SYSE (During system status code 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 "SSC_SUBCMD_STOP_INP" in the automatic operation or the interpolation operation, the Operation completion type is "SSC_FIN_TYPE_INP" even though "SSC_FIN_TYPE_SMZ" is set.
- 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 Motion control board.

■Supported version

API version	Board version	Header file
1.00	01	em4xxstd.h

■Reference

- sscResetIntDriveFin (📖 Page 157 sscResetIntDriveFin)
- sscSetIntDriveFin (📖 Page 158 sscSetIntDriveFin)
- sscGetDriveFinStatus (📖 Page 89 sscGetDriveFinStatus)

3.19 I/O Device Functions

sscGetInputDeviceBit

The input bit device of the motion control station 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 3FFFh)
dev_in [out]	Pointer to 4-byte variable (4 bytes × 1) which stores the status of the input bit device ■Value <ul style="list-style-type: none"> 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

When getting the devices of the general station, use the sscGetBitLinkDevice function/sscGetWordLinkDevice function.

■Supported version

API version	Board version	Header file
1.00	01	em4xxstd.h

■Reference

sscGetInputDeviceWord ( Page 162 sscGetInputDeviceWord)

sscGetInputDeviceWord

The input word device of the motion control station 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 1FFFh)
word_cnt [in]	Word points from the input word device No. (0001h to 2000h)
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

When getting the devices of the general station, use the sscGetBitLinkDevice function/sscGetWordLinkDevice function.

Supported version

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

sscGetInputDeviceBit ( Page 161 sscGetInputDeviceBit)

sscGetInputDeviceDword

The input word device of the motion control station is got on a 2-word basis.

```
int sscGetInputDeviceDword (
    int board_id,
    int channel,
    int word_num,
    int dword_cnt,
    unsigned long *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 1FFEh)
dword_cnt [in]	2-word points from the Input word device No. (0001h to 1000h)
dev_in [out]	Pointer to the array (4 bytes × dword_cnt) which stores the input word device status

■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" + "dword_cnt" designated by the argument exceeds the size of the input device table.

■Point

When getting the devices of the general station, use the sscGetBitLinkDevice function/sscGetWordLinkDevice function.

■Supported version

API version	Board version	Header file
1.00	01	em4xxstd.h

■Reference

None.

sscSetOutputDeviceBit

The output bit device of the motion control station is set on a 1-point basis with the dual port memory exclusive control function of the Motion control 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 3FFFh)
dev_out [in]	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 Motion control board, the timeout time (10 seconds) has elapsed.


Point

- This API function sets the output bit device with the exclusive control function of the Motion control board.
- When setting the devices of the general station, use the sscSetBitLinkDevice function/sscSetWordLinkDevice function.

Supported version

API version	Board version	Header file
1.00	05	em4xxstd.h

Reference

- sscSetOutputDeviceWord ( Page 166 sscSetOutputDeviceWord)
- sscGetOutputDeviceBit ( Page 170 sscGetOutputDeviceBit)

sscSetOutputDeviceBitNonExclusively

The output bit device of the motion control station is set on a 1-point basis without using the dual port memory exclusive control function of the Motion control board.

```
int sscSetOutputDeviceBitNonExclusively (
    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 3FFFh)
dev_out [in]	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.



Point

- This API function does not use the exclusive control function of the Motion control board.
- If the output device signal is updated from the Motion control board by using the other axes start function while executing this API function, the consistency of the data may not be maintained. This condition applies to the case when the host personal computer and Motion control board simultaneously write to the same output device No. In such a case, use the sscSetOutputDeviceBit function to write the output device signal after controlling the possessory right of the output device signal.
- When setting the devices of the general station, use the sscSetBitLinkDevice function/sscSetWordLinkDevice function.

Supported version

API version	Board version	Header file
1.30	05	em4xxstd.h

Reference

- sscSetOutputDeviceBit ( Page 166 sscSetOutputDeviceWord)
- sscGetOutputDeviceBit ( Page 170 sscGetOutputDeviceBit)

sscSetOutputDeviceWord

The output word device of the motion control station is set on a 1-word basis without using the dual port memory exclusive control function of the Motion control 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 1FFFh)
word_cnt [in]	Word points from the output word device No. (0001h to 2000h)
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.
- When setting the devices of the general station, use the sscSetBitLinkDevice function/sscSetWordLinkDevice function.

Supported version

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

- sscSetOutputDeviceBit ( Page 164 sscSetOutputDeviceBit)
- sscGetOutputDeviceWord ( Page 171 sscGetOutputDeviceWord)

sscSetOutputDeviceDword

The output word device of the motion control station is set on a 2-word basis without using the dual port memory exclusive control function of the Motion control board.

```
int sscSetOutputDeviceDword (
    int board_id,
    int channel,
    int word_num,
    int dword_cnt,
    unsigned long *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 1FFh)
dword_cnt [in]	2-word points from the output word device No. (0001h to 1000h)
dev_out [in]	Pointer to the array (4 bytes × dword_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" + "dword_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.
- When setting the devices of the general station, use the sscSetBitLinkDevice function/sscSetWordLinkDevice function.

■Supported version

API version	Board version	Header file
1.00	01	em4xxstd.h

■Reference

None.

sscChangeOutputDeviceWord

The value of a specific bit of the device designated on a 1-word basis among the output word devices of the motion control station is set with the dual port memory exclusive control function of the Motion control board.

For example, use this API function when you want to change the value of a specific bit while controlling the output device signal with the other axes start function.

```
int sscChangeOutputDeviceWord (  
    int board_id,  
    int channel,  
    int word_num,  
    unsigned short target_bits,  
    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 1FFFh)
target_bits [in]	Bit array whose value is to be changed (0000h to FFFFh) (0: Bit non-subject to change, 1: Bit subject to change) Bit0 → □□□1, ... Bit15 → 8□□□
dev_out [in]	Value to be changed (0000h to FFFFh) (For each bit, 0: OFF, 1: 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 Motion control board, the timeout time (10 seconds) has elapsed.

Point




- This API function sets the output word devices with the dual port memory exclusive control function of the Motion control board.
- When setting the devices of the general station, use the sscSetBitLinkDevice function/sscSetWordLinkDevice function.
- An example of changing the value of the output word device (RWw10) from "8100h" to "0110h" (namely, bit4 ON, bit15 OFF) is shown below.

```
sscChangeOutputDeviceWord(board, channel, 0x0010, 0x8010, 0x0010);
```

Supported version

API version	Board version	Header file
1.30	05	em4xxstd.h

Reference

- sscSetOutputDeviceWord ( Page 166 sscSetOutputDeviceWord)
- sscChangeOutputDeviceDword ( Page 169 sscChangeOutputDeviceDword)
- sscGetOutputDeviceWord ( Page 171 sscGetOutputDeviceWord)

sscChangeOutputDeviceDword

The value of a specific bit of the device designated on a 2-word basis among the output word devices of the motion control station is set with the dual port memory exclusive control function of the Motion control board.

For example, use this API function when you want to change the value of a specific bit while controlling the output device signal with the other axes start function.

```
int sscChangeOutputDeviceDword (
    int board_id,
    int channel,
    int word_num,
    unsigned short target_bits,
    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 1FFFh)
target_bits [in]	Bit No. whose value is to be changed (00000000h to FFFFFFFFh) (0: Bit non-subject to change, 1: Bit subject to change) Bit0 → □□□□□□□1, ... Bit31 → 8□□□□□□□
dev_out [in]	Value to be changed (00000000h to FFFFFFFFh) (For each bit, 0: OFF, 1: 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 Motion control board, the timeout time (10 seconds) has elapsed.

Point




- This API function sets the output word devices with the dual port memory exclusive control function of the Motion control board.
- When setting the devices of the general station, use the sscSetBitLinkDevice function/sscSetWordLinkDevice function.
- An example of changing the value of the output word devices (RWw10 and RWw11) from "81008100h" to "01100110h" (namely, bit4 and bit20 ON, bit15 and bit31 OFF) is shown below.

```
sscChangeOutputDeviceDword(board, channel, 0x0010, 0x80108010, 0x00100010);
```

Supported version

API version	Board version	Header file
1.30	05	em4xxstd.h

Reference

- sscSetOutputDeviceDword ( Page 167 sscSetOutputDeviceDword)
- sscChangeOutputDeviceWord ( Page 168 sscChangeOutputDeviceWord)
- sscGetOutputDeviceDword ( Page 172 sscGetOutputDeviceDword)

sscGetOutputDeviceBit

The output bit device of the motion control station 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 3FFFh)
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

When getting the devices of the general station, use the sscGetBitLinkDevice function/sscGetWordLinkDevice function.

Supported version

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

- sscGetOutputDeviceWord ( Page 171 sscGetOutputDeviceWord)
- sscSetOutputDeviceBit ( Page 164 sscSetOutputDeviceBit)

sscGetOutputDeviceWord

The output word device of the motion control station 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 1FFFh)
word_cnt [in]	Word points from the output word device No. (0001h to 2000h)
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

When getting the devices of the general station, use the sscGetBitLinkDevice function/sscGetWordLinkDevice function.

■Supported version

API version	Board version	Header file
1.00	01	em4xxstd.h

■Reference

- sscGetOutputDeviceBit ( Page 170 sscGetOutputDeviceBit)
- sscSetOutputDeviceWord ( Page 166 sscSetOutputDeviceWord)

sscGetOutputDeviceDword

The output word device of the motion control station is got on a 2-word basis.

```
int sscGetOutputDeviceDword (  
    int board_id,  
    int channel,  
    int word_num,  
    int dword_cnt,  
    unsigned long *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 1FFh)
dword_cnt [in]	2-word points from the output word device No. (0001h to 1000h)
dev_out [out]	Pointer to the array (4 bytes × dword_cnt) which stores the output word device status

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" + "dword_cnt" designated by the argument exceeds the size of the output device table.

Point

When getting the devices of the general station, use the sscGetBitLinkDevice function/sscGetWordLinkDevice function.

Supported version

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

None.

3.20 Initialization/Finalization Functions

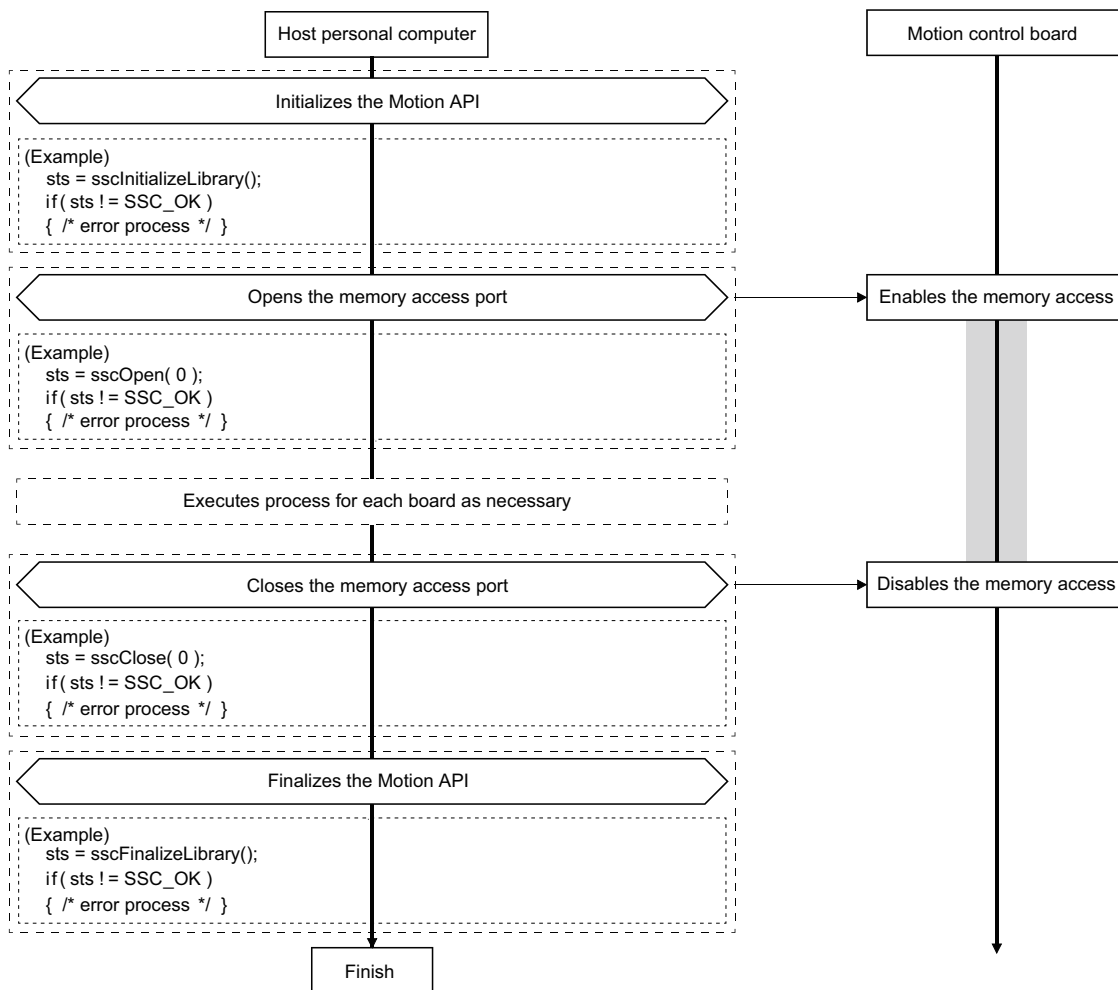
Processing procedure

An example of initialization/finalization of the Motion API processing procedure is shown below.

When using the Motion API, call the Motion API initialization function (sscInitializeLibrary) at first, and call the Motion API finalization function (sscFinalizeLibrary) at the end.

For the Motion API initialization/finalization functions, regardless of the number of the Motion control board to be used, call each function one by one.

As an example, the processing procedure when using the initialization/finalization of the Motion API is shown below.



sscInitializeLibrary

The initialization of the Motion API is executed.

```
int sscInitializeLibrary (  
    void  
);
```

Detailed description

■Argument

None.

■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_ALREADY_INITIALIZED	The initialization of the Motion API (sscInitializeLibrary function) is already executed.

■Point

Before calling other functions of the Motion API, call this API function.

■Supported version

API version	Board version	Header file
1.00	01	em4xxstd.h

■Reference

sscFinalizeLibrary ( Page 175 sscFinalizeLibrary)

sscFinalizeLibrary

The finalization of the Motion API is executed.

```
int sscFinalizeLibrary (  
    void  
);
```

Detailed description

■Argument

None.

■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_NOT_INITIALIZED	The initialization of the Motion API (sscInitializeLibrary function) is not executed.

■Point

When using the Motion API, call this API function before finishing the user program.

■Supported version

API version	Board version	Header file
1.00	01	em4xstd.h

■Reference

sscInitializeLibrary ( Page 174 sscInitializeLibrary)

3.21 Link Device Access Functions

sscGetBitLinkDevice

More than one link device of the general station is got on a 1-bit basis.

```
int sscGetBitLinkDevice (  
    int board_id,  
    int channel,  
    int dev_type,  
    int bit_num,  
    int bit_cnt,  
    int *link_dev  
);
```

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
dev_type	Link device type which gets the bit value ■Value • SSC_RX: RX (Input) • SSC_RY: RY (Output) • SSC_LB: LB (Link relay) • SSC_SB: SB (Link special relay)
bit_num [in]	Link device No. • RX: 0000h to 3FFFh • RY: 0000h to 3FFFh • LB: 0000h to 7FFFh • SB: 0000h to 0FFFh
bit_cnt [in]	Bit points to get • RX: 0001h to 4000h • RY: 0001h to 4000h • LB: 0001h to 8000h • SB: 0001h to 1000h
link_dev [out]	Pointer to the array (4 bytes × bit_cnt) which stores the link device status ■Value • SSC_BIT_OFF: Signal OFF • SSC_BIT_ON: 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_LINK_DEVICE_RANGE_OVER	The "bit_num" + "bit_cnt" designated by the argument exceeds the size of the link device.



■Point

When getting the devices of the motion control station, use the sscGetInputDeviceBit function/sscGetOutputDeviceBit function.

■Supported version

API version	Board version	Header file
1.30	05	em4xxstd.h

■Reference

- [sscGetWordLinkDevice](#) ( Page 178 [sscGetWordLinkDevice](#))
- [sscSetBitLinkDevice](#) ( Page 179 [sscSetBitLinkDevice](#))

sscGetWordLinkDevice

More than one link device of the general station is got on a 1-word basis.

```
int sscGetWordLinkDevice (  
    int board_id,  
    int channel,  
    int dev_type,  
    int word_num,  
    int word_cnt,  
    unsigned short * link_dev  
);
```

Detailed description

Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
dev_type	Link device type which gets the word value ■Value • SSC_RWr: RWr (Input) • SSC_RWw: RWw (Output) • SSC_LW: LW (Link register) • SSC_SW: SW (Link special register)
word_num [in]	Link device No. • RWr: 0000h to 1FFFh • RWw: 0000h to 1FFFh • LW: 0000h to 3FFFh • SW: 0000h to 0FFFh
word_cnt [in]	Word points to get • RWr: 0001h to 2000h • RWw: 0001h to 2000h • LW: 0001h to 4000h • SW: 0001h to 1000h
link_dev [out]	Pointer to the array (2 bytes × word_cnt) which stores the link device status

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_LINK_DEVICE_RANGE_OVER	The "word_num" + "word_cnt" designated by the argument exceeds the size of the link device.



Point

When getting the devices of the motion control station, use the sscGetInputDeviceWord function, sscGetInputDeviceDword function, sscChangeOutputDeviceWord function or sscChangeOutputDeviceDword function.

Supported version

API version	Board version	Header file
1.30	05	em4xxstd.h

Reference

- sscGetBitLinkDevice ( Page 176 sscGetBitLinkDevice)
- sscSetWordLinkDevice ( Page 180 sscSetWordLinkDevice)

sscSetBitLinkDevice

More than one link device of the general station is set on a 1-bit basis.

```
int sscSetBitLinkDevice (
    int board_id,
    int channel,
    int dev_type,
    int bit_num,
    int bit_cnt,
    int *link_dev
);
```

Detailed description

Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
dev_type	Link device type which sets the bit value Value <ul style="list-style-type: none"> SSC_RY: RY (Output) SSC_LB: LB (Link relay) SSC_SB: SB (Link special relay)
bit_num [in]	Link device No. <ul style="list-style-type: none"> RY: 0000h to 3FFFh LB: 0000h to 7FFFh SB: 00h to 1Fh
bit_cnt [in]	Bit points to set <ul style="list-style-type: none"> RY: 0001h to 4000h LB: 0001h to 8000h SB: 01h to 20h
link_dev [in]	Pointer to the array (4 bytes × bit_cnt) which stores the setting data Value <ul style="list-style-type: none"> SSC_BIT_OFF: Signal OFF SSC_BIT_ON: 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_LINK_DEVICE_RANGE_OVER	The "bit_num" + "bit_cnt" designated by the argument exceeds the size of the link device.



Point

When getting the devices of the motion control station, use the sscSetOutputDeviceBit function/ sscSetOutputDeviceBitNonExclusively function.

Supported version

API version	Board version	Header file
1.30	05	em4xxstd.h

Reference

- sscGetBitLinkDevice ( Page 176 sscGetBitLinkDevice)
- sscSetWordLinkDevice ( Page 180 sscSetWordLinkDevice)

sscSetWordLinkDevice

More than one link device of the general station is set on a 1-word basis.

```
int sscSetWordLinkDevice (  
    int board_id,  
    int channel,  
    int dev_type,  
    int word_num,  
    int word_cnt,  
    unsigned short * link_dev  
);
```

Detailed description

Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
dev_type	Link device type which gets the bit value ■Value • SSC_RWr: RWr (Input) • SSC_LW: LW (Link register) • SSC_SW: SW (Link special register)
word_num [in]	Link device No. • RWw: 0000h to 1FFFh • LW: 0000h to 3FFFh • SW: 00h to 1Fh
word_cnt [in]	Word points to get • RWw: 0001h to 2000h • LW: 0001h to 4000h • SW: 01h to 20h
link_dev [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_LINK_DEVICE_RANGE_OVER	The "word_num" + "word_cnt" designated by the argument exceeds the size of the link device.


Point

When getting the devices of the motion control station, use the sscSetOutputDeviceWord function, sscSetOutputDeviceDword function, sscChangeOutputDeviceWord function or sscChangeOutputDeviceDword function.

Supported version

API version	Board version	Header file
1.30	05	em4xxstd.h

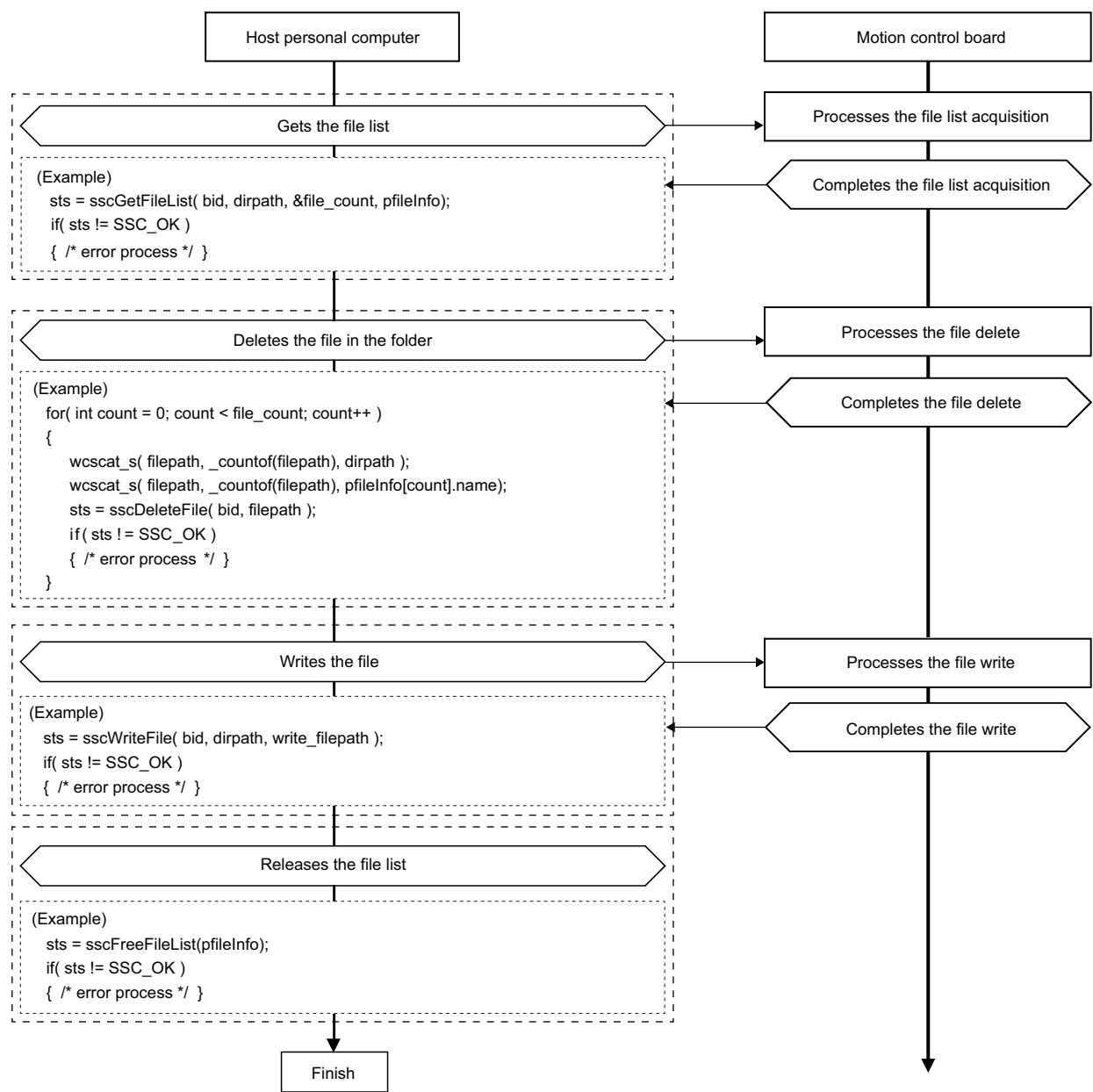
Reference

- sscGetWordLinkDevice ( Page 178 sscGetWordLinkDevice)
- sscSetBitLinkDevice ( Page 179 sscSetBitLinkDevice)

3.22 File Operation Functions

Processing procedure

An example of writing the file processing procedure is shown below.



sscWriteFile

The file is written.

```
int sscWriteFile (  
    int board_id,  
    wchar_t *dst_dirpath,  
    wchar_t *src_filepath  
);
```

Detailed description

Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
dst_dirpath [in]	Pointer to the variable (within 127 characters) which stores the folder path of the write destination Use the Unicode (UTF-16LE) for a character code.
src_filepath [in]	Pointer to the variable (any size) which stores the file path of the file to be written Use the Unicode (UTF-16LE) for a character code.

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_WRITE_FILE	The writing of the file is failed.
SSC_FUNC_ERR_OPEN_FILE	The file cannot be opened.
SSC_FUNC_ERR_CLOSE_FILE	The file could not be closed.
SSC_FUNC_ERR_FILE_SIZE	The file size is 0 byte.
SSC_FUNC_ERR_TIMEOUT_0□ □ = 1 to 8: Timeout location	During the confirmation of response after executing the command to the Motion control board, the timeout time (10 seconds) has elapsed.
SSC_FUNC_ERR_ACCESS_INVALID_DRIVE_PATH	The drive which cannot be accessed is specified.

Point

- This API function can write the file of the network parameters and the device parameters to the Motion control board.
- For the end of the argument dst_dirpath, add the directory identifier (\). If the directory identifier is not added, this API function is failed. (Detailed error code: SSC_FUNC_ERR_ARGUMENT_02)
- The argument dst_dirpath is the absolute path which starts from the drive identifier. Start the path with the directory identifier (\). If the directory identifier is not added, this API function is failed. (Detailed error code: SSC_FUNC_ERR_ARGUMENT_02)
(Example) \rom\
(Example) \rom\
- The following characters can be used for the file path. The multibyte characters cannot be used.

Available character	Range
Alphanumeric	a to z, A to Z, 0 to 9
Symbol	! # \$ ' () - . _ ` space
Delimiter for the directory	\

- For the path which is stored in the argument dst_dirpath, capital letters and lower case letters are not distinguished. However, for the directory identifiers, designate them all in lower case letters.
- For the file name, designate it including the extension.

- The number of characters of the write destination file path (total characters of the folder path of the argument `dst_dirpath` and the file name included in the argument `src_filepath`) is within 127 characters. If the number of characters is 128 or over, this API function is failed. (Detailed error code: `SSC_FUNC_ERR_ARGUMENT_02`)
- This API function waits until the write is completed. The expected write time is shown below. The write time differs depending on the file size and the load status of the system. When the drive capacity gets lower, the write time of the user drive may get longer.

File size	Time
1K bytes × 1 file	1 to 2 seconds
100K bytes × 1 file	2 to 3 seconds
10M bytes × 1 file	230 seconds

- For the `wchar_t*` type argument of the API functions for the Visual C# project, replace it with the string type.
- The network parameter file and the device parameter file can be written using this API function. The write destination path and the file name are shown below.

Type	Write destination path	File name
Network parameter	\rom\	NWParam.bin
		NetworkParameterProject.LZM
		NWParamCommon.bin
Device parameter	\rom\%MELPRJ%\%CC_IE_TSN%\	SLAVExxxxyyy.NSP (xxxx and yyyy are the values in order to distinguish the device station.)

■Supported version

API version	Board version	Header file
1.00	01	em4xxstd.h

■Reference

`sscReadFile` ( Page 184 `sscReadFile`)

sscReadFile

The file is read.

```
int sscReadFile (  
    int board_id,  
    wchar_t *dst_dirpath,  
    wchar_t *src_filepath  
);
```

Detailed description

Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
dst_dirpath [in]	Pointer to the variable which stores the folder path of the read destination Use the Unicode (UTF-16LE) for a character code.
src_filepath [in]	Pointer to the variable (within 127 characters) which stores the file path of the file to be read Use the Unicode (UTF-16LE) for a character code.

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_EXIST_DIRPATH	The folder of the read destination does not exist.
SSC_FUNC_ERR_NOT_EXIST_FILE	The file to be read does not exist.
SSC_FUNC_ERR_READ_FILE	The reading of the file is failed.
SSC_FUNC_ERR_OPEN_FILE	The file cannot be opened.
SSC_FUNC_ERR_CLOSE_FILE	The file could not be closed.
SSC_FUNC_ERR_TIMEOUT_0□ □ = 1 to 8: Timeout location	During the confirmation of response after executing the command to the Motion control board, the timeout time (10 seconds) has elapsed.
SSC_FUNC_ERR_ACCESS_INVALID_DRIVE_PATH	The drive which cannot be accessed is specified.

Point

- This API function can read the file of the network parameters and the device parameters from the Motion control board.
- When the same name file is already existed in the read destination folder, the file is overwritten.
- For the end of the argument dst_dirpath, add the directory identifier (\). If the directory identifier is not added, this API function is failed. (Detailed error code: SSC_FUNC_ERR_ARGUMENT_02)
- The argument src_filepath is the absolute path which starts from the drive identifier. Start the path with the directory identifier (\). If the directory identifier is not added, this API function is failed. (Detailed error code: SSC_FUNC_ERR_ARGUMENT_03)
(Example) \rom\NetworkParameterProject.LZM
- The following characters can be used for the file path. The multibyte characters cannot be used.

Available character	Range
Alphanumeric	a to z, A to Z, 0 to 9
Symbol	! # \$ ' () - . _ ` space
Delimiter for the directory	\

- For the path which stored in the argument src_filepath, capital letters and lower case letters are not distinguished. However, for the directory identifiers, designate them all in lower case letters.

- When the total length of the path, the file name, and the folder path of the read destination designated by the argument exceeds 259 characters, this API function is failed. (Detailed error code: SSC_FUNC_ERR_ARGUMENT_02)
- When the file cannot be saved in the read destination folder, this API function is failed. (Detailed error code: SSC_FUNC_ERR_NOT_EXIST_DIRPATH)
- For the file name, designate it including the extension.
- This API function waits until the read is completed. The expected read time is shown below. The read time differs depending on the file size and the load status of the system.

File size	Time
1K bytes × 1 file	1 to 2 seconds
100K bytes × 1 file	2 to 3 seconds
10M bytes × 1 file	50 seconds

- For the `wchar_t*` type argument of the API functions for the Visual C# project, replace it with the string type.

■Supported version

API version	Board version	Header file
1.00	01	em4xxstd.h

■Reference

sscWriteFile ( Page 182 sscWriteFile)

sscDeleteFile

The file is deleted.

```
int sscDeleteFile (  
    int board_id,  
    wchar_t *filepath  
);
```

Detailed description

Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
filepath [in]	Pointer to the variable (within 127 characters) which stores the file path of the file to be deleted Use the Unicode (UTF-16LE) for a character code.

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_EXIST_FILE	The file to be deleted does not exist.
SSC_FUNC_ERR_DELETE_FILE	The deleting of the file is failed.
SSC_FUNC_ERR_TIMEOUT_0□ □ = 1 to 8: Timeout location	During the confirmation of response after executing the command to the Motion control board, the timeout time (10 seconds) has elapsed.
SSC_FUNC_ERR_ACCESS_INVALID_DRIVE_PATH	The drive which cannot be accessed is specified.

Point

- The argument filepath is the absolute path which starts from the drive identifier. Start the path with the directory identifier (\).
If the directory identifier is not added, this API function is failed. (Detailed error code: SSC_FUNC_ERR_ARGUMENT_02)
(Example) \rom\NetworkParameterProject.LZM
- The following characters can be used for the file path. The multibyte characters cannot be used.

Available character	Range
Alphanumeric	a to z, A to Z, 0 to 9
Symbol	! # \$ ' () - . _ ` space
Delimiter for the directory	\

- Capital letters and lower case letters are not distinguished. However, for the directory identifiers, designate all in lower case letters.
- For the file name, designate it including the extension.
- For the wchar_t* type argument of the API functions for the Visual C# project, replace it with the string type.

Supported version

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

None.

sscCreateDirectory

The directory is created.

```
int sscCreateDirectory (
    int board_id,
    wchar_t *dirpath
);
```

Detailed description

Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
dirpath [in]	Pointer to the variable (within 127 characters) which stores the directory path to be generated Use the Unicode (UTF-16LE) for a character code.

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_CREATE_DIRECTORY	Creating the directory is failed.
SSC_FUNC_ERR_TIMEOUT_0□ □ = 1 to 8: Timeout location	During the confirmation of response after executing the command to the Motion control board, the timeout time (10 seconds) has elapsed.
SSC_FUNC_ERR_ACCESS_INVALID_DRIVE_PATH	The drive which cannot be accessed is specified.

Point

- The argument dirpath is the absolute path which starts from the drive identifier. Start the path with the directory identifier (\). If the directory identifier is not added, this API function is failed. (Detailed error code: SSC_FUNC_ERR_ARGUMENT_02)
- For the end of the argument dirpath, add the directory identifier (\). If the directory identifier is not added, this API function is failed. (Detailed error code: SSC_FUNC_ERR_ARGUMENT_02)
(Example) \rom\NetworkParameterProject\
- The following characters can be used for the file path. The multibyte characters cannot be used.

Available character	Range
Alphanumeric	a to z, A to Z, 0 to 9
Symbol	! # \$ ' () - . _ ` space
Delimiter for the directory	\

- Capital letters and lower case letters are not distinguished. However, for the directory identifiers, designate all in lower case letters.
- For the wchar_t* type argument of the API functions for the Visual C# project, replace it with the string type.
- When a root directory of the drive is designated to the argument dirpath, this API function is failed. (Detailed error code: SSC_FUNC_ERR_CREATE_DIRECTORY)
- Creating multiple layer directories at once is available.
- If the existed directory is designated to the argument dirpath, the directory status does not change, however this API function is succeeded.

Supported version

API version	Board version	Header file
1.20	01	em4xxstd.h

■Reference

sscDeleteDirectory ( Page 189 sscDeleteDirectory)

sscDeleteDirectory

The directory is deleted.

```
int sscDeleteDirectory (
    int board_id,
    wchar_t *dirpath
);
```

Detailed description

Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
dirpath [in]	Pointer to the variable (within 127 characters) which stores the directory path to be deleted Use the Unicode (UTF-16LE) for a character code.

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_EXIST_DIRPATH	The directory path does not exist.
SSC_FUNC_ERR_DELETE_DIRECTORY	The deleting of the directory is failed.
SSC_FUNC_ERR_TIMEOUT_0□ □ = 1 to 8: Timeout location	During the confirmation of response after executing the command to the Motion control board, the timeout time (10 seconds) has elapsed.
SSC_FUNC_ERR_ACCESS_INVALID_DRIVE_PATH	The drive which cannot be accessed is specified.

Point

- The argument dirpath is the absolute path which starts from the drive identifier. Start the path with the directory identifier (\). If the directory identifier is not added, this API function is failed. (Detailed error code: SSC_FUNC_ERR_ARGUMENT_02)
- For the end of the argument dirpath, add the directory identifier (\). If the directory identifier is not added, this API function is failed. (Detailed error code: SSC_FUNC_ERR_ARGUMENT_02)
(Example) \rom\NetworkParameterProject\
- The following characters can be used for the file path. The multibyte characters cannot be used.

Available character	Range
Alphanumeric	a to z, A to Z, 0 to 9
Symbol	! # \$ % ' () - . _ ` space
Delimiter for the directory	\

- Capital letters and lower case letters are not distinguished. However, for the directory identifiers, designate them all in lower case letters.
- For the wchar_t* type argument of the API functions for the Visual C# project, replace it with the string type.
- When a root directory of the drive is designated to the argument dirpath, this API function is failed. (Detailed error code: SSC_FUNC_ERR_DELETE_DIRECTORY)
- The directory which is not empty cannot be deleted. (Detailed error code: SSC_FUNC_ERR_DELETE_DIRECTORY)

Supported version

API version	Board version	Header file
1.20	01	em4xxstd.h

Reference

sscCreateDirectory ( Page 187 sscCreateDirectory)


sscGetFileList

The file list is got.

```
int sscGetFileList (  
    int board_id,  
    wchar_t *dirpath,  
    unsigned int *file_count,  
    FILE_INFO **fileList  
);
```

Detailed description

Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
dirpath [in]	Pointer to the variable (within 127 characters) which stores the folder path of the specified file list. Use the Unicode (UTF-16LE) for a character code.
file_count [out]	Pointer to 4-byte variable (4 bytes × 1) which stores the number of files
fileList [out]	Pointer to the structure array which stores the file list (the number of array elements is Argument (file_count)) For the file information structure, refer to the following.  Page 240 FILE_INFO_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_NOT_EXIST_DIRPATH	The folder pass does not exist.
SSC_FUNC_ERR_GET_FILE_LIST	The acquisition of the file list is failed.
SSC_FUNC_ERR_TIMEOUT_0□ □ = 1 to 8: Timeout location	During the confirmation of response after executing the command to the Motion control board, the timeout time (10 seconds) has elapsed.
SSC_FUNC_ERR_ACCESS_INVALID_DRIVE_PATH	The drive which cannot be accessed is specified.

Point

- The argument fileList is the pointer in the array data range of the file information structure (file list). Make sure to call the sscFreeFileList function, and release the file list range.
- For the end of the argument dirpath, add the directory identifier (\). If the directory identifier is not added, this API function is failed. (Detailed error code: SSC_FUNC_ERR_ARGUMENT_02)
- The argument dirpath is the absolute path which starts from the drive identifier. Start the path with the directory identifier (\). If the directory identifier is not added, this API function is failed. (Detailed error code: SSC_FUNC_ERR_ARGUMENT_02)
(Example) \rom\
- The following characters can be used for the file path. The multibyte characters cannot be used.

Available character	Range
Alphanumeric	a to z, A to Z, 0 to 9
Symbol	! # \$ ' () - . _ ` space
Delimiter for the directory	\

- Capital letters and lower case letters are not distinguished. However, for the directory identifiers, designate them all in lower case letters.
- If the return value of this API function is other than SSC_OK, the value is not stored in the argument file_count and the argument fileList.
- For the wchar_t* type argument of the API functions for the Visual C# project, replace it with the string type.





- An example of the API function for the Visual C# project is shown below.

```
using System.Runtime.InteropServices;
IntPtr file_list = IntPtr.Zero;
string dir_path = "\\rom\\";
uint file_count = 0;
SscApi.sscGetFileList(board_id, dir_path, out file_count, out file_list);
FILE_INFO[] file_info = new FILE_INFO[file_count];
int size = Marshal.SizeOf(typeof(FILE_INFO));
for (int i = 0; i < file_count; i++)
{
    IntPtr addr = (IntPtr)((int)file_list + size * i);
    file_info[i].Initialize();
    file_info[i] = (FILE_INFO)Marshal.PtrToStructure(addr, typeof(FILE_INFO));
}
SscApi.sscFreeFileList(file_list);
```

■Supported version

API version	Board version	Header file
1.00	01	em4xxstd.h

■Reference

- sscWriteFile ( Page 182 sscWriteFile)
- sscReadFile ( Page 184 sscReadFile)
- sscDeleteFile ( Page 186 sscDeleteFile)
- sscFreeFileList ( Page 194 sscFreeFileList)


sscGetFileListEx

The file list is got.

```
int sscGetFileListEx (  
    int board_id,  
    wchar_t *dirpath,  
    unsigned int *file_count,  
    FILE_INFO_EX **fileList  
);
```

Detailed description

Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
dirpath [in]	Pointer to the variable (within 127 characters) which stores the folder path of the specified file list. Use the Unicode (UTF-16LE) for a character code.
file_count [out]	Pointer to 4-byte variable (4 bytes × 1) which stores the number of files
fileList [out]	Pointer to the structure array which stores the file list (the number of array elements is Argument (file_count)) For the file information structure, refer to the following.  Page 240 FILE_INFO_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_NOT_EXIST_DIRPATH	The folder pass does not exist.
SSC_FUNC_ERR_GET_FILE_LIST	The acquisition of the file list is failed.
SSC_FUNC_ERR_TIMEOUT_0□ □ = 1 to 8: Timeout location	During the confirmation of response after executing the command to the Motion control board, the timeout time (10 seconds) has elapsed.
SSC_FUNC_ERR_ACCESS_INVALID_DRIVE_PATH	The drive which cannot be accessed is specified.

Point

- The argument fileList is the pointer in the array data range of the file information structure (file list). Make sure to call the sscFreeFileList function, and release the file list range.
- For the end of the argument dirpath, add the directory identifier (\). If the directory identifier is not added, this API function is failed. (Detailed error code: SSC_FUNC_ERR_ARGUMENT_02)
- The argument dirpath is the absolute path which starts from the drive identifier. Start the path with the directory identifier (\). If the directory identifier is not added, this API function is failed. (Detailed error code: SSC_FUNC_ERR_ARGUMENT_02)
(Example) \rom\
- The following characters can be used for the file path. The multibyte characters cannot be used.

Available character	Range
Alphanumeric	a to z, A to Z, 0 to 9
Symbol	! # \$ ' () - . _ ` space
Delimiter for the directory	\

- Capital letters and lower case letters are not distinguished. However, for the directory identifiers, designate them all in lower case letters.
- If the return value of this API function is other than SSC_OK, the value is not stored in the argument file_count and the argument fileList.
- For the wchar_t* type argument of the API functions for the Visual C# project, replace it with the string type.





- An example of the API function for the Visual C# project is shown below.

```
using System.Runtime.InteropServices;
IntPtr file_list = IntPtr.Zero;
string dir_path = "\\rom\\";
uint file_count = 0;
SscApi.sscGetFileListEx(board_id, dir_path, out file_count, out file_list);
FILE_INFO_EX[] file_info = new FILE_INFO_EX[file_count];
int size = Marshal.SizeOf(typeof(FILE_INFO_EX));
for (int i = 0; i < file_count; i++)
{
    IntPtr addr = (IntPtr)((int)file_list + size * i);
    file_info[i].Initialize();
    file_info[i] = (FILE_INFO_EX)Marshal.PtrToStructure(addr, typeof(FILE_INFO_EX));
}
SscApi.sscFreeFileListEx(file_list);
```

■Supported version

API version	Board version	Header file
1.20	01	em4xxstd.h

■Reference

- sscWriteFile ( Page 182 sscWriteFile)
- sscReadFile ( Page 184 sscReadFile)
- sscDeleteFile ( Page 186 sscDeleteFile)
- sscFreeFileListEx ( Page 195 sscFreeFileListEx)


sscFreeFileList

The file list is released.

```
int sscFreeFileList (  
    FILE_INFO *fileList  
);
```

Detailed description

Argument

Argument	Description
fileList [in]	Pointer to the structure array which stores the file list For the file information structure, refer to the following.  Page 240 FILE_INFO_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.

Point

- Call this API function, and release the file list range which is got by the sscGetFileList function.
- For the example of this API function in the Visual C# project, refer to the following.
 Page 190 sscGetFileList

Supported version

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

sscGetFileList ( Page 190 sscGetFileList)


sscFreeFileListEx

The file list is released.

```
int sscFreeFileListEx (
    FILE_INFO_EX *fileList
);
```

Detailed description

■Argument

Argument	Description
fileList [in]	Pointer to the structure array which stores the file list For the file information structure, refer to the following.  Page 240 FILE_INFO_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.

■Point

- Call this API function, and release the file list range which is got by the sscGetFileListEx function.
- For the example of this API function in the Visual C# project, refer to the following
 Page 192 sscGetFileListEx

■Supported version

API version	Board version	Header file
1.20	01	em4xxstd.h

■Reference

sscGetFileListEx ( Page 192 sscGetFileListEx)

3.23 SLMP Functions



sscSimpReadSlaveObject

The object of the device which is connected to the network is read.

```
int sscSimpReadSlaveObject (  
    int board_id,  
    int channel,  
    int read_type,  
    int read_size,  
    void *read_data,  
    SLMP_SLV_OBJ_CMD *pSimpSlvObjCmd,  
    SLMP_SLV_OBJ_STS *pSimpSlvObjSts  
);
```

Detailed description




■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
read_type [in]	Read type ■Value <ul style="list-style-type: none">SSC_RW_SLAVE_OBJ: Read the designated object by index and subindex using the ReadObject (command: 4020, subcommand: 0001) of SLMP.SSC_RW_SLAVE_OBJ_SUB_BLOCK: Read the designated index object from the designated head subindex sequentially using the ObjectSubIDReadBlock (command: 4020, subcommand: 0005) of SLMP.
read_size [in]	Read size [byte] ■SSC_RW_SLAVE_OBJ <ul style="list-style-type: none">Size of the designated object (1 to 1884) ■SSC_RW_SLAVE_OBJ_SUB_BLOCK <ul style="list-style-type: none">Total size which is read from the designated object sequentially (1 to 1884)
read_data [out]	Pointer to the variable (1 byte × read_size) which stores the read data
pSimpSlvObjCmd [in]	Pointer to 32-byte structure which stores the read/write command data of the slave object For the read/write command data structure of the slave object, refer to the following.  Page 242 SLMP_SLV_OBJ_CMD Structure
pSimpSlvObjSts [out]	Pointer to 32-byte structure which stores the read/write status data of the slave object For the read/write status data structure of the slave object, refer to the following.  Page 243 SLMP_SLV_OBJ_STS 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 Motion control board, the timeout time (1 second) has elapsed.
SSC_FUNC_ERR_SLMP_SEND_RECEIVE	An error occurred during the SLMP communication. An error code is stored in completion_status of the SLMP_SLV_OBJ_STS structure. Refer to the following manuals to confirm the error code and take actions.  Motion Control Board User's Manual (Motion Control)  Motion Control Board User's Manual (Network)
SSC_FUNC_ERR_SLMP_END_CODE	An error occurred during the SLMP communication. An end code of the SLMP response message is stored in end_code of the SLMP_SLV_OBJ_STS structure. Refer to the following manual to confirm the error code and take actions.  MELSEC iQ-R CC-Link IE TSN User's Manual (Application)
SSC_FUNC_ERR_READ_SIZE_MISMATCH	The object of the designated size could not be read. Review the read size.

■Point

- When sending an SLMP message other than the read/write of the object to the device, use the sscSlmpSendCommand function.
- When calling the SLMP function from multiple threads at the same time, the SLMP send/receive is not executed at the same time but executed in order.

■Supported version

API version	Board version	Header file
1.00	01	em4xxstd.h

■Reference

sscSlmpWriteSlaveObject ( Page 198 sscSlmpWriteSlaveObject)

sscSimpWriteSlaveObject

The object of the device which is connected to the network is written.

```
int sscSimpWriteSlaveObject (  
    int board_id,  
    int channel,  
    int write_type,  
    int write_size,  
    void *write_data,  
    SLMP_SLV_OBJ_CMD *pSimpSlvObjCmd,  
    SLMP_SLV_OBJ_STS *pSimpSlvObjSts  
);
```

Detailed description

Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
write_type [in]	Write type ■Value • SSC_RW_SLAVE_OBJ: Write the designated object by index and subindex using the WriteObject (command: 4020, subcommand: 0002) of SLMP. • SSC_RW_SLAVE_OBJ_SUB_BLOCK: Write the designated index object from the designated head subindex sequentially using the ObjectSubIDWriteBlock (command: 4020, subcommand: 0006) of SLMP.
write_size [in]	Write size [byte] ■SSC_RW_SLAVE_OBJ • Size of the designated object (1 to 1884) ■SSC_RW_SLAVE_OBJ_SUB_BLOCK • Total size which is written from the designated object sequentially (1 to 1884)
write_data [in]	Pointer to the variable (1 byte × write_size) which stores the data to be written
pSimpSlvObjCmd [in]	Pointer to 32-byte structure which stores the read/write command data of the slave object For the read/write command data structure of the slave object, refer to the following. 📖 Page 242 SLMP_SLV_OBJ_CMD Structure
pSimpSlvObjSts [out]	Pointer to 32-byte structure which stores the read/write status data of the slave object For the read/write status data structure of the slave object, refer to the following. 📖 Page 243 SLMP_SLV_OBJ_STS 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 Motion control board, the timeout time (1 second) has elapsed.
SSC_FUNC_ERR_SLMP_SEND_RECEIVE	An error occurred during the SLMP communication. An error code is stored in completion_status of the SLMP_SLV_OBJ_STS structure. Refer to the following manuals to confirm the error code and take actions. 📖 Motion Control Board User's Manual (Motion Control) 📖 Motion Control Board User's Manual (Network)
SSC_FUNC_ERR_SLMP_END_CODE	An error occurred during the SLMP communication. An end code of the SLMP response message is stored in end_code of the SLMP_SLV_OBJ_STS structure. Refer to the following manual to confirm the error code and take actions. 📖 IMELSEC iQ-R CC-Link IE TSN User's Manual (Application)

■Point

- When sending an SLMP message other than the read/write of the object to the device, use the `sscSlmpSendCommand` function.
- When calling the SLMP function from multiple threads at the same time, the SLMP send/receive is not executed at the same time but executed in order.

■Supported version

API version	Board version	Header file
1.00	01	em4xxstd.h

■Reference

`sscSlmpReadSlaveObject` ( Page 196 `sscSlmpReadSlaveObject`)





sscSlmpSendCommand

The SLMP message is sent/received to the device which is connected to the network.

```
int sscSlmpSendCommand(  
    int board_id,  
    int channel,  
    SLMP_CONTROL *pSlmpControlData,  
    void *pSlmpRequestData,  
    void *pSlmpResponseData,  
    int response_buffer_size  
);
```

Detailed description

Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
pSlmpControlData [in/out]	Pointer to the structure which stores the SLMP control data For the SLMP control data, refer to the following.  Page 244 SLMP_CONTROL Structure
pSlmpRequestData [in]	Pointer to the variable which stores the SLMP request data ^{*1} For the data length and the detail of the request data, refer to the following manual.  SLMP Reference Manual
pSlmpResponseData [out]	Pointer to the variable which stores the SLMP response data ^{*2} For the data length and the detail of the response data, refer to the following manual.  SLMP Reference Manual
response_buffer_size [in]	The size of the variable (pSlmpResponseData) which stores the SLMP response data [byte] (2 to 2000) For the data length of the response data, refer to the following manual.  SLMP Reference Manual

^{*1} The request data form is shown below.

Offset [byte]	Size [byte]	Item	Description	Setting range
00h	2	Request data length	Sets the data length from the monitoring timer to the request data. (unit: byte)	2 to 2000
02h	2	Monitoring timer	Sets the waiting time from when the external device which received the request message requests to the access destination for processing until when a response is returned. (unit: 250ms) • 0: Unlimited wait • 1 to 65535: 1 to 65535 × 250ms	0 to 65535
04h	Variable length	Request data	Stores the request data of the SLMP message.	—

^{*2} The response data form is shown below.

Offset [byte]	Size [byte]	Item	Description	Setting range
00h	2	Response data length	Stores the data length from the end code to the response data (unit: byte)	—
02h	2	End code	Stores the processing result of the request data. • In normal termination: "0" is stored. • In abnormal termination: The error code of the external device is stored. For details, refer to the specification of the external device.	—
04h	Variable length	Response data	Stores the read data against the request data. (Some request data does not reply to the response 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_ARGUMENT_SLMP_REQUEST	The request data length of the SLMP request data is outside the set range.
SSC_FUNC_ERR_ARGUMENT_SLMP_CONTROL	The arrival monitoring time or the number of resends of the SLMP control data structure is outside the set range.
SSC_FUNC_ERR_TIMEOUT_01	During the confirmation of response after executing the command to the Motion control board, the timeout time (1 second) has elapsed.
SSC_FUNC_ERR_SLMP_RESPONSE_SIZE_OVER	The size of the response data exceeds the size designated in the argument response_buffer_size.
SSC_FUNC_ERR_SLMP_SEND_RECEIVE	An error occurred during the SLMP communication. An error code is stored in completion_status of the SLMP_CONTROL structure. Refer to the following manuals to confirm the error code and take actions. Motion Control Board User's Manual (Motion Control) Motion Control Board User's Manual (Network)

■Point

- This API function waits until the SLMP send/receive is completed. The expected standby time is shown below.
Rough time = Arrival monitoring time × (Number of resends + 1)
- For the monitoring timer of the arrival monitoring time and the request frame of the control data, set them to "Arrival monitoring time ≥ Monitoring timer.
- This API function is completed normally even the target device replies the error response. Even if this API function is completed normally, judge whether it is the normal response or the error response by the end code of the response frame. In case of the error response, refer to the manual of the using SLMP compatible device and take actions.
- This API function sends the request data to the external device which is designated by the IP address of the external device in the control data.
- The communication method is UDP/IP, and the communication is done by the binary code.
- When calling the SLMP functions from multiple threads at the same time, the SLMP send/receive is not executed at the same time but executed in order.
- When executing the read/write of the object against the device, use the sscSlmpReadSlaveObject function/ sscSlmpWriteSlaveObject function.
- The example of this API function (the data read from the remote buffer memory of the remote I/O (NZ2GN2S1-32D)) is shown below.

```
SLMP_CONTROL slmpControlData = {0};
unsigned char slmpRequestData[20] = {0};
unsigned char slmpResponseData[16] = {0};
```

```
int response_buffer_size = sizeof(slmpResponseData) / sizeof(slmpResponseData[0]);
```

```
slmpControlData.completion_type = SSC_SLMP_CHECK_ARRIVAL_ENABLE | SSC_SLMP_STORE_ERROR_INFO_ENABLE;
slmpControlData.channel_use_own_station = 10;
slmpControlData.ip_address_4 = 1;
slmpControlData.ip_address_3 = 3;
slmpControlData.ip_address_2 = 168;
slmpControlData.ip_address_1 = 192;
slmpControlData.port_number = 45239;
slmpControlData.network_number = 0x0000;
slmpControlData.station_number = 0x00FF;
slmpControlData.io_number = SSC_SLMP_IO_CPU_OWN_MANAGEMENT;
slmpControlData.multi_drop_number = 0;
slmpControlData.retry_count = 3;
slmpControlData.arrival_moni_time = 10;
```

```
/* Create SLMP request "Reads data from the remote buffer memory (command: 0x0613, subcommand: 0x0000)". */
*(unsigned short*)&slmpRequestData[0] = 18;          /* data_length[byte] */
*(unsigned short*)&slmpRequestData[2] = 500;          /* watchdog_timer[ms] */
*(unsigned short*)&slmpRequestData[4] = 0x0613;       /* slmp_command */
*(unsigned short*)&slmpRequestData[6] = 0x0000;       /* slmp_sub_command */
*(unsigned long*)&slmpRequestData[14] = 0x00000000;   /* head address */
*(unsigned short*)&slmpRequestData[18] = 0x0001;     /* word length */
```

```
ans = sscSlmpSendCommand(board_id, channel, &slmpControlData, slmpRequestData, slmpResponseData, response_buffer_size);
```

■Supported version

API version	Board version	Header file
1.20	01	em4xstd.h

■Reference

None.

3.24 General Input/Output Functions

sscGetGeneralInputDataBit

The designated general input/output DI data is got on a 1-point basis.

```
int sscGetGeneralInputDataBit (  
    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]	Input pin No. (0 to 3)
din [out]	Pointer to 4-byte variable (4 bytes × 1) which stores the general input DI data status ■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

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

sscGetGeneralInputDataWord ( Page 204 sscGetGeneralInputDataWord)

sscGetGeneralInputDataWord

The designated general input/output DI data is got on a 4-point basis.

```
int sscGetGeneralInputDataWord (  
    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]	Input pin word No. (0)
din [out]	Pointer to 2-byte variable (2 bytes × 1) which stores the general input DI data status (For each bit, 0: input OFF, 1: input ON) <ul style="list-style-type: none">• Bit0: DI0• Bit1: DI1• Bit2: DI2• Bit3: DI3

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

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

sscGetGeneralInputDataBit ( Page 203 sscGetGeneralInputDataBit)

sscSetGeneralOutputDataBit

The designated general input/output DO data is set on a 1-point basis.

```
int sscSetGeneralOutputDataBit (
    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]	Output pin No. (0 to 3)
dout [in]	DO data of the general output ■Value <ul style="list-style-type: none"> • 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

API version	Board version	Header file
1.00	01	em4xstd.h

■Reference

sscSetGeneralOutputDataWord ( Page 207 sscSetGeneralOutputDataWord)

sscSetGeneralOutputDataBitExclusively

The designated general input/output DO data is set on a 1-point basis with the general output exclusive control function of the Motion control board.

```
int sscSetGeneralOutputDataBitExclusively (  
    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]	Output pin No. (0 to 3)
dout [in]	DO data of the general 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_UNUPPORT_BOARD_VERSION	The Motion control board does not support this function.
SSC_FUNC_ERR_TIMEOUT_0□ □ = 1 to 5: Timeout location	During the confirmation of response after executing the command to the Motion control board, the timeout time (10 seconds) has elapsed.




Point

This API function sets the general output with the general output exclusive control function of the Motion control board.

Supported version

API version	Board version	Header file
1.30	05	em4xxstd.h

Reference

- sscSetGeneralOutputDataBit ( Page 205 sscSetGeneralOutputDataBit)
- sscSetGeneralOutputDataWordExclusively ( Page 208 sscSetGeneralOutputDataWordExclusively)
- sscGetGeneralOutputDataBit ( Page 209 sscGetGeneralOutputDataBit)

sscSetGeneralOutputDataWord

The designated general input/output DO data is set on a 4-point basis.

```
int sscSetGeneralOutputDataWord (
    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]	Output pin word No. (0)
dout [in]	DO data of the general output (For each bit, 0: output OFF, 1: output ON) <ul style="list-style-type: none"> • Bit0: DO0 • Bit1: DO1 • Bit2: DO2 • Bit3: DO3

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

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

sscSetGeneralOutputDataBit ( Page 205 sscSetGeneralOutputDataBit)

sscSetGeneralOutputDataWordExclusively

The designated general input/output DO data is set on a 4-point basis with the general output exclusive control function of the Motion control board.

```
int sscSetGeneralOutputDataWordExclusively (  
    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]	Output pin No. (0 to 3)
dout [in]	DO data of the general output (For each bit, 0: output OFF, 1: output ON) • Bit0: DO0 • Bit1: DO1 • Bit2: DO2 • Bit3: DO3

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_UNSUPPORTED_BOARD_VERSION	The Motion control board does not support this function.
SSC_FUNC_ERR_TIMEOUT_0□ □ = 1 to 5: Timeout location	During the confirmation of response after executing the command to the Motion control board, the timeout time (10 seconds) has elapsed.




Point

This API function sets the general output with the general output exclusive control function of the Motion control board.

Supported version

API version	Board version	Header file
1.30	05	em4xxstd.h

Reference

- sscSetGeneralOutputDataBitExclusively ( Page 206 sscSetGeneralOutputDataBitExclusively)
- sscSetGeneralOutputDataWord ( Page 207 sscSetGeneralOutputDataWord)
- sscGetGeneralOutputDataWord ( Page 210 sscGetGeneralOutputDataWord)

sscGetGeneralOutputDataBit

The designated general input/output DO data is got on a 1-point basis.

```
int sscGetGeneralOutputDataBit (
    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]	Output pin No. (0 to 3)
dout [out]	Pointer to 4-byte variable (4 bytes × 1) which stores the DO data of the general output status <div> <div>Value</div> <ul style="list-style-type: none"> SSC_BIT_OFF: Output signal OFF SSC_BIT_ON: Output signal ON </div>

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

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

sscGetGeneralOutputDataWord ( Page 210 sscGetGeneralOutputDataWord)

sscGetGeneralOutputDataWord

The designated general input/output DO data is got on a 4-point basis.

```
int sscGetGeneralOutputDataWord (  
    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]	Output pin word No. (0)
dout [out]	Pointer to 2-byte variable (2 bytes × 1) which stores the DO data of the general output status (For each bit, 0: output OFF, 1: output ON) <ul style="list-style-type: none">• Bit0: DO0• Bit1: DO1• Bit2: DO2• Bit3: DO3

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

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

sscGetGeneralOutputDataBit ( Page 209 sscGetGeneralOutputDataBit)

3.25 Arbitrary User Area Access Functions

sscSetUserArea

The data is written to the arbitrary user area.

```
int sscSetUserArea (  
    int board_id,  
    int channel,  
    int offset,  
    int size,  
    char *data  
);
```

Detailed description

Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
offset [in]	Offset from the head of the arbitrary user area (00000h to 3FFFFh)
size [in]	Data size to be written (00001h to 40000h)
data [in]	Pointer to the variable (designated size) which stores the data to be written

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_ADDRESS_RANGE_OVER	The "offset" + "size" designated by the argument exceeds the accessible range.

Point

None.

Supported version

API version	Board version	Header file
1.20	01	em4xxstd.h

Reference

sscGetUserArea ( Page 212 sscGetUserArea)

sscGetUserArea

The data is read from the arbitrary user area.

```
int sscGetUserArea (  
    int board_id,  
    int channel,  
    int offset,  
    int size,  
    char *data  
);
```

Detailed description

Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
offset [in]	Offset from the head of the arbitrary user area (00000h to 3FFFFh)
size [in]	Data size to be read (00001h to 40000h)
data [in]	Pointer to the variable which stores the read data (designated size)

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_ADDRESS_RANGE_OVER	The "offset" + "size" designated by the argument exceeds the accessible range.

Point

None.

Supported version

API version	Board version	Header file
1.20	01	em4xxstd.h

Reference

sscSetUserArea ( Page 211 sscSetUserArea)

3.26 Event History Functions

sscReadEventHistoryData

The event history is read

```
int sscReadEventHistoryData (  
    int board_id,  
    int channel,  
    wchar_t *dst_filepath  
);
```

Detailed description

Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
dst_filepath [in]	Pointer to the variable which stores the file path of the read destination (any size) Use the Unicode (UTF-16LE) for a character code.

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_UNUPPORT_BOARD_VERSION	The Motion control board does not support this function.
SSC_FUNC_ERR_DESTINATION_FILE_EXTENSION	The file extension of the read destination is incorrect.
SSC_FUNC_ERR_SAVE_DESTINATION_FILE	The saving of the read destination file is failed.
SSC_FUNC_ERR_READ_EVENT_HISTORY	The reading of the event history is failed.
SSC_FUNC_ERR_TIMEOUT_0□ □ = 1 to 5: Timeout location	During the confirmation of response after executing the command to the Motion control board, the timeout time has elapsed.

Point

- For the file in which the event history is read, browse it via the motion test tool.
- When the file already exists in the file path which is specified by the argument dst_filepath, the file is overwritten.
- For the end of the argument dst_filepath, add the extension (.ehs4).
- The data which is added/deleted during the read is not read.
- The API function waits until the read is completed. For the maximum time to read is around 7 to 8 seconds.
The read time differs depending on the event history size and the load status of the system.
- For the wchar_t* type argument of the API functions for the Visual C# project, replace it with the string type.

Supported version

API version	Board version	Header file
1.20	03	em4xxstd.h

Reference

None.

sscClearEventHistoryData

The event history is cleared.

```
int sscClearEventHistoryData (  
    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_UNUPPORT_BOARD_VERSION	The Motion control board does not support this function.
SSC_FUNC_ERR_CLEAR_EVENT_HISTORY	The clearing of the event history is failed.
SSC_FUNC_ERR_TIMEOUT_0□ □ = 1 to 5: Timeout location	During the confirmation of response after executing the command to the Motion control board, the timeout time has elapsed.

Point

None.

Supported version

API version	Board version	Header file
1.20	03	em4xxstd.h

Reference

None.

3.27 Pressure Control Functions


sscSetPrsProfile

The pressure control profile is set.

```
int sscSetPrsProfile (
    int board_id,
    int channel,
    int prs_num,
    PRS_PROFILE *pPrsProfile
);
```

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
prs_num [in]	Pressure control profile No. (1 to 32)
pPrsProfile [in]	Pointer to 3840-byte structure (3840 bytes × 1) which stores the pressure control profile For the pressure control structure, refer to the following.  Page 246 PRS_PROFILE 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_UNUPPORT_BOARD_VERSION	The Motion control board does not support this function.

■Point

A confirmation of the set pressure control profile contents is not performed.

■Supported version

API version	Board version	Header file
1.30	05	em4xxstd.h

■Reference

sscGetPrsProfile ( Page 216 sscGetPrsProfile)


sscGetPrsProfile

The pressure control profile is got.

```
int sscGetPrsProfile (  
    int board_id,  
    int channel,  
    int prs_num,  
    PRS_PROFILE *pPrsProfile  
);
```

Detailed description

Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
prs_num [in]	Pressure control profile No. (1 to 32)
pPrsProfile [out]	Pointer to 3840-byte structure (3840 bytes × 1) which stores the pressure control profile For the pressure control structure, refer to the following.  Page 246 PRS_PROFILE 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_UNSUPPORTED_BOARD_VERSION	The Motion control board does not support this function.

Point

None.

Supported version

API version	Board version	Header file
1.30	05	em4xxstd.h

Reference

sscSetPrsProfile ( Page 215 sscSetPrsProfile)

sscGetPrsStatus

The pressure control status is got.

```
int sscGetPrsStatus (
    int board_id,
    int channel,
    int axnum,
    unsigned short *status,
    unsigned short *feed_step,
    unsigned short *dwell_step,
    unsigned short *release_step,
);
```

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 64)
status [out]	Pointer to 2-byte variable (2 bytes × 1) which stores the pressure control status
feed_step [out]	Pointer to 2-byte variable (2 bytes × 1) which stores the step No. of feeding step in progress
dwell_step [out]	Pointer to 2-byte variable (2 bytes × 1) which stores the step No. of dwelling pressure step in progress
release_step [out]	Pointer to 2-byte variable (2 bytes × 1) which stores the step No. of releasing pressure step in progress

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_UNUPPORT_BOARD_VERSION	The Motion control board does not support this function.

Point

Data is not got for arguments that designate a Null pointer (pressure control status, step No. of feeding step in progress, step No. of dwelling step in progress, and step No. of releasing pressure step in progress).

Supported version

API version	Board version	Header file
1.30	05	em4xxstd.h

Reference

None.

sscGetExecutingPrsNumber

The profile No. of pressure control in progress is got.

```
int sscGetExecutingPrsNumber (  
    int board_id,  
    int channel,  
    int axnum,  
    unsigned short *prs_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 64)
prs_num [out]	Pointer to 2-byte variable (2 bytes × 1) which stores the profile No. of pressure control in progress

■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_UNUPPORT_BOARD_VERSION	The Motion control board does not support this function.

■Point

None.

■Supported version

API version	Board version	Header file
1.30	05	em4xxstd.h

■Reference

None.

sscGetPrsForwardSidePosition

The tip position in pressure control is got.

```
int sscGetPrsForwardSidePosition (
    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 64)
position [out]	Pointer to 4-byte variable (4 bytes × 1) which stores the tip position in pressure control

■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_UNUPPORT_BOARD_VERSION	The Motion control board does not support this function.

■Point

None.

■Supported version

API version	Board version	Header file
1.30	05	em4xxstd.h

■Reference

None.

sscGetPrsSpeedLimit

The speed limit value is got.

```
int sscGetPrsSpeedLimit (  
    int board_id,  
    int channel,  
    int axnum,  
    unsigned long *speed limit  
);
```

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 64)
speed limit [out]	Pointer to 4-byte variable (4 bytes × 1) which stores the speed limit value

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_UNUPPORT_BOARD_VERSION	The Motion control board does not support this function.

Point

None.

Supported version

API version	Board version	Header file
1.30	05	em4xxstd.h

Reference

None.

sscGetPrsPressureCommand

The pressure command value is got.

```
int sscGetPrsPressureCommand (
    int board_id,
    int channel,
    int axnum,
    short *pressure
);
```

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 64)
pressure [out]	Pointer to 2-byte variable (2 bytes × 1) which stores the pressure command value

■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_UNUPPORT_BOARD_VERSION	The Motion control board does not support this function.

■Point

None.

■Supported version

API version	Board version	Header file
1.30	05	em4xxstd.h

■Reference

None.

sscGetPrsLoadCellPressure

The load cell pressure is got.

```
int sscGetPrsLoadCellPressure (  
    int board_id,  
    int channel,  
    int axnum,  
    short *pressure  
);
```

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 64)
pressure [out]	Pointer to 2-byte variable (2 bytes × 1) which stores the load cell pressure value

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_UNUPPORT_BOARD_VERSION	The Motion control board does not support this function.

Point

None.

Supported version

API version	Board version	Header file
1.30	05	em4xxstd.h

Reference

None.

sscGetPrsError

The error information during pressure control is got.

```
int sscGetPrsError (
    int board_id,
    int channel,
    int axnum,
    unsigned short *err_step,
    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. (1 to 64)
err_step [out]	Pointer to 2-byte variable (2 bytes × 1) which stores the error step No.
detail_code [out]	Pointer to 2-byte variable (2 bytes × 1) which stores the error detail No. of switching condition

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_UNUPPORT_BOARD_VERSION	The Motion control board does not support this function.

Point

None.

Supported version

API version	Board version	Header file
1.30	05	em4xxstd.h

Reference

None.

sscSetPrsProfileLimitPosition

The limit position of the designated pressure control profile is set.

```
int sscSetPrsProfileLimitPosition (  
    int board_id,  
    int channel,  
    int prs_num,  
    int process,  
    unsigned short step_num,  
    long position  
);
```

Detailed description

Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
prs_num [in]	Profile No. of pressure control (1 to 32)
process [in]	Process of pressure control ■Value • SSC_PRS_PROCESS_FEED: Feed • SSC_PRS_PROCESS_DWELL: Dwell
step_num [in]	Step No. (1 to 32)
position [in]	Limit position (-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.
SSC_FUNC_ERR_UNSUPPORT_BOARD_VERSION	The Motion control board does not support this function.




Point

None.

Supported version

API version	Board version	Header file
1.30	05	em4xxstd.h

Reference

- sscSetPrsProfile ( Page 215 sscSetPrsProfile)
- sscGetPrsProfile ( Page 216 sscGetPrsProfile)
- sscGetPrsProfileLimitPosition ( Page 225 sscGetPrsProfileLimitPosition)

sscGetPrsProfileLimitPosition

The limit position of the designated pressure control profile is got.

```
int sscGetPrsProfileLimitPosition (
    int board_id,
    int channel,
    int prs_num,
    int process,
    unsigned short step_num,
    long position
);
```

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
prs_num [in]	Profile No. of pressure control (1 to 32)
process [in]	Process of pressure control ■Value <ul style="list-style-type: none"> SSC_PRS_PROCESS_FEED: Feed SSC_PRS_PROCESS_DWELL: Dwell
step_num [in]	Step No. (1 to 32)
position [out]	Pointer to 4-byte variable (4 bytes × 1) which stores the limit position

■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_BOARD_VERSION	The Motion control board does not support this function.




■Point

None.

■Supported version

API version	Board version	Header file
1.30	05	em4xxstd.h

■Reference

- sscSetPrsProfile ( Page 215 sscSetPrsProfile)
- sscGetPrsProfile ( Page 216 sscGetPrsProfile)
- sscSetPrsProfileLimitPosition ( Page 224 sscSetPrsProfileLimitPosition)

sscSetPrsProfileSpeedLimit

The speed limit value of the designated pressure control profile is set.

```
int sscSetPrsProfileSpeedLimit (  
    int board_id,  
    int channel,  
    int prs_num,  
    int process,  
    unsigned short step_num,  
    unsigned long speed_limit  
);
```

Detailed description

Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
prs_num [in]	Profile No. of pressure control (1 to 32)
process [in]	Process of pressure control ■Value • SSC_PRS_PROCESS_FEED: Feed
step_num [in]	Step No. (1 to 32)
speed limit [in]	Speed limit value (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_UNSUPPORT_BOARD_VERSION	The Motion control board does not support this function.




Point

None.

Supported version

API version	Board version	Header file
1.30	05	em4xxstd.h

Reference

- sscSetPrsProfile ( Page 215 sscSetPrsProfile)
- sscGetPrsProfile ( Page 216 sscGetPrsProfile)
- sscGetPrsProfileSpeedLimit ( Page 227 sscGetPrsProfileSpeedLimit)

sscGetPrsProfileSpeedLimit

The speed limit value of the designated pressure control profile is got.

```
int sscGetPrsProfileSpeedLimit (
    int board_id,
    int channel,
    int prs_num,
    int process,
    unsigned short step_num,
    unsigned long *speed_limit
);
```

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
prs_num [in]	Profile No. of pressure control (1 to 32)
process [in]	Process of pressure control ■Value • SSC_PRS_PROCESS_FEED: Feed
step_num [in]	Step No. (1 to 32)
speed limit [out]	Pointer to 4-byte variable (4 bytes × 1) which stores the speed limit value

■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_UNUPPORT_BOARD_VERSION	The Motion control board does not support this function.




■Point

None.

■Supported version

API version	Board version	Header file
1.30	05	em4xstd.h

■Reference

- sscSetPrsProfile ( Page 215 sscSetPrsProfile)
- sscGetPrsProfile ( Page 216 sscGetPrsProfile)
- sscSetPrsProfileSpeedLimit ( Page 227 sscGetPrsProfileSpeedLimit)

sscSetPrsProfilePressureCommand

The pressure command of the designated pressure control profile is set.

```
int sscSetPrsProfilePressureCommand (  
    int board_id,  
    int channel,  
    int prs_num,  
    int process,  
    unsigned short step_num,  
    short pressure_cmd  
);
```

Detailed description

Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
prs_num [in]	Profile No. of pressure control (1 to 32)
process [in]	Process of pressure control ■Value <ul style="list-style-type: none">SSC_PRS_PROCESS_FEED: FeedSSC_PRS_PROCESS_DWELL: DwellSSC_PRS_PROCESS_RELEASE: Pressure release
step_num [in]	Step No. (1 to 32) <ul style="list-style-type: none">Feed: 1 to 32Dwell: 1 to 32Pressure release: 1
pressure_cmd [in]	Pressure command (0 to 32767)

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_UNUPPORT_BOARD_VERSION	The Motion control board does not support this function.




Point

None.

Supported version

API version	Board version	Header file
1.30	05	em4xxstd.h

Reference

- sscSetPrsProfile ( Page 215 sscSetPrsProfile)
- sscGetPrsProfile ( Page 216 sscGetPrsProfile)
- sscGetPrsProfilePressureCommand ( Page 229 sscGetPrsProfilePressureCommand)

sscGetPrsProfilePressureCommand

The pressure command of the designated pressure control profile is got.

```
int sscGetPrsProfilePressureCommand (
    int board_id,
    int channel,
    int prs_num,
    int process,
    unsigned short step_num,
    short *pressure_cmd
);
```

Detailed description

■Argument

Argument	Description
board_id [in]	Board ID No. (0 to 3)
channel [in]	Channel No. (1)
prs_num [in]	Profile No. of pressure control (1 to 32)
process [in]	Process of pressure control ■Value <ul style="list-style-type: none"> SSC_PRS_PROCESS_FEED: Feed SSC_PRS_PROCESS_DWELL: Dwell SSC_PRS_PROCESS_RELEASE: Pressure release
step_num [in]	Step No. (1 to 32) <ul style="list-style-type: none"> Feed: 1 to 32 Dwell: 1 to 32 Pressure release: 1
pressure_cmd [in]	Pointer to 2-byte variable (2 bytes × 1) which stores the pressure command

■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_UNUPPORT_BOARD_VERSION	The Motion control board does not support this function.




■Point

None.

■Supported version

API version	Board version	Header file
1.30	05	em4xxstd.h

■Reference

- sscSetPrsProfile ( Page 215 sscSetPrsProfile)
- sscGetPrsProfile ( Page 216 sscGetPrsProfile)
- sscSetPrsProfilePressureCommand ( Page 228 sscSetPrsProfilePressureCommand)

4 STRUCTURE LIST

4.1 PNT_DATA_EX Structure

This point data structure is used for the automatic operation and the interpolation operation.

```
typedef struct {  
    /* 0000h */  
    long position;  
    unsigned long speed;  
    unsigned short actime;  
    unsigned short dctime;  
    unsigned short dwell;  
    unsigned short subcmd;  
    /* 0010h */  
    unsigned char oas_num[4];  
    unsigned char s_curve;  
    unsigned char prs_num;  
    char reserve1[2];  
    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]

Member	Description
subcmd	<p>Auxiliary command</p> <p>Set the data in the logical sum of each value.</p> <p>■Value</p> <p>[Position command system]</p> <ul style="list-style-type: none"> • SSC_SUBCMD_POS_ABS: Absolute position command • SSC_SUBCMD_POS_INC: Relative position command <p>[Vibration suppression command filter 1 command]</p> <ul style="list-style-type: none"> • SSC_SUBCMD_DAMPING_DISABLE: Vibration suppression command filter 1 disabled • SSC_SUBCMD_DAMPING_ENABLE: Vibration suppression command filter 1 enabled <p>[Deceleration check system]</p> <ul style="list-style-type: none"> • SSC_SUBCMD_STOP_INP: In-position stop • SSC_SUBCMD_STOP_SMZ: Smoothing stop • SSC_SUBCMD_STOP_CONTINUE: Continue operation <p>[Speed switching point specification]</p> <ul style="list-style-type: none"> • SSC_SUBCMD_PNT_SWITCH_AFTER: After point switching • SSC_SUBCMD_PNT_SWITCH_BEFORE: Before point switching <p>[Dwell specification]</p> <ul style="list-style-type: none"> • SSC_SUBCMD_DWELL: Dwell • SSC_SUBCMD_PREDWELL: Predwell <p>[Pass position interrupt specification]</p> <ul style="list-style-type: none"> • SSC_SUBCMD_PASS_POS_DISABLE: Disabled • SSC_SUBCMD_PASS_POS_ENABLE: Enabled <p>[Continuous operation to torque control specification]</p> <ul style="list-style-type: none"> • SSC_SUBCMD_PRESS_DISABLE: Disabled • SSC_SUBCMD_PRESS_ENABLE: Enabled <p>[Loop specification]</p> <ul style="list-style-type: none"> • SSC_SUBCMD_PNT_LOOP_DISABLE: Disabled • SSC_SUBCMD_PNT_LOOP_START: Loop start point • SSC_SUBCMD_PNT_LOOP_END: Loop end point <p>[Interpolation system]</p> <ul style="list-style-type: none"> • 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)
s_curve	S-curve ratio [%] (0 to 100)
prs_num	Profile No. of pressure control (1 to 32)
sub_axnum[4]	<p>Interpolation axis No. (0 to 64)</p> <p>Set the axis No. of the auxiliary axes to be in the same interpolation group.</p> <p>When not using auxiliary axes, set "0".</p>
arc_coord	<p>Arc coordinates [command unit]</p> <p>Set the auxiliary point of the arc or the coordinates of the central point.</p> <p>Settings vary according to the interpolation system.</p>
ac_dc_data[4]	Acceleration/deceleration data [0.1%] (0 to 1000)
subcmd2	<p>Auxiliary command 2</p> <p>Set the data in the logical sum of each value.</p> <p>■Value</p> <p>[Acceleration/deceleration system]</p> <ul style="list-style-type: none"> • SSC_SUBCMD2_ACCDEC_LINE_S: Linear acceleration/deceleration and S-curve acceleration/deceleration • SSC_SUBCMD2_ACCDEC_JERK: Jerk ratio acceleration/deceleration <p>[Pressure control specification]</p> <ul style="list-style-type: none"> • SSC_SUBCMD2_PRS_DISABLE: Pressure control disabled • SSC_SUBCMD2_PRS_ENABLE: Pressure control enabled



■Point

None.

■Supported version

API version	Board version	Header file
1.30	05	em4xstd.h

■Reference

- sscSetPointDataEx ( Page 59 sscSetPointDataEx)
- sscCheckPointDataEx ( Page 60 sscCheckPointDataEx)

4.2 OAS_DATA Structure

This other axis start data structure is used for the other axes start.

```
typedef struct {  
    /* 0000h */  
    unsigned long opt_own;  
    unsigned long opt_observ;  
    long data_own;  
    long data_observ;  
    /* 0010h */  
    char reserve1[24];  
    unsigned long long st_axbit;  
    /* 0030h */  
    unsigned short st_pnt_s;  
    unsigned short st_pnt_e;  
    char reserve2[44];  
    /* 0060h */  
    char reserve3[4];  
    unsigned short dout_ctrl;  
    unsigned short dout_num;  
    unsigned short dout_ctrlbit;  
    unsigned short dout_data;  
    char reserve4[4];  
    /* 0070h */  
    unsigned short dout_timer_ctrl;  
    unsigned short dout_timer;  
    char reserve5[12];  
    /* 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 logical sum of each value to lower 2 bytes, and the observed axis No. (axis No. -1) to upper 2 bytes. ■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. <Example> OAS_DATA oasData; oasData.opt_observ = SSC_OAS_OBSERV_ENABLE SSC_OAS_OBSERV_POSITION_PASS SSC_OAS_OBSERV_JUDGE_COORD_FB SSC_OAS_OBSERV_DATA_MORE (axnum - 1) << 16);
data_own	Axis remaining distance data (or axis pass position data) [command unit]
data_observ	Observed axis specified position data [command unit]
st_axbit	Start axis designation

Member	Description
st_pnt_s	Start axis start point (0 to 2047)
st_pnt_e	Start axis end point (0 to 2047)
dout_ctrl	Output signal specification ■Value • SSC_OAS_DOUT_DISABLE: Enabled • SSC_OAS_DOUT_ENABLE_OUTPUT_BIT_DEVICE: Enabled (Output bit device specification) • SSC_OAS_DOUT_ENABLE_OUTPUT_WORD_DEVICE: Enabled (Output word device specification) • SSC_OAS_DOUT_ENABLE_GENERAL_IO: Enabled (General output specification)
dout_num	Output signal No. (0000h to 1FFFh)
dout_ctrlbit	Output signal enable selection (0000h to FFFFh)
dout_data	Output signal command (0000h to FFFFh)
dout_timer_ctrl	Output signal control (0000h to FFFFh)
dout_timer	Output signal timer [ms] (0 to 65535)



■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

API version	Board version	Header file
1.30	05	em4xxstd.h

■Reference

- sscSetOtherAxisStartData ( Page 121 sscSetOtherAxisStartData)
- sscGetOtherAxisStartData ( Page 122 sscGetOtherAxisStartData)

4.3 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;  
    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;  
    unsigned short normal_rev_torque_time;  
    unsigned short reverse_rev_torque_time;  
    char reserve1[2];  
    /* 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%] (-32768 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
normal_rev_torque_time	Continuous operation to forward rotation torque time constant [ms]
reverse_rev_torque_time	Continuous operation to reverse rotation torque time constant [ms]


Point

None.

Supported version

API version	Board version	Header file
1.30	05	em4xxstd.h

Reference

- sscSetPressData ( Page 65 sscSetPressData)
- sscGetPressData ( Page 66 sscGetPressData)

4.4 SMP_ERR Structure

This sampling error structure is used for the sampling.

```
typedef struct {  
    /* 0000h */  
    unsigned long err_ax;  
    char reserve1[8];  
    /* 0010h */  
    char reserve2[8];  
    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_dat	Data error information
err_bit	Bit error information


Point

None.

Supported version

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

sscGetSamplingError ( Page 134 sscGetSamplingError)

4.5 SMP_DATA_EM Structure

This sampling data structure is used for the sampling.

```
typedef struct {
/* 0000h */
    unsigned long long epoch_seconds;
    unsigned long epoch_nanoseconds;
    short utc_offset;
    short summertime_offset;
/* 0010h */
    long smpdata[32];
/* 0090h */
    unsigned long smpbit[1];
/* 0094h */
} SMP_DATA_EM;
```

Detailed description

Member

Member	Description
epoch_seconds	Epoch time (s)
epoch_nanoseconds	Epoch time (ns)
utc_offset	UTC offset (min)
summertime_offset	Summer time offset (min)
smpdata[32]	Data 1 to 32
smpbit[1]	Bit information

Point

None.

Supported version

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

sscGetSamplingDataEm ( Page 136 sscGetSamplingDataEm)

4.6 INT_CB_DATA Structure

This interrupt data structure is used for the interrupt callback function.

```
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 long long oas_factor_bit;
    char reserve3[8];
    unsigned char oas_factor[64];
/* 0270H */
    unsigned long long pass_factor_bit[2];
    char reserve4[16];
    unsigned char pass_factor[128];
/* 0310H */
    char reserve5[16];
/* 0320H */
    char reserve6[0xE0];
/* 0400H */
} 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
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

■Point

None.

■Supported version

API version	Board version	Header file
1.10	01	em4xxstd.h

■Reference

sscRegisterIntCallback ( Page 143 sscRegisterIntCallback)

4.7 SLAVE_INFO Structure

This device information structure is used for functions to get the controlling axis information.

```
typedef struct {  
    /* 0000h */  
    unsigned char  axis[64];  
    /* 0040h */  
    unsigned char  reserve[16];  
    /* 0050h */  
} SLAVE_INFO;
```

Detailed description

Member

Member	Description
axis[64]	Axis information ■Value <ul style="list-style-type: none">SSC_SLAVE_ON: Controlled axis or status bit ONSSC_SLAVE_OFF: Non-controlled axis or status bit OFF

Point

None.

Supported version

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

sscGetControllingAxis ( Page 52 sscGetControllingAxis)

4.8 FILE_INFO Structure

This file information structure is used for functions to get the file list.

```
typedef struct {  
    /* 0000h */  
    wchar_t name[260];  
    /* 0208h */  
} FILE_INFO;
```

Detailed description

Member

Member	Description
name[260]	File name (within 259 characters) Use the Unicode (UTF-16LE) for a character code.

Point

None.

Supported version

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

sscGetFileList ( Page 190 sscGetFileList)

4.9 FILE_INFO_EX Structure

This file information structure is used for functions to get the file list.

```
typedef struct {
/* 0000h */
    wchar_t name[260];
/* 0208h */
    unsigned long file_size;
    unsigned long attributes;
    FILE_TIME update_time;
    unsigned char reserve[24];
/* 0240h */
} FILE_INFO_EX;
```

Detailed description

Member

Member	Description
name[260]	File name (within 259 characters) Use the Unicode (UTF-16LE) for a character code.
file_size	File size (byte unit)
attributes	Attributes The got data is set in the logical sum of each value. ■Value • SSC_FILE_ATTR_DIRECTORY: directory (When this bit is OFF, the file is shown.)
update_time	Update time For the file update time information structure, refer to the following. ☞ Page 241 FILE_TIME Structure

Point

None.

Supported version

API version	Board version	Header file
1.20	01	em4xxstd.h

Reference

sscGetFileListEx (☞ Page 192 sscGetFileListEx)

4.10 FILE_TIME Structure

This file update time information structure is used for functions to get the file list.

```
typedef struct {  
    /* 0000h */  
    int year;  
    int month;  
    int day;  
    int hour;  
    int minute;  
    int second;  
    /* 0018h */  
} FILE_TIME;
```

Detailed description

Member

Member	Description
year	Year (common era)
month	Month (1 to 12)
day	Day (1 to 31)
hour	Hour (0 to 23)
minute	Minute (0 to 59)
second	Second (0 to 59)


Point

None.

Supported version

API version	Board version	Header file
1.20	01	em4xxstd.h

Reference

FILE_INFO_EX ( Page 240 FILE_INFO_EX Structure)

4.11 SLMP_SLV_OBJ_CMD Structure

This slave object read/write command data structure is used to read/write the objects of the device.

```
typedef struct {  
    /* 0000h */  
    unsigned long ip_address;  
    unsigned short multi_drop_number;  
    unsigned short port_number;  
    unsigned short frame_type;  
    unsigned short timeout;  
    unsigned short retry_count;  
    unsigned char reserve1[2];  
    /* 0010h */  
    unsigned short object_index;  
    unsigned char object_subindex;  
    unsigned char reserve2[13];  
    /* 0020h */  
} SLMP_SLV_OBJ_CMD;
```

Detailed description

Member

Member	Description
ip_address	IP address of the external device • Bit0 to 7: fourth octet • Bit8 to 15: third octet • Bit16 to 23: second octet • Bit24 to 31: first octet
multi_drop_number	Request destination multi drop station No.
port_number	External device port No. ■Value • SSC_SLMP_PORT_MR_J5: Mitsubishi Electric AC servo MELSERVO-J5 series
frame_type	SLMP frame compatible with external device ■Value • SSC_SLMP_FRAME_EXP: station No. extended frame
timeout	Monitoring timer (1 to 65535) [ms] Set the waiting time for the Motion control board to wait for the response after it issued a processing request to the external device.
retry_count	Number of resends (0 to 15) Set the number of resending requests if no response is received even when the waiting time specified by the monitoring time has elapsed.
object_index	Index of the object (0000h to FFFFh)
object_subindex	Subindex of the object Set as follows for the subindex of the object depending on the type designated by the <code>sscSmpReadSlaveObject</code> function or the <code>sscSmpWriteSlaveObject</code> function. ■When designated "SSC_RW_SLAVE_OBJ" • Set the subindex (00h to FFh) of the object. ■When designated "SSC_RW_SLAVE_OBJ_SUB_BLOCK" • Set the subindex (00h to 01h) of the start object.



Point

The function which uses this structure waits until the communication completion. It waits until "timeout × (retry_count + 1) + several seconds" at maximum.

Supported version

API version	Board version	Header file
1.00	01	em4xxstd.h

Reference

- `sscSmpReadSlaveObject` ( Page 196 `sscSmpReadSlaveObject`)
- `sscSmpWriteSlaveObject` ( Page 198 `sscSmpWriteSlaveObject`)



4.12 SLMP_SLV_OBJ_STS Structure

This slave object read/write status data structure is used to read/write the object of the device.

```
typedef struct {
/* 0000h */
    unsigned short end_code;
    unsigned short data_size;
    unsigned short completion_status;
    unsigned char reserve1[10];
/* 0010h */
    unsigned char reserve2[16];
/* 0020h */
} SLMP_SLV_OBJ_STS;
```

Detailed description

■Member

Member	Description
end_code	End code <ul style="list-style-type: none"> • In normal termination: "0" is stored. • In abnormal termination: The error code which is set by the external device is stored.
data_size	Data size [byte] <ul style="list-style-type: none"> ■sscSmpReadSlaveObject function <ul style="list-style-type: none"> • The data size which is read is stored. ■sscSmpWriteSlaveObject function <ul style="list-style-type: none"> • "0" is stored.
completion_status	Completion status <ul style="list-style-type: none"> • 0: normal • Other than 0: abnormal (error code) For the detail of the error code, refer to "ALARM No." in the following manual.  Motion Control Board User's Manual (Motion Control) Or, refer to "List of Error Codes" in the following manual.  Motion Control Board User's Manual (Network)


■Point

None.

■Supported version

API version	Board version	Header file
1.00	01	em4xxstd.h

■Reference

- sscSmpReadSlaveObject ( Page 196 sscSmpReadSlaveObject)
- sscSmpWriteSlaveObject ( Page 198 sscSmpWriteSlaveObject)

4.13 SLMP_CONTROL Structure

This SLMP control data structure is used to send SLMP messages.

```
typedef struct {  
    /* 0000h */  
    unsigned short completion_type;  
    unsigned short completion_status;  
    unsigned short channel_use_own_station;  
    unsigned long ip_address;  
    unsigned short port_number;  
    unsigned short network_number;  
    unsigned short station_number;  
    /* 0010h */  
    unsigned short io_number;  
    unsigned short multi_drop_number;  
    unsigned short retry_count;  
    unsigned short arrival_moni_time;  
    unsigned short clock_set_flag;  
    unsigned char clock_data_year_lower;  
    unsigned char clock_data_month;  
    unsigned char clock_data_day;  
    unsigned char clock_data_hour;  
    unsigned char clock_data_minute;  
    unsigned char clock_data_second;  
    /* 0020h */  
    unsigned char clock_data_week;  
    unsigned char clock_data_year_upper;  
    unsigned long error_ip_address;  
    unsigned char reserve1[2];  
    /* 0028h */  
} SLMP_CONTROL;
```

Detailed description

Member

Member	Description
completion_type [in]	<p>Execution/error completion type</p> <p>Set the data in the logical sum of each value.</p> <p>■Value</p> <p>[Execution type]</p> <ul style="list-style-type: none">SSC_SLMP_CHECK_ARRIVAL_DISABLE: Without arrival check (The instruction is regarded as completed when a request message is sent from own station.)SSC_SLMP_CHECK_ARRIVAL_ENABLE: With arrival check (The instruction is regarded as completed when a response message is received from the external device) <p>[Error completion type]</p> <ul style="list-style-type: none">SSC_SLMP_STORE_ERROR_INFO_DISABLE: No data at error completion storedSSC_SLMP_STORE_ERROR_INFO_ENABLE: Data at error completion stored <p>When setting "Without arrival check", a response message is not received. This value is set when using the command with which a response message is not received or when not referring to the response message.</p> <p>The data at error completion is stored in the member of this structure. (the clock data (the last 2 digits of the year) to the IP address of the error detected device)</p>
completion_status [out]	<p>Completion status</p> <ul style="list-style-type: none">0: normalOther than 0: abnormal (error code) <p>For the detail of the error code, refer to "ALARM No." in the following manual.</p> <p> Motion Control Board User's Manual (Motion Control)</p> <p>Or, refer to "List of Error Codes" in the following manual.</p> <p> Motion Control Board User's Manual (Network)</p>
channel_use_own_station [in]	<p>Channel for own station</p> <p>■Value</p> <ul style="list-style-type: none">1: Channel in which the serial No. is not set2 to 9: Channel in which the serial No. is set10 to 17: Station No. extended frame <p>The serial No. is set when sending multiple request messages to the same SLMP-compatible device. The serial No. to be set is decided automatically by the system. For the detail of the serial No., refer to the following manual.</p> <p> SLMP Reference Manual</p>

Member	Description
ip_address [in]	IP address of the external device <ul style="list-style-type: none"> • Bit0 to 7: fourth octet • Bit8 to 15: third octet • Bit16 to 23: second octet • Bit24 to 31: first octet
port_number [in]	External device port No. For the port No. to designate, refer to the manual of the external device. Value <ul style="list-style-type: none"> • SSC_SLMP_PORT_MR_J5: Mitsubishi Electric AC servo MELSERVO-J5 series
network_number [in]	Request destination network No. Set the fixed No.0000h (fixed).
station_number [in]	Request destination station No. Set the fixed No.00FFh (fixed).
io_number [in]	Request destination unit I/O No. Value <ul style="list-style-type: none"> • SSC_SLMP_IO_CPU_OWN_MANAGEMENT: Own station/control CPU
multi_drop_number [in]	Request destination multi drop station No. When there is no multi drop No., the No. is 0000h (fixed).
retry_count [in/out]	At executing function <ul style="list-style-type: none"> • Set the number of resending requests (0 to 15) if no response is received even when the waiting time specified by the monitoring timer has elapsed. At completion of executing function <ul style="list-style-type: none"> • The number of resends performed (result)(0 to 15) is stored.
arrival_moni_time [in]	Arrival monitoring time (0 to 32767) [second] <ul style="list-style-type: none"> • 0: 10 seconds • 1 to 32767: 1 to 32767 seconds Specify the monitoring time until the processing completion. If the processing is not completed within the monitoring time, resends are performed for the number of times specified in the number of resends.
clock_set_flag [out]	Clock set flag The validity status (valid or invalid) of the clock data and the IP address of the error detected device are stored. Value <ul style="list-style-type: none"> • 0: Invalid • 1: Valid
clock_data_year_lower [out]	Clock data (last 2 digits of the year): 00h to 99h
clock_data_month [out]	Clock data (month): 01h to 12h
clock_data_day [out]	Clock data (day): 01h to 31h
clock_data_hour [out]	Clock data (hour): 00h to 23h
clock_data_minute [out]	Clock data (minute): 00h to 59h
clock_data_second [out]	Clock data (second): 00h to 59h
clock_data_week [out]	Clock data (day of week): 00h to 06h
clock_data_year_upper [out]	Clock data (first 2 digits of the year): 00h to 99h
error_ip_address [out]	IP address of error detected device The IP address of the error detected device is stored. <ul style="list-style-type: none"> • Bit0 to 7: fourth octet • Bit8 to 15: third octet • Bit16 to 23: second octet • Bit24 to 31: first octet

Point

- The clock data and the IP address of the error detected device are cleared at the normal completion.
- The clock data is stored in binary-coded decimal.
- The output member variable of this structure is stored only at the normal completion or at the completion with the following error codes.
SSC_FUNC_ERR_SLMP_RESPONSE_SIZE_OVER, SSC_FUNC_ERR_SLMP_SEND_RECEIVE

Supported version

API version	Board version	Header file
1.20	01	em4xxstd.h

Reference

sscSlmpSendCommand ( Page 200 sscSlmpSendCommand

4.14 PRS_PROFILE Structure

This pressure control profile structure is used for pressure control.

```
typedef struct {  
    /* 0000h */  
    PRS_SWITCH_TO_PRESSURE switch_to_pressure;  
    /* 0030h */  
    PRS_CMN_SETTING common_setting;  
    /* 0040h */  
    PRS_FEED_DATA feed_data;  
    /* 0750h */  
    PRS_SWITCH_TO_DWELL switch_to_dwell;  
    /* 0780h */  
    PRS_DWELL_DATA dwell_data;  
    unsigned char reserve1[40];  
    /* 0EB8h */  
    PRS_RELEASE_DATA release_data;  
    unsigned char reserve2[40];  
    /* 0F00h */  
} PRS_PROFILE;
```

Detailed description

■Member

Member	Description
switch_to_pressure	Mode switching (position to pressure) For the mode switching (position to pressure) structure, refer to the following. ↗ Page 247 PRS_SWITCH_TO_PRESSURE structure
common_setting	Common setting (Feed / Dwell) For the common setting (Feed / Dwell) structure, refer to the following. ↗ Page 248 PRS_CMN_SETTING structure
feed_data	Feed data For the feed data structure, refer to the following. ↗ Page 249 PRS_FEED_DATA structure
switch_to_dwell	Switching (feed to dwell) For the switching (feed to dwell) structure, refer to the following. ↗ Page 251 PRS_SWITCH_TO_DWELL structure
dwell_data	Dwell data For the dwell data structure, refer to the following. ↗ Page 252 PRS_DWELL_DATA structure
release_data	Pressure release data For the pressure release data structure, refer to the following. ↗ Page 254 PRS_RELEASE_DATA structure

■Point

None.

■Supported version

API version	Board version	Header file
1.30	05	em4xxstd.h

■Reference

- sscSetPrsProfile ([↗](#) Page 215 sscSetPrsProfile)
- sscGetPrsProfile ([↗](#) Page 216 sscGetPrsProfile)

PRS_SWITCH_TO_PRESSURE structure

This mode switching (position to pressure) structure is used for the pressure control profile structure.

```
typedef struct {
/* 0000h */
    long limit_position;
    unsigned short option1;
    unsigned short timeout;
/* 0008h */
    PRS_SWITCH switch_condition;
    unsigned char reserve[8];
/* 0030h */
} PRS_SWITCH_TO_PRESSURE;
```

Detailed description

Member

Member	Description
limit_position	Limit position [command unit] (-2147483648 to 2147483647)
option1	Switching option 1 Set data in the logical sum of each value. ■Value [Select movement] <ul style="list-style-type: none"> SSC_PRS_MODE_FEED_DWELL: Feed/dwell SSC_PRS_MODE_RELEASE: Pressure release [Forced switch to pressure mode] <ul style="list-style-type: none"> SSC_PRS_FORCED_SWITCH_DISABLE: Disabled SSC_PRS_FORCED_SWITCH_ENABLE: Enabled [Select switching delay] <ul style="list-style-type: none"> SSC_PRS_SWITCH_DELAY_ENABLE: Switching delay enabled SSC_PRS_SWITCH_DELAY_DISABLE: Switching delay disabled
timeout	Timeout duration [ms] (0 to 65535)
switch_condition	Pressure control switching condition For the pressure control switching condition structure, refer to the following. ➞ Page 255 PRS_SWITCH structure

Point

None.

Supported version

API version	Board version	Header file
1.30	05	em4xxstd.h

Reference

PRS_PROFILE structure (➞ Page 246 PRS_PROFILE Structure)

PRS_CMN_SETTING structure

This common settings on feeding/dwelling pressure structure is used for the pressure control profile structure.

```
typedef struct {  
    /* 0000h */  
    long end_position;  
    unsigned long speed_limit;  
    short pressure_cmd;  
    unsigned char reserve[6];  
    /* 0010h */  
} PRS_CMN_SETTING;
```

Detailed description

■Member

Member	Description
end_position	Final position (SE) [command unit] (-2147483648 to 2147483647)
speed_limit	Referenced speed limit value [speed unit] (1 to 2147483647)
pressure_cmd	Referenced pressure command value (1 to 32767)

■Point

None.

■Supported version

API version	Board version	Header file
1.30	05	em4xxstd.h

■Reference

PRS_PROFILE structure ([🔗](#) Page 246 PRS_PROFILE Structure)


PRS_FEED_DATA structure

This feed data structure is used for the control pressure profile structure.

```
typedef struct {
/* 0000h */
    unsigned short step_count;
    unsigned short cmn_option;
    unsigned long init_speed_limit;
    short init_pressure;
    unsigned char reserve[6];
/* 0010h */
    PRS_FEED_STEP step[32];
/* 0710h */
} PRS_FEED_DATA;
```

Detailed description

Member

Member	Description
step_count	Number of steps (1 to 32)
cmn_option	Feeding common option 1 Set data in the logical sum of each value. ■Value [Selection of initial speed limit value] <ul style="list-style-type: none"> SSC_PRS_SELECT_SPEED_LIMIT_SWITCHING: Command speed while switching value SSC_PRS_SELECT_SPEED_LIMIT_INIT: Initial speed limit value [Selection of initial pressure command value] <ul style="list-style-type: none"> SSC_PRS_INIT_PRESSURE_CMD_DISABLE: Disabled SSC_PRS_INIT_PRESSURE_CMD_ENABLE: Enabled
init_speed_limit	Initial speed limit value of initial command [speed unit] (0 to 2147483647)
init_pressure	Initial pressure command value of initial command (0 to 32767)
step[32]	Step No. of feeding step 1 to 32 For the step No. of feeding step structure, refer to the following.  Page 250 PRS_FEED_STEP structure

Point

None.

Supported version

API version	Board version	Header file
1.30	05	em4xxstd.h

Reference

PRS_PROFILE structure ( Page 246 PRS_PROFILE Structure)


PRS_FEED_STEP structure

This feed data structure is used for the feed data structure.

```
typedef struct {  
    /* 0000h */  
    long limit_position;  
    unsigned long speed_limit;  
    short pressure_cmd;  
    unsigned char reserve1[2];  
    unsigned long speed_limit_acc_dec;  
    /* 0010h */  
    unsigned long pressure_cmd_acc_dec;  
    unsigned char reserve2[4];  
    /* 0018h */  
    PRS_SWITCH switch_condition;  
    /* 0038h */  
} PRS_FEED_STEP;
```

Detailed description

■Member

Member	Description
limit_position	Limit position [command unit] (-2147483648 to 2147483647)
speed_limit	Speed limit value [speed unit] (0 to 2147483647)
pressure_cmd	Pressure command value (0 to 32767)
speed_limit_acc_dec	Speed limit time constant [ms] (0 to 8388608)
pressure_cmd_acc_dec	Pressure command time constant [ms] (0 to 8388608)
switch_condition	Pressure control switching conditions For the pressure control switching conditions structure, refer to the following.  Page 255 PRS_SWITCH structure

■Point

None.

■Supported version

API version	Board version	Header file
1.30	05	em4xxstd.h

■Reference

PRS_FEED_DATA structure ( Page 249 PRS_FEED_DATA structure)


PRS_SWITCH_TO_DWELL structure

This switching (feed to dwell) structure is used for the pressure control profile structure.

```
typedef struct {
/* 0000h */
    long short option1;
    short switch_pressure;
    unsigned long switch_pressure_time;
/* 0008h */
    PRS_SWITCH switch_condition;
    unsigned char reserve[8];
/* 0030h */
} PRS_SWITCH_TO_DWELL;
```

Detailed description

■Member

Member	Description
option1	Switching option 1 Set data in the logical sum of each value. ■Value [Forced switch to pressure mode] <ul style="list-style-type: none"> • SSC_PRS_DWELL_FORCED_SWITCH_DISABLE: Disabled • SSC_PRS_DWELL_FORCED_SWITCH_EABLE: Enabled [Selection of switching pressure] <ul style="list-style-type: none"> • SSC_PRS_DWELL_SWITCH_DISABLE: Disabled • SSC_PRS_DWELL_SWITCH_ENABLE: Enabled
switch_pressure	Switching pressure in switching pressure setting (0 to 32767)
switch_pressure_time	Switching time in switching pressure setting [ms] (0 to 327670)
switch_condition	Pressure control switching conditions For the pressure control switching conditions structure, refer to the following.  Page 255 PRS_SWITCH structure

■Point

None.

■Supported version

API version	Board version	Header file
1.30	05	em4xxstd.h

■Reference

PRS_PROFILE structure ( Page 246 PRS_PROFILE Structure)


PRS_DWELL_DATA structure

This dwell data structure is used for the pressure control profile structure.

```
typedef struct {  
    /* 0000h */  
    unsigned short step_count;  
    unsigned short cmn_option;  
    unsigned char reserve[12];  
    /* 0010h */  
    PRS_DWELL_STEP step[32];  
    /* 0710h */  
} PRS_DWELL_DATA;
```

Detailed description

■Member

Member	Description
step_count	Number of steps (1 to 32)
switch_pressure	Dwelling pressure common option 1 ■Value [Mode resetting] <ul style="list-style-type: none">• SSC_PRS_RESET_MODE_ENABLE: With mode resetting• SSC_PRS_RESET_MODE_DISABLE: Without mode resetting
step[32]	Step No. of dwelling pressure step 1 to 32 For the step No. of dwelling pressure step structure, refer to the following.  Page 253 PRS_DWELL_STEP structure

■Point

None.

■Supported version

API version	Board version	Header file
1.30	05	em4xxstd.h

■Reference

PRS_PROFILE structure ( Page 246 PRS_PROFILE Structure)


PRS_DWELL_STEP structure

This dwell step structure is used for the dwell data structure.

```
typedef struct {
/* 0000h */
    long limit_position;
    unsigned long speed_limit;
    short pressure_cmd;
    unsigned short option1;
    unsigned long speed_limit_acc_dec;
/* 0010h */
    unsigned long pressure_cmd_acc_dec;
    unsigned char reserve[4];
/* 0018h */
    PRS_SWITCH switch_condition;
/* 0710h */
} PRS_DWELL_STEP;
```

Detailed description

■Member

Member	Description
limit_position	Limit position [command unit] (-2147483648 to 2147483647)
speed_limit	Speed limit value [speed unit] (0 to 2147483647)
pressure_cmd	Pressure command value (0 to 32767)
option1	Dwelling pressure option 1 ■Value [Dwelling pressure time constant] • SSC_PRS_SWITCH_DELAY_ENABLE: Time constant is enabled • SSC_PRS_SWITCH_DELAY_DISABLE: Time constant is disabled
speed_limit_acc_dec	Speed limit time constant [ms] (0 to 8388608)
pressure_cmd_acc_dec	Pressure command time constant [ms] (0 to 8388608)
switch_condition	Pressure control switching conditions For the pressure control switching conditions structure, refer to the following.  Page 255 PRS_SWITCH structure

■Point

None.

■Supported version

API version	Board version	Header file
1.30	05	em4xxstd.h

■Reference

PRS_DWELL_DATA structure ( Page 252 PRS_DWELL_DATA structure)

PRS_RELEASE_DATA structure

This pressure release data structure is used for the pressure control profile structure.

```
typedef struct {  
    /* 0000h */  
    long end_position;  
    unsigned long speed_limit;  
    unsigned long init_speed_limit;  
    short init_pressure;  
    unsigned char reserve1[2];  
    /* 0010h */  
    unsigned long speed_limit_acc_dec;  
    unsigned long speed_limit_end_dec;  
    unsigned char reserve2[8];  
    /* 0020h */  
} PRS_RELEASE_DATA;
```

Detailed description

■Member

Member	Description
end_position	Final position (SE2) [command unit] (-2147483648 to 2147483647)
speed_limit	Referenced speed limit value [speed unit] (1 to 2147483647)
init_speed_limit	Initial speed limit value (V0) (0 to 2147483647)
init_pressure	Initial pressure command value (PR0) (0 to 32767)
speed_limit_acc_dec	Speed limit time constant [ms] (0 to 8388608)
speed_limit_end_dec	Speed limit stopping time constant [ms] (0 to 8388608)

■Point

None.

■Supported version

API version	Board version	Header file
1.30	05	em4xxstd.h

■Reference

PRS_PROFILE structure ([↩](#) Page 246 PRS_PROFILE Structure)

PRS_SWITCH structure

This pressure control switching conditions structure is used for structures such as the position control to pressure control switching conditions structure.

```
typedef struct {
/* 0000h */
    unsigned char condition_select[4];
    long judgement_value[4];
/* 0014h */
    unsigned short established_condition;
    unsigned char reserve[10];
/* 0020h */
} PRS_SWITCH;
```

Detailed description

■Member

Member	Description
condition_select[4]	<p>Switching condition options 1 to 4 Set data in the logical sum of each value.</p> <p>■Value [Select data]</p> <ul style="list-style-type: none"> • SSC_PRS_SELECT_DEFAULT_DATA: Default selected data • SSC_PRS_SELECT_AXIS_BIT: No condition (switching by axis bit signal)*¹ • SSC_PRS_SELECT_CMD_POS: Command position [command unit] • SSC_PRS_SELECT_FB_POS: F/B position [command unit] • SSC_PRS_SELECT_LOAD_CELL: Load cell pressure (pressure feedback) • SSC_PRS_SELECT_STEP_TIME: Step running time [ms]*² • SSC_PRS_SELECT_CMD_SPD: Command speed [speed unit] • SSC_PRS_SELECT_FB_SPD: F/B speed [speed unit] • SSC_PRS_SELECT_IMMEDIATELY: No condition (switching immediately)*¹ <p>[Switching conditions decision]</p> <ul style="list-style-type: none"> • SSC_PRS_VALUE_OR_MORE: Selected data turns decided value or more • SSC_PRS_VALUE_OR_LESS: Selected data turns decided value or less • SSC_PRS_DIFF_VALUE_OR_MORE: Changed value of selected data turns decided value or more • SSC_PRS_DIFF_VALUE_OR_LESS: Changed value of selected data turns decided value or less • SSC_PRS_PASS_VALUE_INCREASE: Selected data gets increased and passes through decided value • SSC_PRS_PASS_VALUE_DECREASE: Selected data gets decreased and passes through decided value
judgement_value[4]	<p>Judgement value 1 to 4</p> <ul style="list-style-type: none"> • Default selected data: -2147483648 to 2147483647 • No condition (switching by axis bit signal): No setting required • Command position [command unit]: -2147483648 to 2147483647 • F/B position [command unit]: -2147483648 to 2147483647 • Load cell pressure (pressure feedback): -2147483648 to 2147483647 • Step running time [ms]: -2147483648 to 2147483647 • Command speed [speed unit]: 0 to 32767 • F/B speed [speed unit]: 0 to 32767 • No condition (switching immediately): No setting required

Member	Description
established_condition	<p>Establishment conditions</p> <p>■Value</p> <ul style="list-style-type: none"> • SSC_PRS_CONDITION_1: Switching condition 1 only • SSC_PRS_CONDITION_1AND2: Switching condition 1 AND Switching condition 2 • SSC_PRS_CONDITION_1OR2: Switching condition 1 OR Switching condition 2 • SSC_PRS_CONDITION_1AND2AND3: Switching condition 1 AND Switching condition 2 AND Switching condition 3 • SSC_PRS_CONDITION_1OR2OR3: Switching condition 1 OR Switching condition 2 OR Switching condition 3 • SSC_PRS_CONDITION__1AND2_OR3: (Switching condition 1 AND Switching condition 2) OR Switching condition 3 • SSC_PRS_CONDITION__1OR2_AND3: (Switching condition 1 OR Switching condition 2) AND Switching condition 3 • SSC_PRS_CONDITION_1AND2AND3AND4: Switching condition 1 AND Switching condition 2 AND Switching condition 3 AND Switching condition 4 • SSC_PRS_CONDITION_1OR2OR3OR4: Switching condition 1 OR Switching condition 2 OR Switching condition 3 OR Switching condition 4 • SSC_PRS_CONDITION__1OR2OR3_AND4: (Switching condition 1 OR Switching condition2 OR Switching condition3) AND Switching condition 4 • SSC_PRS_CONDITION__1AND2AND3_OR4: (Switching condition 1 AND Switching condition 2 AND Switching condition 3) OR Switching condition 4 • SSC_PRS_CONDITION__1AND2_OR3_AND4: ((Switching condition 1 AND Switching condition 2) OR Switching condition 3) AND Switching condition 4 • SSC_PRS_CONDITION__1OR2_AND3_AND4: ((Switching condition 1 OR Switching condition 2) AND Switching condition 3) AND Switching condition 4 • SSC_PRS_CONDITION__1AND2_OR3_OR4: ((Switching condition 1 AND Switching condition 2) OR Switching condition 3) OR Switching condition 4 • SSC_PRS_CONDITION__1OR2_AND3_OR4: ((Switching condition 1 OR Switching condition 2) AND Switching condition 3) OR Switching condition 4 • SSC_PRS_CONDITION__1AND2_OR_3AND4_: (Switching condition 1 AND Switching condition 2) OR (Switching condition 3 AND Switching condition 4) • SSC_PRS_CONDITION__1OR2_OR_3AND4_: (Switching condition 1 OR Switching condition 2) OR (Switching condition 3 AND Switching condition 4) • SSC_PRS_CONDITION__1AND2_AND_3OR4_: (Switching condition 1 AND Switching condition 2) AND (Switching condition 3 OR Switching condition 4) • SSC_PRS_CONDITION__1OR2_AND_3OR4_: (Switching condition 1 OR Switching condition 2) AND (Switching condition 3 OR Switching condition 4)

*1 It is not required to set the switching conditions decision.

*2 For the switching conditions decisions, "SSC_PRS_DIFF_VALUE_OR_MORE"/"SSC_PRS_DIFF_VALUE_OR_LESS"/"SSC_PRS_PASS_VALUE_DECREASE" cannot be set.





■Point

None.

■Supported version

API version	Board version	Header file
1.30	05	em4xxstd.h

■Reference

- PRS_SWITCH_TO_PRESSURE structure ( Page 247 PRS_SWITCH_TO_PRESSURE structure)
- PRS_FEED_STEP structure ( Page 250 PRS_FEED_STEP structure)
- PRS_SWITCH_TO_DWELL structure ( Page 251 PRS_SWITCH_TO_DWELL structure)
- PRS_DWELL_STEP structure ( Page 253 PRS_DWELL_STEP structure)

5 BIT DEFINITION LIST

The following tables list the bit definitions to be specified in the sscSetCommandBitSignalEX function, the sscGetStatusBitSignalEx function, and the sscWaitStatusBitSignalEx function.

5.1 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_3	For manufacturer setting
SSC_CMDBIT_SYS_4	
SSC_CMDBIT_SYS_5	
SSC_CMDBIT_SYS_6	
SSC_CMDBIT_SYS_7	
SSC_CMDBIT_SYS_8	
Bit No. (constant)	Signal name
SSC_CMDBIT_SYS_SEMI	Software forced stop
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_ITFE	Interrupt processing fast 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_33	For manufacturer setting
SSC_CMDBIT_SYS_34	
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_49	For manufacturer setting
SSC_CMDBIT_SYS_50	
SSC_CMDBIT_SYS_51	
SSC_CMDBIT_SYS_52	
SSC_CMDBIT_SYS_53	
SSC_CMDBIT_SYS_54	
SSC_CMDBIT_SYS_55	
SSC_CMDBIT_SYS_56	
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_SMPs	Sampling start
SSC_CMDBIT_SYS_66	For manufacturer setting
SSC_CMDBIT_SYS_67	
SSC_CMDBIT_SYS_68	
SSC_CMDBIT_SYS_69	
SSC_CMDBIT_SYS_70	
SSC_CMDBIT_SYS_71	
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_ASYNC1	Non-synchronous command (group 1)
SSC_CMDBIT_SYS_ASYNC2	Non-synchronous command (group 2)
SSC_CMDBIT_SYS_ASYNC3	Non-synchronous command (group 3)
SSC_CMDBIT_SYS_ASYNC4	Non-synchronous command (group 4)
SSC_CMDBIT_SYS_ASYNC5	Non-synchronous command (group 5)
SSC_CMDBIT_SYS_ASYNC6	Non-synchronous command (group 6)
SSC_CMDBIT_SYS_ASYNC7	Non-synchronous command (group 7)
SSC_CMDBIT_SYS_ASYNC8	Non-synchronous command (group 8)
Bit No. (constant)	Signal name
SSC_CMDBIT_SYS_89	For manufacturer setting
SSC_CMDBIT_SYS_90	
SSC_CMDBIT_SYS_91	
SSC_CMDBIT_SYS_92	
SSC_CMDBIT_SYS_93	
SSC_CMDBIT_SYS_94	
SSC_CMDBIT_SYS_95	
SSC_CMDBIT_SYS_96	
Bit No. (constant)	Signal name
SSC_CMDBIT_SYS_RBR	Reboot preparation
SSC_CMDBIT_SYS_RBS	Execute rebooting
SSC_CMDBIT_SYS_CRST	System alarm reset
SSC_CMDBIT_SYS_100	For manufacturer setting
SSC_CMDBIT_SYS_SMON	System monitor command
SSC_CMDBIT_SYS_SMONR	System monitor latch command
SSC_CMDBIT_SYS_103	For manufacturer setting
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_SPWRT	Parameter write command
SSC_CMDBIT_SYS_114	For manufacturer setting
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_SPRD	Parameter read command
SSC_CMDBIT_SYS_122	For manufacturer setting
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_SMP SW	Sampling setting write command
SSC_CMDBIT_SYS_130	For manufacturer setting
SSC_CMDBIT_SYS_131	
SSC_CMDBIT_SYS_132	
SSC_CMDBIT_SYS_SMP SR	Sampling setting read command
SSC_CMDBIT_SYS_134	For manufacturer setting
SSC_CMDBIT_SYS_135	
SSC_CMDBIT_SYS_136	
Bit No. (constant)	Signal name
SSC_CMDBIT_SYS_137	For manufacturer setting
SSC_CMDBIT_SYS_138	
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_145	For manufacturer setting
SSC_CMDBIT_SYS_146	
SSC_CMDBIT_SYS_147	
SSC_CMDBIT_SYS_148	
SSC_CMDBIT_SYS_149	
SSC_CMDBIT_SYS_150	
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_AERST	All error reset
SSC_CMDBIT_SYS_158	For manufacturer setting
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_177	For manufacturer setting
SSC_CMDBIT_SYS_178	
SSC_CMDBIT_SYS_179	
SSC_CMDBIT_SYS_180	
SSC_CMDBIT_SYS_181	
SSC_CMDBIT_SYS_182	
SSC_CMDBIT_SYS_183	
SSC_CMDBIT_SYS_184	
Bit No. (constant)	Signal name
SSC_CMDBIT_SYS_185	For manufacturer setting
SSC_CMDBIT_SYS_186	
SSC_CMDBIT_SYS_187	
SSC_CMDBIT_SYS_188	
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	
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	

5.2 System Status Bit

Bit No. (constant)	Signal name
SSC_STSBIT_SYS_ITO	Outputting factor of interrupt
SSC_STSBIT_SYS_2	For manufacturer setting
SSC_STSBIT_SYS_3	
SSC_STSBIT_SYS_HRIF	Highly response I/F enabled
SSC_STSBIT_SYS_5	For manufacturer setting
SSC_STSBIT_SYS_MSIO	MSI enabled
SSC_STSBIT_SYS_HSMO	High speed operation mode enabled
SSC_STSBIT_SYS_8	For manufacturer setting
Bit No. (constant)	Signal name
SSC_STSBIT_SYS_EMIO	Being executed forced stop
SSC_STSBIT_SYS_10	For manufacturer setting
SSC_STSBIT_SYS_11	
SSC_STSBIT_SYS_PPIAL	Batch judgment of passing position interrupt condition
SSC_STSBIT_SYS_SEO	System error detected
SSC_STSBIT_SYS_14	For manufacturer setting
SSC_STSBIT_SYS_EMID	External forced stop disabled
SSC_STSBIT_SYS_16	For manufacturer setting
Bit No. (constant)	Signal name
SSC_STSBIT_SYS_17	For manufacturer setting
SSC_STSBIT_SYS_18	
SSC_STSBIT_SYS_19	
SSC_STSBIT_SYS_20	
SSC_STSBIT_SYS_21	
SSC_STSBIT_SYS_22	
SSC_STSBIT_SYS_23	
SSC_STSBIT_SYS_24	
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_31	
SSC_STSBIT_SYS_32	
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_49	For manufacturer setting
SSC_STSBIT_SYS_50	
SSC_STSBIT_SYS_51	
SSC_STSBIT_SYS_52	
SSC_STSBIT_SYS_53	
SSC_STSBIT_SYS_54	
SSC_STSBIT_SYS_55	
SSC_STSBIT_SYS_56	
Bit No. (constant)	Signal name
SSC_STSBIT_SYS_57	For manufacturer setting
SSC_STSBIT_SYS_58	
SSC_STSBIT_SYS_59	
SSC_STSBIT_SYS_60	
SSC_STSBIT_SYS_61	
SSC_STSBIT_SYS_62	
SSC_STSBIT_SYS_63	
SSC_STSBIT_SYS_64	
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_69	For manufacturer setting
SSC_STSBIT_SYS_70	
SSC_STSBIT_SYS_71	
SSC_STSBIT_SYS_72	
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_IPCH	Changeable interpolation group
SSC_STSBIT_SYS_80	For manufacturer setting

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 completed
SSC_STSBIT_SYS_RBNG	Reboot preparation error
SSC_STSBIT_SYS_CALM	Current system alarm
SSC_STSBIT_SYS_100	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_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_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_116	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_120	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_125	For manufacturer setting
SSC_STSBIT_SYS_126	
SSC_STSBIT_SYS_127	
SSC_STSBIT_SYS_128	
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_132	For manufacturer setting
SSC_STSBIT_SYS_SRFIN	Sampling setting read completed
SSC_STSBIT_SYS_SREN	Sampling setting No. error
SSC_STSBIT_SYS_135	For manufacturer setting
SSC_STSBIT_SYS_136	
Bit No. (constant)	Signal name
SSC_STSBIT_SYS_137	For manufacturer setting
SSC_STSBIT_SYS_138	
SSC_STSBIT_SYS_139	
SSC_STSBIT_SYS_140	
SSC_STSBIT_SYS_141	
SSC_STSBIT_SYS_142	
SSC_STSBIT_SYS_143	
SSC_STSBIT_SYS_144	
Bit No. (constant)	Signal name
SSC_STSBIT_SYS_145	For manufacturer setting
SSC_STSBIT_SYS_146	
SSC_STSBIT_SYS_147	
SSC_STSBIT_SYS_148	
SSC_STSBIT_SYS_149	
SSC_STSBIT_SYS_150	
SSC_STSBIT_SYS_151	
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_AERO	Processing all error reset
SSC_STSBIT_SYS_AERF	All error reset completed
SSC_STSBIT_SYS_AERE	All error reset error
SSC_STSBIT_SYS_160	For manufacturer setting

Bit No. (constant)	Signal name
SSC_STSBIT_SYS_EMC0	Forced stop factor 0 (Being executed external forced stop)
SSC_STSBIT_SYS_EMC1	Forced stop factor 1 (Being executed software forced stop)
SSC_STSBIT_SYS_EMC2	Forced stop factor 2 (User watchdog error detected)
SSC_STSBIT_SYS_164	For manufacturer setting
SSC_STSBIT_SYS_165	
SSC_STSBIT_SYS_EMC5	Forced stop factor 5 (Preparing reboot)
SSC_STSBIT_SYS_EMC6	Forced stop factor 6 (System status code error detected)
SSC_STSBIT_SYS_168	For manufacturer setting
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_177	For manufacturer setting
SSC_STSBIT_SYS_178	
SSC_STSBIT_SYS_179	
SSC_STSBIT_SYS_180	
SSC_STSBIT_SYS_181	
SSC_STSBIT_SYS_182	
SSC_STSBIT_SYS_183	
SSC_STSBIT_SYS_184	
Bit No. (constant)	Signal name
SSC_STSBIT_SYS_185	For manufacturer setting
SSC_STSBIT_SYS_186	
SSC_STSBIT_SYS_187	
SSC_STSBIT_SYS_188	
SSC_STSBIT_SYS_189	
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	
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	

5.3 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_5	
SSC_CMDBIT_AX_SRST	Drive unit 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	Interpolation operation mode
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_65	For manufacturer setting
SSC_CMDBIT_AX_66	
SSC_CMDBIT_AX_67	
SSC_CMDBIT_AX_68	
SSC_CMDBIT_AX_69	
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_85	
SSC_CMDBIT_AX_86	
SSC_CMDBIT_AX_87	
SSC_CMDBIT_AX_88	
Bit No. (constant)	Signal name
SSC_CMDBIT_AX_89	For manufacturer setting
SSC_CMDBIT_AX_90	
SSC_CMDBIT_AX_91	
SSC_CMDBIT_AX_92	
SSC_CMDBIT_AX_93	
SSC_CMDBIT_AX_94	
SSC_CMDBIT_AX_95	
SSC_CMDBIT_AX_96	
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 switching command
SSC_CMDBIT_AX_102	For manufacturer setting
SSC_CMDBIT_AX_103	
SSC_CMDBIT_AX_104	
Bit No. (constant)	Signal name
SSC_CMDBIT_AX_PSMFSW	Pressure control mode forced switching
SSC_CMDBIT_AX_PSDFSW	Dwelling pressure forced switching
SSC_CMDBIT_AX_107	For manufacturer setting
SSC_CMDBIT_AX_PSMRST	Pressure control mode resetting
SSC_CMDBIT_AX_109	For manufacturer setting
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_121	For manufacturer setting
SSC_CMDBIT_AX_122	
SSC_CMDBIT_AX_123	
SSC_CMDBIT_AX_124	
SSC_CMDBIT_AX_125	
SSC_CMDBIT_AX_126	
SSC_CMDBIT_AX_127	
SSC_CMDBIT_AX_128	
Bit No. (constant)	Signal name
SSC_CMDBIT_AX_129	For manufacturer setting
SSC_CMDBIT_AX_130	
SSC_CMDBIT_AX_131	
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_161	For manufacturer setting
SSC_CMDBIT_AX_162	
SSC_CMDBIT_AX_163	
SSC_CMDBIT_AX_164	
SSC_CMDBIT_AX_165	
SSC_CMDBIT_AX_166	
SSC_CMDBIT_AX_167	
SSC_CMDBIT_AX_168	
Bit No. (constant)	Signal name
SSC_CMDBIT_AX_169	For manufacturer setting
SSC_CMDBIT_AX_170	
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	
Bit No. (constant)	Signal name
SSC_CMDBIT_AX_MON	Monitor command
SSC_CMDBIT_AX_MONR	Monitor latch command
SSC_CMDBIT_AX_259	For manufacturer setting
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_PWRT	Parameter write command
SSC_CMDBIT_AX_266	For manufacturer setting
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_PRD	Parameter read command
SSC_CMDBIT_AX_274	For manufacturer setting
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	
Bit No. (constant)	Signal name
SSC_CMDBIT_AX_385	For manufacturer setting
SSC_CMDBIT_AX_386	
SSC_CMDBIT_AX_387	
SSC_CMDBIT_AX_388	
SSC_CMDBIT_AX_389	
SSC_CMDBIT_AX_390	
SSC_CMDBIT_AX_391	
SSC_CMDBIT_AX_392	
Bit No. (constant)	Signal name
SSC_CMDBIT_AX_393	For manufacturer setting
SSC_CMDBIT_AX_394	
SSC_CMDBIT_AX_395	
SSC_CMDBIT_AX_396	
SSC_CMDBIT_AX_397	
SSC_CMDBIT_AX_398	
SSC_CMDBIT_AX_399	
SSC_CMDBIT_AX_400	

Bit No. (constant)	Signal name
SSC_CMDBIT_AX_401	For manufacturer setting
SSC_CMDBIT_AX_402	
SSC_CMDBIT_AX_403	
SSC_CMDBIT_AX_404	
SSC_CMDBIT_AX_405	
SSC_CMDBIT_AX_406	
SSC_CMDBIT_AX_407	
SSC_CMDBIT_AX_408	
Bit No. (constant)	Signal name
SSC_CMDBIT_AX_409	For manufacturer setting
SSC_CMDBIT_AX_410	
SSC_CMDBIT_AX_411	
SSC_CMDBIT_AX_412	
SSC_CMDBIT_AX_413	
SSC_CMDBIT_AX_414	
SSC_CMDBIT_AX_415	
SSC_CMDBIT_AX_416	
Bit No. (constant)	Signal name
SSC_CMDBIT_AX_417	For manufacturer setting
SSC_CMDBIT_AX_418	
SSC_CMDBIT_AX_419	
SSC_CMDBIT_AX_420	
SSC_CMDBIT_AX_421	
SSC_CMDBIT_AX_422	
SSC_CMDBIT_AX_423	
SSC_CMDBIT_AX_424	
Bit No. (constant)	Signal name
SSC_CMDBIT_AX_425	For manufacturer setting
SSC_CMDBIT_AX_426	
SSC_CMDBIT_AX_427	
SSC_CMDBIT_AX_428	
SSC_CMDBIT_AX_429	
SSC_CMDBIT_AX_430	
SSC_CMDBIT_AX_431	
SSC_CMDBIT_AX_432	
Bit No. (constant)	Signal name
SSC_CMDBIT_AX_433	For manufacturer setting
SSC_CMDBIT_AX_434	
SSC_CMDBIT_AX_435	
SSC_CMDBIT_AX_436	
SSC_CMDBIT_AX_437	
SSC_CMDBIT_AX_438	
SSC_CMDBIT_AX_439	
SSC_CMDBIT_AX_440	

Bit No. (constant)	Signal name
SSC_CMDBIT_AX_441	For manufacturer setting
SSC_CMDBIT_AX_442	
SSC_CMDBIT_AX_443	
SSC_CMDBIT_AX_444	
SSC_CMDBIT_AX_445	
SSC_CMDBIT_AX_446	
SSC_CMDBIT_AX_447	
SSC_CMDBIT_AX_448	
Bit No. (constant)	Signal name
SSC_CMDBIT_AX_449	For manufacturer setting
SSC_CMDBIT_AX_450	
SSC_CMDBIT_AX_451	
SSC_CMDBIT_AX_452	
SSC_CMDBIT_AX_453	
SSC_CMDBIT_AX_454	
SSC_CMDBIT_AX_455	
SSC_CMDBIT_AX_456	
Bit No. (constant)	Signal name
SSC_CMDBIT_AX_457	For manufacturer setting
SSC_CMDBIT_AX_458	
SSC_CMDBIT_AX_459	
SSC_CMDBIT_AX_460	
SSC_CMDBIT_AX_461	
SSC_CMDBIT_AX_462	
SSC_CMDBIT_AX_463	
SSC_CMDBIT_AX_464	
Bit No. (constant)	Signal name
SSC_CMDBIT_AX_465	For manufacturer setting
SSC_CMDBIT_AX_466	
SSC_CMDBIT_AX_467	
SSC_CMDBIT_AX_468	
SSC_CMDBIT_AX_469	
SSC_CMDBIT_AX_470	
SSC_CMDBIT_AX_471	
SSC_CMDBIT_AX_472	
Bit No. (constant)	Signal name
SSC_CMDBIT_AX_473	For manufacturer setting
SSC_CMDBIT_AX_474	
SSC_CMDBIT_AX_475	
SSC_CMDBIT_AX_476	
SSC_CMDBIT_AX_477	
SSC_CMDBIT_AX_478	
SSC_CMDBIT_AX_479	
SSC_CMDBIT_AX_480	

Bit No. (constant)	Signal name
SSC_CMDBIT_AX_481	For manufacturer setting
SSC_CMDBIT_AX_482	
SSC_CMDBIT_AX_483	
SSC_CMDBIT_AX_484	
SSC_CMDBIT_AX_485	
SSC_CMDBIT_AX_486	
SSC_CMDBIT_AX_487	
SSC_CMDBIT_AX_488	
Bit No. (constant)	Signal name
SSC_CMDBIT_AX_489	For manufacturer setting
SSC_CMDBIT_AX_490	
SSC_CMDBIT_AX_491	
SSC_CMDBIT_AX_492	
SSC_CMDBIT_AX_493	
SSC_CMDBIT_AX_494	
SSC_CMDBIT_AX_495	
SSC_CMDBIT_AX_496	
Bit No. (constant)	Signal name
SSC_CMDBIT_AX_497	For manufacturer setting
SSC_CMDBIT_AX_498	
SSC_CMDBIT_AX_499	
SSC_CMDBIT_AX_500	
SSC_CMDBIT_AX_501	
SSC_CMDBIT_AX_502	
SSC_CMDBIT_AX_503	
SSC_CMDBIT_AX_504	
Bit No. (constant)	Signal name
SSC_CMDBIT_AX_505	For manufacturer setting
SSC_CMDBIT_AX_506	
SSC_CMDBIT_AX_507	
SSC_CMDBIT_AX_508	
SSC_CMDBIT_AX_509	
SSC_CMDBIT_AX_510	
SSC_CMDBIT_AX_511	
SSC_CMDBIT_AX_512	

5.4 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	Drive unit alarm
SSC_STSBIT_AX_SWRN	Drive unit 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 interpolation operation mode
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_DCDD	Driver command discard detected

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	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	Acceleration time constant change error
SSC_STSBIT_AX_TDCE	Deceleration time constant change error
SSC_STSBIT_AX_PCE	Position change error
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_65	For manufacturer setting
SSC_STSBIT_AX_66	
SSC_STSBIT_AX_67	
SSC_STSBIT_AX_68	
SSC_STSBIT_AX_69	
SSC_STSBIT_AX_70	
SSC_STSBIT_AX_71	
SSC_STSBIT_AX_PRSMO	Pressing in progress
Bit No. (constant)	Signal name
SSC_STSBIT_AX_73	For manufacturer setting
SSC_STSBIT_AX_SINP	Drive unit 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_84	
SSC_STSBIT_AX_85	
SSC_STSBIT_AX_86	
SSC_STSBIT_AX_87	
SSC_STSBIT_AX_88	
Bit No. (constant)	Signal name
SSC_STSBIT_AX_89	For manufacturer setting
SSC_STSBIT_AX_90	
SSC_STSBIT_AX_91	
SSC_STSBIT_AX_92	
SSC_STSBIT_AX_93	
SSC_STSBIT_AX_94	
SSC_STSBIT_AX_95	
SSC_STSBIT_AX_96	
Bit No. (constant)	Signal name
SSC_STSBIT_AX_97	For manufacturer setting
SSC_STSBIT_AX_98	
SSC_STSBIT_AX_99	
SSC_STSBIT_AX_100	
SSC_STSBIT_AX_CTLMCF	Control mode switching completed
SSC_STSBIT_AX_CTLMCE	Control mode switching error
SSC_STSBIT_AX_103	For manufacturer setting
SSC_STSBIT_AX_PRSOVR	Out of torque range during pressing
Bit No. (constant)	Signal name
SSC_STSBIT_AX_PSCMO	Pressure control mode in progress
SSC_STSBIT_AX_PSFD	Feeding/dwelling pressure
SSC_STSBIT_AX_PSDW	Dwelling pressure
SSC_STSBIT_AX_PSPAS	Pressure attainment switching during feed
SSC_STSBIT_AX_PSRL	Releasing pressure
SSC_STSBIT_AX_PSDFSC	Condition of final step dwelling pressure satisfied
SSC_STSBIT_AX_PSMCF	Pressure control mode switching completed
SSC_STSBIT_AX_PSMCE	Pressure control mode switching error
Bit No. (constant)	Signal name
SSC_STSBIT_AX_PSDCF	Dwelling pressure switching completed
SSC_STSBIT_AX_PSDCE	Dwelling pressure switching error
SSC_STSBIT_AX_115	For manufacturer setting
SSC_STSBIT_AX_116	
SSC_STSBIT_AX_117	
SSC_STSBIT_AX_118	
SSC_STSBIT_AX_PSMRC	Pressure control mode resetting completed
SSC_STSBIT_AX_PSMRE	Pressure control mode resetting error

Bit No. (constant)	Signal name
SSC_STSBIT_AX_121	For manufacturer setting
SSC_STSBIT_AX_122	
SSC_STSBIT_AX_123	
SSC_STSBIT_AX_124	
SSC_STSBIT_AX_125	
SSC_STSBIT_AX_126	
SSC_STSBIT_AX_127	
SSC_STSBIT_AX_128	
Bit No. (constant)	Signal name
SSC_STSBIT_AX_129	For manufacturer setting
SSC_STSBIT_AX_130	
SSC_STSBIT_AX_131	
SSC_STSBIT_AX_132	
SSC_STSBIT_AX_133	
SSC_STSBIT_AX_134	
SSC_STSBIT_AX_135	
SSC_STSBIT_AX_136	
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_161	For manufacturer setting
SSC_STSBIT_AX_162	
SSC_STSBIT_AX_163	
SSC_STSBIT_AX_164	
SSC_STSBIT_AX_165	
SSC_STSBIT_AX_166	
SSC_STSBIT_AX_167	
SSC_STSBIT_AX_168	
Bit No. (constant)	Signal name
SSC_STSBIT_AX_169	For manufacturer setting
SSC_STSBIT_AX_170	
SSC_STSBIT_AX_171	
SSC_STSBIT_AX_172	
SSC_STSBIT_AX_173	
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	
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_263	For manufacturer setting
SSC_STSBIT_AX_264	
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_268	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_272	For manufacturer setting
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_277	For manufacturer setting
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	
Bit No. (constant)	Signal name
SSC_STSBIT_AX_385	For manufacturer setting
SSC_STSBIT_AX_386	
SSC_STSBIT_AX_387	
SSC_STSBIT_AX_388	
SSC_STSBIT_AX_389	
SSC_STSBIT_AX_390	
SSC_STSBIT_AX_391	
SSC_STSBIT_AX_392	
Bit No. (constant)	Signal name
SSC_STSBIT_AX_393	For manufacturer setting
SSC_STSBIT_AX_394	
SSC_STSBIT_AX_395	
SSC_STSBIT_AX_396	
SSC_STSBIT_AX_397	
SSC_STSBIT_AX_398	
SSC_STSBIT_AX_399	
SSC_STSBIT_AX_400	

Bit No. (constant)	Signal name
SSC_STSBIT_AX_401	For manufacturer setting
SSC_STSBIT_AX_402	
SSC_STSBIT_AX_403	
SSC_STSBIT_AX_404	
SSC_STSBIT_AX_405	
SSC_STSBIT_AX_406	
SSC_STSBIT_AX_407	
SSC_STSBIT_AX_408	
Bit No. (constant)	Signal name
SSC_STSBIT_AX_409	For manufacturer setting
SSC_STSBIT_AX_410	
SSC_STSBIT_AX_411	
SSC_STSBIT_AX_412	
SSC_STSBIT_AX_413	
SSC_STSBIT_AX_414	
SSC_STSBIT_AX_415	
SSC_STSBIT_AX_416	
Bit No. (constant)	Signal name
SSC_STSBIT_AX_417	For manufacturer setting
SSC_STSBIT_AX_418	
SSC_STSBIT_AX_419	
SSC_STSBIT_AX_420	
SSC_STSBIT_AX_421	
SSC_STSBIT_AX_422	
SSC_STSBIT_AX_423	
SSC_STSBIT_AX_424	
Bit No. (constant)	Signal name
SSC_STSBIT_AX_425	For manufacturer setting
SSC_STSBIT_AX_426	
SSC_STSBIT_AX_427	
SSC_STSBIT_AX_428	
SSC_STSBIT_AX_429	
SSC_STSBIT_AX_430	
SSC_STSBIT_AX_431	
SSC_STSBIT_AX_432	
Bit No. (constant)	Signal name
SSC_STSBIT_AX_433	For manufacturer setting
SSC_STSBIT_AX_434	
SSC_STSBIT_AX_435	
SSC_STSBIT_AX_436	
SSC_STSBIT_AX_437	
SSC_STSBIT_AX_438	
SSC_STSBIT_AX_439	
SSC_STSBIT_AX_440	

Bit No. (constant)	Signal name
SSC_STSBIT_AX_441	For manufacturer setting
SSC_STSBIT_AX_442	
SSC_STSBIT_AX_443	
SSC_STSBIT_AX_444	
SSC_STSBIT_AX_445	
SSC_STSBIT_AX_446	
SSC_STSBIT_AX_447	
SSC_STSBIT_AX_448	
Bit No. (constant)	Signal name
SSC_STSBIT_AX_449	For manufacturer setting
SSC_STSBIT_AX_450	
SSC_STSBIT_AX_451	
SSC_STSBIT_AX_452	
SSC_STSBIT_AX_453	
SSC_STSBIT_AX_454	
SSC_STSBIT_AX_455	
SSC_STSBIT_AX_456	
Bit No. (constant)	Signal name
SSC_STSBIT_AX_457	For manufacturer setting
SSC_STSBIT_AX_458	
SSC_STSBIT_AX_459	
SSC_STSBIT_AX_460	
SSC_STSBIT_AX_461	
SSC_STSBIT_AX_462	
SSC_STSBIT_AX_463	
SSC_STSBIT_AX_464	
Bit No. (constant)	Signal name
SSC_STSBIT_AX_465	For manufacturer setting
SSC_STSBIT_AX_466	
SSC_STSBIT_AX_467	
SSC_STSBIT_AX_468	
SSC_STSBIT_AX_469	
SSC_STSBIT_AX_470	
SSC_STSBIT_AX_471	
SSC_STSBIT_AX_472	
Bit No. (constant)	Signal name
SSC_STSBIT_AX_473	For manufacturer setting
SSC_STSBIT_AX_474	
SSC_STSBIT_AX_475	
SSC_STSBIT_AX_476	
SSC_STSBIT_AX_477	
SSC_STSBIT_AX_478	
SSC_STSBIT_AX_479	
SSC_STSBIT_AX_480	

Bit No. (constant)	Signal name
SSC_STSBIT_AX_481	For manufacturer setting
SSC_STSBIT_AX_482	
SSC_STSBIT_AX_483	
SSC_STSBIT_AX_484	
SSC_STSBIT_AX_485	
SSC_STSBIT_AX_486	
SSC_STSBIT_AX_487	
SSC_STSBIT_AX_488	
Bit No. (constant)	Signal name
SSC_STSBIT_AX_489	For manufacturer setting
SSC_STSBIT_AX_490	
SSC_STSBIT_AX_491	
SSC_STSBIT_AX_492	
SSC_STSBIT_AX_493	
SSC_STSBIT_AX_494	
SSC_STSBIT_AX_495	
SSC_STSBIT_AX_496	
Bit No. (constant)	Signal name
SSC_STSBIT_AX_497	For manufacturer setting
SSC_STSBIT_AX_498	
SSC_STSBIT_AX_499	
SSC_STSBIT_AX_500	
SSC_STSBIT_AX_501	
SSC_STSBIT_AX_502	
SSC_STSBIT_AX_503	
SSC_STSBIT_AX_504	
Bit No. (constant)	Signal name
SSC_STSBIT_AX_505	For manufacturer setting
SSC_STSBIT_AX_506	
SSC_STSBIT_AX_507	
SSC_STSBIT_AX_508	
SSC_STSBIT_AX_509	
SSC_STSBIT_AX_510	
SSC_STSBIT_AX_511	
SSC_STSBIT_AX_512	

6 INTERRUPT EVENT FACTOR LIST

The following lists the interrupt event factors.

System interrupt	
Event factor	Content
SSC_INT_SYS_SYSE	During system status code error
SSC_INT_SYS_CALM	During system alarm
SSC_INT_SYS_EMIO	During forced stop
SSC_INT_SYS_SEO	During system error
SSC_INT_SYS_04	For manufacturer setting
SSC_INT_SYS_05	
SSC_INT_SYS_06	
SSC_INT_SYS_07	
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	Drive unit alarm
SSC_INT_AX_SWRN	Drive unit 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_16	For manufacturer setting
SSC_INT_AX_17	
SSC_INT_AX_18	
SSC_INT_AX_19	
SSC_INT_AX_20	
SSC_INT_AX_21	
SSC_INT_AX_22	
SSC_INT_AX_PRSMO	Pressing in progress
SSC_INT_AX_24	For manufacturer setting
SSC_INT_AX_SINP	Drive unit in-position
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	

7 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 Motion control 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	<p>The axis No. and the command bit No./status bit No./alarm type do not correspond.</p> <p><Example></p> <ul style="list-style-type: none"> • 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_□ □ = 01 to 13: Timeout location	<p>A timeout occurred.</p> <p>Consider changing the set timeout value for API functions which have timeout parameters.</p>
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	SSC_FUNC_ERR_NOT_FOUND_BOARD	The Motion control board cannot be found. Confirm that the device driver is installed, and restart the host personal computer.
00021011H	SSC_FUNC_ERR_GET_CHANNEL_NUM	The mount channel information could not be got. The operating system may not recognize the Motion control board properly. Confirm that the Motion control 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 Motion API that combines with the device driver contained in the EM Motion SDK.
00021014H	SSC_FUNC_ERR_PCIE_NOT_INITIALIZED	The initialization of the Motion control board is not completed. Wait a while and try the execution again.
00021016H	SSC_FUNC_ERR_TIME_IS_NOT_SET	The time setting of the Motion control board is not completed. Restart the host personal computer.
00021019H	SSC_FUNC_ERR_MEMORY_ACCESS_DISABLE	It is under condition that the Motion control board cannot be accessed. Wait a while and try the execution again.
0002101AH	SSC_FUNC_ERR_NOT_EXIST_BOARD_ID	The Motion control board of the designated board ID could not be found. Confirm the board ID selection (dip switch) of the Motion control board.
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.
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.

Parameter functions error

Value	Constant definition	Cause/countermeasure
00040000H	SSC_FUNC_ERR_STS_BIT_PREN□ □ = 1 to 2: Array No. of the parameter read Nos. (for 2 Nos.)	A parameter read error occurred. <ul style="list-style-type: none"> 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.)
00040001H		
00040002H	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.
00040003H		
00040020H	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. <ul style="list-style-type: none"> 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.)
00040021H		
00040022H	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.
00040023H		
00040024H	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.
00040025H		
00040026H	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.
00040027H		

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. <ul style="list-style-type: none"> 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 Motion control board receives the signal after the start operation is requested).
00060020H	SSC_FUNC_ERR_NOW_ALARM_SERVO	A drive unit 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 switching 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 16" 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. <ul style="list-style-type: none"> • 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.

Alarm functions error

Value	Constant definition	Cause/countermeasure
00065000H	SSC_FUNC_ERR_STS_BIT_AERE	The all error reset error signal (AERE) turned ON.

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.

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.
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.

I/O device functions error

Value	Constant definition	Cause/countermeasure
000E0000H	SSC_FUNC_ERR_DVI_TABLE_RANGE_OVER	The "device No." + "point" designated by the argument exceeds the size of the input device table.
000E0001H	SSC_FUNC_ERR_DVO_TABLE_RANGE_OVER	The "device No." + "point" designated by the argument exceeds the size of the output device table.

Initialization/Finalization functions error

Value	Constant definition	Cause/countermeasure
00100000H	SSC_FUNC_ERR_NOT_INITIALIZED	The initialization of the Motion API (sscInitializeLibrary function) is not executed.
00100001H	SSC_FUNC_ERR_ALREADY_INITIALIZED	The initialization of the Motion API (sscInitializeLibrary function) is already executed.

Link device functions error

Value	Constant definition	Cause/countermeasure
00110000H	SSC_FUNC_ERR_LINK_DEVICE_RANGE_OVER	The "device No." + "point" designated by the argument exceeds the size of the link device.

File operation functions error

Value	Constant definition	Cause/countermeasure
00130000H	SSC_FUNC_ERR_NOT_EXIST_DIRPATH	The folder does not exist.
00130001H	SSC_FUNC_ERR_NOT_EXIST_FILE	The file does not exist.
00130002H	SSC_FUNC_ERR_OPEN_FILE	The file cannot be opened.
00130003H	SSC_FUNC_ERR_WRITE_FILE	The writing of the file is failed.
00130004H	SSC_FUNC_ERR_READ_FILE	The reading of the file is failed.
00130005H	SSC_FUNC_ERR_DELETE_FILE	The deleting of the file is failed.
00130006H	SSC_FUNC_ERR_GET_FILE_LIST	The acquisition of the file list is failed.
00130008H	SSC_FUNC_ERR_ACCESS_INVALID_DRIVE_PATH	The drive which cannot be accessed is specified.
00130009H	SSC_FUNC_ERR_CLOSE_FILE	The file could not be closed.
0013000AH	SSC_FUNC_ERR_CREATE_DIRECTORY	Creating the directory is failed.
0013000BH	SSC_FUNC_ERR_DELETE_DIRECTORY	The deleting of the directory is failed.
0013000CH	SSC_FUNC_ERR_FILE_SIZE	The file size is 0 byte.

SLMP functions error

Value	Constant definition	Cause/countermeasure
00140001H	SSC_FUNC_ERR_SLMP_END_CODE	An error occurred during the SLMP communication. Since an end code of the SLMP response message is stored to the end_code of which is stored the execution result of the function, refer to the following manual to check the end code and take actions. MELSEC iQ-R CC-Link IE TSN User's Manual (Application)
00140002H	SSC_FUNC_ERR_READ_SIZE_MISMATCH	The object of the designated size could not be read. Review the read size.
00140003H	SSC_FUNC_ERR_SLMP_SEND_RECEIVE	An error occurred during the SLMP communication. Since an error code is stored to the completion_status of which is stored the execution result of the function, refer to the following manuals to check the error code and take actions. Motion Control Board User's Manual (Motion Control) Motion Control Board User's Manual (Network)
00140004H	SSC_FUNC_ERR_ARGUMENT_SLMP_REQUEST	The request data length of the SLMP request data is outside the set range.
00140005H	SSC_FUNC_ERR_ARGUMENT_SLMP_CONTROL	The arrival monitoring time or the number of resends of the SLMP control data structure is outside the set range.
00140006H	SSC_FUNC_ERR_SLMP_RESPONSE_SIZE_OVER	The size of the response data exceeds the size designated in the argument response_buffer_size.

Arbitrary user area access functions error

Value	Constant definition	Cause/countermeasure
00011001H	SSC_FUNC_ERR_ADDRESS_RANGE_OVER	The "offset" + "size" designated by the argument exceeds the accessible range.

Event history functions error

Value	Constant definition	Cause/countermeasure
00160000H	SSC_FUNC_ERR_DESTINATION_FILE_EXTENSION	The file extension of the read destination is incorrect.
00160001H	SSC_FUNC_ERR_READ_EVENT_HISTORY	The reading of the event history is failed.
00160002H	SSC_FUNC_ERR_CLEAR_EVENT_HISTORY	The clearing of the event history is failed.
00160003H	SSC_FUNC_ERR_SAVE_DESTINATION_FILE	The saving of the file of the read destination is failed.

APPENDIX

Appendix 1 Differences between MR-MC341 and MR-EM441G

Differences between Motion API/MR-EM441G

Differences between the MR-MC341 API library and the MR-EM441G Motion API are shown below.

Item	MR-MC341	MR-EM441G
Maximum station No.	16	0 ^{*1}
I/O bit device No.	0000h to 23FFh	0000h to 3FFFh
I/O word device No.	0000h to 23FFh	0000h to 1FFFh

*1 The number of connectable remote stations is "0".

API functions added to the MR-EM441G Motion API

New API functions added to the MR-EM441G Motion API are shown below.

Function type	Function name	Function content	Reference section
Information functions	sscGetBoardVersionEx	Gets the Motion control board version information.	Page 34 sscGetBoardVersionEx
	sscGetOperationCycleMonitorEx	Gets the operation cycle monitor data.	Page 35 sscGetOperationCycleMonitorEx
	sscGetBoardSerialNumberW	Gets the Motion control board serial No.	Page 37 sscGetBoardSerialNumberW
	sscGetMotionApiVersion	Gets the Motion API version.	Page 38 sscGetMotionApiVersion
System functions	sscGetSystemStatusCodeEx	Gets the system status code.	Page 51 sscGetSystemStatusCodeEx
Command/status functions	sscGetStatusBits	Gets the status bit all at once.	Page 57 sscGetStatusBits
	sscCheckBitStatus	Gets the status bit specified from the status bit array.	Page 58 sscCheckBitStatus
Continuous operation to torque control data functions	sscSetPressTargetTorque	Sets the target torque of continuous operation to torque control data.	Page 67 sscSetPressTargetTorque
	sscGetPressTargetTorque	Gets the target torque of continuous operation to torque control data.	Page 68 sscGetPressTargetTorque
Alarm functions	sscGetSystemErrorCode	Gets the system error code.	Page 104 sscGetSystemErrorCode
	sscResetAllError	Executes the all error reset.	Page 105 sscResetAllError
General monitor functions	sscSetMonitorEm	Starts the monitoring.	Page 108 sscSetMonitorEm
	sscGetMonitorEm	Gets the monitoring data.	Page 110 sscGetMonitorEm
User watchdog functions	sscCheckPCleBusConnection asdfadf	Confirms the PCle bus connection status.	Page 120 sscCheckPCleBusConnection
Sampling functions	sscGetSamplingDataEm	Gets the sampling data.	Page 136 sscGetSamplingDataEm

Function type	Function name	Function content	Reference section
I/O device functions	sscGetInputDeviceDword	Gets the input word device of the motion control station on a 2-word basis.	Page 163 sscGetInputDeviceDword
	sscSetOutputDeviceBitNonExclusively	Sets the output bit device of the motion control station on a 1-point basis without using the dual port memory exclusive control function of the Motion control board.	Page 165 sscSetOutputDeviceBitNonExclusively
	sscSetOutputDeviceDword	Sets the output word device on a 2-word basis without using the dual port memory exclusive control function of the Motion control board.	Page 167 sscSetOutputDeviceDword
	sscChangeOutputDeviceWord	Sets the value of a specific bit of the device designated on a 1-word basis among the output word devices of the motion control station with the dual port memory exclusive control function of the Motion control board.	Page 168 sscChangeOutputDeviceWord
	sscChangeOutputDeviceDword	Sets the value of a specific bit of the device designated on a 2-word basis among the output word devices of the motion control station with the dual port memory exclusive control function of the Motion control board.	Page 169 sscChangeOutputDeviceDword
	sscGetOutputDeviceDword	Gets the output word device of the motion control station on a 2-word basis.	Page 171 sscGetOutputDeviceWord
Initialize/finalize functions	sscInitializeLibrary	Executes the process of initialization of the Motion API.	Page 174 sscInitializeLibrary
	sscFinalizeLibrary	Executes the process of finalization of the Motion API.	Page 175 sscFinalizeLibrary
Link device access functions	sscGetBitLinkDevice	Gets more than one link device of the general station on a 1-bit basis.	Page 176 sscGetBitLinkDevice
	sscGetWordLinkDevice	Gets more than one link device of the general station on a 1-word basis.	Page 178 sscGetWordLinkDevice
	sscSetBitLinkDevice	Sets more than one link device of the general station on a 1-bit basis.	Page 179 sscSetBitLinkDevice
	sscSetWordLinkDevice	Sets more than one link device of the general station on a 1-word basis.	Page 180 sscSetWordLinkDevice
File operation functions	sscWriteFile	Writes the file.	Page 182 sscWriteFile
	sscReadFile	Reads the file.	Page 184 sscReadFile
	sscDeleteFile	Deletes the file.	Page 186 sscDeleteFile
	sscCreateDirectory	Creates the directory.	Page 187 sscCreateDirectory
	sscDeleteDirectory	Deletes the directory.	Page 189 sscDeleteDirectory
	sscGetFileList	Gets the file list.	Page 190 sscGetFileList
	sscGetFileListEx		Page 192 sscGetFileListEx
	sscFreeFileList	Release the file list.	Page 194 sscFreeFileList
	sscFreeFileListEx		Page 195 sscFreeFileListEx
SLMP functions	sscSimpReadSlaveObject	Reads the object of the device which is connected to the network.	Page 196 sscSimpReadSlaveObject
	sscSimpWriteSlaveObject	Writes the object of the device which is connected to the network.	Page 198 sscSimpWriteSlaveObject
	sscSimpSendCommand	Sends/receives the SLMP message to the device which is connected to the network.	Page 200 sscSimpSendCommand

Function type	Function name	Function content	Reference section
General input/output functions	sscGetGeneralInputDataBit	Gets the designated general input/output DI data on a 1-point basis.	Page 203 sscGetGeneralInputDataBit
	sscGetGeneralInputDataWord	Gets the designated general input/output DI data on a 4-point basis.	Page 204 sscGetGeneralInputDataWord
	sscSetGeneralOutputDataBit	Sets the designated general input/output DO data on a 1-point basis.	Page 205 sscSetGeneralOutputDataBit
	sscSetGeneralOutputDataBitExclusively	Sets the designated general input/output DO data on a 1-point basis with the general output exclusive control function of the Motion control board.	Page 206 sscSetGeneralOutputDataBitExclusively
	sscSetGeneralOutputDataWord	Sets the designated general input/output DO data on a 4-point basis.	Page 207 sscSetGeneralOutputDataWord
	sscSetGeneralOutputDataWordExclusively	Sets the designated general input/output DO data on a 4-point basis with the general output exclusive control function of the Motion control board.	Page 208 sscSetGeneralOutputDataWordExclusively
	sscGetGeneralOutputDataBit	Gets the designated general input/output DO data on a 1-point basis.	Page 209 sscGetGeneralOutputDataBit
	sscGetGeneralOutputDataWord	Gets the designated general input/output DO data on a 4-point basis.	Page 210 sscGetGeneralOutputDataWord
Arbitrary user area access functions	sscSetUserArea	Writes the data in the arbitrary user area.	Page 211 sscSetUserArea
	sscGetUserArea	Reads the data in the arbitrary user area.	Page 212 sscGetUserArea
Event history functions	sscReadEventHistoryData	Reads the event history.	Page 213 sscReadEventHistoryData
	sscClearEventHistoryData	Clear the event history.	Page 214 sscClearEventHistoryData
Pressure control functions	sscSetPrsProfile	Sets the pressure control profile.	Page 215 sscSetPrsProfile
	sscGetPrsProfile	Gets the pressure control profile.	Page 216 sscGetPrsProfile
	sscGetPrsStatus	Gets the pressure control status.	Page 217 sscGetPrsStatus
	sscGetExecutingPrsNumber	Gets the profile No. of pressure control in progress.	Page 218 sscGetExecutingPrsNumber
	sscGetPrsForwardSidePosition	Gets the tip position in pressure control.	Page 219 sscGetPrsForwardSidePosition
	sscGetPrsSpeedLimit	Gets the speed limit value.	Page 220 sscGetPrsSpeedLimit
	sscGetPrsPressureCommand	Gets the pressure command value.	Page 221 sscGetPrsPressureCommand
	sscGetPrsLoadCellPressure	Gets the load cell pressure.	Page 222 sscGetPrsLoadCellPressure
	sscGetPrsError	Gets the error information during pressure control.	Page 223 sscGetPrsError
	sscSetPrsProfileLimitPosition	Sets the limit position of the designated pressure control profile.	Page 224 sscSetPrsProfileLimitPosition
	sscGetPrsProfileLimitPosition	Gets the limit position of the designated pressure control profile.	Page 225 sscGetPrsProfileLimitPosition
	sscSetPrsProfileSpeedLimit	Sets the speed limit value of the designated pressure control profile.	Page 226 sscSetPrsProfileSpeedLimit
	sscGetPrsProfileSpeedLimit	Gets the speed limit value of the designated pressure control profile.	Page 227 sscGetPrsProfileSpeedLimit
	sscSetPrsProfilePressureCommand	Sets the pressure command of the designated pressure control profile.	Page 228 sscSetPrsProfilePressureCommand
	sscGetPrsProfilePressureCommand	Gets the pressure command of the designated pressure control profile.	Page 229 sscGetPrsProfilePressureCommand

MR-EM441G replacements for MR-MC341 API functions

- The following MR-MC341 functions shown in the table below cannot be used with MR-EM441G. Replace these functions with the appropriate MR-EM441G API functions.

MR-EM441G replacements for MR-MC341 API functions are shown in the table below.

Function type	Function name	
	MR-MC341 API functions	MR-EM441G API functions
Information functions	sscGetBoardVersion	sscGetBoardVersionEx
	sscGetOperationCycleMonitor	sscGetOperationCycleMonitorEx
General monitor functions	sscSetMonitor	sscSetMonitorEm
	sscGetMonitor	sscGetMonitorEm
Sampling functions	sscGetSamplingData	sscGetSamplingDataEm
Transient transmit functions	sscSendReceiveTransientData	sscSmpReadSlaveObject/sscSmpWriteSlaveObject

- While the following MR-MC341 API functions can be used with MR-EM441G, we recommend using MR-EM441G functions with MR-EM441G. MR-EM441G replacements for MR-MC341 API functions are shown in the table below.

Function type	Function name	
	MR-MC341 API functions	MR-EM441G API functions
System functions	sscGetSystemStatusCode	sscGetSystemStatusCodeEx

- 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-EM441G models.

Function type	Function name	Structure
System functions	sscGetControllingAxis	SLAVE_INFO structure
Sampling functions	sscGetSamplingError	SMP_ERR structure
Interrupt functions	sscRegisterIntCallback	INT_CB_DATA structure

INDEX

A

Alarm functions	102
Arbitrary user area access functions	211

C

Change functions	92
Command/Status functions	53
Continuous operation to torque control data functions	65

D

Device functions	30
----------------------------	----

E

Event history functions	213
-----------------------------------	-----

F

File operation functions	181
FILE_INFO structure	239
FILE_INFO_EX structure	240
FILE_TIME structure	241

G

General input/output functions	203
General monitor functions	106

H

High speed monitor functions	111
----------------------------------------	-----

I

I/O device functions	161
Information functions	33
Initialization/Finalization functions	173
INT_CB_DATA structure	237
Interrupt functions	137

L

Link device access functions	176
----------------------------------------	-----

O

OAS_DATA structure	232
Operating functions	69
Other axes start functions	121

P

Parameter functions	39
Pass position interrupt functions	126
PNT_DATA_EX structure	230
Point table functions	59
PRESS_DATA structure	234

Pressure control functions	215
PRS_CMN_SETTING structure	248
PRS_DWELL_DATA structure	252
PRS_DWELL_STEP structure	253
PRS_FEED_DATA structure	249
PRS_FEED_STEP structure	250
PRS_PROFILE structure	246
PRS_RELEASE_DATA structure	254
PRS_SWITCH structure	255
PRS_SWITCH_TO_DWELL structure	251
PRS_SWITCH_TO_PRESSURE structure	247

S

Sampling functions	130
SLAVE_INFO structure	238
SLMP functions	196
SLMP_CONTROL structure	244
SLMP_SLV_OBJ_CMD structure	242
SLMP_SLV_OBJ_STS structure	243
SMP_DATA_EM structure	236
SMP_ERR structure	235
sscAutoStart	75
sscChange2Parameter	42
sscChangeAutoAccTime	99
sscChangeAutoDecTime	101
sscChangeAutoPosition	93
sscChangeAutoSpeed	97
sscChangeControlMode	91
sscChangeLinearPosition	94
sscChangeManualAccTime	98
sscChangeManualDecTime	100
sscChangeManualPosition	92
sscChangeManualSpeed	96
sscChangeOutputDeviceDword	169
sscChangeOutputDeviceWord	168
sscChangeParameter	40
sscChangeWdCounter	119
sscCheck2Parameter	45
sscCheckBitStatus	58
sscCheckIntPassPositionData	127
sscCheckParameter	44
sscCheckPCleBusConnection	120
sscCheckPointDataEx	60
sscCheckPointOffset	62
sscClearEventHistoryData	214
sscClose	32
sscCreateDirectory	187
sscDataSetStart	81
sscDeleteDirectory	189
sscDeleteFile	186
sscDriveRapidStop	84
sscDriveRapidStopNoWait	85
sscDriveStop	82
sscDriveStopNoWait	83
sscFinalizeLibrary	175
sscFreeFileList	194
sscFreeFileListEx	195
sscGetAlarm	102
sscGetBitLinkDevice	176
sscGetBoardSerialNumber	36
sscGetBoardSerialNumberW	37

sscGetBoardVersionEx	34
sscGetCmdSpeedFast	114
sscGetControlCycle	33
sscGetControllingAxis	52
sscGetCurrentCmdPositionFast	111
sscGetCurrentFbFast	116
sscGetCurrentFbPositionFast	112
sscGetDriveFinStatus	89
sscGetDriveMode	88
sscGetDrivingPointNumber	63
sscGetExecutingPassNumber	129
sscGetExecutingPrsNumber	218
sscGetFbSpeedFast	115
sscGetFileList	190
sscGetFileListEx	192
sscGetGeneralInputDataBit	203
sscGetGeneralInputDataWord	204
sscGetGeneralOutputDataBit	209
sscGetGeneralOutputDataWord	210
sscGetInputDeviceBit	161
sscGetInputDeviceDword	163
sscGetInputDeviceWord	162
sscGetIoStatusFast	113
sscGetLastError	28
sscGetMonitorEm	110
sscGetMotionApiVersion	38
sscGetMountChannel	29
sscGetOperationCycleMonitorEx	35
sscGetOtherAxisStartData	122
sscGetOtherAxisStartStatus	125
sscGetOutputDeviceBit	170
sscGetOutputDeviceDword	172
sscGetOutputDeviceWord	171
sscGetPressData	66
sscGetPressTargetTorque	68
sscGetPrsError	223
sscGetPrsForwardSidePosition	219
sscGetPrsLoadCellPressure	222
sscGetPrsPressureCommand	221
sscGetPrsProfile	216
sscGetPrsProfileLimitPosition	225
sscGetPrsProfilePressureCommand	229
sscGetPrsProfileSpeedLimit	227
sscGetPrsSpeedLimit	220
sscGetPrsStatus	217
sscGetSamplingDataEm	136
sscGetSamplingError	134
sscGetSamplingParameter	133
sscGetSamplingStatus	135
sscGetStatusBits	57
sscGetStatusBitSignalEx	55
sscGetSystemErrorCode	104
sscGetSystemStatusCode	50
sscGetSystemStatusCodeEx	51
sscGetUserArea	212
sscGetWordLinkDevice	178
sscHomeReturnStart	76
ssclncStart	73
ssclnitalizeLibrary	174
ssclntDisable	142
ssclntEnable	141
ssclntEnd	140
ssclnterpolationStart	79
ssclntStart	139
sscJogStart	69
sscJogStop	71

sscJogStopNoWait	72
sscLinearStart	77
sscOpen	31
sscOtherAxisStartAbortOff	124
sscOtherAxisStartAbortOn	123
sscReadEventHistoryData	213
sscReadFile	184
sscReboot	48
sscRegisterIntCallback	143
sscResetAlarm	103
sscResetAllError	105
sscResetIntDriveFin	157
sscResetIntEvent	145
sscResetIntOasEvent	149
sscResetIntPassPosition	153
sscSetBitLinkDevice	179
sscSetCommandBitSignalEx	54
sscSetDriveMode	86
sscSetGeneralOutputDataBit	205
sscSetGeneralOutputDataBitExclusively	206
sscSetGeneralOutputDataWord	207
sscSetGeneralOutputDataWordExclusively	208
sscSetIntDriveFin	158
sscSetIntEvent	146
sscSetIntOasEvent	150
sscSetIntPassPosition	154
sscSetIntPassPositionData	126
sscSetLatestPointNumber	64
sscSetMonitorEm	108
sscSetOtherAxisStartData	121
sscSetOutputDeviceBit	164
sscSetOutputDeviceBitNonExclusively	165
sscSetOutputDeviceDword	167
sscSetOutputDeviceWord	166
sscSetPointDataEx	59
sscSetPointOffset	61
sscSetPressData	65
sscSetPressTargetTorque	67
sscSetPrsProfile	215
sscSetPrsProfileLimitPosition	224
sscSetPrsProfilePressureCommand	228
sscSetPrsProfileSpeedLimit	226
sscSetSamplingParameter	132
sscSetStartingPassNumber	128
sscSetUserArea	211
sscSetWordLinkDevice	180
sscSmpReadSlaveObject	196
sscSmpSendCommand	200
sscSmpWriteSlaveObject	198
sscStartSampling	130
sscStopMonitor	109
sscStopSampling	131
sscSystemStart	49
sscUnregisterIntCallback	144
sscWaitIntDriveFin	159
sscWaitIntEvent	147
sscWaitIntOasEvent	151
sscWaitIntPassPosition	155
sscWaitStatusBitSignalEx	56
sscWdDisable	118
sscWdEnable	117
sscWriteFile	182
Support functions	28
System functions	47

U

User watchdog functions	117
-----------------------------------	-----

REVISIONS

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

Revision date	*Manual number	Description
August 2023	IB(NA)-0300601ENG-A	First edition
June 2024	IB(NA)-0300601ENG-B	<p>■Added functions</p> <p>Continuous operation to torque control data functions, Operating functions (sscChangeControlMode), I/O device functions (sscSetOutputDeviceBitNonExclusively, sscChangeOutputDeviceWord, sscChangeOutputDeviceDword), Link device access functions (sscSetWordLinkDevice), General Input/output functions (sscSetGeneralOutputDataBitExclusively, sscSetGeneralOutputDataWordExclusively), Pressure control functions, PRESS_DATA structure, PRS_PROFILE structure</p> <p>■Added or modified parts</p> <p>SAFETY PRECATIONS, TERMS, GENERIC TERMS AND ABBREVIATIONS, Chapter 1, Section 1.1, Chapter 2, Section 3.2, 3.6, 3.7, 3.8, 3.9, 3.15, 3.16, 3.18, 3.19, 3.21, 3.22, 3.23, 3.24, 3.27, 4.1, 4.2, 4.3, 4.14, 5.2, 5.3, 5.4, Chapter 6, Chapter 7, Appendix 1, INDEX, INFORMATION AND SERVICES, TRADEMASRKS</p>

Japanese manual number: IB-0300594-B

This manual confers no industrial property rights or any rights of any other kind, nor does it confer any patent licenses. Mitsubishi Electric Corporation cannot be held responsible for any problems involving industrial property rights which may occur as a result of using the contents noted in this manual.

© 2023 MITSUBISHI ELECTRIC CORPORATION

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 Motion control board, its applications should be those that may not result in a serious damage even if any failure or malfunction occurs in the Motion control board, and a backup or fail-safe function should operate on an external system to the Motion control board when any failure or malfunction occurs.

(2) Our Motion control 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

CANopen[®] is registered Community Trademarks of CAN in Automation e.V.

Microsoft, Visual C++, Visual C#, and Windows are trademarks of the Microsoft group of companies.

Unicode is either a registered trademark or a trademark of Unicode, Inc. in the United States and other countries.

PCI Express is either a registered trademark or a trademark of PCI-SIG.

The company names, system names and product names mentioned in this manual are either registered trademarks or trademarks of their respective companies.

In some cases, trademark symbols such as [™] or [®] are not specified in this manual.

IB(NA)-0300601ENG-B(2406)MEE

MODEL: EM441-U-API-E

MODEL CODE: 1XB060

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

When exported from Japan, this manual does not require application to the
Ministry of Economy, Trade and Industry for service transaction permission.

Specifications subject to change without notice.