Panasonic®

Motion Controller

GM1 Series

Reference Manual

Instruction Edition

WUME-GM1PGR-01

2021.2

(MEMO)

Introduction

Thank you for purchasing a Panasonic product. Before you use the product, please carefully read through the installation instructions and the manuals, and understand them in detail to use the product properly.

Types of Manuals

- This manual describes ladder instructions, function instructions, and function block instructions.
- There are different types of manuals for the GM1 series, as listed below. Refer to the appropriate manual according to your need.
- These manuals can be downloaded from our website: https://industrial.panasonic.com/ac/e/motor/motion-controller/mc/gm1/index.jsp

Manuals for GM1 series

Manual name	Manual code	Description
GM1 Controller RTEX User's Manual (Setup)	WUME-GM1RTXSU	Explains wiring between the GM1 and its peripheral devices, installation method, and operation check method.
GM1 Controller RTEX User's Manual (Operation)	WUME-GM1RTXOP	Explains how to use GM Programmer and PANATERM Lite for GM, set up each function, create projects, and perform other operations.
GM1 Series Reference Manual (Hardware)	WUME-GM1H	Explains the functions and performance of each GM1 unit.
GM1 Series Reference Manual (Instruction)	WUME-GM1PGR	Explains the specifications of each instruction that can be used with the GM1 Series.

Copyright / Trademarks

Note on the Software

The "PANATERM Lite for GM" is not incorporated into the GM Programmer as of January 2021. The features will be added when the GM Programmer is upgraded.

(MEMO)

Table of Contents

(MEMO)

1 List of Instructions

1.1 List of Ladder Instructions

The following table lists contact and coil ladder instructions that can be used in ladder diagram programs for GM Programmer.

Name	Code	Description	Page
NO contact	-| |-	This instruction outputs a BOOL-type input from the left to the right. If the variable of the contact is TRUE, then the input value from the left is output. If the variable of the contact is FALSE, then FALSE is output.	"P.2-2"
NC contact	-|/|-	This instruction outputs the negated value of the BOOL-type input from the left to the right. If the variable of the contact is TRUE, then FALSE is output. If the variable of the contact is FALSE, then the input value from the left is output.	"P.2-3"
Rising edge detection	-|P|-	When a rising edge is detected in the BOOL-type input from the left, TRUE is output for one cycle only.	"P.2-3"
Falling edge detection	-|N|-	When a falling edge is detected in the BOOL-type input from the left, TRUE is output for one cycle only	"P.2-4"
Parallel NO contact	-	NO contacts can be wired in parallel. The contacts wired in parallel are treated as OR logic. If the output of one or more contacts is TRUE, TRUE is output.	"P.2-5"
Parallel NC contact	-	NC contacts can be wired in parallel. The contacts wired in parallel are treated as OR logic. If the output of one or more contacts is TRUE, TRUE is output.	"P.2-6"
Coil	-( )	A BOOL-type input from the left can be saved. If the input is TRUE, then TRUE is saved. If the input is FALSE, then FALSE is saved.	"P.2-7"
Negated coil	-(/)	The negated value of the BOOL-type input from the left can be saved. If the input is TRUE, then FALSE is saved. If the input is FALSE, then TRUE is saved.	"P.2-7"
Set coil	-(S)	If the BOOL-type input from the left becomes TRUE, TRUE is saved. It can be used together with the reset coil.	"P.2-8"
Reset coil	-(R)	If the BOOL-type input from the left becomes TRUE, FALSE is saved. It can be used together with the set coil.	"P.2-9"

1.2 List of Function Instructions

This section provides lists of the functions used by the GM Programmer. These functions can be used without declaring them with variables.

■ Substitution instruction

Name	Function	Description	Page
MOVE	Substitution	Substitutes the input argument values with the output argument.	"P.3-3"

■ Arithmetic operation instructions

Name	Function	Description	Page
ADD	Addition	Adds the input arguments.	"P.3-4"
SUB	Subtraction	Subtracts the input arguments.	"P.3-5"
MUL	Multiplication	Multiplies the input arguments.	"P.3-6"
DIV	Division	Divides the input arguments.	"P.3-7"
MOD	Mod	Outputs the remainder of the input argument.	"P.3-8"

■ Boolean operation instructions

Name	Function	Description	Page
AND	Logical AND	Outputs the logical AND of the input arguments.	"P.3-9"
OR	Logical OR	Outputs the logical OR of the input arguments.	"P.3-10"
XOR	Exclusive OR	Outputs the Exclusive OR of the input arguments.	"P.3-11"
NOT	Negation	Outputs the negation of the input argument.	"P.3-10"
AND_THEN	Logical AND	Outputs the logical AND of the input arguments.	"P.3-12"
OR_ELSE	Logical OR	Outputs the logical OR of the input arguments.	"P.3-13"

■ Comparison operation instructions

Name	Function	Description	Page
EQ	“Equal” comparison	Compares the two input arguments and, if they are equal to each other, outputs TRUE.	"P.3-15"
NE	“Not Equal” comparison	Compares the two input arguments and, if they are not equal to each other, outputs TRUE.	"P.3-15"
LT	“Less Than” comparison	Compares the two input arguments and, if the first argument is less than the second argument, outputs TRUE.	"P.3-16"
LE	“Less Than or Equal” comparison	Compares the two input arguments and, if the first argument is less than the second argument or equal, outputs TRUE.	"P.3-17"
GT	“Greater Than” comparison	Compares the two input arguments and, if the first argument is greater than the second argument, outputs TRUE.	"P.3-18"

Name	Function	Description	Page
GE	“Greater Than Or Equal” comparison	Compares the two input arguments and, if the first argument is greater than the second argument or equal, outputs TRUE.	"P.3-19"

■ Bit shift instructions

Name	Function	Description	Page
SHL	Shift left	Shifts the input argument to the left by the specified number of bits. Inserts “0” from the least significant bit to the specified bit and outputs the data.	"P.3-21"
SHR	Shift right	Shifts the input argument to the right by the specified number of bits. Inserts “0” from the most significant bit to the specified bit and outputs the data.	"P.3-22"
ROL	Rotate left	Shifts the input argument to the left by the specified number of bits. Inserts the value in excess from the most significant bit into the data starting from the least significant bit and outputs the data.	"P.3-22"
ROR	Rotate right	Shifts the input argument to the right by the specified number of bits. Inserts the value in excess from the least significant bit into the data starting from the most significant bit and outputs the data.	"P.3-23"

■ Selection instructions

Name	Function	Description	Page
SEL	Binary selector	Outputs “IN0” when the input argument G is FALSE and “IN1” when the input argument G is TRUE.	"P.3-25"
MUX	Multiplexer	Outputs the input argument value depending on the input argument K (0,1,2,…).	"P.3-26"
LIMIT	Limiter	Limits the value of the input argument IN between the input arguments MN and MX and outputs the data.	"P.3-27"
MAX	Maximum value	Outputs the maximum value of the input argument.	"P.3-28"
MIN	Minimum value	Outputs the minimum value of the input argument.	"P.3-28"

■ Numerical operation instructions

Name	Function	Description	Page
ABS	Absolute value	Outputs the absolute value.	"P.3-30"
SQRT	Square root	Outputs the the square root (√) of a number.	"P.3-30"
LN	Natural logarithm	Outputs the natural logarithm ($\log_e X$) of a number.	"P.3-31"
LOG	Common logarithm	Outputs the common logarithm ($\log_{10} X$) of a number.	"P.3-32"
EXP	Natural exponent	Outputs the natural exponent (e^X) of a number.	"P.3-33"
EXPT	Exponentiation	Outputs the exponentiation of a number (a^n).	"P.3-33"
SIN	Trigonometric function (sine)	Outputs the result of the sine function calculation.	"P.3-34"
COS	Trigonometric function (cosine)	Outputs the result of the cosine function calculation.	"P.3-35"

Name	Function	Description	Page
TAN	Trigonometric function (tangent)	Outputs the result of the tangent function calculation.	"P.3-36"
ASIN	Trigonometric function (arc sine)	Outputs the result of the arc sine function calculation.	"P.3-37"
ACOS	Trigonometric function (arc cosine)	Outputs the result of the arc cosine function calculation.	"P.3-38"
ATAN	Trigonometric function (arc tangent)	Outputs the result of the arc tangent function calculation.	"P.3-38"

■ Data type conversion instructions

Name	Function	Description	Page
<Type 1>_TO_<Type 2>	Data type conversion	Converts type 1 input argument to type 2.	"P.3-40"
TRUNC	Data type conversion	Changes the real number to the DINT-type data.	"P.3-41"
TRUNC_INT	Data type conversion	Changes the real number to the INT-type data.	"P.3-41"

■ Character string instructions

The standard library is required.

Name	Function	Description	Page
LEN	Length of a character string	Outputs the length of a character string.	"P.3-43"
LEFT	Extracting characters from the left end	Extracts a character string consisting of the specified number of characters from the left of the character string.	"P.3-43"
RIGHT	Extracting characters from the right end	Extracts a character string consisting of the specified number of characters from the right of the character string.	"P.3-44"
MID	Extracting characters from the specified position	Extracts a character string consisting of the specified number of characters from the specified position of the character string.	"P.3-45"
CONCAT	Concatenating character strings	Concatenates two character strings.	"P.3-46"
INSERT	Inserting a character string	Inserts another character string into the specified position of one character string.	"P.3-47"
DELETE	Deleting a character string	Deletes a character string consisting of the specified number of characters from the specified position of the character string.	"P.3-47"
REPLACE	Replacing a character string	Replaces a character string, consisting of the specified number of characters from the specified position of the character string, with another character string.	"P.3-48"
FIND	Search for a character string	Searches for a specified character string in the character strings and outputs the position.	"P.3-50"

Other instructions

Name	Function	Description	Page
SIZEOF	Get the size	Outputs the size (in units of byte) of the input argument.	"P.3-51"
ADR	Get the address	Outputs the address of the input argument.	"P.3-51"

SD memory card slot instruction

Name	Function	Description	Page
SYS_GetSDCoverState	Get SD card cover open / close state	Gets an open / close state of the card cover for the SD memory card slot.	"P.3-53"
SYS_GetSDAccessRdy	Get SD card access ready state	Gets the state whether an access to the SD memory card is allowed.	"P.3-53"

1.3 List of Function Block Instructions

This section provides lists of the function blocks used by the GM Programmer. These function blocks can be used with declaring the instances with variables.

1.3.1 Basic Instructions

■ Timer instructions

Name	Function	Description	Page
TON	Timer ON	Starts the timer when the input argument changes from FALSE to TRUE and, after an elapse of the specified time, the output argument outputs TRUE.	"P.4-2"
TOF	Timer OFF	Starts the timer when the input argument changes from TRUE to FALSE and, after an elapse of the specified time, the output argument outputs FALSE.	"P.4-3"
TP	Timer pulse	Starts the timer when the input argument changes from FALSE to TRUE until the specified time elapses. Outputs TRUE to the output argument while the timer keeps counting.	"P.4-4"
RTC	Realtime clock	Starts counting time from the specified date and time when the input argument changes from FALSE to TRUE. Outputs TRUE to the output argument while the clock keeps counting time.	"P.4-6"

■ Counter instructions

Name	Function	Description	Page
CTU	Up counter	Starts incrementing the counter value at the rising edge of the input argument CU and, after counting the specified number of count values, outputs TRUE.	"P.4-7"
CTD	Down counter	Starts decrementing from the specified number of count value at the rising edge of the input argument CD. Outputs TRUE when it reaches 0.	"P.4-8"
CTUD	Up-down counter	Starts incrementing the counter value at the rising edge of the input argument CU and, after counting the specified number of count values, outputs TRUE. Starts decrementing the counter value at the rising edge of the input argument CD and, when it reaches 0, outputs TRUE.	"P.4-9"

■ Edge detection instructions

Name	Function	Description	Page
R_TRIG	Rising edge detection	Outputs TRUE for one cycle only when detecting a rising edge.	"P.4-11"
F_TRIG	Falling edge detection	Outputs TRUE for one cycle only when detecting a falling edge.	"P.4-11"

■ Bistable circuit instructions

Name	Function	Description	Page
SR	Set-priority bistable circuit	If the input argument SET1 is TRUE, outputs TRUE. If the input argument RESET is TRUE, outputs FALSE. If both SET1 and RESET1 are TRUE, outputs TRUE	"P.4-13"
RS	Reset-priority bistable circuit	If the input argument SET1 is TRUE, outputs TRUE. If the input argument RESET is TRUE, outputs FALSE. If both SET1 and RESET1 are TRUE, outputs FALSE.	"P.4-14"

1.3.2 Motion Control Function Blocks (Single Axis Control)

■ Servo ON

Name	Function	Description	Page
MC_Power	Servo ON	Sets the axis to the servo ON state to be ready for operation.	"P.5-2"

■ Home return

Name	Function	Description	Page
PMC_Home	Home return	Performs home return operation on the axis. Uses the home return function of the amplifier.	"P.5-4"

■ Control switch

Name	Function	Description	Page
SMC_SetControllerMode	Control mode setup	Sets up the control mode for controlling the position, velocity, and torque.	"P.5-7"

■ Stop

Name	Function	Description	Page
MC_Stop	Forced stop	Causes the axis to make a deceleration stop. After stopping, the axis remains stopped while Execute is TRUE.	"P.5-8"
MC_Halt	Stop	Causes the axis to make a deceleration stop. After the axis is stopped or while the axis is being decelerated, other motion instructions can be executed.	"P.5-9"

■ JOG / Inching

Name	Function	Description	Page
MC_Jog	Jogging	Causes the axis to keep traveling in a forward or reverse direction at a	"P.5-10"

Name	Function	Description	Page
		constant velocity while the input is TRUE.	
SMC_Inch	Inching	Causes the axis to travel in a forward or reverse direction for a specified relative distance when the input turns TRUE.	"P.5-11"

■ Position control

Name	Function	Description	Page
MC_MoveAbsolute	Absolute value positioning	Causes the axis to travel to a position specified as an absolute position.	"P.5-13"
MC_MoveRelative	Relative value positioning	Causes the axis to travel to a position specified as a relative position.	"P.5-14"
MC_MoveAdditive	Change target position	Adds a relative distance to the target position of the immediately preceding instruction.	"P.5-15"
MC_MoveSuperImposed	Superimposed positioning	Adds a relative distance, a velocity, an acceleration, and a deceleration to the operations of the immediately preceding instruction.	"P.5-17"
MC_PositionProfile	Position profile move	Causes the axis to operate according to the profile data that consists of a combination of position and time.	"P.5-19"
SMC_MoveContinuousAbsolute	Absolute value position velocity move	Executes absolute value positioning and, after the axis reaches the target position, causes the axis to keep moving at a specified velocity.	"P.5-22"
SMC_MoveContinuousRelative	Relative value position velocity move	Executes relative value positioning and, after the axis reaches the target position, causes the axis to keep moving at a specified velocity.	"P.5-23"

■ Velocity control

Name	Function	Description	Page
MC_MoveVelocity	Velocity control	Specifies the velocity of the axis.	"P.5-25"
MC_VelocityProfile	Velocity profile move	Causes the axis to operate according to the profile data that consists of a combination of velocity and time.	"P.5-26"
MC_AccelerationProfile	Acceleration profile move	Causes the axis to operate according to the profile data that consists of a combination of time and acceleration.	"P.5-27"

■ Torque control

Name	Function	Description	Page
PMC_SetTorque	Torque control	Specifies the torque of the axis.	"P.5-30"

1.3.3 Motion Control Function Blocks (Synchronous Control)

■ Cam operation

Name	Function	Description	Page
MC_CamIn	Start cam control	Starts cam synchronous operation.	"P.6-2"
MC_CamOut	Cancel cam operation	Cancels cam synchronous operation.	"P.6-5"
MC_CamTableSelect	Cam table selection	Specifies the cam table for cam synchronous operation.	"P.6-6"
SMC_GetTappetValue	Tappet output	Outputs the tappet set in the cam table.	"P.6-8"

■ Gear operation

Name	Function	Description	Page
MC_GearIn	Start gear operation	Starts gear synchronous operation.	"P.6-11"
MC_GearInPos	Position specified gear operation	Starts gear synchronous operation from the specified absolute position.	"P.6-12"
MC_GearOut	Cancel gear operation	Cancels the gear synchronous operation.	"P.6-14"

■ Phase correction

Name	Function	Description	Page
MC_Phasing	Master axis phase correction	Corrects the phase between the master and slave axes.	"P.6-16"

1.3.4 Motion Control Function Blocks (Multi-axis Control)

■ Interpolation control

Name	Function	Description	Page
PMC_ Interpolator2D	2-axis interpolation control	Specifies the CNC pattern and performs 2-axis interpolation control.	"P.7-2"
PMC_ Interpolator3D	3-axis interpolation control	Specifies the CNC pattern and performs 3-axis interpolation control.	"P.7-3"
PMC_ PositionTracker	Trajectory display data generation	Generates the data for displaying the trajectory of the target axis visually.	"P.7-5"

1.3.5 Motion Control Function Blocks (Motion Communication Control)

■ RTEX

Name	Function	Description	Page
RTEX_ClearAmpAlarm	Clear amplifier alarm	Clears the amplifier's alarm.	"P.8-2"
RTEX_ReadAmpAlarm	Read amplifier alarm	Reads the amplifier's alarm.	"P.8-4"
RTEX_ReadAmpState	Amplifier alarm status	Reads the amplifier's alarm status.	"P.8-5"
RTEX_ReadAmpData	Amplifier monitor	Reads the amplifier's monitor data.	"P.8-6"
RTEX_ReadAmpParameter	Read amplifier parameter	Reads the amplifier's parameters.	"P.8-7"
RTEX_WriteAmpParameter	Write amplifier parameter	Writes the amplifier's parameters.	"P.8-8"
RTEX_ClearAmpMultiTurnData	Clear the multi-turn data	Clears the multi-turn data of the amplifier.	"P.8-9"
RTEX_ClearAmpPositionalDeviation	Clear amplifier deviation counter	Clears the deviation counter of the amplifier.	"P.8-10"
RTEX_ReadNot	Read POT of amplifier	Reads the amplifier's POT status.	"P.8-11"
RTEX_ReadPot	Read NOT of amplifier	Reads the amplifier's NOT status.	"P.8-11"
RTEX_GetTrackingCommandError	Error	Measures the number of sent RTEX commands and the number of lost RTEX commands.	"P.8-12"

1.3.6 Motion Control Function Blocks (Auxiliary Function)

■ Motion auxiliary function (Monitoring)

Name	Function	Description	Page
MC_ReadActualPosition	Read actual position	Reads the actual position data of the axis.	"P.9-2"
MC_ReadActualVelocity	Read actual velocity	Reads the actual velocity of the axis.	"P.9-2"
PMC_ReadActualTorque	Read actual torque	Read the actual torque value of the axis.	"P.9-3"
MC_ReadStatus	Read status	Reads the status information of the axis.	"P.9-4"
SMC_InPosition	In-position judgment	Compares the actual position of the AMP with the command value and judges whether the position is within the specified range.	"P.9-6"
SMC_ReadFBError	Read oldest error	Reads the oldest function block error information.	"P.9-7"
SMC_ClearFBError	Clear oldest error	Clears the oldest FB error information.	"P.9-8"
SMC_CheckAxisCommunication	Check axis communication state	Checks the communication state of the axis.	"P.9-8"
SMC_CheckLimits	Check exceeding limits	Checks whether the velocity, acceleration, or deceleration is in	"P.9-9"

Name	Function	Description	Page
		excess of the dynamic limit set value of the device.	
SMC_GetMaxSetAccDec	Measure maximum acceleration / deceleration	Measures the maximum value of the axis acceleration / deceleration command.	"P.9-10"
SMC_GetMaxSetVelocity	Measure maximum velocity	Measures the maximum value of the axis velocity command.	"P.9-11"
SMC_GetTrackingError	Measure tracking error	Measures the tracking error of the actual position for the axis command position.	"P.9-11"
SMC_MeasureDistance	Measures turnaround travel distance	Measures the travel distance.	"P.9-12"
SMC_ReadSetPosition	Read axis set position	Reads the set command position of the axis.	"P.9-13"

■ Motion auxiliary function (Change / reset)

Name	Function	Description	Page
MC_Reset	Reset axis error	Resets the state transition error of the axis.	"P.9-14"
MC_SetPosition	Change actual position	Changes the actual command position of the axis.	"P.9-14"

■ Motion auxiliary function (Other functions)

Name	Function	Description	Page
PMC_ReadLatchPosition	Monitor AMP latch position	Monitors the AMP latch position.	"P.9-16"
PMC_StopLatchPosition	Stop AMP latch monitoring	Stops monitoring the AMP latch position.	"P.9-18"
MC_DigitalCamSwitch	Enable digital cam switch	Performs ON / OFF control on the digital output according to the axis position.	"P.9-21"
SMC_BacklashCompensation	Compensate backlash	Compensates the backlash.	"P.9-25"

1.3.7 Function Blocks (Others)

■ COM port (General-purpose communication)

The following table lists the function blocks that are used to perform general-purpose communication with the COM port.

Name	Function	Description	Page
COM.Open	Open COM port	Opens the COM port.	"P.10-4"
COM.Close	Close COM port	Closes the COM port.	"P.10-6"
COM.Read	Read COM port	Reads data from the COM port.	"P.10-7"
COM.Write	Write COM port	Writes data to the COM port.	"P.10-8"

Name	Function	Description	Page
COM.ERROR	Error ID	This is an enumeration type error ID that is output when the COM port (general-purpose communication) function block is executed.	"P.10-8"

■ COM port (Modbus COM)

The following table lists the instructions that are used to perform ModbusRTU communication with the COM port.

Name	Function	Description	Page
IoDrvModbusComPort	ModbusComPort device	This is a function block that controls the Modbus_Master_COM_Port device.	"P.10-10"
IoDrvModbus.ModbusChannel	Start sending Modbus command	Sends the command set in the Modbus Slave channel of the ModbusSlaveCOM_Port device.	"P.10-10"
IoDrvModbus.ModbusRequest	Modbus request	Processes the Modbus command specified by I/O without using the ModbusMasterComPort device.	"P.10-11"
IoDrvModbus.ModbusRequest2	Modbus request 2	Like the ModbusRequest, processes the Modbus command specified by I/O without using the ModbusMasterComPort device.	"P.10-12"
IoDrvModbus.ModbusSlaveComPort	ModbusSlaveComPort device	This is a function block that controls the Modbus_Slave_COM_Port device.	"P.10-14"
IoDrvModbus.MB_ErrorCodes	Error code	This is an enumeration type error code that is output when the function block for Modbus communication instruction that uses the COM port is executed.	"P.10-14"

■ LAN port (IoDrvEthernet)

The following table lists the library functions that are used for the network interface to perform communication with the LAN port.

Name	Function	Description	Page
IoDrvEthernet	Ethernet device	This is a function block that acquires the status of the LANPort device.	"P.10-16"
IoDrvEthernet.IPARRAY_TO_INADDR	From array type to union type	This is a function that converts an array type IP address to an INADDR (union type).	"P.10-16"
IoDrvEthernet.IPARRAY_TO_IPSTRING	From array type to character string type	This is a function that converts an array type IP address to a character string type.	"P.10-17"
IoDrvEthernet.IPARRAY_TO_UDINT	From array type to UDINT type	This is a function that converts an array type IP address to a UDINT type.	"P.10-17"
IoDrvEthernet.IPSTRING_TO_UDINT	From character string type to UDINT type	This is a function that converts a character string type IP address to a UDINT type.	"P.10-18"

Name	Function	Description	Page
IoDrvEthernet.UDINT_TO_IPARRAY	From UDINT type to array type	This is a function that converts a UDINT type IP address to an array type.	"P.10-18"
IoDrvEthernet.UDINT_TO_IPSTRING	From UDINT type to character string type	This is a function that converts a UDINT type IP address to an array type.	"P.10-18"

■ LAN port (General-purpose communication)

The following table lists the library functions that are used to perform general-purpose communication with the LAN port using the TCP or UDP protocol.

Name	Function	Description	Page
NBS.TCP_Client	Connect to TCP client	Connects to the TCP/IP client.	"P.10-20"
NBS.TCP_Connection	Connect TCP	Establishes the connection of the client connecting to the connection port opened by TCP_Server.	"P.10-20"
NBS.TCP_Read	Receive TCP data	Acquires data received by the connection port that is established by TCP_Connection.	"P.10-21"
NBS.TCP_Server	Connect TCP server	Opens the specified port as a TCP/IP connection port.	"P.10-22"
NBS.TCP_Write	Send TCP data	Sends data to the connection port that is established by TCP_Connection.	"P.10-23"
NBS.UDP_Peer	Open UDP port	Opens the UDP/IP port.	"P.10-23"
NBS.UDP_Receive	Receive UDP data	Receives data to the connection handle acquired by UDP_Peer.	"P.10-24"
NBS.UDP_Send	Send UDP data	Sends data to the connection handle acquired by UDP_Peer.	"P.10-26"
NBS.ERROR	Error code	This is an enumeration type error code that is output when the function block for communication instruction that uses the LAN port is executed.	"P.10-25"

■ LAN port (Modbus TCP)

The following table lists the library functions that are used to perform ModbusTCP communication with the LAN port.

Name	Function	Description	Page
IoDrvModbusTCP	ModbusTCP device	This is a function block that controls the Modbus_TCP_Master device.	"P.10-28"
IoDrvModbusTCP.ModbusChannel	Start sending Modbus command	Sends the command set in the Modbus Slave channel of the ModbusTCP_Slave device.	"P.10-28"
IoDrvModbusTCP.ModbusRequest	Modbus request	Processes the Modbus command specified by I/O without using the Modbus_TCP_Slave device.	"P.10-29"
IoDrvModbusTCPSlave	ModbusTCPSlave device	This is a function block that controls the Modbus__TCP_Slave device.	"P.10-30"

Name	Function	Description	Page
IoDrvModbus.MB_ErrorCodes	Error code	This is an enumeration type error code that is output when the function block for Modbus communication instruction that uses the LAN port is executed.	"P.10-31"

■ LAN port (EtherNet/IP)

The following table lists instructions that are used to control EtherNet/IP scanner and adapter functions using the GM1 controller.

Name	Function	Overview	Page
IoDrvEtherNetIP	EtherNet/IP scanner device	This is a function block that controls the EtherNet/IP scanner device.	"P.10-33"
RemoteAdapter	Remote adapter device	This is a function block for the Remote adapter device linked to the EtherNet/IP scanner device.	"P.10-34"
IoDrvEtherNetIPAdapter	EtherNet/IP adapter device	This is a function block that controls the EtherNet/IP adapter device.	"P.10-36"
Module	EtherNet/IP module device	This is a function block that controls the EtherNet/IP module device.	"P.10-37"
Apply_Attributes	Apply_Attributes service	This is a function block that calls Apply_Attributes service of the CIP object instance.	"P.10-38"
Generic_Service	Execute generic service	This is a function block that executes generic services with the EtherNet/IP adapter.	"P.10-39"
Get_Attribute_Single	Inquire specific attributes of a specific instance	This is a function block that inquires specific attributes of a specific instance of the CIP object	"P.10-40"
Get_Attributes_All	Inquire all attributes of a specific instance	This is a function block that inquires all attributes of a specific instance of the CIP object	"P.10-41"
Set_Attribute_Single	Set specific attributes of a specific instance	This is a function block that sets specific attributes of a specific instance of the CIP object	"P.10-42"
Set_Attributes_All	Set all attributes of a specific instance	This is a function block that sets all attributes of a specific instance of the CIP object	"P.10-43"
NOP	NOP service	This is a function block that executes the NOP service of a specific instance of the CIP object	"P.10-44"
Reset	Reset service	This is a function block that executes the Reset service of a specific instance of the CIP object	"P.10-45"
Start	Start service	This is a function block that executes the Start service of a specific instance of the CIP object	"P.10-45"
Stop	Stop service	This is a function block that executes the Stop service of a specific instance of the CIP object	"P.10-46"

Name	Function	Overview	Page
ENIP.ERROR	Message service instruction error code		"P.10-47"
ENIP.CIPClass	Service class code		"P.10-49"

■ SD card operation (File operation)

Files in the SD card inserted in the SD memory card slot can be operated.

Name	Function	Description	Page
FILE.Open	Open file	Opens a file or creates a new file.	"P.10-52"
FILE.Close	Close file	Closes a file.	"P.10-53"
FILE.Read	Read file	Reads data from the file opened by the Open instruction.	"P.10-53"
FILE.Write	Write file	Writes data to the file opened by the Open instruction.	"P.10-54"
FILE.Flush	Flush file	Flushes buffer contents to the file opened by the Open instruction.	"P.10-55"
FILE.Copy	Copy file	Copies a file.	"P.10-56"
FILE.Rename	Rename file	Changes a file name.	"P.10-57"
FILE.Delete	Delete file	Deletes a file.	"P.10-58"
FILE.EOF	EOF of file	Determines whether the current offset of a file is EOF (End Of File) or not.	"P.10-58"
FILE.GetAttribute	Get file attribute	Gets file attributes (compressed, hidden, normal, read only).	"P.10-59"
FILE.GetPos	Get file offset	Gets the current offset of a file.	"P.10-60"
FILE.GetSize	Get file size	Gets the file size.	"P.10-60"
FILE.GetTime	Get file update time	Get the update time of a file	"P.10-61"
FILE.SetPos	Set file offset	Sets the offset of a file.	"P.10-62"

■ SD card operation (Directory operation)

Directories in the SD card inserted in the SD memory card slot can be operated.

Name	Function	Description	Page
FILE.DirCreate	Create directory	Creates a directory with a specified name.	"P.10-64"
FILE.DirOpen	Open directory	Opens a directory.	"P.10-64"
FILE.DirClose	Close directory	Closes a directory	"P.10-65"
FILE.DirCopy	Copy directory	Copies a directory.	"P.10-66"
FILE.DirRename	Rename directory	Renames a directory	"P.10-67"
FILE.DirRemove	Delete directory	Deletes a directory.	"P.10-67"
FILE.DirList	Directory list	Outputs a list of directories and files inside the directory.	"P.10-68"

■ Clock setting

The following table lists the function blocks that are used to set the clock of the GM1 Controller.

Name	Function	Description	Page
SYS_GetTime	Get time	This is a function block that gets the current local time	"P.10-70"
SYS_SetTime	Set time	This is a function block that sets the current local time.	"P.10-70"
SYS_GetTimezone	Get time zone information	This is a function block that gets the time zone information.	"P.10-71"
SYS_SetTimezone	Set time zone information	This is a function block that sets the time zone information.	"P.10-71"
SYS_DateConcat	Convert from UINT type to DATE type	This is a function that converts a UINT type date to a DATE type.	"P.10-72"
SYS_DateSplit	Convert from DATE type to UINT type	This is a function that converts a DATE type date to a UINT type.	"P.10-73"
SYS_DTConcat	Convert from UINT type to DT type	This is a function that converts a UINT type date and time to a DT type.	"P.10-73"
SYS_DTSplit	Convert from UINT type to DT type	This is a function that converts a UINT type date and time to a DT type.	"P.10-74"
SYS_GetDayOfWeek	Get day of the week	This is a function that gets the day of the week from the DATE type date.	"P.10-75"
SYS_TODConcat	Convert from UINT type to TOD type	This is a function that converts a UINT type time with milliseconds to a TOD type.	"P.10-75"
SYS_TODSplit	Convert from UINT type to TOD type	This is a function that converts a TOD type time with milliseconds to a UINT type.	"P.10-76"
ERROR	Clock instruction error code		"P.10-77"

■ System data

Name	Function	Description	Page
SYS_GetSystemError	Get system error	Gets the information of a system error that has occurred in the GM1 Controller.	"P.10-78"

1.4 List of Function Block Instructions that Cannot Be Used with the GM1

■ Instructions not available for Modbus

The following function blocks in the IoDrvModbusTCP, IoDrvModbusTCPSlave, IoDrvModbus, and IoDrvModbusSerialSlave libraries are not available for the GM1 Controller.

Name	Function	Alternative function	Page
ModbusTCPSlaveBase	-	-	-
ModbusTCPSlaveUnit	-	-	-
ModbusTCPSlaveUnit_Diag	-	-	-
IoDrvModbusTCP_Diag	-	-	-
ModbusTCPSlave_Diag	-	-	-
ModbusTCPDeviceDiag	-	-	-
IoDrvModbusComPort_Diag	-	-	-
ModbusSlaveComPort_Diag	-	-	-
IoDrvModbusSerialSlave	-	-	-
ModbusSerialDeviceDiag	-	-	-
ModbusServer	-	-	-

■ Instructions not available for general-purpose communication

The following function blocks in the CAA NBS(Net Base Services) library are not available for the GM1 Controller.

Name	Function	Alternative function	Page
TCP_ReadBuffer	-	-	-
TCP_WriteBuffer	-	-	-
UDP_ReceiveBuffer	-	-	-
UDP_SendBuffer	-	-	-
DummyJob	-	-	-

■ Instructions not available for EtherNet/IP

The following function blocks in the IoDrvEtherNetIP and IoDrvEtherNetIPAdapter libraries are not available for the GM1 Controller.

Name	Function	Alternative function	Page
IoDrvEtherNetIP_diag	-	-	-
RemoteAdapter_diag	-	-	-
AdapterDiagnosis	-	-	-
IoDrvEtherNetIPAdapter_Diag	-	-	-
Module_Diag	-	-	-

■ Instructions not available for motion control

The following function blocks in the SM3_Basic library are not available for the GM1 Controller. Alternative functions are listed, if available.

Name	Function	Alternative function	Page
MC_TouchProbe	Enable external latch	PMC_ReadLatchPosition	"P.9-16"
MC_AbortTrigger	Disable external latch	MC_StopLatchPosition	"P.9-18"
SMC_Commissioning	Commissioning status	Commissioning function of the GM Programmer	-
SMC_SetCustomRampType	Set acceleration / deceleration custom operation	-	-
SMC_CAM_ObjectManager	Manage cam data	-	-
SMC3_CommunicateDriveParameter	Communication setting	RTEX_ReadAmpParameter	"P.8-7"
SMC3_ReadDriveParameter	Read drive parameter	RTEX_ReadAmpParameter	"P.8-7"
SMC3_ReadParameter	Read parameter	RTEX_ReadAmpParameter	"P.8-7"
SMC3_WriteDriveParameter	Write drive parameter	RTEX_WriteAmpParameter	"P.8-8"
SMC3_WriteParameter	Write parameter	RTEX_WriteAmpParameter	"P.8-8"
MC_ReadBoolParameter	Read BOOL-type parameter	RTEX_ReadAmpParameter	"P.8-7"
MC_ReadParameter	Read parameter	RTEX_ReadAmpParameter	"P.8-7"
MC_WriteBoolParameter	Write BOOL-type parameter	RTEX_WriteAmpParameter	"P.8-8"
MC_WriteParameter	Write parameter	RTEX_WriteAmpParameter	"P.8-8"
MC_ReadActualTorque	Read actual torque	PMC_ReadActualTorque	"P.9-3"
SMC_SetTorque	Set torque	PMC_SetTorque	"P.5-30"
MC_Home	Home return	PMC_Home	"P.5-4"
SMC_VIRTUAL_AXIS	Set virtual axis	-	-
SMC3_BrakeStatus	Get brake status	-	-
SMC3_BrakeControl	Brake control	-	-
SMC3_PersistPosition	Persist actual axis position	-	-
SMC3_PersistPositionLogical	Persist logical axis position	-	-
SMC3_PersistPositionSingleturn	Persist actual axis position with a range	-	-
SMC_PerfStat	Calculate performance statistics	-	-
SMC_SeriesStat	Calculate increment statistics	-	-
SMC_AxisDiagnosticLog	Log axis parameter	-	-
SMC3_ReinitDrive		-	-
FB_Template_Edge		-	-
FB_Template_EdgeAbort		-	-
FB_Template_EdgeAbortTimeout		-	-
SMC_StartupDrive		-	-

Name	Function	Alternative function	Page
SMC_ChangeDynamicLimits		-	-
SMC_ChangeGearingRatio		-	-
SMC_SetMovementType		-	-
SMC_SetRampType		-	-
SMC_SetSoftwareLimits		-	-
SMC_CAMBounds		-	-
SMC_CAMBounds_Pos		-	-
SMC_CamEditor		-	-
SMC_CamRegister		-	-
SMC_GetCamSlaveSetPositio n		-	-
SMC_ReadCAM		-	-
SMC_WriteCAM		-	-
MC_ReadAxisError		-	-
SMC_PerfTimerSum		-	-
SMC_FollowPosition		-	-
SMC_FollowPositionVelocity		-	-
SMC_FollowSetValues		-	-
SMC_FollowVelocity		-	-
SMC_Homing		-	-

2 Ladder Instructions

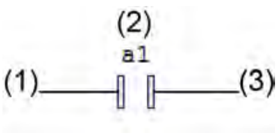
2.1 Ladder Instructions

This section describes ladder instructions that can be used for ladder diagram program (LD program).

2.1.1 NO Contact

If the variable corresponding to the contact is TRUE, then the input value is output. If the variable is FALSE, then FALSE is output.

■ Icon



■ Parameter

No.	Scope	Type	Description
(1)	Input	BOOL	Input to the NO contact
(2)	Variable name	BOOL	Variable that corresponds to the NO contact
(3)	Output	BOOL	Output from the NO contact

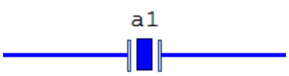
■ Input method

Use one of the following methods to input the NO contact.

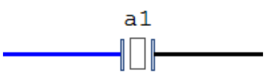
- From the tool box, select **Ladder elements**> **NO contact** and drag to "Start from here".
- Right-click on the network, and, from the displayed menu, select **Insert Contact**.
- Click the icon on the tool bar.
- From the menu, select **FBD / LD / IL**>**Insert Contact**.
- Press the shortcut keys <Ctrl+k> simultaneously.

■ Program example

If the variable (a1) corresponding to the NO contact is TRUE, then the value input to the NO contact (TRUE) is output as is.



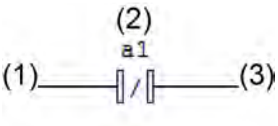
If the variable (a1) corresponding to the contact is FALSE, then FALSE is output.



2.1.2 NC Contact

If the variable corresponding to the contact is TRUE, then FALSE is output. If the variable is FALSE, then the input value is output.

■ Icon



■ Parameter

No.	Scope	Type	Description
(1)	Input	BOOL	Input to the NC contact
(2)	Variable name	BOOL	Variable that corresponds to the NC contact
(3)	Output	BOOL	Output from the NC contact

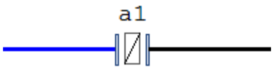
■ Input method

Use one of the following methods to input the NC contact.

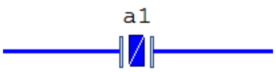
- From the tool box, select **Ladder elements**> **NC contact** and drag to "Start from here".
- Right-click on the network, and, from the displayed menu, select "Insert NC contact".
- Click the Icon on the tool bar.
- From the menu, select **FBD / LD / IL**>**Insert NC contact**.

■ Program example

If the variable (a1) corresponding to the NC contact is TRUE, then FALSE is output.



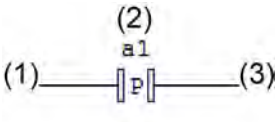
If the variable (a1) corresponding to the NC contact is FALSE, then the value input to the NC contact (TRUE) is output as is.



2.1.3 Rising Edge Detection Contact

If a rising edge is detected in the variable corresponding to the contact, then the input value is output for one cycle only.

■ Icon



■ Parameter

No.	Scope	Type	Description
(1)	Input	BOOL	Input to the contact
(2)	Variable name	BOOL	Variable that corresponds to the rising edge detection contact
(3)	Output	BOOL	Output from the contact

■ Input method

Input the rising edge detection contact by first inputting the NO contact and then changing the NO contact.

Select the NO contact just input and then perform one of the following operations.

- Right-click and, from the displayed menu, select **Edge detection** .
- From the menu, select **FBD / LD / IL**> **Edge detection** .
- Press the shortcut keys <Ctrl+e> simultaneously.
- Click the on the tool bar.

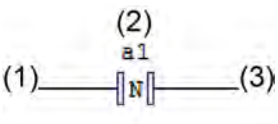
■ Program example

The following program is designed to detect the rising edge with the variable (a1) corresponding to the rising edge detection contact and to output TRUE for one cycle only.

2.1.4 Falling Edge Detection Contact

If a falling edge is detected in the variable corresponding to the contact, then the input value is output for one cycle only.

■ Icon



■ Parameter

No.	Scope	Type	Description
(1)	Input	BOOL	Input to the contact

No.	Scope	Type	Description
(2)	Variable name	BOOL	Variable that corresponds to the falling edge detection contact
(3)	Output	BOOL	Output from the contact

■ Input method

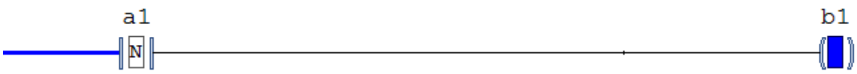
Input the falling edge detection contact by first inputting the NO contact and then changing the NO contact.

Select the NO contact just input and then perform one of the following operations.

- Right-click and, from the displayed menu, select **Edge detection** twice.
- From the menu, select **FBD / LD / IL**> **Edge detection** twice.
- Press the shortcut keys Ctrl+e simultaneously twice.
- Click the on the tool bar twice.

■ Program example

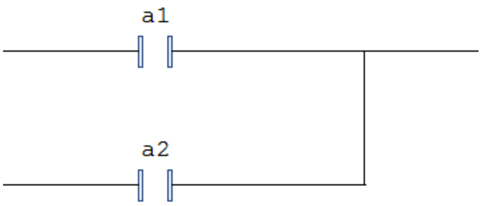
The following program is designed to detect the falling edge with the variable (a1) corresponding to the falling edge detection contact and to output TRUE for one cycle only.



2.1.5 Parallel NO Contact

NO contacts can be input in parallel to the initial contact. Of the contacts wired in parallel, if the output of one or more contacts is TRUE, TRUE is output.

■ Icon



■ Input method

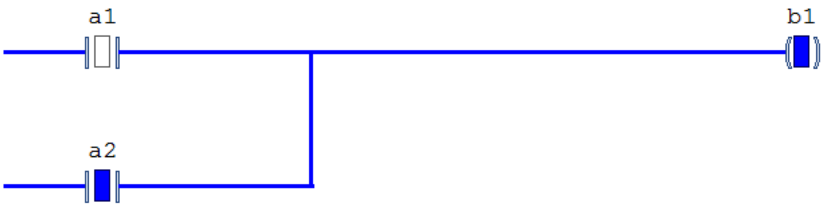
To input a parallel NO contact below the initial contact, select **Ladder elements** >**Parallel NO contact** from the tool box and drag to the position indicated with "▼" next to the contact.

Or, with the contact selected, perform one of the following operations.

- Right-click, and, from the displayed menu, select **Insert contact in parallel (below)**.
- From the menu, select **FBD/LD/IL**>**Insert contact in parallel (below)**.
- Press the shortcut keys <Ctrl+r> simultaneously.
- Click the on the tool bar.

■ Program example

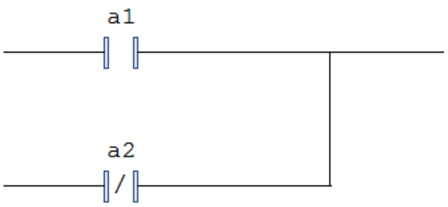
This program is designed to input one NO contact in parallel to the NO contact. TRUE is output because the NO contact below is TRUE.



2.1.6 Parallel NC Contact

NC contacts can be input in parallel to the initial contact. Of the contacts wired in parallel, if the output of one or more contacts is TRUE, TRUE is output.

■ Icon



■ Input method

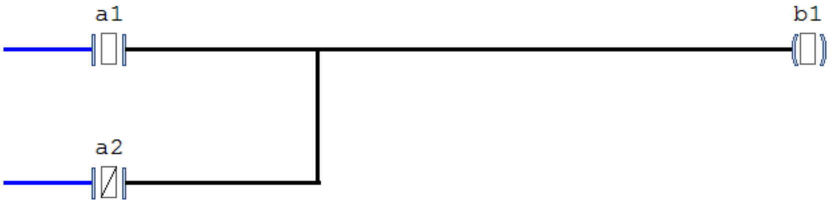
To input a parallel NC contact below the initial contact, select **Ladder elements** >**Parallel NC contact** from the tool box and drag to the position indicated with "▼" next to the contact.

Or, with the contact selected, perform one of the following operations.

- Right-click, and, from the displayed menu, select "Insert NC contact in parallel (below)".
- From the menu, select **FBD / LD / IL**>**Insert NC contact in parallel (below)**.
- Click the on the tool bar.

■ Program example

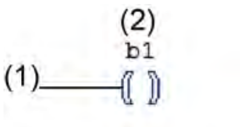
This program is designed to input one NC contact in parallel to the NO contact. FALSE is output because the outputs of both contacts are FALSE.



2.1.7 Coil

The input value is saved in the variable corresponding to the coil. If the input value is TRUE, then TRUE is saved. If the input value is FALSE, then FALSE is saved.

■ Icon



■ Parameter

No.	Scope	Type	Description
(1)	Input	BOOL	Input to the coil
(2)	Variable name	BOOL	Name of the variable that corresponds to the coil

■ Input method

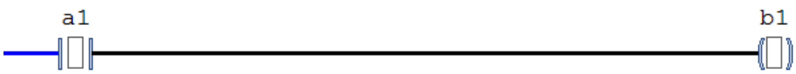
Use one of the following methods to input the coil.

- From the tool box, select **Ladder elements**> **Coil** and drag to "Add output or jump here" (when connecting to a contact).
- Right-click on the network, and, from the displayed menu, select **Insert Coil** .
- Click the icon on the tool bar.
- From the menu, select **FBD / LD / IL**>**Insert Coil** .
- Press the shortcut keys <Ctrl+a> simultaneously.

■ Program example

This program is designed to input the output from the NO contact to the coil.
TRUE is saved in the variable (b1) because the input to the coil is TRUE.

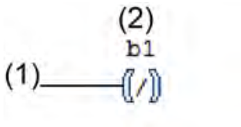
FALSE is saved in the variable (b1) because the input to the coil is FALSE.



2.1.8 Negated Coil

The negated value of the input is saved in the variable corresponding to the coil. If the input value is TRUE, then FALSE is saved. If the input value is FALSE, then TRUE is saved.

■ Icon



■ Parameter

No.	Scope	Type	Description
(1)	Input	BOOL	Input to the negated coil
(2)	Variable name	BOOL	Name of the variable that corresponds to the negated coil

■ Input method

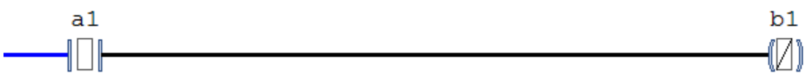
The negated coil can be input by inputting a coil and changing it.
With the input coil selected, perform one of the following operations.

- Right-click and, from the displayed menu, select **Negation**.
- From the menu, select **FBD / LD / IL**>**Negation**.
- Press the shortcut keys <Ctrl+n> simultaneously.
- Click the on the tool bar.

■ Program example

This program is designed to input the output from the NO contact to the negated coil.
FALSE is saved in the variable (b1) because the input to the coil is TRUE.

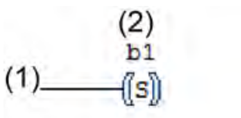
TRUE is saved in the variable (b1) because the input to the coil is FALSE.



2.1.9 Set Coil

When the input value turns TRUE, TRUE is saved in the variable corresponding to the coil.
TRUE is held until the input to the reset coil that corresponds to the same variable turns TRUE.

■ Icon



■ Parameter

No.	Scope	Type	Description
(1)	Input	BOOL	Input to the set coil.
(2)	Variable name	BOOL	Name of the variable that corresponds to the set coil

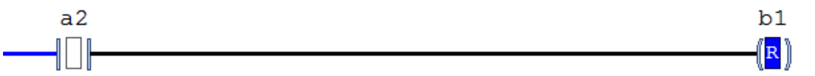
■ Input method

Use one of the following methods to input the set coil.

- From the tool box, select **Ladder elements**>**Set coil** and drag to "Add output or jump here" (when connecting to the contact).
- Right-click on the network, and, from the displayed menu, select "Insert Set Coil ".
- Click the icon on the tool bar.
- From the menu, select **FBD / LD / IL**>**Insert Set Coil** .

■ Program example

This program is designed to input the output from the NO contact to the set coil and the reset coil.
TRUE is saved in the set coil variable (b1) because the input to the set coil is TRUE.



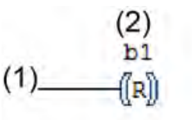
Info.

- Each set coil should be accompanied by a reset coil.

2.1.10 Reset Coil

When the input value turns TRUE, FALSE is saved in the variable corresponding to the coil. FALSE is held until the input to set coil that corresponds to the same variable turns TRUE.

■ Icon



■ Parameter

No.	Scope	Type	Description
(1)	Input	BOOL	Input to the reset coil.
(2)	Variable name	BOOL	Name of the variable that corresponds to the reset coil

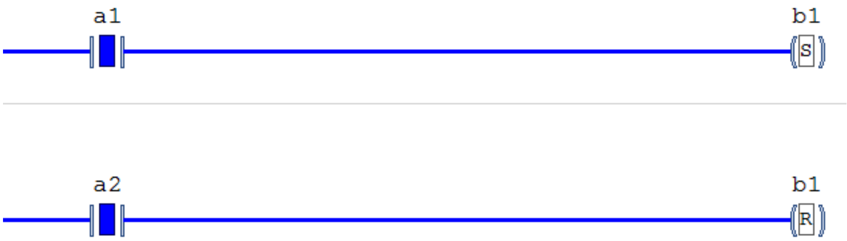
■ Input method

Use one of the following methods to input the reset coil.

- From the tool box, select **Ladder elements**>**Reset Coil** and drag to "Add output or jump here" (when connecting to a contact).
- Right-click on the network, and, from the displayed menu, select **Reset Coil**.
- Click the icon on the tool bar.
- From the menu, select **FBD / LD / IL**>**Insert reset Reset Coil** .

■ Program example

This program is designed to input the output from the NO contact to the set coil and the reset coil.
FALSE is saved in the variable (b1) because the input to the reset coil is TRUE.



Info.

- Each set coil should be accompanied by a reset coil.

3 Functions

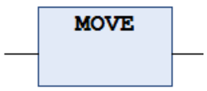
3.1 Substitution Instruction

Use the substitution instruction to substitute a variable with a value of another variable.

3.1.1 MOVE (Substitution)

This is a function that substitutes the value of a variable specified in the input for a variable specified in the output.

■ Icon



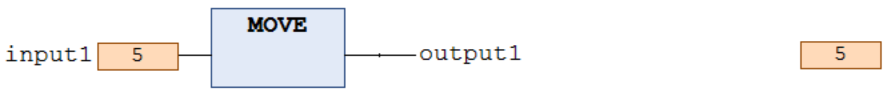
■ Parameter

Scope	Type	Description
Input	All	Specifies the variable of the substitution source.
Output	All	Specifies the variable of the substitution target.

■ Program example

This program is designed to substitute the value of input variable “input1” for the output variable “output1”.

LD program



ST program

```
output1 5 := MOVE(input1 5);
```

It is also possible to substitute the value using an operator (:=).

```
output1 5 := input1 5;
```

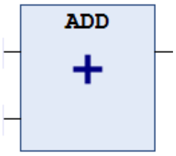
3.2 Arithmetic Operation Instructions

Arithmetic operation instructions can be used to perform calculation such as four arithmetic operations.

3.2.1 ADD (Addition)

This is a function that adds input arguments and outputs the sum.

■ Icon



■ Parameter

Scope	Type	Description
Input	(Note 1)	Specifies the variables to be added.
Output	(Note 1)	Outputs the sum of variables specified in the input.

(Note 1) Usable data type

BYTE, WORD, DWORD, LWORD, SINT, USINT, INT, UINT, DINT, UDINT, LINT, ULINT, REAL, LREAL, TIME, TIME_OF_DAY, DATE_AND_TIME

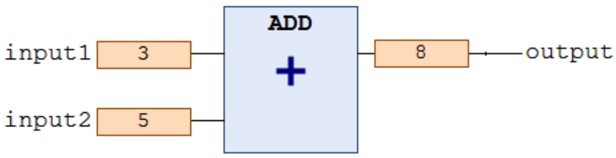
Time type data can be added in the following combinations.

- TIME + TIME = TIME
- TIME_OF_DAY + TIME = TIME_OF_DAY
- DATE_AND_TIME + TIME = DATE_AND_TIME

■ Program example

This program is designed to output the sum of input variables “input1” and “input2” to the output variable “output”.

LD program



ST program

It is possible to add the values using “+” operator.

```
output 8 := input1 3 + input2 5;
```

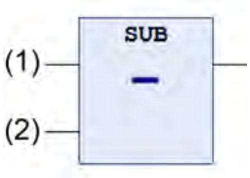
Info.

- If you want to increase input arguments in the LD program, right-click on the ADD function, and, on the displayed menu, select "Add Input".

3.2.2 SUB (Subtraction)

This is a function that subtracts input arguments and outputs the difference.

■ Icon



■ Parameter

Scope	Number	Type	Description
Input	(1), (2)	(Note 1)	Specifies the variables to be subtracted.
Output	-	(Note 1)	Outputs the value obtained by subtracting the input (2) from the input (1).

(Note 1) Usable data types

BYTE, WORD, DWORD, LWORD, SINT, USINT, INT, UINT, DINT, UDINT, LINT, ULINT, REAL, LREAL, TIME, TIME_OF_DAY, DATE, DATE_AND_TIME

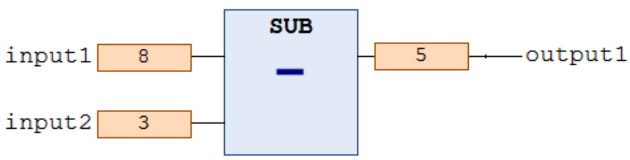
For time type data, subtraction can be performed in the following combinations. Note that negative time cannot be calculated.

- TIME - TIME = TIME
- DATE - DATE = TIME
- TOD - TIME = TOD
- TOD - TOD = TIME
- DT - TIME = DT
- DT - DT = TIME

■ Program example

This program is designed to output the difference between the input variables “input1” and “input2” to the output variable “output1”.

LD program



ST program

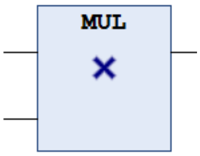
It is possible to subtract the values using “-” operator.

```
output1 5 := input1 8 - input2 3;
```

3.2.3 MUL (Multiplication)

This is a function that multiplies input arguments and outputs the product.

■ Icon



■ Parameter

Scope	Type	Description
Input	(Note 1)	Specifies the variables to be multiplied.
Output	(Note 1)	Outputs the product of variables specified in the input.

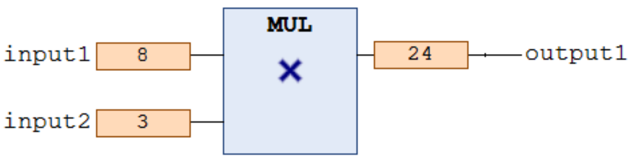
(Note 1) Usable data type

BYTE, WORD, DWORD, LWORD, SINT, USINT, INT, UINT, DINT, UDINT, LINT, ULINT, REAL, LREAL, TIME

■ Program example

This program is designed to output the product of the input variables “input1” and “input2” to the output variable “output1”.

LD program



ST program

It is possible to multiply the values using “*” operator.

```
output1 24 := input1 8 * input2 3;
```

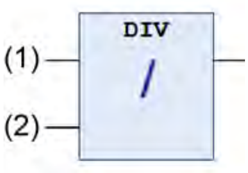
Info.

- If you want to increase input arguments in the LD program, right-click on the MUL function, and, on the displayed menu, select "Add Input".
- TIME type data cannot be multiplied by REAL type, LREAL type, or TIME type data.

3.2.4 DIV (Division)

This is a function that divides input arguments and outputs the quotient.

■ Icon



■ Parameter

Scope	No.	Type	Description
Input	(1), (2)	(Note 1)	Specifies the variables to be divided.
Output	-	(Note 1)	Outputs the quotient obtained by dividing the input (2) by the input (1).

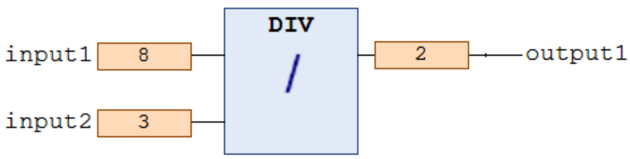
(Note 1) Usable data types

BYTE, WORD, DWORD, LWORD, SINT, USINT, INT, UINT, DINT, UDINT, LINT, ULINT, REAL, LREAL, TIME

■ Program example

This program is designed to output the quotient of the INT type input variables “input1” and “input2” to the INT type output variable “output1”.

LD program



ST program

It is possible to divide the values using the division operator ("/") .

```
output1 2 := input1 8 / input2 3;
```

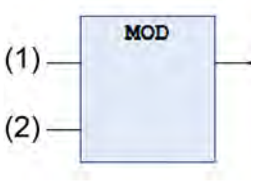
Info.

- TIME type variables can be divided by integer type variables.
- When a variable is divided by a DINT, LINT, REAL, or LREAL type variable, it can be checked if 0 is used in the calculation. (Refer to “Auto Check POU” in the “SMC Tool Introduction Guide”.)

3.2.5 MOD (Remainder)

This is a function that divides input arguments and outputs the remainder.

■ Icon



■ Parameter

Scope	No.	Type	Description
Input	(1), (2)	(Note 1)	Specifies the variables to be divided.
Output	-	(Note 1)	Outputs the remainder obtained by dividing the input (2) by the input (1).

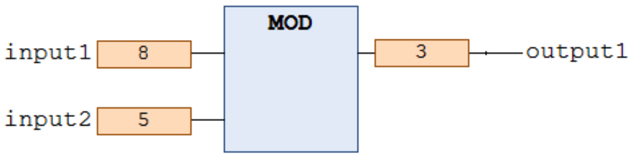
(Note 1) Usable data type

BYTE, WORD, DWORD, LWORD, SINT, USINT, INT, UINT, DINT, UDINT, LINT, ULINT

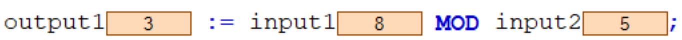
■ Program example

This program is designed to output the remainder obtained from dividing the INT type input variables “input1” and “input2” to the INT type output variable “output1”.

LD program



ST program



3.3 Boolean Operation Instructions

Boolean operation instructions can be used to perform bool operations such as logical AND or logical OR.

3.3.1 AND (Logical AND)

This is a function that outputs logical AND of the input arguments.

■ Icon

■ Parameter

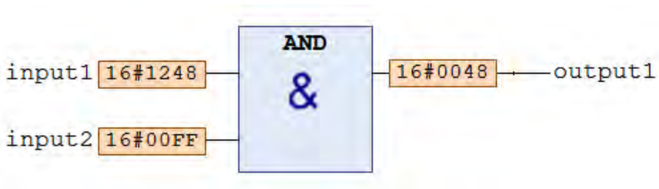
Scope	Type	Description
Input	(Note 1)	Specifies the variables to be used to obtain logical AND.
Output	(Note 1)	Outputs the logical AND of the variables specified in the input.

(Note 1) Usable data type
BOOL, BYTE, WORD, DWORD, LWORD

■ Program example

This program is designed to output the logical AND of the WORD type input variables “input1” and “input2” to the output variable “output1”.
The execution result is displayed in a hexadecimal number.

LD program



ST program

```
output1 16#0048 := input1 16#1248 AND input2 16#00FF ;
```

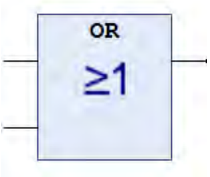
Info.

- If you want to increase input arguments in the LD program, right-click on the AND function, and, on the displayed menu, select "Add Input".

3.3.2 OR (Logical OR)

This is a function that outputs logical OR of the input arguments.

■ Icon



■ Parameter

Scope	Type	Description
Input	(Note 1)	Specifies the variables to be used to obtain logical OR.
Output	(Note 1)	Outputs the logical OR of the variables specified in the input.

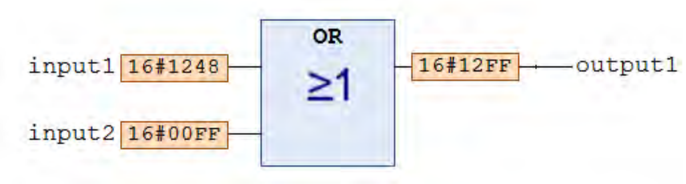
(Note 1) Usable data type

BOOL, BYTE, WORD, DWORD, LWORD

■ Program example

This program is designed to output the logical OR of the WORD type input variables “input1” and “input2” to the output variable “output1”.
The execution result is displayed in a hexadecimal number.

LD program



ST program

```
output1 16#12FF := input1 16#1248 OR input2 16#00FF ;
```

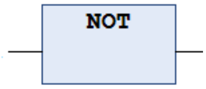
Info.

- If you want to increase input arguments in the LD program, right-click on the OR function, and, on the displayed menu, select "Add Input".

3.3.3 NOT (Negation)

This is a function that outputs the negation of the input argument.

■ Icon



■ Parameter

Scope	Type	Description
Input	(Note 1)	Specifies the variable to be used to obtain the negation.
Output	(Note 1)	Outputs the negation of the variable specified in the input.

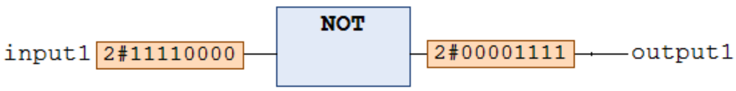
(Note 1) Usable data type

BOOL, BYTE, WORD, DWORD, LWORD

■ Program example

This program is designed to output the negation of the BYTE type input variable “input1” to the output variable “output1”.
The execution result is displayed in a binary number.

LD program



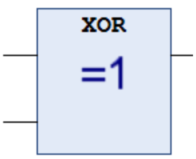
ST program

```
output1 2#00001111 := NOT input1 2#11110000 ;
```

3.3.4 XOR (Exclusive OR)

This is a function that outputs exclusive OR of the input arguments.

■ Icon



■ Parameter

Scope	Type	Description
Input	(Note 1)	Specifies the variables to be used to obtain exclusive OR.
Output	(Note 1)	Outputs the exclusive OR of the variables specified in the input.

Scope	Type	Description
		Outputs 0 if both input bits are 1 or 0. Outputs 1 if one of the two input bits is 1 and the other bit is 0.

(Note 1) Usable data type

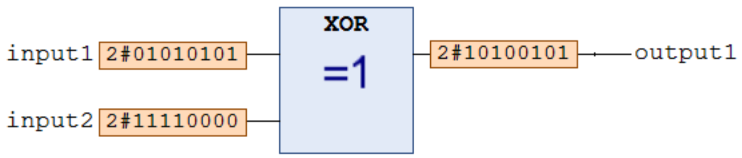
BOOL, BYTE, WORD, DWORD, LWORD

■ Program example

This program is designed to output the exclusive OR of the BYTE type input variables “input1” and “input2” to the output variable “output1”.

The execution result is displayed in a binary number.

LD program



ST program

```
output1 2#10100101 := input1 2#01010101 XOR input2 2#11110000;
```

3.3.5 AND_THEN (Logical AND)

This is a conditional AND evaluation function of the input operand.

■ Usable data types

BOOL, BIT

■ Program example

This program is designed to compare the value of the variable accessed by pwAddress (pointer) with wExpected if the pwAddress is not NULL and, if they are the same, substitute with the value of wNewValue.

As default values, “5” is stored in the variable “test1” accessed by pwAddress, “5” in wExpected, and “3” in wNewValue.

As an initial step, judgment is made whether pwAddress is NULL or not. Since it is not NULL, comparison is made between the value of “test1” and the value of wExpected as the next step. Since these two values are both “5”, TRUE is assigned. As a result, the value of wNewValue “3” is stored in the “test1” and the xFlag flag is set to TRUE.

ST program

[Declaration section]

```
VAR
     pwAddress    : POINTER TO WORD;
     wExpected    : WORD := 5;
     wNewValue    : WORD := 3;
     xFlag        : BOOL;
     test1        : WORD := 5;
END_VAR
```

[Implementation section]

```
pwAddress 16#F1D10BBE := ADR(test1 16#0003 );

IF pwAddress 16#F1D10BBE <> 0 AND_THEN pwAddress^ 16#0003 = wExpected 16#0005 THEN
    pwAddress^ 16#0003 := wNewValue 16#0003 ;
    xFlag TRUE := TRUE;
ELSE
    xFlag TRUE := FALSE;
END_IF
```

Info.

- Expressions of other operands are executed only when the first operand is TRUE.
  Therefore, if no value is stored in pwAddress in the above example, the initial NULL judgment turns FALSE. As a result, no judgment is performed on operands after the AND_THEN operator.

3.3.6 OR_ELSE (Logical OR)

This is a conditional OR evaluation function of the input operand.

■ Usable data types

BOOL, BIT

■ Program example

16#000000FF is stored in the variable dw.

"dw.8" that represents bit 8 of dw is FALSE and "dw.1" that represents bit 1 is TRUE.

Therefore, the operation result flag bX is TRUE.

Note that the third input expression is not executed and bEver remains FALSE.

ST program

[Declaration section]

```
VAR
     bEver    : BOOL;
     bX       : BOOL;
     dw       : DWORD := 16#000000FF;
END_VAR
```

[Implementation section]

```
bEver FALSE := FALSE;
bX TRUE := dw 16#000000FF .8 FALSE OR_ELSE dw 16#000000FF .1 TRUE OR_ELSE dw 16#000000FF .1 TRUE OR_ELSE (bEver FALSE := TRUE);
```

Info.

- In case of OR_ELSE, when one of the operands is evaluated TRUE, all other operator expressions are not evaluated.

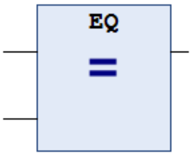
3.4 Comparison Operation Instructions

Comparison operation instructions can be used to compare two arguments.

3.4.1 EQ (“Equal” Comparison)

This is a function that compares two input arguments and determines if they are the same value.

■ Icon



■ Parameter

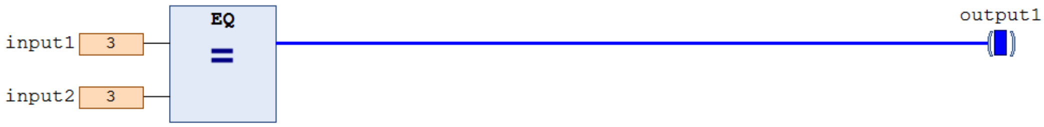
Scope	Type	Description
Input	All	Specifies the variables to be compared.
Output	BOOL	Outputs TRUE if the input variable values are the same. Outputs FALSE if they are different.

■ Program example

This program is designed to compare the input variables “input1” and “input2” and output the result to the output variable “output1”.

LD program

TRUE is output because the input variable values “input1” and “input2” are the same.



ST program

Use the operator (=) to compare the values.

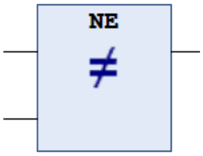
FALSE is output because the input variable values “input1” and “input2” are different.

```
output1 FALSE := (input1 3 = input2 5);
```

3.4.2 NE (“Not Equal” Comparison)

This is a function that compares two input arguments and determines if they are not the same.

■ Icon



■ Parameter

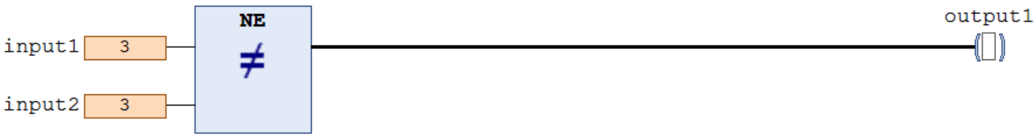
Scope	Type	Description
Input	All	Specifies the variables to be compared.
Output	BOOL	Outputs TRUE if the input variable values are different. Outputs FALSE if they are the same.

■ Program example

This program is designed to compare the input variables “input1” and “input2” and output the result to the output variable “output1”.

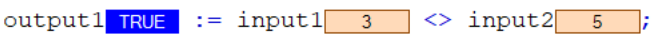
LD program

FALSE is output because the input variable values “input1” and “input2” are the same.



ST program

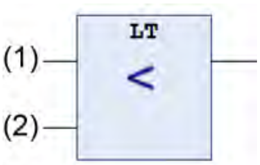
Use the operator (<>) to compare the values.
TRUE is output because the input variable values “input1” and “input2” are different.



3.4.3 LT (“Less Than” Comparison)

This is a function that compares two input arguments and determines if the first argument is less than the second argument.

■ Icon



■ Parameter

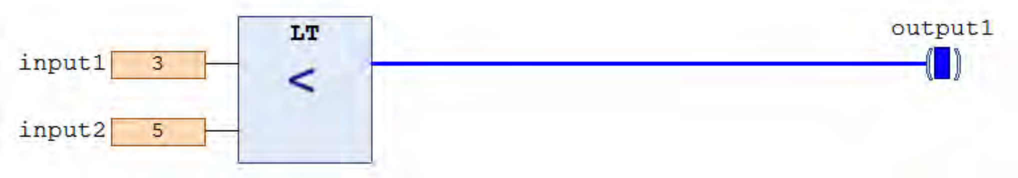
Scope	No.	Type	Description
Input	(1), (2)	All	Specifies the variables to be compared.
Output	-	BOOL	Outputs TRUE if the value of input (1) is less than the value of input (2). Otherwise, outputs FALSE.

■ Program example

This program is designed to compare the input variables “input1” and “input2” and output the result to the output variable “output1”.

LD program

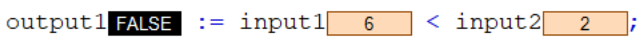
TRUE is output because the input variable “input1” is less than the input variable “input2”.



ST program

Use the operator (<) to compare the values.

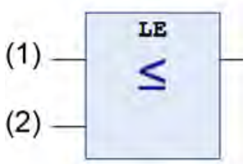
FALSE is output because the input variable “input1” is not less than the input variable “input2”.



3.4.4 LE (“Less Than or Equal” Comparison)

This is a function that compares two input arguments and determines if the first argument is less than or equal to the second argument.

■ Icon



■ Parameter

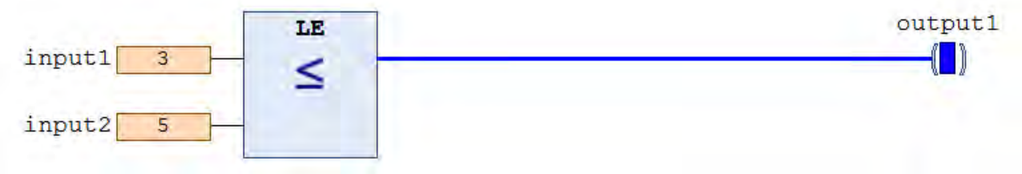
Scope	No.	Type	Description
Input	(1), (2)	All	Specifies the variables to be compared.
Output	-	BOOL	Outputs TRUE if the value of input (1) is less than or equal to the value of input (2). Otherwise, outputs FALSE.

■ Program example

This program is designed to compare the input variables “input1” and “input2” and output the result to the output variable “output1”.

LD program

TRUE is output because the input variable “input1” is less than or equal to the input variable “input2”.



ST program

Use the operator (<=) to compare the values.

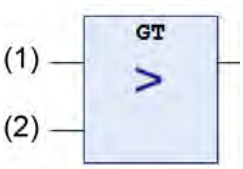
FALSE is output because the input variable “input1” is not less than or equal to the input variable “input2”.

```
output1 FALSE := input1 6 <= input2 2 ;
```

3.4.5 GT (“Greater Than” Comparison)

This is a function that compares two input arguments and determines if the first argument is greater than the second argument.

■ Icon



■ Parameter

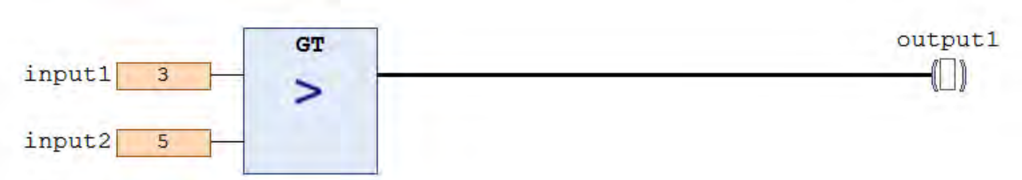
Scope	No.	Type	Description
Input	(1), (2)	All	Specifies the variables to be compared.
Output	-	BOOL	Outputs TRUE if the value of input (1) is greater than the value of input (2). Otherwise, outputs FALSE.

■ Program example

This program is designed to compare the input variables “input1” and “input2” and output the result to the output variable “output1”.

LD program

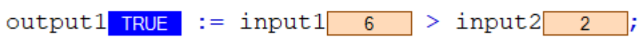
FALSE is output because the input variable “input1” is not greater than the input variable “input2”.



ST program

Use the operator (>) to compare the values.

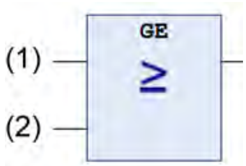
TRUE is output because the input variable “input1” is greater than the input variable “input2”.



3.4.6 GE (“Greater Than Or Equal” Comparison)

This is a function that compares two input arguments and determines if the first argument is greater than or equal to the second argument.

■ Icon



■ Parameter

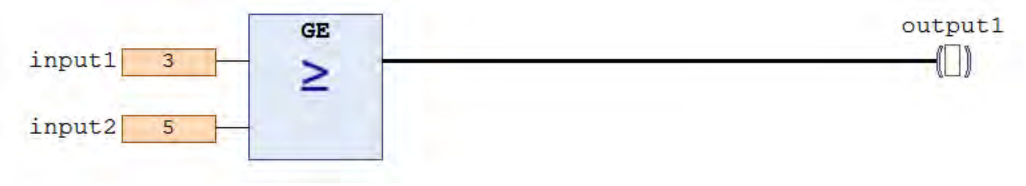
Scope	No.	Type	Description
Input	(1), (2)	All	Specifies the variables to be compared.
Output	-	BOOL	Outputs TRUE if the value of input (1) is greater than or equal to the value of input (2). Otherwise, outputs FALSE.

■ Program example

This program is designed to compare the input variables “input1” and “input2” and output the result to the output variable “output1”.

LD program

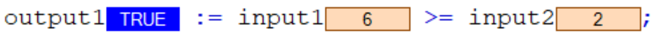
FALSE is output because the input variable “input1” is not greater than or equal to the input variable “input2”.



ST program

Use the operator (>=) to compare the values.

TRUE is output because the input variable “input1” is greater than or equal to the input variable “input2”.



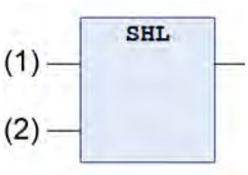
3.5 Bit Shift Instructions

Bit shift instructions can be used to perform bit shift operation on input arguments.

3.5.1 SHL (Shift Left)

This is a function that shifts the input argument to the left by the specified number of bits and outputs the shifted value. “0” is inserted from the least significant bit up to the bit position shifted by the shift quantity.

■ Icon



■ Parameter

Scope	No.	Type	Description
Input	(1)	(Note 1)	Specifies the variable on which bit shift is performed.
	(2)	(Note 1)	Specifies the number of times bit shift is performed (shift quantity).
Output	-	(Note 1)	Outputs the value bit shifted to the left from the value of input (1) by the quantity specified in the input (2).

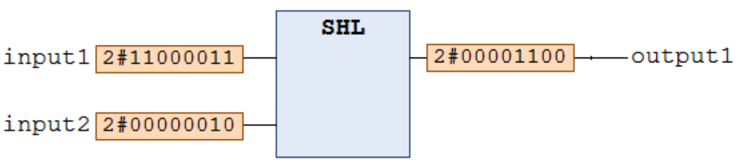
(Note 1) Usable data type

BYTE, WORD, DWORD, LWORD, SINT, USINT, INT, UINT, DINT, UDINT, LINT, ULINT

■ Program example

This program is designed to output the value that is shifted to the left from the value (2#11000011) of input variable “input1” by the number of bits (2 bits) specified in “input2” to the output variable “output1”.

LD program



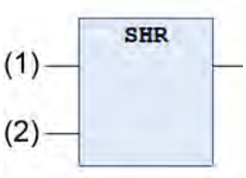
ST program

```
output1 2#00001100 := SHL(input1 2#11000011, input2 2#00000010);
```

3.5.2 SHR (Shift Right)

This is a function that shifts the input argument to the right by the specified number of bits and outputs the shifted value. “0” is inserted from the most significant bit up to the bit position shifted by the shift quantity.

■ Icon



■ Parameter

Scope	No.	Type	Description
Input	(1)	(Note 1)	Specifies the variable on which bit shift is performed.
	(2)	(Note 1)	Specifies the number of times bit shift is performed (shift quantity).
Output	-	(Note 1)	Outputs the value bit shifted to the right from the value of input (1) by the quantity specified in the input (2).

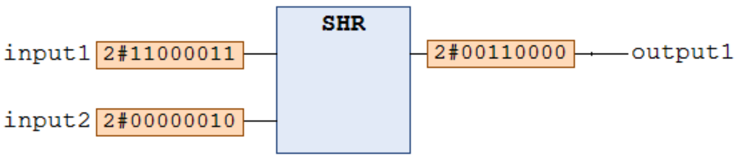
(Note 1) Usable data type

BYTE, WORD, DWORD, LWORD, SINT, USINT, INT, UINT, DINT, UDINT, LINT, ULINT

■ Program example

This program is designed to output the value that is shifted to the right from the value (2#11000011) of input variable “input1” by the number of bits (2 bits) specified in “input2” to the output variable “output1”.

LD program



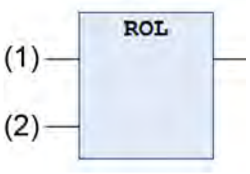
ST program

```
output1 2#00110000 := SHR(input1 2#11000011, input2 2#00000010);
```

3.5.3 ROL (Rotate Left)

This is a function that shifts the input argument to the left by the specified number of bits and outputs the shifted value. The bit value that has overflowed the most significant bit when the bit is shifted is inserted into the data starting from the least significant bit up to the bit position shifted by the shift quantity.

■ Icon



■ Parameter

Scope	No.	Type	Description
Input	(1)	(Note 1)	Specifies the variable on which bit shift is performed.
	(2)	(Note 1)	Specifies the number of times bit shift is performed (shift quantity).
Output	-	(Note 1)	Outputs the value rotated and shifted to the left from the value of input (1) by the quantity specified in the input (2).

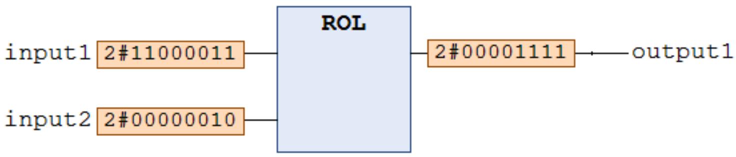
(Note 1) Usable data type

BYTE, WORD, DWORD, LWORD, SINT, USINT, INT, UINT, DINT, UDINT, LINT, ULINT

■ Program example

This program is designed to output the value that is rotated and shifted to the left from the value (2#11000011) of input variable “input1” by the number of bits (2 bits) specified in “input2” to the output variable “output1”.

LD program



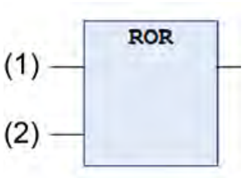
ST program

```
output1 2#00001111 := ROL(input1 2#11000011 , input2 2#00000010 );
```

3.5.4 ROR (Rotate Right)

This is a function that shifts the input argument to the right by the specified number of bits and outputs the shifted value. The bit value that has overflowed the least significant bit when the bit is shifted is inserted into the data starting from the most significant bit up to the bit position shifted by the shift quantity.

Icon



Parameter

Scope	No.	Type	Description
Input	(1)	(Note 1)	Specifies the variable on which bit shift is performed.
	(2)	(Note 1)	Specifies the number of times bit shift is performed (shift quantity).
Output	-	(Note 1)	Outputs the value rotated and shifted to the right from the value of input (1) by the quantity specified in the input (2).

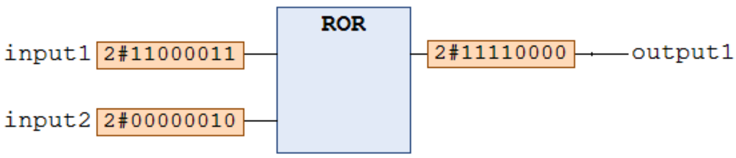
(Note 1) Usable data type

BYTE, WORD, DWORD, LWORD, SINT, USINT, INT, UINT, DINT, UDINT, LINT, ULINT

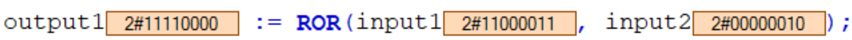
Program example

This program is designed to output the value that is rotated and shifted to the right from the value (2#11000011) of input variable “input1” by the number of bits (2 bits) specified in “input2” to the output variable “output1”.

LD program



ST program



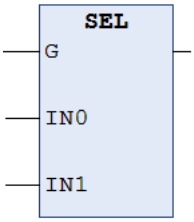
3.6 Selection Instructions

Selection instructions can be used to select arguments from multiple input arguments according to specified conditions and to output them.

3.6.1 SEL (Binary Selector)

This is a function that outputs the value of the input argument IN0 or IN1 depending on whether the input argument G is true or false.

■ Icon



■ Parameter

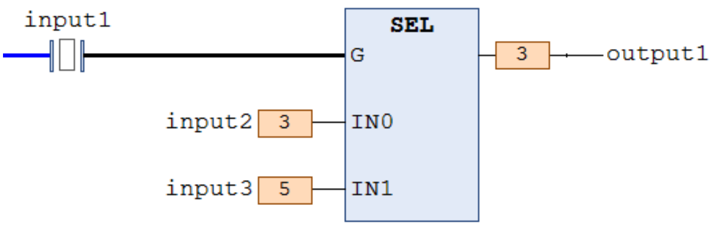
Scope	Name	Type	Description
Input	G	BOOL	Conditions for selecting the contents to be output
	IN0	All	Specifies the variable to be output if G is FALSE.
	IN1	All	Specifies the variable to be output if G is TRUE.
Output	-	All	Outputs the value of IN0 or IN1 depending on the value of G.

■ Program example

This program is designed to output the value of the input variable “input2” or “input3” to the output variable “output1” depending on the value of the input variable “input1”.

LD program

This program is designed to output the value of “input2” (IN0) because the value of “input1” is FALSE.



ST program

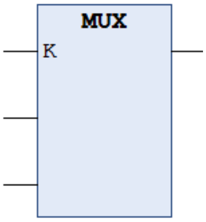
This program is designed to output the value of “input3” (IN1) to the "output1" because the value of “input1” is TRUE.

```
output1 5 := SEL(input1 TRUE ,input2 3 ,input3 5 );
```

3.6.2 MUX (Multiplexer)

This is a function that selectively outputs the input arguments depending on the value of the input argument K.

■ Icon



■ Parameter

Scope	Name	Type	Description
Input	K	(Note 1)	Specifies the value (K = 0, 1, 2...) to select the value to output.
	-	All	Specifies the value to be output depending on K.
Output	-	All	Outputs one of the input arguments depending on the value of K.

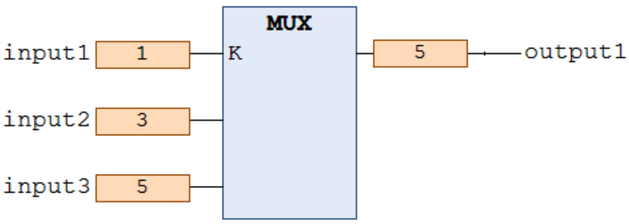
(Note 1) Usable data type
BYTE, WORD, DWORD, LWORD, SINT, USINT, INT, UINT, DINT, UDINT, LINT, ULINT

■ Program example

This program is designed to output the value of the input variable “input2” or “input3” to the output variable “output1” depending on the value of the input variable “input1”.

LD program

This program is designed to output the value of “input3” to “output1” depending on the value (1) of “input1”.



ST program

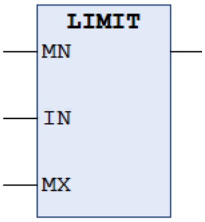
This program is designed to output the value of “input 2” to the “output 1” depending on the value (0) of “input 1”.

```
output1 3 := MUX(input1 0, input2 3, input3 5);
```

3.6.3 LIMIT (Limiter)

This is a function that limits the input value with the lower and upper limit values and outputs a restricted value.

■ Icon



■ Parameter

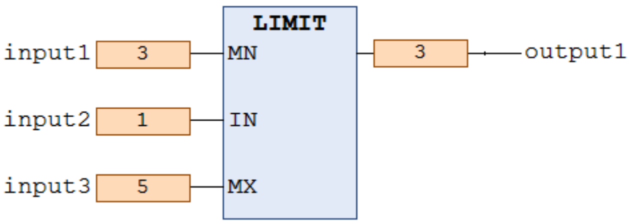
Scope	Name	Type	Description
Input	MN	All	Specifies the lower limit of the value to be output.
	IN	All	Specifies the input values to be restricted.
	MX	All	Specifies the upper limit of the value to be output.
Output	-	All	Outputs values according to the following conditions. IN ≤ MN: Outputs "MN". MN ≤ IN ≤ MX: Outputs "IN". MX ≤ IN: Outputs "MX".

■ Program example

This program is designed to limit the value range of the input variable “input2” with the input variable “input1” (lower limit) and the input variable “input3” (upper limit) and to output the limited value to the output variable “output1”.

LD program

This program is designed to output “3” to “output1” because the value (1) of “input2” (IN) is less than or equal to the lower limit (3) specified in “input1” (MN).



ST program

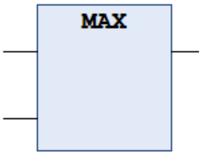
This program is designed to output “5” to “output1” because the value (8) of “input2” is greater than or equal to the upper limit (5) specified in “input3”.

```
output1 5 := LIMIT(input1 3 , input2 8 , input3 5 );
```

3.6.4 MAX (Maximum Value)

This is a function that outputs the maximum value of the input arguments.

■ Icon



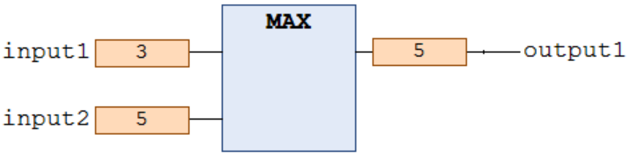
■ Parameter

Scope	Type	Description
Input	All	Specifies the values from which to obtain the maximum value.
Output	All	Outputs the maximum value of the input values.

■ Program example

This program is designed to output the maximum value of the input variables to the output variable “output1”.

LD program



ST program

```
output1 5 := MAX(input1 3, input2 5);
```

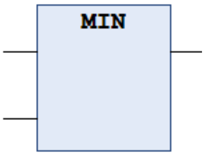
Info.

- If you want to increase input arguments in the LD program, right-click on the MAX function, and, on the displayed menu, select "Add Input".

3.6.5 MIN (Minimum Value)

This is a function that outputs the minimum value of the input arguments.

■ Icon



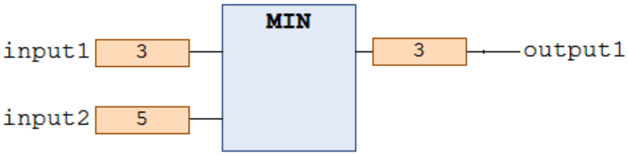
■ Parameter

Scope	Type	Description
Input	All	Specifies the values from which to obtain the minimum value.
Output	All	Outputs the minimum value of the input values.

■ Program example

This program is designed to output the minimum value of the input variables to the output variable “output1”.

LD program



ST program

```
output1 3 := MIN(input1 3 , input2 5 );
```

Info.

- If you want to increase input arguments in the LD program, right-click on the MIN function, and, on the displayed menu, select "Add Input".

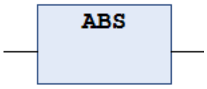
3.7 Numerical Operation Instructions

Numerical operation instructions can be used to perform various numerical calculations.

3.7.1 ABS (Absolute Value)

This is a function that outputs the absolute value of the input argument.

■ Icon



■ Parameter

Scope	Type	Description
Input	(Note 1)	Specifies the value from which to obtain the absolute value.
Output	(Note 1)	Outputs the absolute value of the input argument.

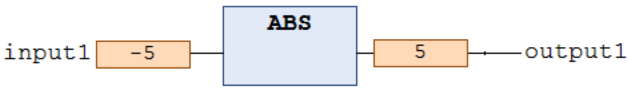
(Note 1) Usable data type

BYTE, WORD, DWORD, LWORD, SINT, USINT, INT, UINT, DINT, UDINT, LINT, ULINT, REAL, LREAL

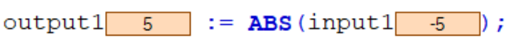
■ Program example

This program is designed to output the absolute value of the input variable “input1” to the output variable “output1”.

LD program



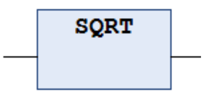
ST program



3.7.2 SQRT (Square Root)

This is a function that outputs the square root (√) of the input argument.

■ Icon



■ Parameter

Scope	Type	Description
Input	(Note 1)	Specifies the value from which to obtain the square root.
Output	(Note 2)	Outputs the square root of the input argument.

(Note 1) Usable data type

BYTE, WORD, DWORD, LWORD, SINT, USINT, INT, UINT, DINT, UDINT, LINT, ULINT, REAL, LREAL

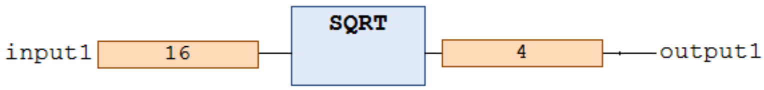
(Note 2) Usable data type

REAL (if the input is REAL), LREAL

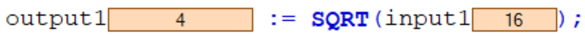
■ Program example

This program is designed to output the square root of the input variable “input1” to the output variable “output1”.

LD program



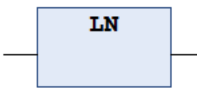
ST program



3.7.3 LN (Natural Logarithm)

This is a function that outputs the natural logarithm ($\log_e X$) of the input argument (X).

■ Icon



■ Parameter

Scope	Type	Description
Input	(Note 1)	Specifies the value from which to obtain the natural logarithm.
Output	(Note 2)	Outputs the natural logarithm of the input argument.

(Note 1) Usable data type

BYTE, WORD, DWORD, LWORD, SINT, USINT, INT, UINT, DINT, UDINT, LINT, ULINT, REAL, LREAL

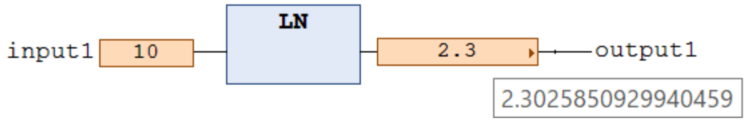
(Note 2) Usable data type

REAL (if the input is REAL), LREAL

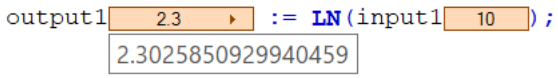
■ Program example

This program is designed to output the natural logarithm ($\log_e 10$) of the input variable “input1” (10) to the output variable “output1”.

LD program



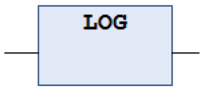
ST program



3.7.4 LOG (Common Logarithm)

This is a function that outputs the common logarithm ($\log_{10}X$) of the input argument (X).

■ Icon



■ Parameter

Scope	Type	Description
Input	(Note 1)	Specifies the value from which to obtain the common logarithm.
Output	(Note 2)	Outputs the common logarithm of the input argument.

(Note 1) Usable data type

BYTE, WORD, DWORD, LWORD, SINT, USINT, INT, UINT, DINT, UDINT, LINT, ULINT, REAL, LREAL

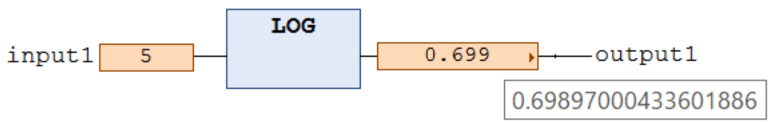
(Note 2) Usable data type

REAL (if the input is REAL), LREAL

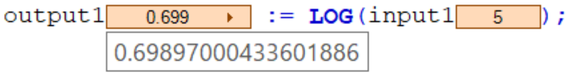
■ Program example

This program is designed to output the common logarithm ($\log_{10}5$) of the input variable “input1” (5) to the output variable “output1”.

LD program



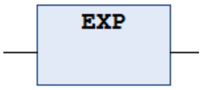
ST program



3.7.5 EXP (Natural Exponent)

This is a function that outputs the natural exponent (e^X) of the input argument (X).

■ Icon



■ Parameter

Scope	Type	Description
Input	(Note 1)	Specifies the value from which to obtain the natural exponent.
Output	(Note 2)	Outputs the natural exponent of the input argument.

(Note 1) Usable data type

BYTE, WORD, DWORD, LWORD, SINT, USINT, INT, UINT, DINT, UDINT, LINT, ULINT, REAL, LREAL

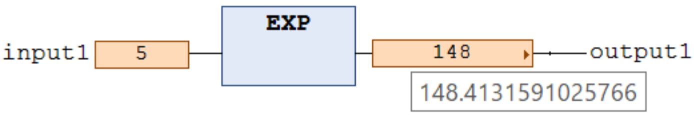
(Note 2) Usable data type

REAL (if the input is REAL), LREAL

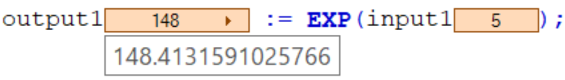
■ Program example

This program is designed to output the natural exponent of the input variable “input1” to the output variable “output1”.

LD program



ST program



3.7.6 EXPT (Exponentiation)

This is a function that outputs the exponentiation (a^n) of the input arguments (a, n).

■ Icon

■ Parameter

Scope	No.	Type	Description
Input	(1)	(Note 1)	Inputs the base of exponentiation.
	(2)	(Note 1)	Inputs the exponent of exponentiation.
Output	(3)	(Note 2)	Outputs the exponentiation obtained from the input arguments. Outputs a^n in the following case. Input (1): a Input (2): n

(Note 1) Usable data type

BYTE, WORD, DWORD, LWORD, SINT, USINT, INT, UINT, DINT, UDINT, LINT, ULINT, REAL, LREAL

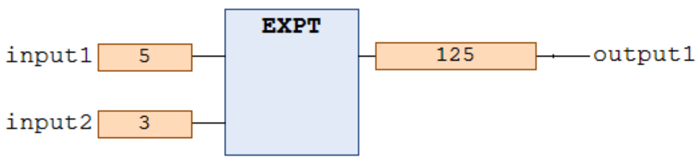
(Note 2) Usable data type

REAL (if the input is REAL), LREAL

■ Program example

This program is designed to output the exponentiation (5^3 = 125) obtained from the input variables “input1” and “input2” to the output variable “output1”.

LD program



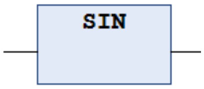
ST program

```
output1 125 := EXPT(input1 5 , input2 3 );
```

3.7.7 SIN (Trigonometric Function Sine)

This is a function that outputs the value of the trigonometric function sine. The unit of the input argument is radian.

■ Icon



■ Parameter

Scope	Type	Description
Input	(Note 1)	Specifies the value (unit: radian) from which to obtain the trigonometric function sine.
Output	(Note 2)	Outputs the value of sine of the input argument.

(Note 1) Usable data type

BYTE, WORD, DWORD, LWORD, SINT, USINT, INT, UINT, DINT, UDINT, LINT, ULINT, REAL, LREAL

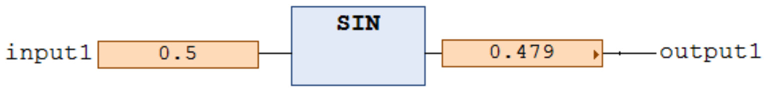
(Note 2) Usable data type

REAL (if the input is REAL), LREAL

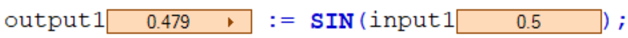
■ Program example

This program is designed to output the value of the trigonometric function sine obtained from the input variable “input1” to the output variable ”output1”.

LD program



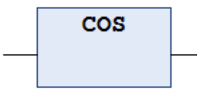
ST program



3.7.8 COS (Trigonometric Function Cosine)

This is a function that outputs the value of the trigonometric function cosine. The unit of the input argument is radian.

■ Icon



■ Parameter

Scope	Type	Description
Input	(Note 1)	Specifies the value (unit: radian) from which to obtain the trigonometric function cosine.
Output	(Note 2)	Outputs the value of cosine of the input argument.

(Note 1) Usable data type

BYTE, WORD, DWORD, LWORD, SINT, USINT, INT, UINT, DINT, UDINT, LINT, ULINT, REAL, LREAL

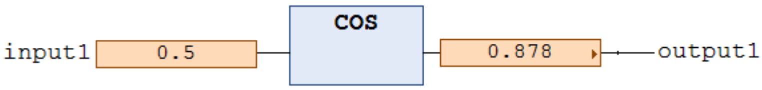
(Note 2) Usable data type

REAL (if the input is REAL), LREAL

■ Program example

This program is designed to output the value of the trigonometric function cosine obtained from the input variable “input1” to the output variable ”output1”.

LD program

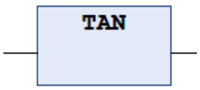


ST program

3.7.9 TAN (Trigonometric Function Tangent)

This is a function that outputs the value of the trigonometric function tangent. The unit of the input argument is radian.

■ Icon



■ Parameter

Scope	Type	Description
Input	(Note 1)	Specifies the value (unit: radian) from which to obtain the trigonometric function tangent.
Output	(Note 2)	Outputs the value of tangent of the input argument.

(Note 1) Usable data type

BYTE, WORD, DWORD, LWORD, SINT, USINT, INT, UINT, DINT, UDINT, LINT, ULINT, REAL, LREAL

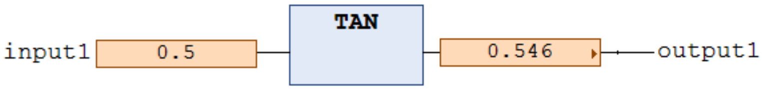
(Note 2) Usable data type

REAL (if the input is REAL), LREAL

■ Program example

This program is designed to output the value of the trigonometric function tangent obtained from the input variable “input1” to the output variable ”output1”.

LD program

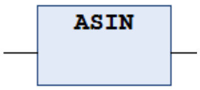


ST program

3.7.10 ASIN (Trigonometric Function Arc Sine)

This is a function that outputs the value of the trigonometric function arc sine. The unit of the input argument is radian.

■ Icon



■ Parameter

Scope	Type	Description
Input	(Note 1)	Specifies the value (unit: radian) from which to obtain the trigonometric function arc sine.
Output	(Note 2)	Outputs the value of arc sine of the input argument.

(Note 1) Usable data type

BYTE, WORD, DWORD, LWORD, SINT, USINT, INT, UINT, DINT, UDINT, LINT, ULINT, REAL, LREAL

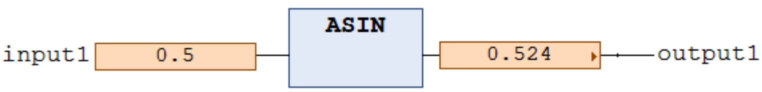
(Note 2) Usable data type

REAL (if the input is REAL), LREAL

■ Program example

This program is designed to output the value of the trigonometric function arc sine obtained from the input variable “input1” to the output variable ”output1”.

LD program

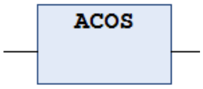


ST program

3.7.11 ACOS (Trigonometric Function Arc Cosine)

This is a function that outputs the value of the trigonometric function arc cosine. The unit of the input argument is radian.

■ Icon



■ Parameter

Scope	Type	Description
Input	(Note 1)	Specifies the value (unit: radian) from which to obtain the trigonometric function arc cosine.
Output	(Note 2)	Outputs the value of arc cosine of the input argument.

(Note 1) Usable data type

BYTE, WORD, DWORD, LWORD, SINT, USINT, INT, UINT, DINT, UDINT, LINT, ULINT, REAL, LREAL

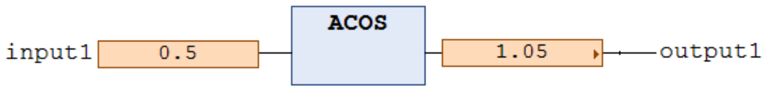
(Note 2) Usable data type

REAL (if the input is REAL), LREAL

■ Program example

This program is designed to output the value of the trigonometric function arc cosine obtained from the input variable “input1” to the output variable ”output1”.

LD program

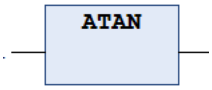


ST program

3.7.12 ATAN (Trigonometric Function Arc Tangent)

This is a function that outputs the value of the trigonometric function arc tangent. The unit of the input argument is radian.

■ Icon



■ Parameter

Scope	Type	Description
Input	(Note 1)	Specifies the value (unit: radian) from which to obtain the trigonometric function arc tangent.
Output	(Note 2)	Outputs the value of arc tangent of the input argument.

(Note 1) Usable data type

BYTE, WORD, DWORD, LWORD, SINT, USINT, INT, UINT, DINT, UDINT, LINT, ULINT, REAL, LREAL

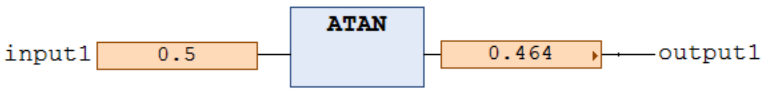
(Note 2) Usable data type

REAL (if the input is REAL), LREAL

■ Program example

This program is designed to output the value of the trigonometric function arc tangent obtained from the input variable “input1” to the output variable ”output1”.

LD program



ST program

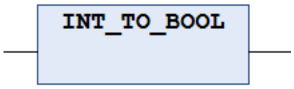
3.8 Data Type Conversion Instructions

Data type conversion instructions can be used to convert the data type of a variable.

3.8.1 "Type 1"_TO_"Type 2" ("Type 1">"Type 2" Conversion)

This is a function that converts the data type of the input argument "Type 1" to another data type "Type 2". Conversion from a larger size data type to a smaller size data type is not performed automatically. It is necessary to convert the data type using this instruction.

■ Icon



■ Parameter

Scope	Type	Description
Input	All	Specifies the variable required to be converted to a different type.
Output	All	Outputs the variable converted to a different type.

■ Types of functions and conversion examples

Types of functions	Conversion example		Description
	Input	Output	
INT_TO_BOOL	5	TRUE	If other than 0, outputs TRUE.
UINT_TO_USINT	300 (16#012C)	44 (16#2C)	Outputs lower eight bits out of the 16 bits of UINT.
REAL_TO_INT	3.5	4	Outputs data after rounding decimals to the nearest whole number.
TIME_TO_STRING	t#2s5ms	'T#2s5ms'	Outputs the TIME constant in the STRING type.
TOD_TO_UDINT	tod#00:30:20	182000	Outputs the milliseconds from 00:00:00.
DATE_TO_UDINT	d#2000-1-1	946684800	Outputs the seconds from 1970-1-1.
STRING_TO_WORD	'123'	123	Outputs the character string of a numerical value.

■ Program example

LD program

The following is a program example of the INT_TO_BOOL function.

ST program

The following is a program example of the TOD_TO_UDINT function.

```
output1 1820000 := TOD_TO_UDINT(input1 TOD#0:30:20);
```

3.8.2 TRUNC (Real Number to DINT Conversion)

This is a function that converts a real number type input to a DINT type.

■ Icon

■ Parameter

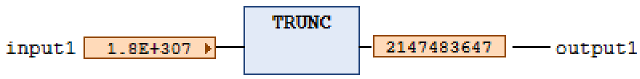
Scope	Type	Description
Input	REAL, LREAL	Real number type value
Output	DINT	Outputs the value converted to the DINT type from the input argument.

■ Program example

This program is designed to convert the LREAL type input variable “input1” to the DINT type output variable “output1” and output the converted data.

Input1 := 1.7976931348623157E+307;

LD program



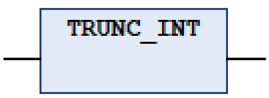
ST program

```
output1 2147483647 := TRUNC(input1 1.8E+307);
```

3.8.3 TRUNC_INT (Real Number to INT Conversion)

This is a function that converts a real number type input to an INT type.

■ Icon



■ Parameter

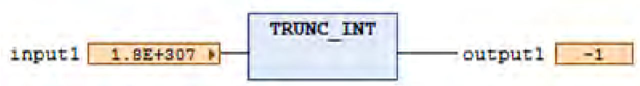
Scope	Type	Description
Input	REAL, LREAL	Real number type value
Output	INT	Outputs the value converted to the INT type from the input argument.

■ Program example

This program is designed to convert the LREAL type input variable “input1” to the INT type output variable “output1” and output the converted data.

Input1 := 1.7976931348623157E+307;

LD program



ST program

```
output1 -1 := TRUNC_INT(input1 1.8E+307 );
```

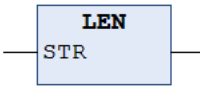
3.9 Character string instructions

Character string instructions can be used to perform various operations on character strings.

3.9.1 LEN (Length of a Character String)

This is a function that outputs the length of a character string.

■ Icon



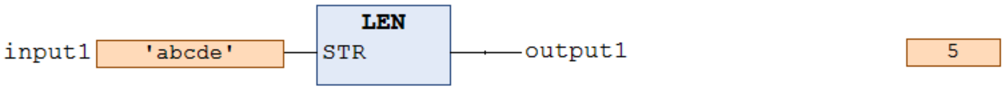
■ Parameter

Scope	Name	Type	Description
Input	STR	STRING	Specifies the character string from which to obtain the length.
Output	-	INT	Outputs the character string length of the input argument.

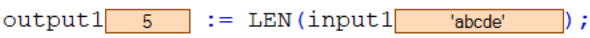
■ Program example

This program is designed to output the character string length of the input variable “input1” to the output variable ”output1”.

LD program



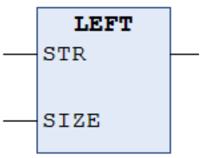
ST program



3.9.2 LEFT (Extracting Characters from the Left End)

This is a function that extracts a character string consisting of the specified number of characters from the left end of the character string and outputs the extracted data.

■ Icon



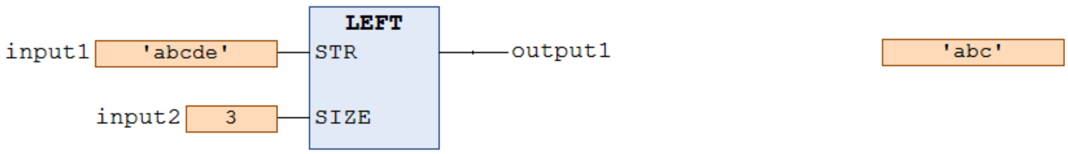
Parameter

Scope	Name	Type	Description
Input	STR	STRING	Specifies the character string from which a character string is to be extracted.
	SIZE	INT	Specifies the number of characters to be extracted from the left.
Output	-	STRING	Extracts a character string consisting of the number of characters specified in SIZE from STR and outputs the extracted data.

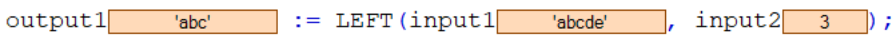
Program example

This program is designed to extract a character string consisting of the number of characters (3 characters) specified in “input2” from the left end of the character string of the input variable “input1” and to output the extracted character string to the output variable ”output1”.

LD program



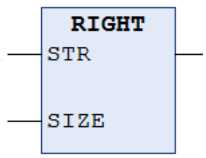
ST program



3.9.3 RIGHT (Extracting Characters from the Right End)

This is a function that extracts a character string consisting of the specified number of characters from the right end of the character string and outputs the extracted data.

Icon



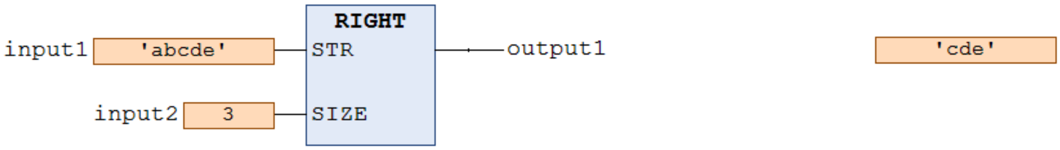
Parameter

Scope	Name	Type	Description
Input	STR	STRING	Specifies the character string from which a character string is to be extracted.
	SIZE	INT	Specifies the number of characters to be extracted from the right.
Output	-	STRING	Extracts a character string consisting of the number of characters specified in SIZE from STR and outputs the extracted data.

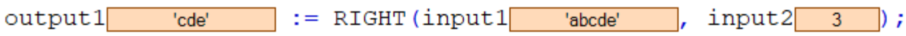
■ Program example

This program is designed to extract a character string consisting of the number of characters (3 characters) specified in “input2” from the right end of the character string of the input variable “input1” and to output the extracted character string to the output variable ”output1”.

LD program



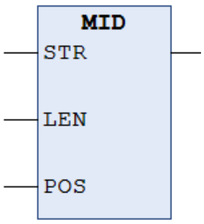
ST program



3.9.4 MID (Extracting Characters from the Specified Position)

This is a function that extracts a character string consisting of the specified number of characters from the specified position of the character string and outputs the extracted data.

■ Icon



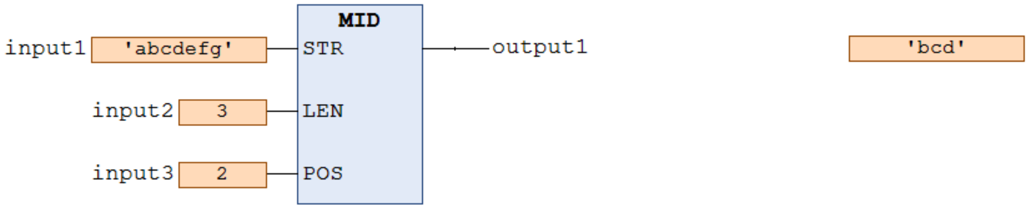
■ Parameter

Scope	Name	Type	Description
Input	STR	STRING	Specifies the character string from which a character string is to be extracted.
	LEN	INT	Specifies the number of characters to be extracted.
	POS	INT	Specified the position from which extraction is to be started.
Output	-	STRING	Extracts a character string consisting of the number of characters specified in LEN from STR starting from the position specified in POS and outputs the extracted data.

■ Program example

This program is designed to extract a character string consisting of the number of characters (3 characters) specified in “input2” from the character string of the input variable “input1”, starting from the position (2nd character from the left end) specified in “input3”, and to output the extracted data to the output variable ”output1”.

LD program



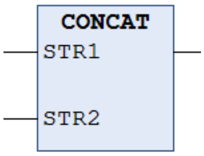
ST program

```
output1 'bcd' := MID(input1 'abcdefg', input2 3, input3 2);
```

3.9.5 CONCAT (Concatenating Character Strings)

This is a function that concatenates the character strings.

■ Icon



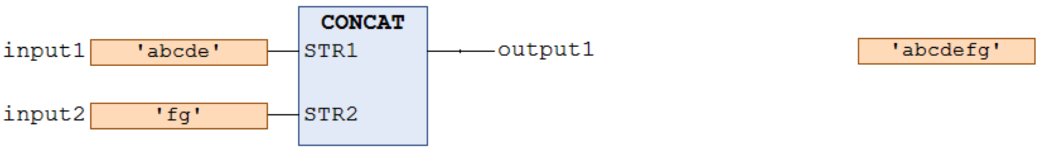
■ Parameter

Scope	Name	Type	Description
Input	STR1	STRING	Specifies the character string to be concatenated.
	STR2	STRING	Specifies the character string to be concatenated.
Output	-	STRING	Concatenate the STR2 character string to the right of the STR1 character string and output the concatenated data.

■ Program example

This program is designed to concatenate the character string of “input2” to the character string of the input variable “input1” and to output the concatenated data to the output variable ”output1”.

LD program



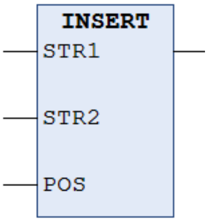
ST program

```
output1 'abcdefg' := CONCAT(input1 'abcde', input2 'fg');
```

3.9.6 INSERT (Inserting a Character String)

This is a function that inserts a character string in the specified position and outputs the inserted data.

■ Icon



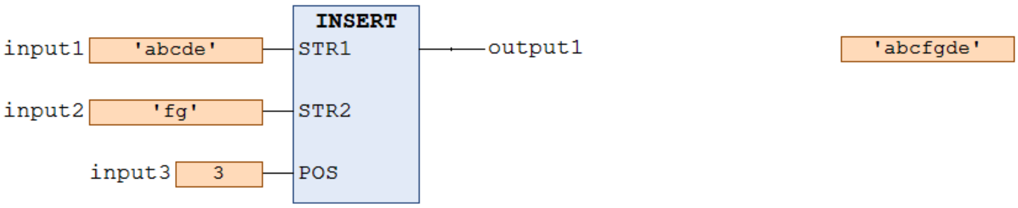
■ Parameter

Scope	Name	Type	Description
Input	STR1	STRING	Specifies the character string in which a character string is to be inserted.
	STR2	STRING	Specifies the character string to be inserted.
	POS	INT	Specifies the position to be inserted. n-th character from the left
Output	-	STRING	Inserts the STR2 character string in the position specified in POS in the STR1 character string.

■ Program example

This program is designed to insert the character string of “input2” in the position (3rd character from the left end) specified in “input3” from the left of the the character string of the input variable “input1” and to output the inserted data to the output variable ”output1”.

LD program



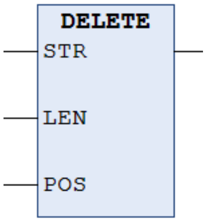
ST program

```
output1 'abcfgde' := INSERT(input1 'abcde', input2 'fg', input3 3);
```

3.9.7 DELETE (Deleting a Character String)

This is a function that deletes a character string from the specified position and outputs the deleted data.

■ Icon



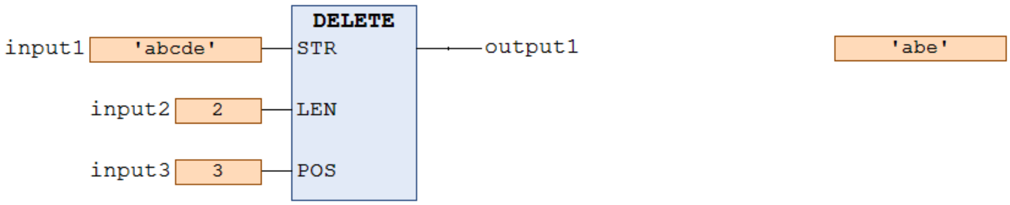
■ Parameter

Scope	Name	Type	Description
Input	STR	STRING	Specifies the character string from which a character string is to be deleted.
	LEN	INT	Specifies the length of the character string to be deleted.
	POS	INT	Specified the position from which deletion is to be started. n-th character from the left
Output	-	STRING	Deletes a character string consisting of the number of characters specified in LEN from the left end of the STR character string starting from the position specified in POS and outputs the deleted data.

■ Program example

This program is designed to delete a character string consisting of the number of characters (2 characters) specified in “input2” from the character string of the input variable “input1” starting from the position (3rd character from the left) specified in “input3” and to output the deleted data to the output variable “output1”.

LD program



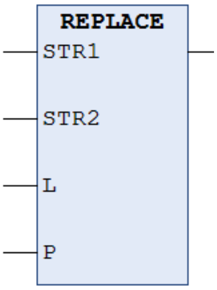
ST program

```
output1 'abe' := DELETE(input1 'abcde', input2 2, input3 3);
```

3.9.8 REPLACE (Replacing a Character String)

This is a function that replaces the character strings and outputs the replaced character strings.

■ Icon



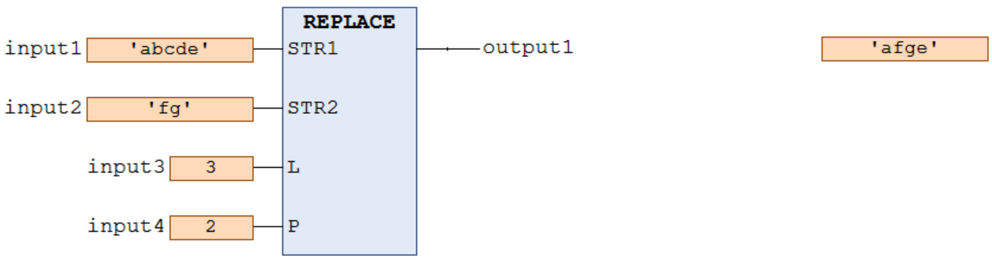
■ Parameter

Scope	Name	Type	Description
Input	STR1	STRING	Specifies the character string to be replaced.
	STR2	STRING	Specifies the character string to be added by replacement.
	L	INT	Specifies the number of characters to be deleted by replacement.
	P	INT	Specifies the position where the character string specified by STR2 is to be added by replacement.
Output	-	STRING	Replaces the number of characters specified in L with the character string specified in STR2 from the left end of the character string specified in STR1 starting from the position specified in P and outputs the replaced data.

■ Program example

This program is designed to replace a character string consisting of the number of characters specified in “input3” with the character string specified in “input2” from the position specified in “input4” in the character string of the input variable “input1” and to output the replaced data to the output variable “output1”.

LD program



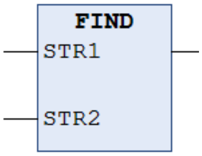
ST program

```
output1 'afge' := REPLACE(input1 'abcde', input2 'fg',
           input3 3 , input4 2 );
```

3.9.9 FIND (Search for Character String)

This is a function that searches for a specified character string and outputs the searched position.

■ Icon



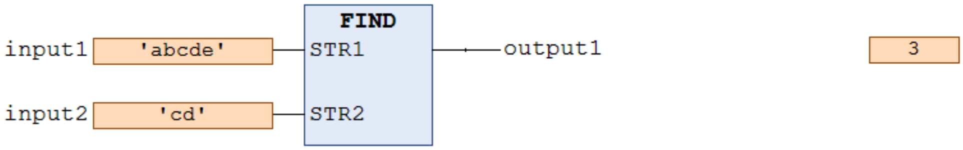
■ Parameter

Scope	Name	Type	Description
Input	STR1	STRING	Specifies the character string in which a character string is to be searched for.
	STR2	STRING	Specifies s character string to search for in the character string specified in STR1.
Output	-	INT	Searches for the character string specified in STR2 in the character string specified in STR1 and outputs the position from the left end.

■ Program example

This program is designed to search for the character string specified in “input2” in the character string of the input variable “input1” and to output the position from the left to the output variable ”output1”.

LD program



ST program

Info.

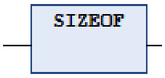
- Outputs 0 if the character string is not found.
- If the character string is found in multiple places, the position found first (the leftmost position) is output.

3.10 Other Instructions

3.10.1 SIZEOF (Get the Size)

This is a function that outputs the size (number of bytes) of the input argument.

■ Icon



■ Parameter

Scope	Type	Description
Input	(Note 1)	Specifies the argument whose size is to be calculated.
Output	(Note 1)	Outputs the size of (1).

(Note 1) Usable data types

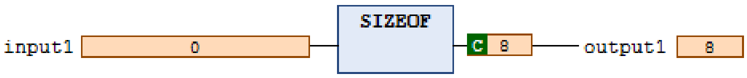
All standard data types

(BOOL, BYTE, WORD, DWORD, LWORD, SINT, USINT, INT, UINT, DINT, UDINT, LINT, ULINT, REAL, LREAL, TIME, LTIME, DATE, TIME_OF_DAY, DATE_AND_TIME, STRING, WSTRING)

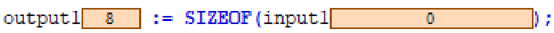
■ Program example

This program is designed to output the size of the ULINT type input variable “input1” to the UINT type output variable “output1”.

LD program



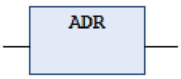
ST program



3.10.2 ADR (Get the Address)

This is a function that outputs the address of the variable.

■ Icon



■ Parameter

Scope	Type	Description
Input	(Note 1)	Input the variable from which to get the address.
Output	(Note 1)	Outputs the address (pointer) of the input variable.

(Note 1) Usable data types

All standard data types

(BOOL, BYTE, WORD, DWORD, LWORD, SINT, USINT, INT, UINT, DINT, UDINT, LINT, ULINT, REAL, LREAL, TIME, LTIME, DATE, TIME_OF_DAY, DATE_AND_TIME, STRING, WSTRING)

■ Usable data type

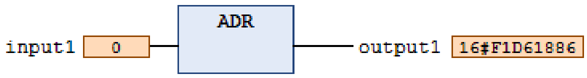
All standard data types

(BOOL, BYTE, WORD, DWORD, LWORD, SINT, USINT, INT, UINT, DINT, UDINT, LINT, ULINT, REAL, LREAL, TIME, LTIME, DATE, TIME_OF_DAY, DATE_AND_TIME, STRING, WSTRING)

■ Program example

This program is designed to output the address of the input variable “input1” to the output variable “output1”.

LD program



ST program

```
output1 16#F1D61884  := ADR(input1 0 );
```

3.11 SD Memory Card Slot Instruction

3.11.1 SYS_GetSDCoverState (Get SD Card Cover Open / Close State)

This is a function that gets an open / close state of the card cover for the SD memory card slot.

■ Icon

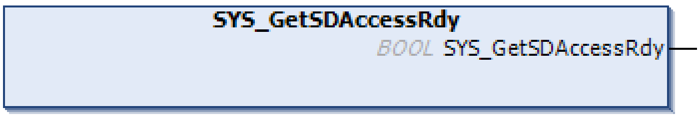
■ Parameter

Scope	Name	Type	Description
Output	SYS_GetSDCoverState	BOOL	TRUE: The card cover is closed. FALSE: The card cover is open.

3.11.2 SYS_GetSDAccessRdy (Get SD Card Access Ready State)

This is a function block that gets the state whether an access to the SD memory card is allowed.

■ Icon



■ Parameter

Scope	Name	Type	Description
Output	SYS_GestSDAccessRdy	BOOL	TRUE: Access to the SD memory card is enabled. FALSE: Access to the SD memory card is disabled.

(MEMO)

4 Function Blocks (Basic Instructions)

4.1 Timer Instructions

Timer instructions can be used to perform timer operations.

4.1.1 TON (Timer ON)

This is a function block (FB) that starts the timer when the input becomes TRUE. After a specified time elapses, the output becomes TRUE.

■ Icon

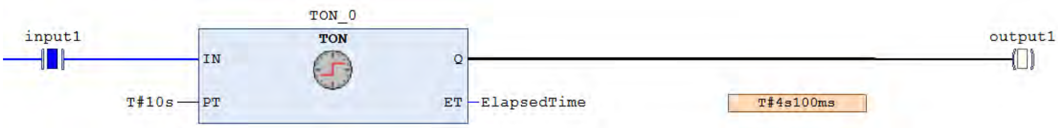
■ Parameter

Scope	Name	Type	Description
Input	IN	BOOL	Starts the timer when FALSE becomes TRUE and the timer continues counting while it remains TRUE. Resets the timer when it becomes FALSE.
	PT	TIME	Specifies the timer time.
Output	Q	BOOL	Outputs TRUE when the time specified in the input argument PT elapses.
	ET	TIME	Specifies the elapsed time of the timer.

■ Program example

This program is designed to start the timer when the input variable “input1” becomes TRUE and, after an elapse of 10 seconds, to cause the output variable “output1” to become TRUE. The instance name is TON_0.

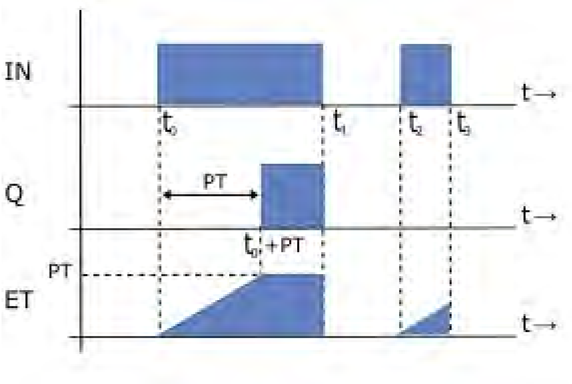
LD program



ST program

```
TON_0(IN TRUE := input1 TRUE , PT T#10s := T#10S,
     Q FALSE => output1 FALSE , ET T#3s659ms => ElapsedTime T#3s659ms );
```

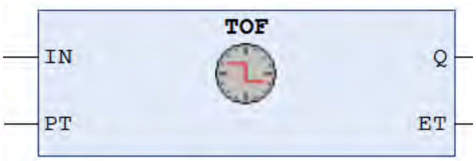
■ **Time-sequence diagram**



4.1.2 TOF (Timer OFF)

This is a function block (FB) that starts the timer when the input becomes FALSE. After a specified time elapses, the output becomes FALSE.

■ **Icon**



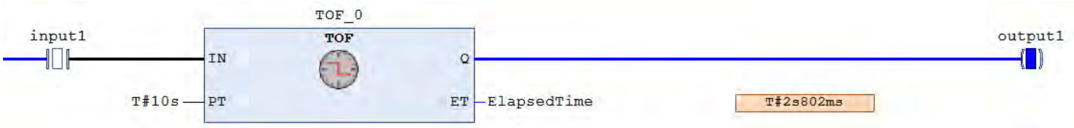
■ **Parameter**

Scope	Name	Type	Description
Input	IN	BOOL	Starts the timer when TRUE becomes FALSE and the timer continues counting while it remains FALSE. Resets the timer when it becomes TRUE.
	PT	TIME	Specifies the timer time.
Output	Q	BOOL	Outputs FALSE when the time specified in the input argument PT elapses.
	ET	TIME	Specifies the elapsed time of the timer.

■ **Program example**

This program is designed to start the timer when the input variable “input1” changes from TRUE to FALSE and, after an elapse of 10 seconds, to cause the output variable “output1” to become FALSE. The instance name is TOF_0.

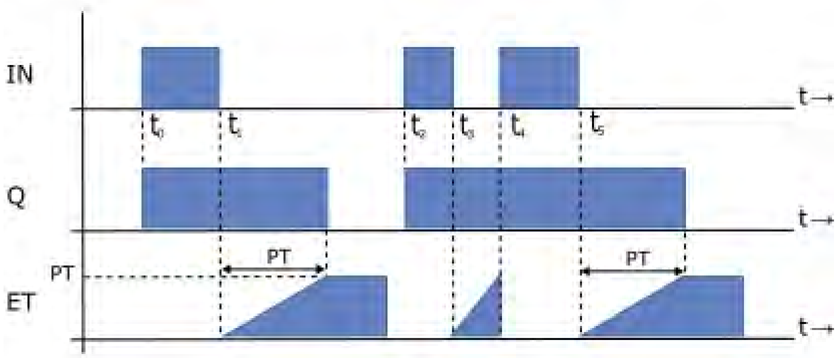
LD program



ST program

```
TOF_0(IN FALSE := input1 FALSE,
      PT T#10s := T#10S,
      Q TRUE => output1 TRUE,
      ET T#2s113ms => ElapsedTime T#2s113ms);
```

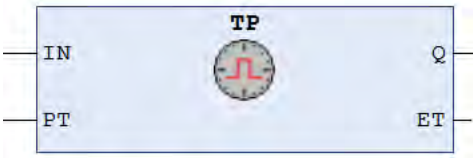
■ **Time-sequence diagram**



4.1.3 TP (Timer Pulse)

This is a function block that starts the timer at the rising edge. The output remains TRUE while the timer keeps counting. After a specified time elapses, the output becomes FALSE.

■ **Icon**



■ **Parameter**

Scope	Name	Type	Description
Input	IN	BOOL	Starts the timer when FALSE changes to TRUE (rising edge).

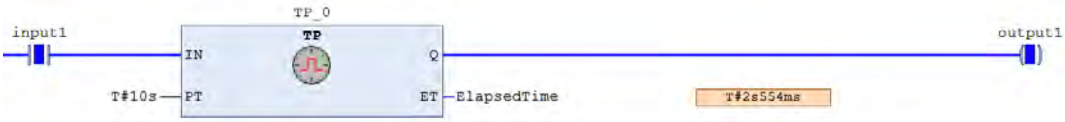
Scope	Name	Type	Description
			Resets the timer when the timer expires and TRUE changes to FALSE .
	PT	TIME	Specifies the timer time.
Output	Q	BOOL	Outputs TRUE from when the timer is started until when the time specified in the input argument PT elapses. Outputs FALSE after the specified time elapses.
	ET	TIME	Specifies the elapsed time of the timer.

■ Program example

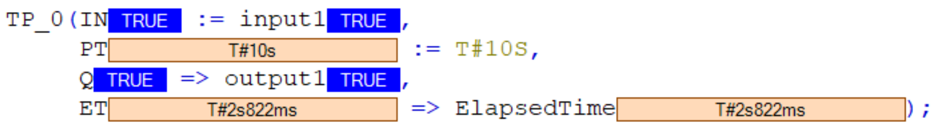
This program is designed to start the timer when the input variable “input1” changes from FALSE to TRUE and, during the time from when the timer is started to when the timer expires (for 10 seconds), to cause the output variable “output1” to remain TRUE.

The instance name is TP_0.

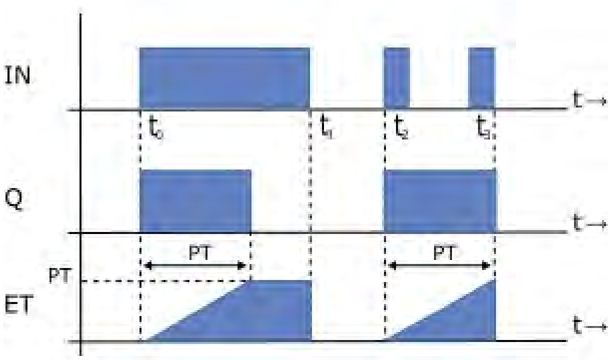
LD program



ST program



■ Time-sequence diagram



4.1.4 RTC (Realtime Clock)

This is a function block that starts counting time at the rising edge starting from the specified date and time. The output remains TRUE while the time counting continues. After a specified time elapses, the output becomes FALSE.

■ Icon

■ Parameter

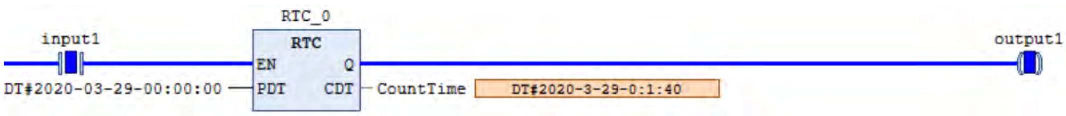
Scope	Name	Type	Description
Input	EN	BOOL	Starts counting time from the date and time specified in the input argument PDT when FALSE changes to TRUE (rising edge). When TRUE changes to FALSE, DT#1970-01-01-00:00:00 is set in the output argument CDT.
	PDT	DATE_AND_TIME	Date and time when time counting starts
Output	Q	BOOL	Outputs TRUE while time counting continues.
	CDT	DATE_AND_TIME	Outputs the time count time from the date and time specified in the input argument PDT.

■ Program example

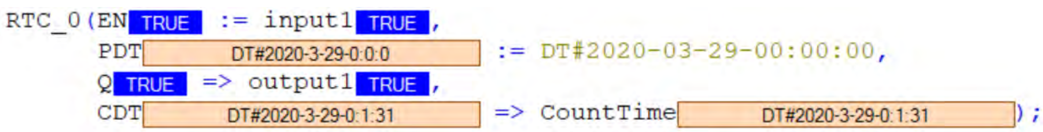
This program is designed to start counting time, starting from 0 o'clock of March 29, 2020, when the input variable “input1” changes from FALSE to TRUE, and, to cause the output variable “output1” to remain TRUE while time counting continues.

The instance name is RTC_0.

LD program



ST program



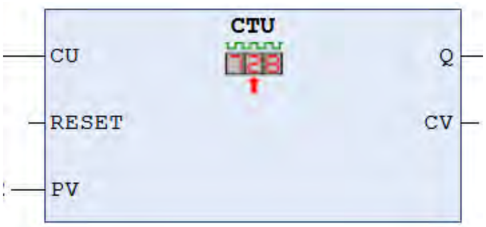
4.2 Counter Instructions

Counter instructions can be used to perform counter operations.

4.2.1 CTU (Up Counter)

This is a function block that increments the counter value by 1 every time the rising edge occurs.

■ Icon



■ Parameter

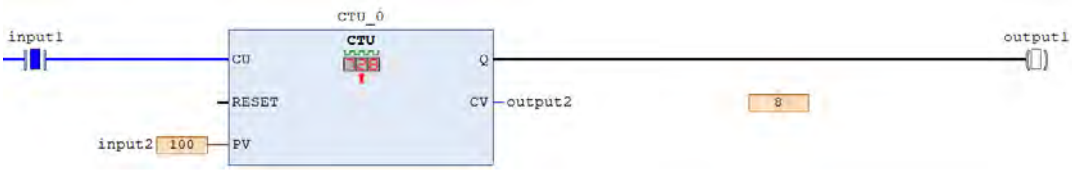
Scope	Name	Type	Description
Input	CU	BOOL	Increments the value of the output argument CV by 1 when FALSE changes to TRUE (rising edge).
	RESET	BOOL	If TRUE, 0 is set in the output argument CV.
	PV	WORD	Target value of CV
Output	Q	BOOL	Outputs TRUE when the CV value reaches the PV value.
	CV	WORD	Outputs the current counter value.

■ Program example

This program is designed to increment the value of the output variable “output2” by 1 every time the input variable “input1” changes from FALSE to TRUE. The program is designed to cause the output variable “output1” to change to TRUE when the value (100) of the input variable “input2” is counted up.

The instance name is CTU_0.

LD program



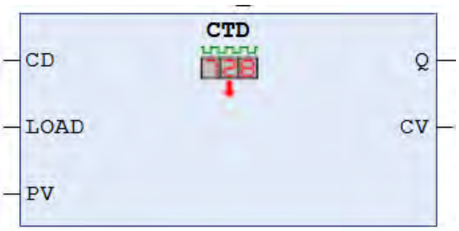
ST program

```
CTU_0(
      CU TRUE := input1 TRUE ,
      RESET:= ,
      PV 100 := 100,
      Q FALSE => output1 FALSE ,
      CV 8 => output2 8 );
```

4.2.2 CTD (Down Counter)

This is a function block that decrements the counter value by 1 every time the rising edge occurs.

■ Icon



■ Parameter

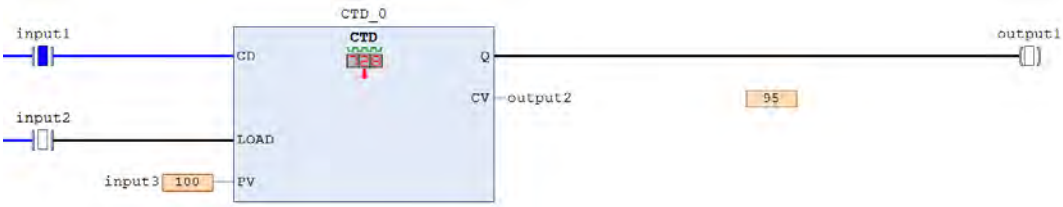
Scope	Name	Type	Description
Input	CD	BOOL	Decrements the value of the output argument CV by 1 when FALSE changes to TRUE (rising edge).
	LOAD	BOOL	If TRUE, the value specified in PV is set in the output argument CV.
	PV	WORD	Initial value of the counter value
Output	Q	BOOL	Outputs TRUE when the CV value becomes 0.
	CV	WORD	Outputs the current counter value.

■ Program example

This program is designed to decrement the value of the output variable “output2” by 1 every time the input variable “input1” changes from FALSE to TRUE, and to cause the output variable “output1” to change to TRUE when the value becomes 0. The initial value (100) to count down from is specified in the input variable “input3”.

The instance name is CTD_0.

LD program



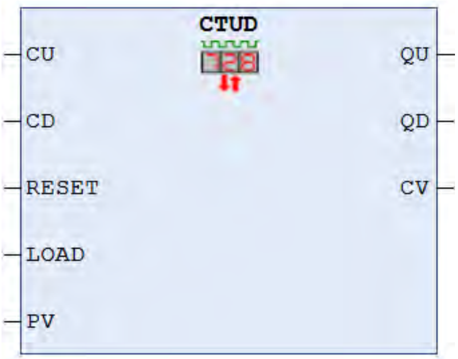
ST program

```
CTD_0(
    CD TRUE := input1 TRUE ,
    LOAD FALSE := input2 FALSE ,
    PV 100 := input3 100 ,
    Q FALSE => output1 FALSE ,
    CV 95 => output2 95 );
```

4.2.3 CTUD (Up-down Counter)

This is a function block that increments or decrements the counter value by 1 every time the rising edge occurs.

■ Icon



■ Parameter

Scope	Name	Type	Description
Input	CU	BOOL	Increments the value of the output argument CV by 1 when FALSE changes to TRUE (rising edge).
	CD	BOOL	Decrements the value of the output argument CV by 1 when FALSE changes to TRUE (rising edge).
	RESET	BOOL	If TRUE, 0 is set in the output argument CV.
	LOAD	BOOL	If TRUE, the value specified in PV is set in the output argument CV.
	PV	WORD	Initial value of the counter value
Output	QU	BOOL	Outputs TRUE when the CV value reaches the PV value.

Scope	Name	Type	Description
	QD	BOOL	Outputs TRUE when the CV value becomes 0.
	CV	WORD	Outputs the current counter value.

■ Program example

Every time the input variable “input1” changes from FALSE to TRUE, the value of the output variable “output3” is incremented by 1.

Every time the input variable “input2” changes from FALSE to TRUE, the value of the output variable “output3” is decremented by 1

When the output variable “output3” becomes greater than or equal to the input variable “input5”, the output variable “output1” becomes TRUE.

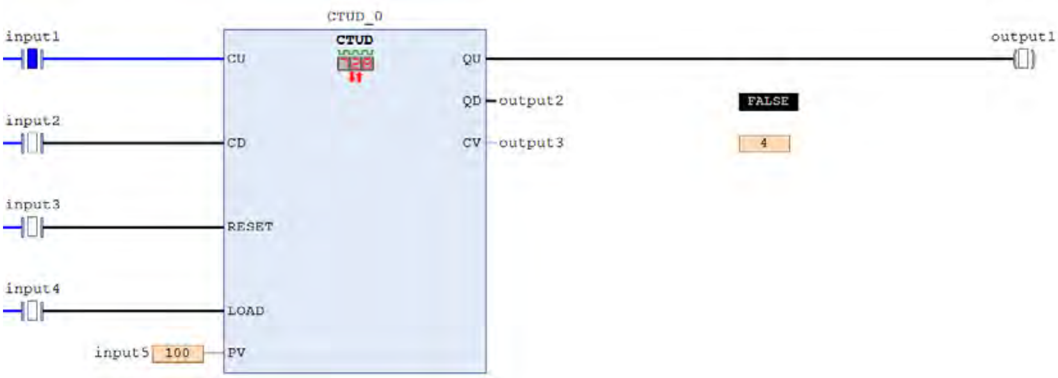
When the output variable “output3” becomes 0, the output variable “output2” becomes TRUE.

When the input variable “input3” becomes TRUE, the output variable “output3” becomes 0.

When the input variable “input4” becomes TRUE, the value (100) of the input variable “input5” is set in the output variable “output3”.

The instance name is CTUD_0.

LD program



ST program

```
CTUD_0(
    CU TRUE := input1 TRUE ,
    CD FALSE := input2 FALSE ,
    RESET FALSE := input3 FALSE ,
    LOAD FALSE := input4 FALSE ,
    PV 100 := input5 100 ,
    QU FALSE => output1 FALSE ,
    QD FALSE => output2 FALSE ,
    CV 3 => output3 3 );
```

4.3 Edge Detection Instructions

Edge detection instructions can be used to perform edge detection.

4.3.1 R_TRIG (Rising Edge Detection)

This is a function block that detects a rising edge.

■ Icon

■ Parameter

Scope	Name	Type	Description
Input	CLK	BOOL	Input that detects a rising edge
Output	Q	BOOL	Outputs TRUE for one cycle only when a rising edge is detected in the input argument CLK.

■ Program example

When the input variable “input1” changes from FALSE to TRUE, the output variable “output1” becomes TRUE for one cycle only.
The instance name is R_TRIG_0.

LD program

ST program

```
R_TRIG_0(
      CLK TRUE := input1 TRUE ,
      Q TRUE => output1 TRUE );
```

4.3.2 F_TRIG (Falling Edge Detection)

This is a function block that detects a falling edge.

■ Icon

■ Parameter

Scope	Name	Type	Description
Input	CLK	BOOL	Input that detects a falling edge
Output	Q	BOOL	Outputs TRUE for one cycle only when a falling edge is detected in the input argument CLK.

■ Program example

When the input variable “input1” changes from FALSE to TRUE, the output variable “output1” becomes TRUE for one cycle only.

The instance name is F_TRIG_0.

LD program

ST program

```
F_TRIG_0(
    CLK FALSE := input1 FALSE ,
    Q TRUE => output1 TRUE );
```

4.4 Bistable Circuit Instructions

Bistable circuit instructions can be used to perform edge detection.

4.4.1 SR (Set-priority Bistable Circuit)

This is a function block that realizes a bistable (flip-flop) circuit. The priority is given to the set input.

■ Icon

■ Parameter

Scope	Name	Type	Description
Input	SET1	BOOL	Specifies the set input for a bistable circuit.
	RESET	BOOL	Specifies the reset input for a bistable circuit.
Output	Q1	BOOL	When the input argument SET1 becomes TRUE, outputs and holds TRUE. When the input argument RESET becomes TRUE, outputs and holds FALSE. When both SET1 and RESET1 are TRUE, outputs and holds TRUE.

■ Program example

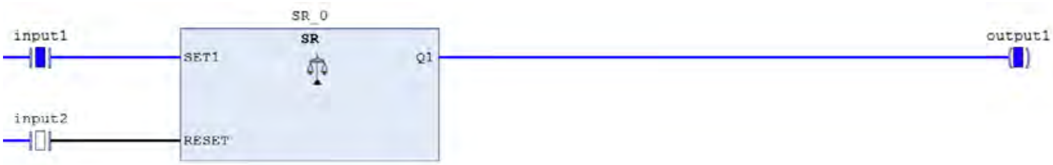
When the input variable “input1” becomes TRUE, the output variable “output1” becomes TRUE.

Even if the input variable “input1” becomes FALSE, “output1” remains TRUE.

When the input variable “input1” is FALSE and if input variable “input2” becomes TRUE, the output variable “output1” becomes FALSE.

The instance name is SR_0.

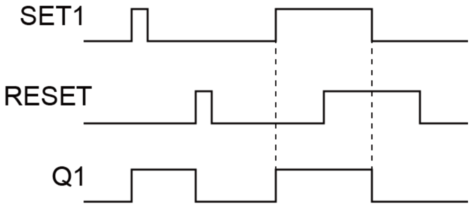
LD program



ST program

```
SR_0(
    SET1 TRUE := input1 TRUE ,
    RESET FALSE := input2 FALSE ,
    Q1 TRUE => output1 TRUE );
```

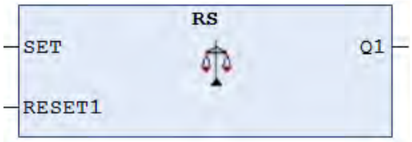
■ Time-sequence diagram



4.4.2 RS (Reset-priority Bistable Circuit)

This is a function block that realizes a bistable (flip-flop) circuit. The priority is given to the reset input.

■ Icon



■ Parameter

Scope	Name	Type	Description
Input	SET1	BOOL	Specifies the set input for a bistable circuit.
	RESET	BOOL	Specifies the reset input for a bistable circuit.
Output	Q1	BOOL	When the input argument SET1 becomes TRUE, outputs and holds TRUE. When the input argument RESET becomes TRUE, outputs and holds FALSE. When both SET1 and RESET1 are TRUE, outputs and holds FALSE.

■ Program example

When the input variable “input1” becomes TRUE, the output variable “output1” becomes TRUE.

Even if the input variable “input1” becomes FALSE, “output1” remains TRUE.

When the input variable “input1” is FALSE and if the input variable “input2” becomes TRUE, the output variable “output1” becomes FALSE.

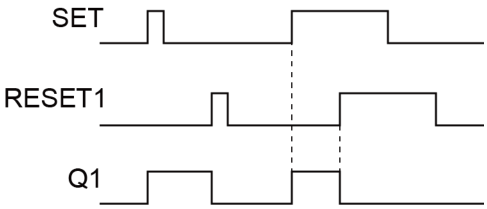
The instance name is RS_0.

LD program

ST program

```
RS_0(
     SET TRUE := input1 TRUE ,
     RESET1 FALSE := input2 FALSE ,
     Q1 TRUE => output1 TRUE );
```

■ **Time-sequence diagram**



(MEMO)

5 Motion Control Function Blocks (Single Axis Control)

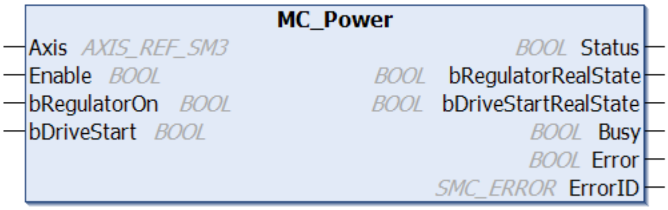
This section describes motion control function blocks for the single axis.

5.1 Servo ON

5.1.1 MC_Power (Servo ON)

This is a function block (FB) that sets the axis to the servo ON state to be ready for operation.

■ Icon

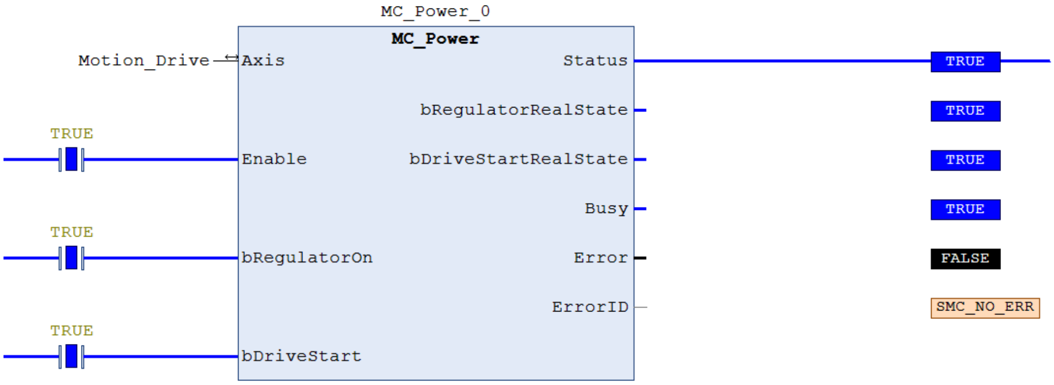


■ Parameter

Scope	Name	Type	Initial	Description
Input / output	Axis	AXIS_REF_SM3	-	Specifies the axis.
Input	Enable	BOOL	FALSE	TRUE: The FB can be executed.
	bRegulatorOn	BOOL	FALSE	TRUE: Servo ON FALSE: Servo OFF
	bDriveStart	BOOL	FALSE	TRUE: Quick stop is disabled.
Output	Status	BOOL	FALSE	TRUE: The axis can be executed.
	bRegulatorRealState	BOOL	FALSE	TRUE: The FB is ready to be executed.
	bDriveStartRealState	BOOL	FALSE	TRUE: Operation is not stopped due to quick stop.
	Busy	BOOL	FALSE	TRUE: Execution of the FB is not completed.
	Error	BOOL	FALSE	TRUE: An error has occurred within the FB.
	ErrorID	SMC_ERROR	FALSE	An error ID is output.

■ Program example

The following program is designed to set the axis that corresponds to the input variable Motion_Drive to the servo ON state.



Info.

- If the input arguments Enable, bRegulatorOn, and bDriveStart are TRUE and the output argument Status is FALSE, a hardware problem may occur.
- Reference manual

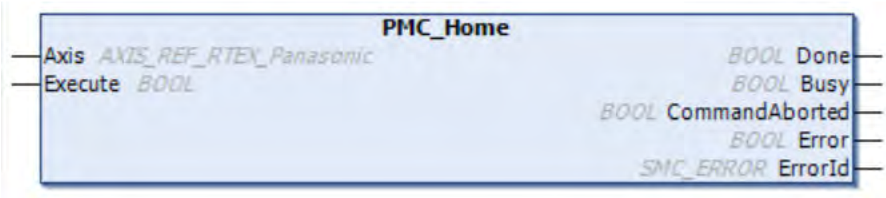
  GM1 Controller RTEX User's Manual (Operation Edition)

5.2 Home Return

5.2.1 PMC_Home (Home Return)

This is a function block (FB) that performs home return of the axis. The home return function of the servo amplifier is used.

■ Icon



■ Parameter

Scope	Name	Type	Initial	Description
Input / output	Axis	AXIS_REF_RTEX_Panasonic	-	Specifies the axis.
Input	Execute	BOOL	FALSE	TRUE: Starts execution at the rising edge. FALSE: Stops processing.
Output	Done	BOOL	FALSE	TRUE: Execution is completed and transitioned to the Standstill state.
	Busy	BOOL	FALSE	TRUE: Execution of the FB is not completed.
	CommandAborted	BOOL	FALSE	TRUE: An interruption from other FB has occurred.
	Error	BOOL	FALSE	TRUE: An error has occurred within the FB.
	ErrorID	SMC_ERROR	0	An error ID is output.

■ Execution operation

- Execute = TRUE: Starts the home return mode. Execute = FALSE: Ends the home return mode.
- When PMC_Home is successfully completed (when Done changes to TRUE), the home return mode is automatically ended.
- When PMC_Home is abnormally terminated (when Error changes to TRUE), end the home return mode by setting Execute = FALSE and making a call.

■ Execution errors

The PMC_Home function block outputs the following errors.

Error	Description
SMC_WRONG_CONTROLLER_MODE	Executed in a mode other than the position control mode. Change to SMC_position using SMC_SetControllerMode.
SMC_DI_HOMING_ERROR	The version of the amplifier paired with an absolute encoder is lower than V1.24.
	Trigger setting is incorrect.
	Amplifier parameters (Pr4.00 to Pr4.07) are incorrect.
	Abnormal state in HOME, POT, or NOT is detected.
	The home return cannot be completed even if POT and NOT settings were inverted three times or more.
	The home return was completed at an incorrect position.
SMC_MS_DIRECTION_NOT_APPLICABLE	The return direction setting is incorrect.
SMC_AXIS_NOT_READY_FOR_MOTION	The axis is in a state (Stopping, Disabled, or Errorstop) where PMC_Home cannot be executed.
SMC_REGULATOR_OR_START_NOT_SET	The servo was turns OFF and the brake was applied.
SMC_3SH_INVALID_VELACC_VALUES	The input target velocity, home return creep speed, acceleration, or deceleration is incorrect.
SMC_AXIS_REF_CHANGED_DURING_OPERATION	The Axis was changed during operation.

■ Execution conditions

- As the PMC_Home function block uses the RTEX home return command, it cannot be executed together with PMC_ReadLatchPosition or PMC_StopLatchPosition.
- If PMC_Home is executed while PMC_ReadLatchPosition or PMC_StopLatchPosition is being executed, the CommandAborted parameter becomes TRUE. Furthermore, if PMC_Home of another instance is executed while one PMC_Home is being executed, the CommandAborted parameter of the PMC_Home executed later becomes TRUE.

■ Amplifier parameter conditions

When using PMC_Home, set amplifier parameters as shown in the following table.

Parameter	Parameter name	Setting A	Setting B
Pr4.00	SI1 input selection	SI-MON5	SI-MON5
Pr4.01	SI2 input selection	POT	
Pr4.02	SI3 input selection	NOT	
Pr4.03	SI4 input selection	SI-MON1	SI-MON1
Pr4.04	SI5 input selection	HOME	HOME
Pr4.05	SI6 input selection	EXT2	POT
Pr4.06	SI7 input selection	EXT3	NOT
Pr4.07	SI8 input selection	SI-MON4	SI-MON4

Return methods that can be executed for the settings A and B are as shown in the following table.

Return method	Setting A	Setting B
DOG method 1	○	○(Note 2)
DOG method 2	×(Note 1)	○
DOG method 3	○	○(Note 2)
Limit method 1	○	○(Note 2)
Limit method 2	×(Note 1)	○
Home return method	○	○(Note 2)
Stop-on-contact method 1	○	○(Note 2)
Stop-on-contact method 2	○	○
Data setting method	○	○
High-speed home return method	○	○

(Note 1) When using POT, NOT, or HOME as a home reference trigger, assign them as follows.
HOME: SI5 input selection
POT: SI6 input selection
NOT: SI7 input selection

(Note 2) When EXT2 or EXT3 is used as a home reference trigger, it can be used only for the above setting A.

Info.

- Reference manual
  GM1 Controller RTEX User's Manual (Operation Edition)

5.3 Control Switch

5.3.1 SMC_SetControllerMode (Control Mode Setting)

This is a function block (FB) that sets up the control mode for controlling the position, velocity, and torque.

■ Icon

■ Parameter

Scope	Name	Type	Initial	Description
Input / output	Axis	AXIS_REF_SM3	-	Specifies the axis.
Input	bExecute	BOOL	FALSE	Sets up the control mode at the rising edge.
	nControllerMode	SMC_CONTROLLER_MODE	SMC_position	Specifies the control mode.
Output	bDone	BOOL	FALSE	TRUE: Control mode setup is completed.
	bBusy	BOOL	FALSE	TRUE: Execution of the FB is not completed.
	bError	BOOL	FALSE	TRUE: An error has occurred within the FB.
	nErrorID	SMC_ERROR	0	An error ID is output.

■ SMC_CONTROLLER_MODE (Enumeration type)

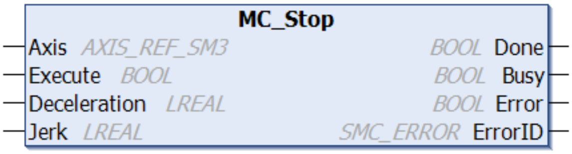
Name	Value	Description
SMC_nocontrol	0	Usage prohibited
SMC_torque	1	Torque mode
SMC_velocity	2	Velocity mode
SMC_position	3	Position mode
SMC_current	4	Usage prohibited

5.4 Stop

5.4.1 MC_Stop (Forced Stop)

This is a function block (FB) that causes the axis to make a deceleration stop. After stopping, the axis remains stopped while Execute is TRUE. While the axis is being decelerated or while it is stopped, other function blocks cannot be executed.

■ Icon



■ Parameter

Scope	Name	Type	Initial	Description
Input / output	Axis	AXIS_REF_SM3	-	Specifies the axis.
Input	Execute	BOOL	FALSE	Starts execution at the rising edge. While it is TRUE, other FB cannot be executed.
	Deceleration	LREAL	0	Specifies the deceleration (u/s^2).
	Jerk	LREAL	0	Specifies the jerk (u/s^3).
Output	Done	BOOL	FALSE	TRUE: The axis velocity has reached 0.
	Busy	BOOL	FALSE	TRUE: The FB is in operation.
	Error	BOOL	FALSE	TRUE: An error has occurred.
	ErrorID	SMC_ERROR	0	An error ID is output.

Info.

- In the torque control mode (SMC_torque), the axis cannot be stopped using MC_Stop. For stopping methods, refer to PMC_SetTorque.
- Reference manual

  GM1 Controller RTEX User's Manual (Operation Edition)

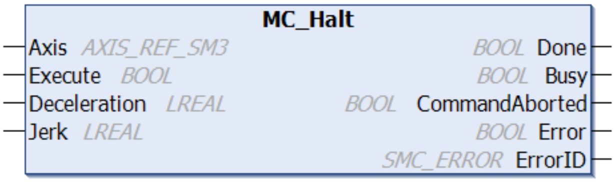
REFERENCE

5.8.1 PMC_SetTorque (Torque Control)

5.4.2 MC_Halt (Halt)

This is a function block (FB) that causes the axis to make a deceleration stop. After the axis is stopped or while the axis is being decelerated, other motion instructions can be executed.

■ Icon



■ Parameter

Scope	Name	Type	Initial	Description
Input / output	Axis	AXIS_REF_SM3	-	Specifies the axis.
Input	Execute	BOOL	FALSE	Starts execution at the rising edge.
	Deceleration	LREAL	0	Specifies the deceleration (u/s^2).
	Jerk	LREAL	0	Specifies the jerk (u/s^3).
Output	Done	BOOL	FALSE	TRUE: The axis velocity has reached 0.
	Busy	BOOL	FALSE	TRUE: The FB is in operation.
	CommandAborted	BOOL	FALSE	TRUE: An interruption from other FB has occurred.
	Error	BOOL	FALSE	TRUE: An error has occurred.
	ErrorID	SMC_ERROR	0	An error ID is output.

Info.

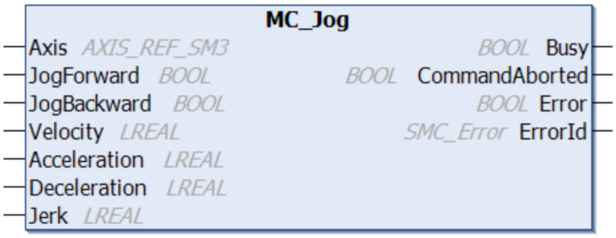
- Reference manual
  GM1 Controller RTEX User's Manual (Operation Edition)

5.5 JOG / Inching

5.5.1 MC_Jog (Jogging)

This is a function block (FB) that causes the axis to keep traveling in a forward or reverse direction at a constant velocity. While the input is TRUE, the axis keeps traveling in a forward or reverse direction at a constant velocity.

■ Icon



■ Parameter

Scope	Name	Type	Initial	Description
Input / output	Axis	AXIS_REF_SM3	-	Specifies the axis.
Input	JogForward	BOOL	FALSE	While JogForward is TRUE, the axis travels in a forward direction. If JogBackaward is TRUE at the same time, the axis does not operate.
	JogBackward	BOOL	FALSE	While JogBackward is TRUE, the axis travels in a reverse direction. If JogForward is TRUE at the same time, the axis does not operate.
	Velocity	LREAL	0	Specifies the velocity (u/s).
	Acceleration	LREAL	0	Specifies the acceleration (u/s^2).
	Deceleration	LREAL	0	Specifies the deceleration (u/s^2).
	Jerk	LREAL	0	Specifies the jerk (u/s^3).
Output	Busy	BOOL	FALSE	TRUE: Execution of the FB is not completed.
	CommandAborted	BOOL	FALSE	TRUE: An interruption from other FB has occurred.
	Error	BOOL	FALSE	TRUE: An error has occurred within the FB.
	ErrorID	SMC_ERROR	0	An error ID is output.

Info.

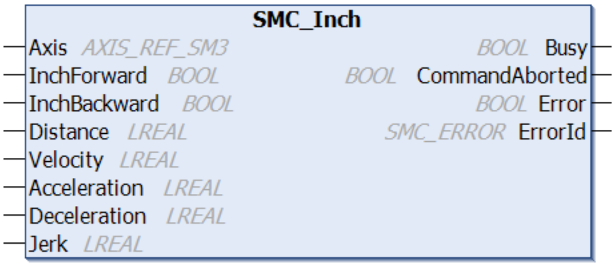
- Reference manual

  GM1 Controller RTEX User's Manual (Operation Edition)

5.5.2 SMC_Inch (Inching)

This is a function block (FB) that causes the axis to travel in a forward or reverse direction for a specified relative distance. When the input turns TRUE, the axis travels in a forward or reverse direction for a specified relative distance.

■ Icon



■ Parameter

Scope	Name	Type	Initial	Description
Input / output	Axis	AXIS_REF_SM3	-	Specifies the axis.
Input	InchForward	BOOL	FALSE	When the input changes from FALSE to TRUE, the axis starts traveling in a forward direction for the distance specified in Distance. When the input changes to FALSE before the axis travels the specified distance, the axis stops traveling. If InchBackaward is TRUE at the same time, the axis does not operate.
	InchBackward	BOOL	FALSE	When the input changes from FALSE to TRUE, the axis starts traveling in a reverse direction for the distance specified in Distance. When the input changes to FALSE before the axis travels the specified distance, the axis stops traveling. If InchForward is TRUE at the same time, the axis does not operate.
	Distance	LREAL	0	Specifies the travel distance (u).
	Velocity	LREAL	0	Specifies the velocity (u/s).
	Acceleration	LREAL	0	Specifies the acceleration (u/s^2).
	Deceleration	LREAL	0	Specifies the deceleration (u/s^2).

Scope	Name	Type	Initial	Description
	Jerk	LREAL	0	Specifies the jerk (u/s^3).
Output	Busy	BOOL	FALSE	TRUE: Execution of the FB is not completed.
	CommandAborted	BOOL	FALSE	TRUE: An interruption from other FB has occurred.
	Error	BOOL	FALSE	TRUE: An error has occurred within the FB.
	ErrorID	SMC_ERROR	0	An error ID is output.

Info.

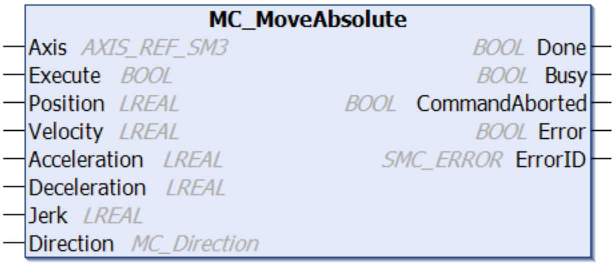
- Reference manual
  GM1 Controller RTEX User's Manual (Operation Edition)

5.6 Position Control

5.6.1 MC_MoveAbsolute (Absolute Value Positioning)

This is a function block (FB) that causes the axis to travel to a position specified as an absolute position.

■ Icon



■ Parameter

Scope	Name	Type	Initial	Description
Input / output	Axis	AXIS_REF_SM3	-	Specifies the axis.
Input	Execute	BOOL	FALSE	Starts execution at the rising edge.
	Position	LREAL	0	Specifies the target position (u).
	Velocity	LREAL	0	Specifies the maximum velocity (u/s).
	Acceleration	LREAL	0	Specifies the acceleration (u/s^2).
	Deceleration	LREAL	0	Specifies the deceleration (u/s^2).
	Jerk	LREAL	0	Specifies the jerk (u/s^3).
	Direction	MC_Direction	shortest	Specifies the traveling direction of the axis. Direction can be specified only for the modulo type. For the finite axis, the specification is ignored.
Output	Done	BOOL	FALSE	TRUE: The axis has reached the target position.
	Busy	BOOL	FALSE	TRUE: The FB is in operation.
	CommandAborted	BOOL	FALSE	TRUE: An interruption from other FB has occurred.
	Error	BOOL	FALSE	TRUE: An error has occurred.
	ErrorID	SMC_ERROR	0	An error ID is output.

■ **MC_Direction (Enumeration type)**

Name	Value	Description
positive	1	Travels in the positive direction.
negative	-1	Travels in the negative direction.
shortest	0	Travels in the shortest direction from the current command position to the target command position when MC_MoveAbsolute is executed.
fastest	3	Travels in the fastest direction from the current command position to the target command position when MC_MoveAbsolute is executed. If the axis is being driven by another function block when MC_MoveAbsolute is executed, selects the fastest direction within the GM1. If the axis is being stopped when MC_MoveAbsolute is executed, makes the same movement as for the shortest.
current	2	Travels to the current direction. If the axis is being driven by another function block when MC_MoveAbsolute is executed, travels in the same direction. If the axis is being stopped when MC_MoveAbsolute is executed, travels in the direction moved by the previously executed function block.

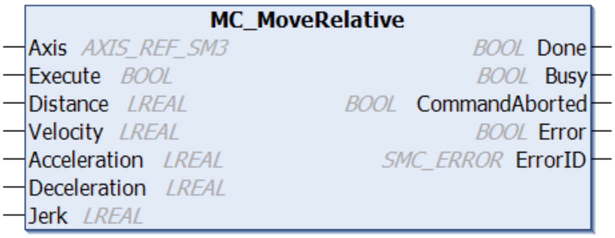
Info.

- Reference manual
  GM1 Controller RTEX User's Manual (Operation Edition)

5.6.2 MC_MoveRelative (Relative Value Positioning)

This is a function block (FB) that causes the axis to travel to a position specified as a relative position.

■ **Icon**



■ **Parameter**

Scope	Name	Type	Initial	Description
Input / output	Axis	AXIS_REF_SM3	-	Specifies the axis.

Scope	Name	Type	Initial	Description
Input	Execute	BOOL	FALSE	Starts execution at the rising edge.
	Distance	LREAL	0	Specifies the relative distance (u).
	Velocity	LREAL	0	Specifies the maximum velocity (u/s).
	Acceleration	LREAL	0	Specifies the acceleration (u/s^2).
	Deceleration	LREAL	0	Specifies the deceleration (u/s^2).
	Jerk	LREAL	0	Specifies the jerk (u/s^3).
Output	Done	BOOL	FALSE	TRUE: The axis has traveled the specified relative distance.
	Busy	BOOL	FALSE	TRUE: The FB is in operation.
	CommandAborted	BOOL	FALSE	TRUE: An interruption from other FB has occurred.
	Error	BOOL	FALSE	TRUE: An error has occurred.
	ErrorID	SMC_ERROR	0	An error ID is output.

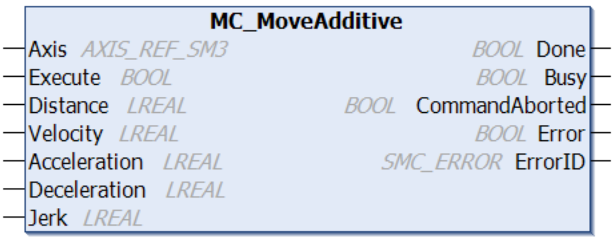
Info.

- Reference manual
  GM1 Controller RTEX User's Manual (Operation Edition)

5.6.3 MC_MoveAdditive (Target Position Change)

This is a function block (FB) that adds a relative distance to the target position of the immediately preceding instruction. No addition is made to the velocity, acceleration, or deceleration; they change to the specified values.

■ Icon



■ Parameter

Scope	Name	Type	Initial	Description
Input / output	Axis	AXIS_REF_SM3	-	Specifies the axis.
Input	Execute	BOOL	FALSE	Starts execution at the rising edge.
	Distance	LREAL	0	Specifies the relative distance (u) to be added.

Scope	Name	Type	Initial	Description
	Velocity	LREAL	0	Specifies the maximum velocity (u/s).
	Acceleration	LREAL	0	Specifies the acceleration (u/s^2).
	Deceleration	LREAL	0	Specifies the deceleration (u/s^2).
	Jerk	LREAL	0	Specifies the jerk (u/s^3).
Output	Done	BOOL	FALSE	TRUE: The axis has traveled the specified relative distance.
	Busy	BOOL	FALSE	TRUE: The FB is in operation.
	CommandAborted	BOOL	FALSE	TRUE: An interruption from other FB has occurred.
	Error	BOOL	FALSE	TRUE: An error has occurred.
	ErrorID	SMC_ERROR	0	An error ID is output.

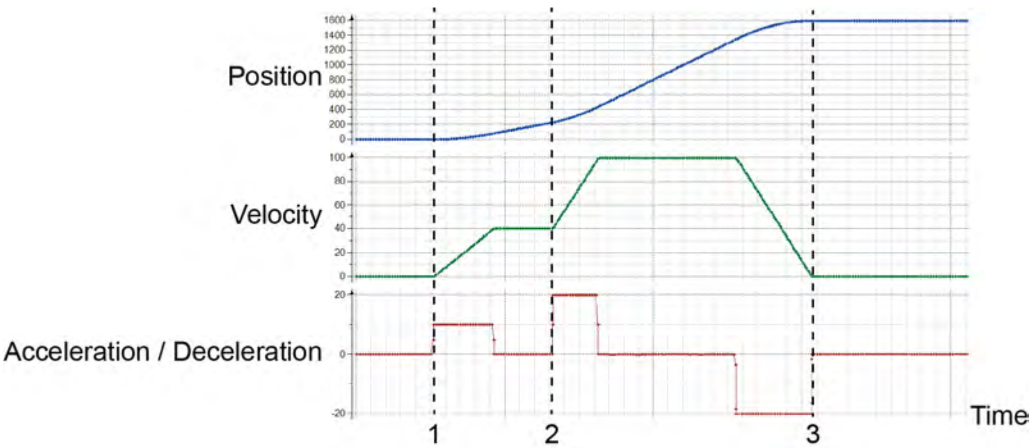
■ Operations when the function block is executed

This example shows the position, velocity, acceleration, and deceleration traces when MC_MoveAdditive is executed while MC_MoveRelative (Relative Value Positioning) is being executed.

Function block input parameters

Execution sequence	Function blocks	Distance (Relative distance)	Velocity (Velocity)	Acceleration (Acceleration)	Deceleration (Deceleration)
1	MC_MoveRelative	1000	40	10	10
2	MC_MoveAdditive	600	100	20	20

Trace



1. MC_MoveRelative is started.
2. MC_MoveAdditive is started.
   MC_MoveRelative is interrupted at the timing when MC_MoveAddive is started. The output parameter CommandAborted of MC_MoveRelative turns TRUE.

The velocity and acceleration / deceleration change to values (velocity: 100, acceleration / deceleration: 20) specified by MC_MoveAdditive.

3. MC_MoveAdditive is completed.

   The axis travels to the position where the relative distance specified by MC_MoveAdditive is added to the position specified by MC_MoveRelative (1000+600=1600).

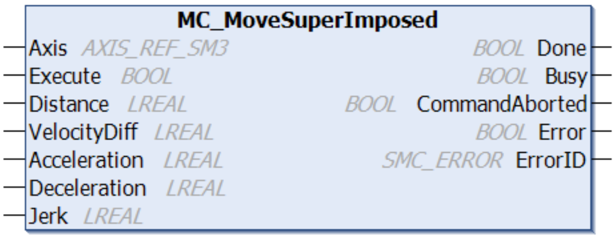
Info.

- Use the MC_MoveSuperImposed function block to cause the axis to travel after adding the velocity, acceleration, and deceleration to the previously executed instruction.

5.6.4 MC_MoveSuperImposed (Superimposed positioning)

This is a function block (FB) that adds a relative distance, a velocity, an acceleration, and a deceleration to the operations of the immediately preceding instruction.

■ Icon



■ Parameter

Scope	Name	Type	Initial	Description
Input / output	Axis	AXIS_REF_SM3	-	Specifies the axis.
Input	Execute	BOOL	FALSE	Starts execution at the rising edge.
	Distance	LREAL	0	Specifies the relative distance (u).
	VelocityDiff	LREAL	0	Specifies the maximum velocity (u/s).
	Acceleration	LREAL	0	Specifies the acceleration (u/s^2).
	Deceleration	LREAL	0	Specifies the deceleration (u/s^2).
	Jerk	LREAL	0	Specifies the jerk (u/s^3).
Output	Done	BOOL	FALSE	TRUE: The axis has traveled the specified relative distance.
	Busy	BOOL	FALSE	TRUE: The FB is in operation.
	CommandAborted	BOOL	FALSE	TRUE: An interruption from other FB has occurred.
	Error	BOOL	FALSE	TRUE: An error has occurred.
	ErrorID	SMC_ERROR	0	An error ID is output.

Note

- When MC_MoveSuperImposed is being executed (Busy = TRUE), be sure to call at every interval.

  If a call is not made, the axis may perform an unexpected operation.

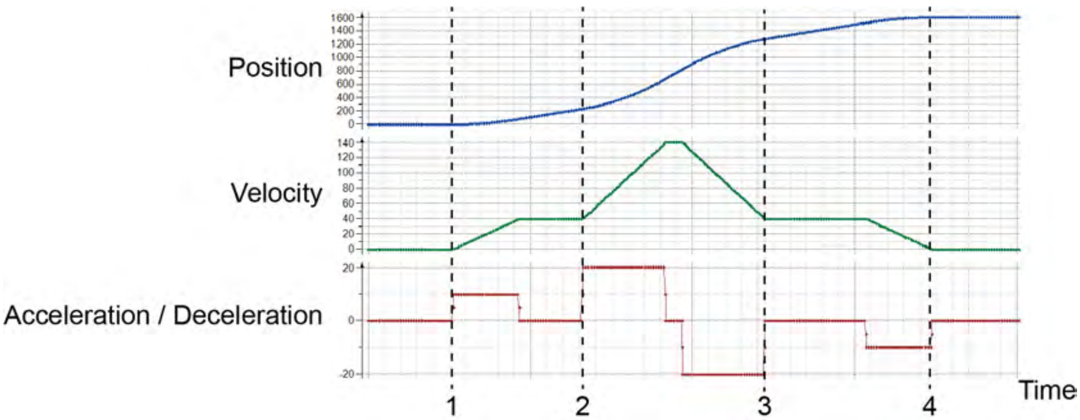
■ Operations when the function block is executed

This example shows the position, velocity, and acceleration traces when MC_MoveSuperImposed is executed while MC_MoveRelative (Relative Value Positioning) is being executed.

Function block input parameters

Execution sequence	Function blocks	Distance (Relative distance)	Velocity (Velocity)	Acceleration (Acceleration)	Deceleration (Acceleration)
1	MC_MoveRelative	1000	40	10	10
2	MC_MoveSuperImposed	600	100	20	20

Trace



1. MC_MoveRelative is started.
2. MC_MoveSuperImposed is started.

   The specified velocity, acceleration, and deceleration are added at the timing when MC_MoveSuperImposed is started.
3. MC_MoveSuperImposed is completed.
4. MC_MoveRelative is completed.

   The axis travels to the position where the relative distance specified by MC_MoveSuperImposed is added to the position specified by MC_MoveRelative (1000+600=1600).

Info.

- Use the MC_MoveAdditive function block to cause the axis to travel at the specified values without adding the velocity, acceleration, and deceleration to the previously executed instruction.

5.6.5 MC_PositionProfile (Position Profile Move)

This is a function block (FB) that causes the axis to operate according to the profile data that consists of a combination of position and time.

■ Icon

■ Parameter

Scope	Name	Type	Initial	Description
Input / output	Axis	AXIS_REF_SM3	-	Specifies the axis.
	TimePosition	MC_TP_REF	-	Specifies the time/position profile.
Input	Execute	BOOL	FALSE	Starts execution at the rising edge.
	ArraySize	INT	0	A member of the Input / output TimePosition. Specifies the number of points to be executed by FB in the array of time and position that are specified by MC_TP_ARRAY.
	PositionScale	LREAL	1	Position scaling
	Offset	LREAL	0	Position offset
Output	Done	BOOL	FALSE	TRUE: The movements specified by the profile are completed.
	Busy	BOOL	FALSE	TRUE: Execution of the FB is not completed.
	CommandAborted	BOOL	FALSE	TRUE: An interruption from other FB has occurred.
	Error	BOOL	FALSE	TRUE: An error has occurred within the FB.
	ErrorID	SMC_ERROR	0	An error ID is output.

■ MC_TP_REF (Structure)

Member	Type	Description
Number_of_pairs	INT	Not used
isAbsolute	BOOL	Methods of specifying the position of profile data TRUE: Specified in an absolute value. FALSE: Specified in a relative value.
MC_TP_Array	ARRAY [1..100] OF SMC_TP	Time and position profile data (1st point to 100th point)

■ SMC_TP (Structure)

Member	Type	Description
delta_time	TIME	Time of the profile data
position	LREAL	Position of the profile data

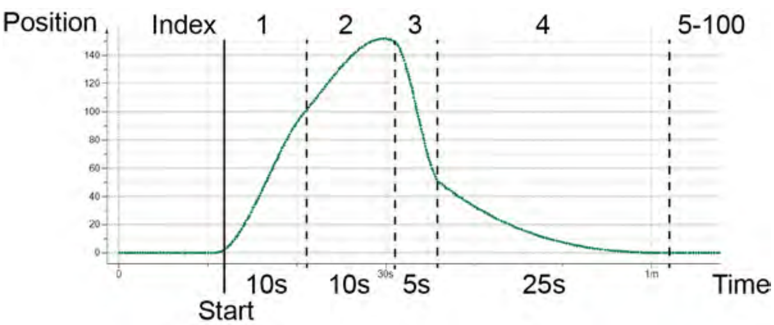
■ Operations when the function block is executed

This example shows time and position traces when the MC_PositionProfile function block is executed with the following parameter settings.

Parameter

Index of the MC_TP_Array array, a member of the input TimePosition	Delta_time (time)	Position (absolute position)
1	Time#10s	100
2	Time#10s	150
3	Time#5s	50
4	Time#25s	0
5 to 100	Time#0ms	0

Trace



Note

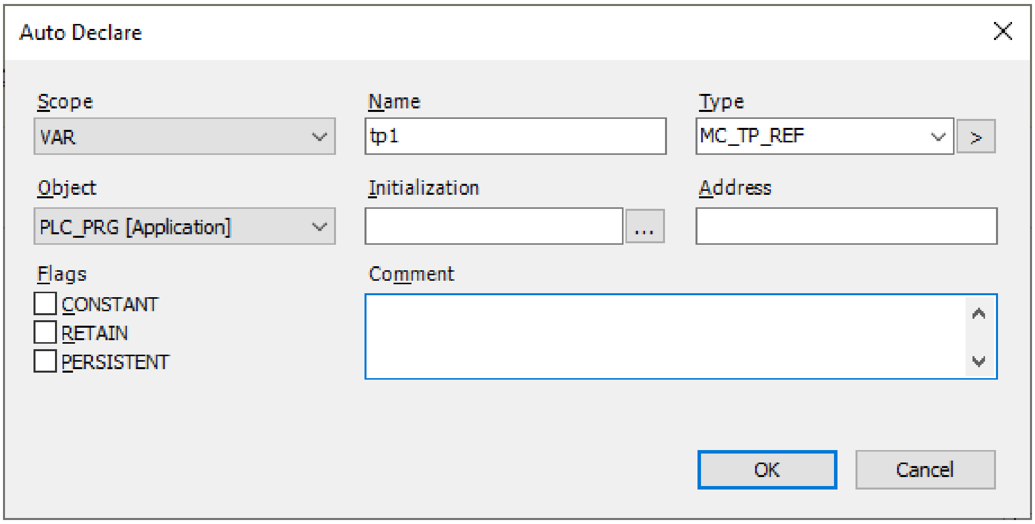
- While the axis keeps driving, do not specify delta_time to 0 ms.
  Otherwise, it may cause an abnormal operation.

5.6.6 Default Setting for Variables of the MC_TP_REF Type Structure

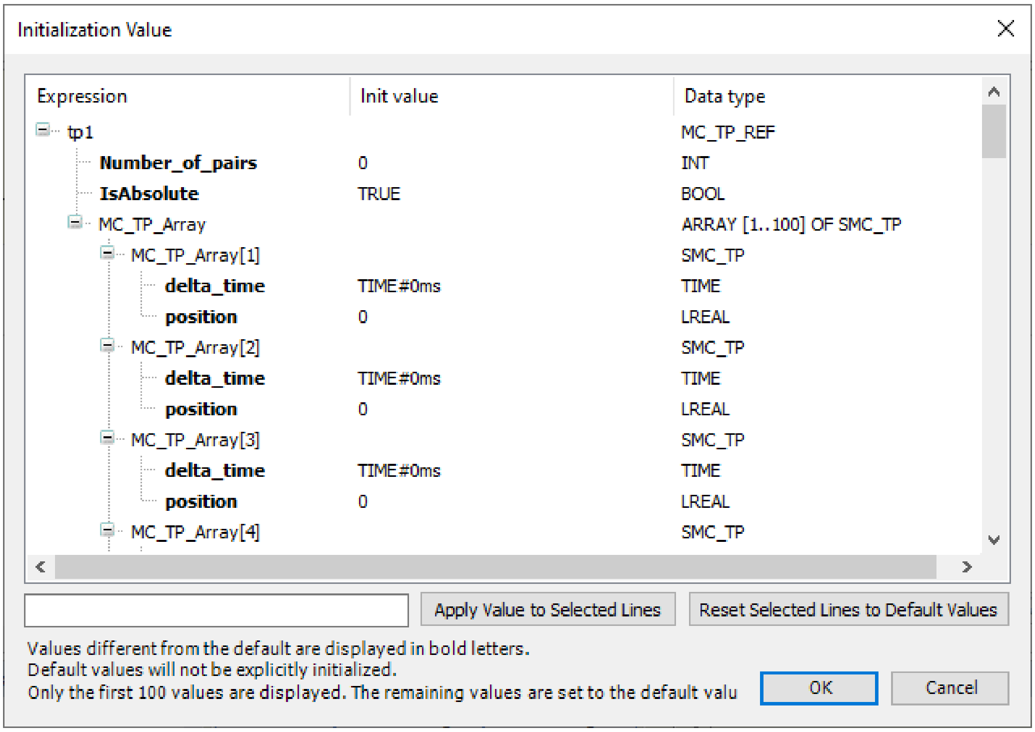
To enter the value of the input TimePosition, it is necessary to make default setting for variables of the MC_TP_REF type structure.

1 2 Procedure

1. When the input variable to TimePosition is declared, "Automatic Declaration" dialog box is displayed. Click ... displayed next to the "Initial Value" field.



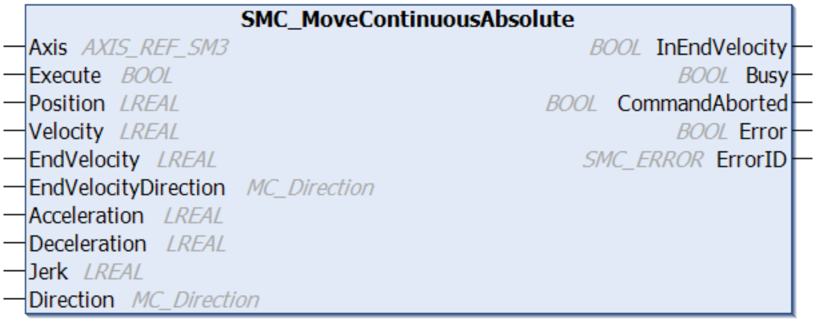
2. The "Initial Value" dialog box is displayed and, on the dialog box, you can set the default value for every member of the variable type (MC_TP_REF).



5.6.7 SMC_MoveContinuousAbsolute (Absolute Value Position Velocity Move)

This is a function block (FB) that executes absolute value positioning and, after the axis reaches the target position, causes the axis to keep moving at a specified velocity.

■ Icon



■ Parameter

Scope	Name	Type	Initial	Description
Input / output	Axis	AXIS_REF_SM3	-	Specifies the axis.
Input	Execute	BOOL	FALSE	Starts execution at the rising edge.
	Position	LREAL	0	Specifies the target position (u).
	Velocity	LREAL	0	Specifies the velocity (u/s) until the axis reaches the target position.
	EndVelocity	LREAL	0	Specifies the velocity (u/s) after the axis reaches the target position.
	EndVelocityDirection	MC_Direction	current	Specifies the traveling direction after the axis reaches the target. Specifies either “positive”, “negative”, or “current”. If “fastest” or “shortest” is specified, an error occurs.
	Acceleration	LREAL	0	Specifies the acceleration (u/s^2).
	Deceleration	LREAL	0	Specifies the deceleration (u/s^2).
	Jerk	LREAL	0	Specifies the jerk (u/s^3).
	Direction	MC_Direction	shortest	Specifies the traveling direction of the axis until the axis reaches the target position. Possible to specify only for the modulo type. For the finite axis, the specification is ignored.
Output	InEndVelocity	BOOL	FALSE	TRUE: The axis has reached the target position (Position).
	Busy	BOOL	FALSE	TRUE: Execution of the FB is not completed.

Scope	Name	Type	Initial	Description
	CommandAborted	BOOL	FALSE	TRUE: An interruption from other FB has occurred.
	Error	BOOL	FALSE	TRUE: An error has occurred within the FB.
	ErrorID	SMC_ERROR	0	An error ID is output.

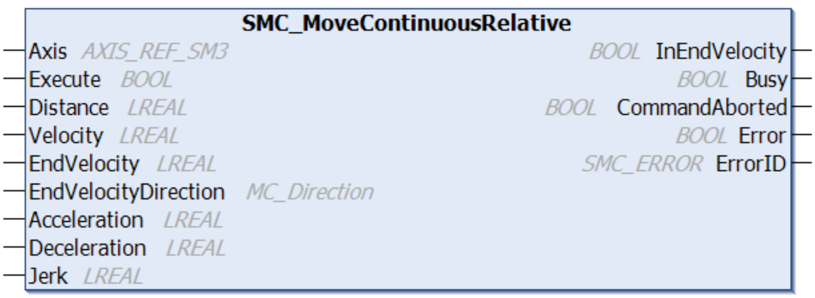
■ **MC_Direction (Enumeration type)**

Name	Value	Description
positive	1	Travels in the positive direction.
negative	-1	Travels in the negative direction.
shortest	0	Travels in the shortest direction from the current command position to the target command position when SMC_MoveContinuousAbsolute is executed.
fastest	3	Travels in the fastest direction from the current command position to the target command position when SMC_MoveContinuousAbsolute is executed. If the axis is being driven by another function block when SMC_MoveContinuousAbsolute is executed, selects the fastest direction within the GM1. If the axis is being stopped when SMC_MoveContinuousAbsolute is executed, makes the same movement as for the shortest.
current	2	Travels to the current direction. If the axis is being driven by another function block when SMC_MoveContinuousAbsolute is executed, travels in the same direction. If the axis is being stopped when SMC_MoveContinuousAbsolute is executed, travels in the direction moved by the previously executed function block.

5.6.8 SMC_MoveContinuousRelative (Relative Value Position Velocity Move)

This is a function block (FB) that executes relative value positioning and, after the axis reaches the target position, causes the axis to keep moving at a specified velocity.

■ **Icon**



■ Parameter

Scope	Name	Type	Initial	Description
Input / output	Axis	AXIS_REF_SM3	-	Specifies the axis.
Input	Execute	BOOL	FALSE	Starts execution at the rising edge.
	Distance	LREAL	0	Specifies the relative distance (u).
	Velocity	LREAL	0	Specifies the velocity (u/s).
	EndVelocity	LREAL	0	Specifies the velocity (u/s) after the axis travels the relative distance.
	EndVelocityDirection	MC_Direction	current	Specifies the traveling direction after the axis travels the relative distance. Specifies either “positive”, “negative”, or “current”. If “fastest” or “shortest” is specified, an error occurs.
	Acceleration	LREAL	0	Specifies the acceleration (u/s^2).
	Deceleration	LREAL	0	Specifies the deceleration (u/s^2).
	Jerk	LREAL	0	Specifies the jerk (u/s^3).
Output	InEndVelocity	BOOL	FALSE	TRUE: The axis has traveled the specified relative distance and has reached the specified velocity.
	Busy	BOOL	FALSE	TRUE: Execution of the FB is not completed.
	CommandAborted	BOOL	FALSE	TRUE: An interruption from other FB has occurred.
	Error	BOOL	FALSE	TRUE: An error has occurred within the FB.
	ErrorID	SMC_ERROR	0	An error ID is output.

■ MC_Direction (Enumeration type)

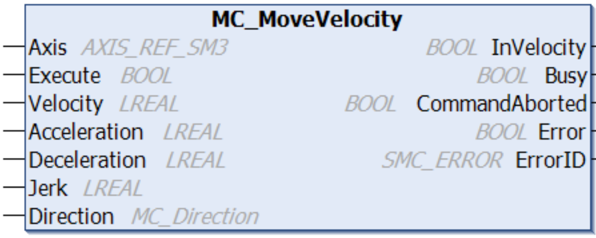
Name	Value	Description
positive	1	Travels in the positive direction.
negative	-1	Travels in the negative direction.
shortest	0	Not available. Do not specify this.
fastest	3	Not available. Do not specify this.
current	2	Travels to the current direction. Possible to use only for the modulo axis.

5.7 Velocity Control

5.7.1 MC_MoveVelocity (Velocity Control)

This is a function block (FB) that specifies the velocity of the axis. The axis keeps moving at the specified velocity and direction.

■ Icon



■ Parameter

Scope	Name	Type	Initial	Description
Input / output	Axis	AXIS_REF_SM3	-	Specifies the axis.
Input	Execute	BOOL	FALSE	Starts execution at the rising edge.
	Velocity	LREAL	0	Specifies the velocity (u/s).
	Acceleration	LREAL	0	Specifies the acceleration (u/s^2).
	Deceleration	LREAL	0	Specifies the deceleration (u/s^2).
	Jerk	LREAL	0	Specifies the jerk (u/s^3).
	Direction	MC_Direction	current	Specifies the traveling direction of the axis. “positive”, “negative”, or “current” (An error occurs when “fastest” or “shortest” is selected)
Output	InVelocity	BOOL	FALSE	TRUE: The axis has reached the specified velocity for the first time.
	Busy	BOOL	FALSE	TRUE: The FB is in operation.
	CommandAborted	BOOL	FALSE	TRUE: An interruption from other FB has occurred.
	Error	BOOL	FALSE	TRUE: An error has occurred.
	ErrorID	SMC_ERROR	0	An error ID is output.

■ MC_Direction (Enumeration type)

Name	Value	Description
positive	1	Travels in the positive direction.
negative	-1	Travels in the negative direction.

Name	Value	Description
shortest	0	Not available. Do not specify this.
fastest	3	Not available. Do not specify this.
current	2	Travels to the current direction. Possible to use only for the modulo axis.

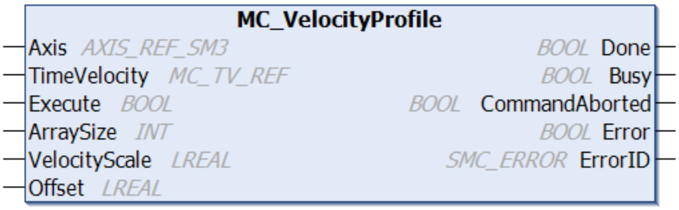
Info.

- To stop, set Velocity to 0 and execute again.
- Reference manual
  GM1 Controller RTEX User's Manual (Operation Edition)

5.7.2 MC_VelocityProfile (Velocity Profile Movement)

This is a function block (FB) that causes the axis to operate according to the profile data that consists of a combination of time and velocity.

■ Icon



■ Parameter

Scope	Name	Type	Initial	Description
Input / output	Axis	AXIS_REF_SM3	-	Specifies the axis.
	TimeVelocity	MC_TV_REF	-	Specifies the time / velocity profile.
Input	Execute	BOOL	FALSE	Starts execution at the rising edge.
	ArraySize	INT	0	A member of input TimeVelocity. Specifies the number of points to be executed by FB in the array of time and velocity that are specified by MC_TV_ARRAY.
	VelocityScale	LREAL	1	Velocity scaling
	Offset	LREAL	0	Velocity offset (u)
Output	Done	BOOL	FALSE	TRUE: The movements specified by the profile are completed.
	Busy	BOOL	FALSE	TRUE: Execution of the FB is not completed.
	CommandAborted	BOOL	FALSE	TRUE: An interruption from other FB has occurred.

Scope	Name	Type	Initial	Description
	Error	BOOL	FALSE	TRUE: An error has occurred within the FB.
	ErrorID	SMC_ERROR	0	An error ID is output.

■ MC_TV_REF (Structure)

Member	Type	Description
Number_of_pairs	INT	Not used
isAbsolute	BOOL	Methods of specifying the velocity of profile data TRUE: Specified in an absolute value. FALSE: Specified in a relative value.
MC_TV_Array	ARRAY [1..100] OF SMC_TV	Time and velocity profile data (1st point to 100th point)

■ SMC_TV (Structure)

Member	Type	Description
delta_time	TIME	Time of the profile data
velocity	LREAL	Velocity of profile data

Regarding the method for entering defaults for variables of the MC_TV_REF Type Structure, refer to “Default Setting for Variables of the MC_TP_REF Type Structure”.

Note

- While the axis keeps driving, do not specify delta_time to 0 ms.
  Otherwise, it may cause an abnormal operation.

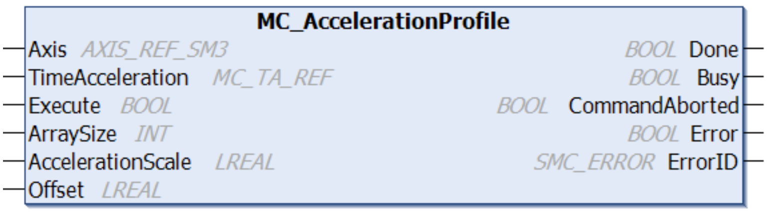
REFERENCE

5.6.6 Default Setting for Variables of the MC_TP_REF Type Structure

5.7.3 MC_AccelerationProfile (Acceleration Profile Movement)

This is a function block (FB) that causes the axis to operate according to the profile data that consists of a combination of time and acceleration.

■ Icon



■ Parameter

Scope	Name	Type	Initial	Description
Input / output	Axis	AXIS_REF_SM3	-	Specifies the axis.
	TimeAcceleration	MC_TA_REF	-	Specifies the time / acceleration profile.
Input	Execute	BOOL	FALSE	Starts execution at the rising edge.
	ArraySize	INT	0	A member of the input TimeAccleration. Specifies the number of points to be executed by FB in the array of time and acceleration that are specified by MC_TA_ARRAY.
	AccelerationScale	LREAL	1	Acceleration scaling
	Offset	LREAL	0	Acceleration offset (u)
Output	Done	BOOL	FALSE	TRUE: The movements specified by the profile are completed.
	Busy	BOOL	FALSE	TRUE: Execution of the FB is not completed.
	CommandAborted	BOOL	FALSE	TRUE: An interruption from other FB has occurred.
	Error	BOOL	FALSE	TRUE: An error has occurred within the FB.
	ErrorID	SMC_ERROR	0	An error ID is output.

■ MC_TA_REF (Structure)

Member	Type	Description
Number_of_pairs	INT	Not used
isAbsolute	BOOL	Methods of specifying the acceleration of profile data TRUE: Specified in an absolute value. FALSE: Specified in a relative value.
MC_TA_Array	ARRAY [1..100] OF SMC_TA	Time and acceleration profile data (1st point to 100th point)

■ SMC_TA (Structure)

Member	Type	Description
delta_time	TIME	Time of the profile data
Acceleration	LREAL	Acceleration of profile data

Regarding the method for entering defaults for variables of the MC_TA_REF Type Structure, refer to “Default Setting for Variables of the MC_TP_REF Type Structure”.

Note

- While the axis keeps driving, do not specify delta_time to 0 ms.
  Otherwise, it may cause an abnormal operation.

REFERENCE

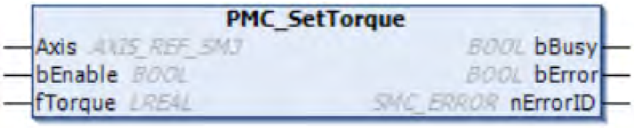
5.6.6 Default Setting for Variables of the MC_TP_REF Type Structure

5.8 Torque Control

5.8.1 PMC_SetTorque (Torque Control)

This is a function block (FB) that specifies the torque of the axis.

■ Icon



■ Parameter

Scope	Name	Type	Initial	Description
Input / output	Axis	AXIS_REF_SM3	-	Specifies the axis.
Input	bEnable	BOOL	FALSE	TRUE: The FB can be executed.
	fTorque	LREAL	0	Specifies the torque (%).
Output	bBusy	BOOL	FALSE	TRUE: Execution of the FB is not completed.
	bError	BOOL	FALSE	TRUE: An error has occurred within the FB.
	nErrorID	SMC_ERROR	0	An error ID is output.

Info.

- This function block can be executed only in the torque control mode (SMC_torque). Change to the torque control mode in advance using MC_SetControllerMode.
- When executing torque control using this function block, disable bit 0 (Two-degree-of-freedom control mode) of the MINAS amplifier parameter Pr6.47 (Function enhancement setting 2).
- When stopping the axis while the axis is being controlled in the torque control mode (SMC_torque), set fTorque to 0 and execute again. Or, change to the position control mode using SMC_SetControllerMode and then stop the axis using MC_Stop.
- Reference manual

  GM1 Controller RTEX User's Manual (Operation Edition)

REFERENCE

5.3.1 SMC_SetControllerMode (Control Mode Setting)
5.4.1 MC_Stop (Forced Stop)

6 Motion Control Function Blocks (Synchronous Control)

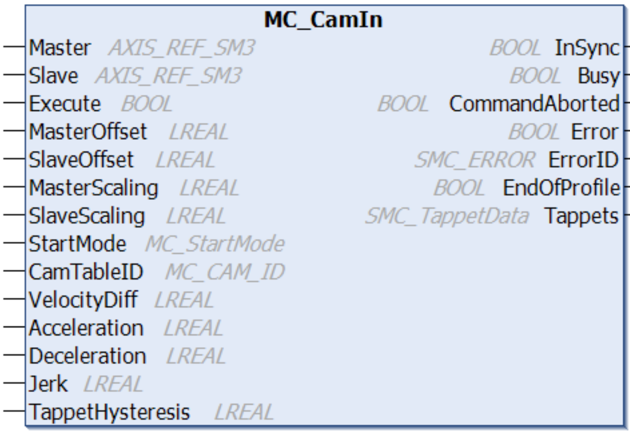
This section describes motion control function blocks to perform synchronous processing.

6.1 Cam Operation

6.1.1 MC_CamIn (Start Cam Operation)

This is a function block (FB) that starts cam synchronous operation. The master axis and the slave axis operate in synchronization according to the cam table.

■ Icon



■ Parameter

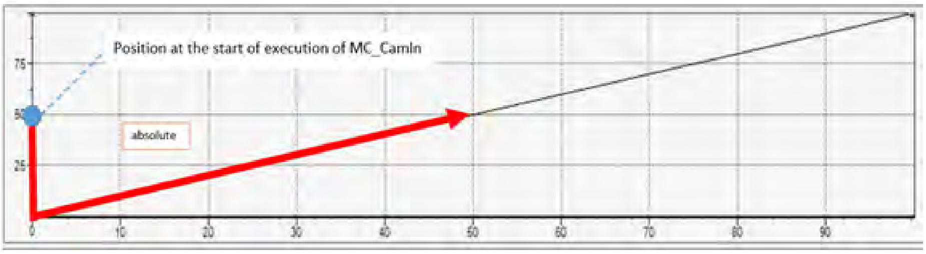
Scope	Name	Type	Initial	Comment
Input / output	Master	AXIS_REF_SM3	-	Specifies the master axis.
	Slave	AXIS_REF_SM3	-	Specifies the slave axis.
Input	Execute	BOOL	FALSE	Starts execution at the rising edge.
	MasterOffset	LREAL	0	Master axis table offset
	SlaveOffset	LREAL	0	Slave axis table offset
	MasterScaling	LREAL	1	Master axis profile scaling factor
	SlaveSacling	LREAL	1	Slave axis profile scaling factor
	StartMode	MC_StartMode	absolute	Start mode
	CamTableID	MC_CAM_ID	0	Cam table ID Specifies the output for MC_CamTableSelect.
	VelocityDiff	LREAL	0	Specifies the maximum velocity difference (u/s) when StartMode is set to ramp_in.
	Acceleration	LREAL	0	Specifies the acceleration (u/s^2) when StartMode is set to ramp_in.
	Deceleration	LREAL	0	Specifies the deceleration (u/s^2) when StartMode is set to ramp_in.

Scope	Name	Type	Initial	Comment
	Jerk	LREAL	0	Specifies jerk (u/s^3) when StartMode is set to ramp_in.
	TappetHysteresis	LREAL	0	Specifies the hysteresis value (position) of the tappet.
Output	InSync	BOOL	FALSE	TRUE: The cam is in synchronization for the first time.
	Busy	BOOL	FALSE	TRUE: Execution of the FB is not completed.
	CommandAborted	BOOL	FALSE	TRUE: An interruption from other FB has occurred.
	Error	BOOL	FALSE	TRUE: An error has occurred within the FB.
	ErrorID	SMC_ERROR	0	An error ID is output.
	EndOfProfile	BOOL	FALSE	A pulse is output every time the cam profile period of the slave axis ends.
	Tappets	SMC_TAPPETDATA		Output tapped data Used as the input for SMC_GetTappetValue.

■ MC_StartMode (Enumeration type)

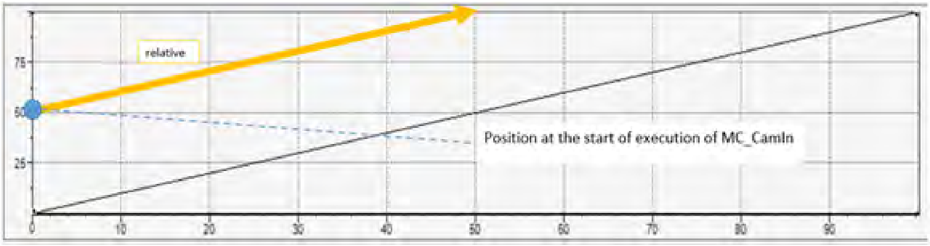
- absolute

  If the slave axis position is offset from the position on the cam table when MC_CamIn is executed, the position jumps. When starting a new cycle, the cam does not consider the current slave axis position. Therefore, if the slave axis position relative to the mater axis at the start point is offset from the slave axis position relative to the mater axis at the end point, the slave axis position may jump.



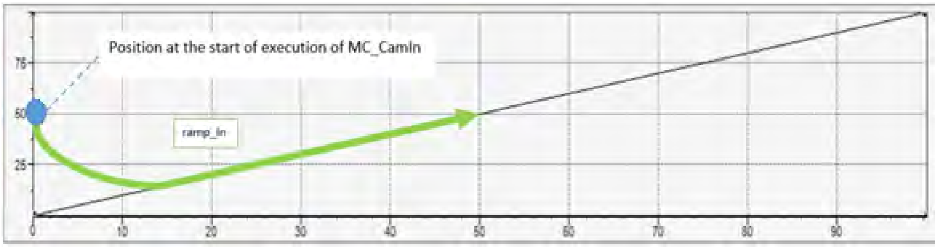
- relative

  If the slave axis position is offset from the position on the cam table when MC_CamIn is executed, the position jumps. In consideration of the current slave axis position, new cam operation is started. The slave axis position at the time when the previous cycle is completed is reflected as SlaveOffset on the cam operation. It must be noted, however, that the position may jump if the master axis position is not 0 at the start position.



- ramp_in, ramp_in_neg, ramp_in_pos

  When MC_CamIn is executed, acceleration and deceleration is performed according to VelocityDiff, Acceleration, and Deceleration to reach the synchronized state (InSync). If the slave axis is a modulo axis, correction is made only in the positive direction when the mode is set to ramp_in_pos, while correction is made only in the negative direction when the mode is set to ramp_in_neg. With the finite axis, ramp_in_pos and ramp_in_neg are treated as ramp_in.



The final StartMode is determined by MC_CamIn.StartMode and MC_CamTableSelect.SlaveAbsolute.

MC_CamIn.StartMode	MC_CamTableSelect.SlaveAbsolute	StartMode
absolute	TRUE	absolute
absolute	FALSE	relative
relative	TRUE	relative
relative	FALSE	relative
ramp_in	TRUE	ramp_in absolute
ramp_in	FALSE	ramp_in relative
ramp_in_pos	TRUE	ramp_in_pos absolute
ramp_in_pos	FALSE	ramp_in_pos relative
ramp_in_neg	TRUE	ramp_in_neg absolute
ramp_in_neg	FALSE	ramp_in_neg relative

■ Changing the scale and offset in the cam table (MasterOffset, SlaveOffset, MasterScaling, SlaveSacling)

A cam table can be created by using the Cam Table Editor on the GM Programmer. The cam table created by using the Cam Table Editor determines the relationship between the master axis position and the slave axis position that operate on the POU,

SlavePosition = CAM(MasterPosition)

The scale and offset in the cam table can be changed on the POU by setting the MC_CamIn parameters including MasterOffset, SlaveOffset, sterScaling, and SlaveSacling. At that time, the slave axis position is determined as follows according to the setting of each parameter.

SlavePosition = SlaveSacling * CAM(MasterPosition * MasterScaling + MasterOffset) + SlaveOffset

■ Tappet

For information on setting the tappet and TappetHysteresis of MC_CamIn, refer to the description on SMC_GetTappetValue

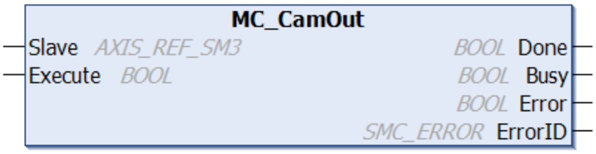
■ SMC_TAPPETDATA (Structure)

Member	Type	Description
pTaps	ARRAY [0..2] OF POINTER TO SMC_CAMTappet	Used internally for the output of MC_CamIn and for the input of SMC_GetTappetValue.
dwCycleTime	DWORD	
byChannels	BYTE	
bRestart	BOOL	

6.1.2 MC_CamOut (Cancel Cam Operation)

This is a function block (FB) that cancels synchronous operation of the cam. Synchronized operation between the master axis and slave axis is canceled.

■ Icon



■ Parameter

Scope	Name	Type	Initial	Description
Input / output	Slave	AXIS_REF_SM3	-	Specifies the slave axis to be released.
Input	Execute	BOOL	FALSE	Starts execution at the rising edge.
Output	Done	BOOL	FALSE	TRUE: Synchronization cancellation is completed.
	Busy	BOOL	FALSE	TRUE: Execution of the FB is not completed.
	Error	BOOL	FALSE	TRUE: An error has occurred within the FB.
	ErrorID	SMC_ERROR	0	An error ID is output.

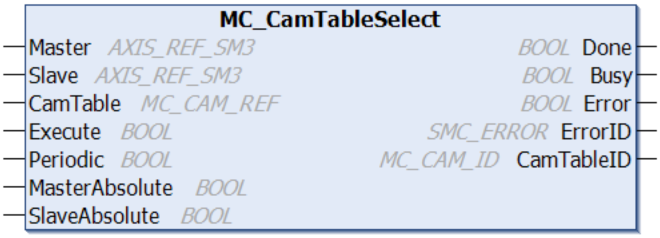
Info.

- The slave axis operation continues even after the cam operation is canceled. Execute MC_Halt or MC_Stop to stop the slave axis.
- Reference manual
  GM1 Controller RTEX User's Manual (Operation Edition)

6.1.3 MC_CamTableSelect (Select Cam Table)

This is a function block (FB) that specifies the cam table to be used for synchronous operation of the cam. When the cam table to be used for synchronized operation between the master axis and slave axis is selected, a cam table ID is output.

■ Icon



■ Parameter

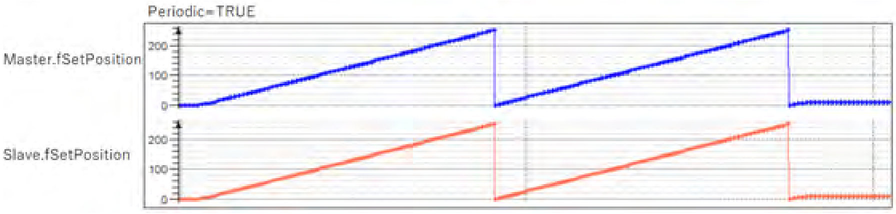
Scope	Name	Type	Initial	Description
Input / output	Master	AXIS_REF_SM3	-	Specifies the master axis.
	Slave	AXIS_REF_SM3	-	Specifies the slave axis.
	CamTable	MC_CAM_REF	-	Specifies the cam table.
Input	Execute	BOOL	FALSE	Starts execution at the rising edge.
	Periodic	BOOL	TRUE	Movement of the slave axis TRUE: Repeat execution FALSE: 1-period execution
	MasterAbsolute	BOOL	TRUE	TRUE: Absolute position of the master axis FALSE: Relative position of the master axis
	SlaveAbsolute	BOOL	TRUE	TRUE: Absolute position of the slave axis FALSE: Relative position of the slave axis
Output	Done	BOOL	FALSE	TRUE: Selection is completed.
	Busy	BOOL	FALSE	TRUE: Execution of the FB is not completed.
	Error	BOOL	FALSE	TRUE: An error has occurred within the FB.
	ErrorID	SMC_ERROR	0	An error ID is output.

Scope	Name	Type	Initial	Description
	CamTableID	MC_CAM_ID		Cam table ID Used as the input for MC_CamIn.

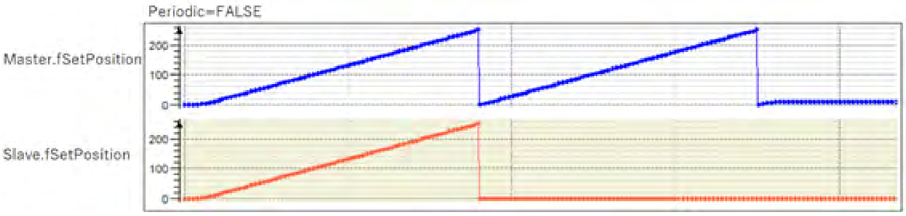
■ Periodic (Periodic cam control)

If Periodic of MC_CamTableSelect is set to TRUE, cam operation is repeatedly performed. The cam is automatically restarted when reaching the end position. If Periodic is set to FALSE, when the master axis reaches the end position, EndOfProfile of MC_CamIn changes to TRUE and the slave axis stops at the current position. When the master axis enters the cam position range again, the slave axis starts moving according to the cam table.

If periodic = TRUE



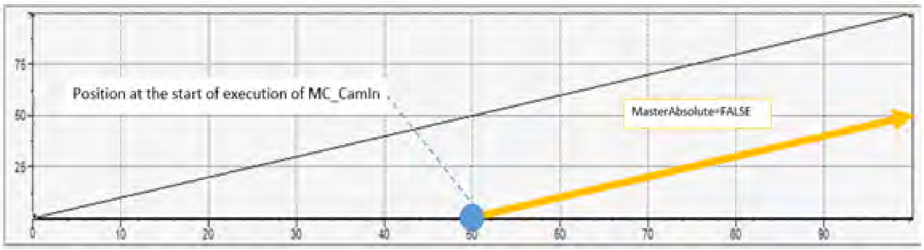
If periodic = FALSE



■ MasterAbsolute

If MasterAbsolute is set to TRUE, the cam starts from the current master axis position. Any position within the cam table range specified for the master axis can be set as the starting position. If the starting position is outside the cam table range, an error occurs.

If MasterAbsolute is set to FALSE, the cam is relocated to the current master axis position. The zero point of the master axis in the cam table is relocated to the current master axis position. This mode is allowed only when the value 0 is within the master axis range.



■ **SlaveAbsolute**

Refer to MC_CamIn.

Info.

- Reference manual
  GM1 Controller RTEX User's Manual (Operation Edition)

6.1.4 SMC_GetTappetValue (Tappet Output)

This is a function block (FB) that outputs the tappet data set using the cam table. Specify the output Tappets of the MC_CamIn function block as the input for this FB and obtain the tappet output for one track.

■ **Icon**

■ **Parameter**

Scope	Name	Type	Initial	Description
Input / output	Tappets	SMC_TAPPETDATA	-	Specifies the output Tappets of MC_CamIn.
Input	iID	INT	0	Track ID of the tappet
	bInitValue	BOOL	FALSE	Default of the tappet
	bSetInitValueAtReset	BOOL	FALSE	TRUE: The tappet output is Initialized when MC_CamIn is executed. FALSE: The tappet output is held when MC_CamIn is executed.
Output	bTappet	BOOL	FALSE	Tappet output

■ SMC_TAPPETDATA (Structure)

Member	Type	Description
pTaps	ARRAY [0..2] OF POINTER TO SMC_CAMTappet	Used internally for the output of MC_CamIn and for the input of SMC_GetTappetValue.
dwCycleTime	DWORD	
byChannels	BYTE	
bRestart	BOOL	

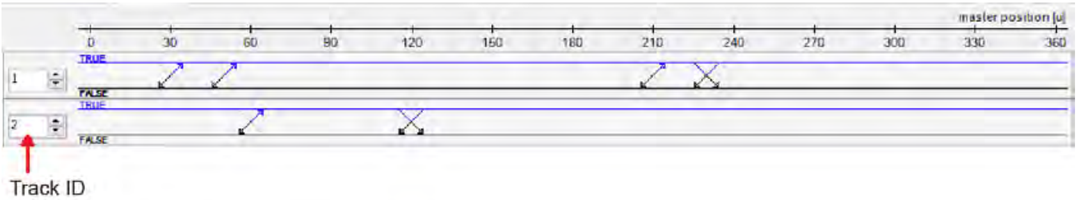
■ Tappet settings

Set tappet settings on the "Tappet" tab or "Tappet Table" tab window.

The settings that are set on these two tab windows are linked to each other. They can be set from either window.

Set switches required for the tracks (100 max).
The following is an example where three switches are set for two tracks.

"Tappet"



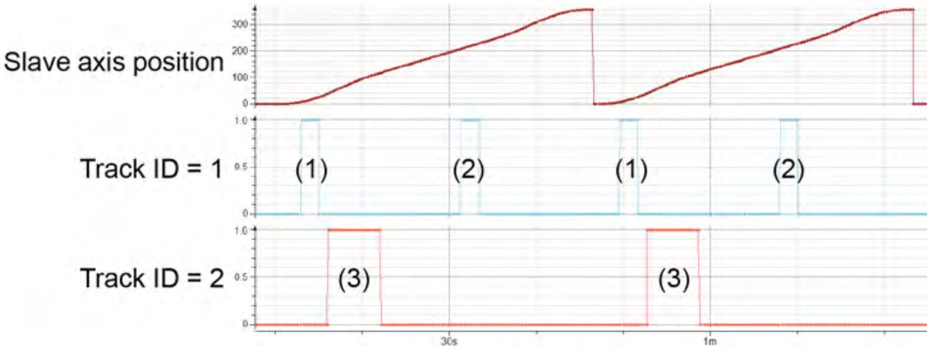
"Tappet Table"

	Track ID	X	positive pass	negative pass	
	1				
		30	switch ON	switch OFF	Switch (1)
		50	switch ON	switch OFF	
		210	switch ON	switch OFF	Switch (2)
		230	switch OFF	switch OFF	
	2				
		60	switch ON	switch OFF	Switch (3)
		120	switch OFF	switch OFF	

■ Operations when the function block is executed

The operations that take place when, after setting the tappet, the SMC_GetTappetValue function block is executed and the tappet (bTappet) is output. The axis is set to the modulo (modulo value: 360).

Trace
(1) to (3) are switch numbers.



■ TappetHysteresis of MC_CamIn

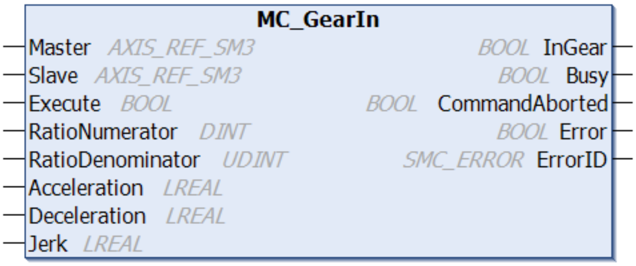
By setting TappetHysteresis, mechanical chattering can be filtered. When the cam reaches the tappet position, tappet processing is performed. Once the cam moves out of the tappet position range set in TappetHysteresis, tappet processing is not performed unless the cam reaches the tappet position again.

6.2 Gear Operation

6.2.1 MC_GearIn (Start Gear Operation)

This is a function block (FB) that starts synchronous operation of the gears. Specify the gear ratio between the master axis and slave axis and start moving the gears for synchronous operation of the gears.

■ Icon



■ Parameter

Scope	Name	Type	Initial	Description
Input / output	Master	AXIS_REF_SM3	-	Specifies the master axis.
	Slave	AXIS_REF_SM3	-	Specifies the slave axis.
Input	Execute	BOOL	FALSE	Starts execution at the rising edge.
	RatioNumerator	DINT	1	Specifies the gear ratio (numerator).
	RatioDenominator	UDINT	1	Specifies the gear ratio (denominator).
	Acceleration	LREAL	0	Maximum acceleration (u/s^2) until gear synchronization is completed
	Deceleration	LREAL	0	Maximum deceleration (u/s^2) until gear synchronization is completed
	Jerk	LREAL	0	Maximum jerk (u/s^3) until gear synchronization is completed
Output	InGear	BOOL	FALSE	TRUE: Gear synchronization is completed.
	Busy	BOOL	FALSE	TRUE: The FB is in operation.
	CommandAborted	BOOL	FALSE	TRUE: An interruption from other FB has occurred.
	Error	BOOL	FALSE	TRUE: An error has occurred within the FB.
	ErrorID	SMC_ERROR	0	An error ID is output.

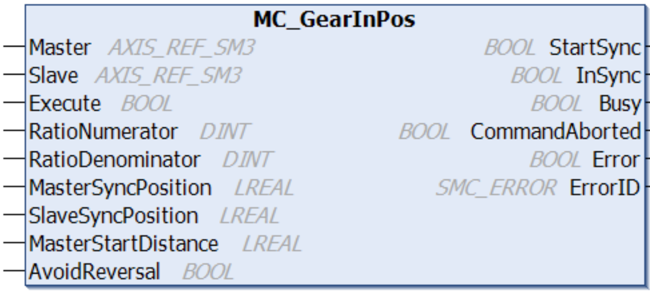
Info.

- Reference manual
  GM1 Controller RTEX User's Manual (Operation Edition)

6.2.2 MC_GearInPos (Position Specified Gear Operation)

This is a function block (FB) that starts synchronous operation of the gears from the specified absolute position.

■ Icon



■ Parameter

Scope	Name	Type	Initial	Description
Input / output	Master	AXIS_REF_SM3	-	Specifies the master axis.
	Slave	AXIS_REF_SM3	-	Specifies the slave axis.
Input	Execute	BOOL	FALSE	Starts execution at the rising edge.
	RatioNumerator	DINT	1	Specifies the gear ratio (numerator).
	RatioDenominator	DINT	1	Specifies the gear ratio (denominator).
	MasterSyncPosition	LREAL	0	Master axis position to start synchronization
	SlaveSyncPosition	LREAL	0	Slave axis position to start synchronization
	MasterStartDistance	LREAL	0	When the master axis moves forward from the MasterSyncPosition position for the distance specified by MasterStartDistance, the slave axis starts moving for the synchronized operation with the master axis. If MasterStartDistance is zero, the slave axis immediately starts moving for synchronization.
	AvoidReversal	BOOL	FALSE	FALSE: Reverse rotation of the slave axis is possible.

Scope	Name	Type	Initial	Description
				TRUE: Reverse rotation of the slave axis is impossible.
Output	StartSync	BOOL	FALSE	TRUE: Gear synchronization is started.
	InSync	BOOL	FALSE	TRUE: Gear synchronization is completed.
	Busy	BOOL	FALSE	TRUE: The FB is in operation.
	CommandAborted	BOOL	FALSE	TRUE: An interruption from other FB has occurred.
	Error	BOOL	FALSE	TRUE: An error has occurred within the FB.
	ErrorID	SMC_ERROR	0	An error ID is output.

■ Operations when the function block is executed

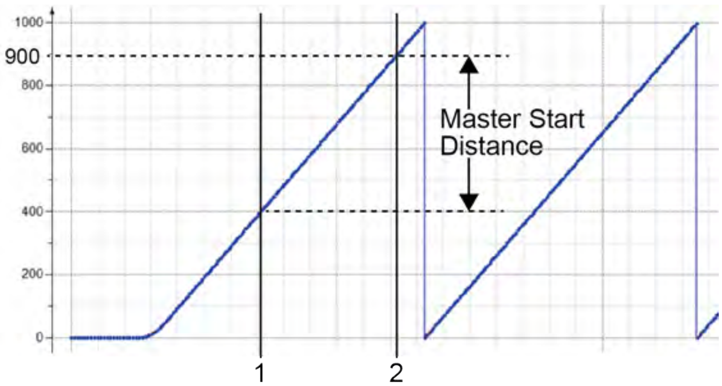
This example shows the trace when the MC_GearInPos function block is executed with the following conditions.

Execution condition

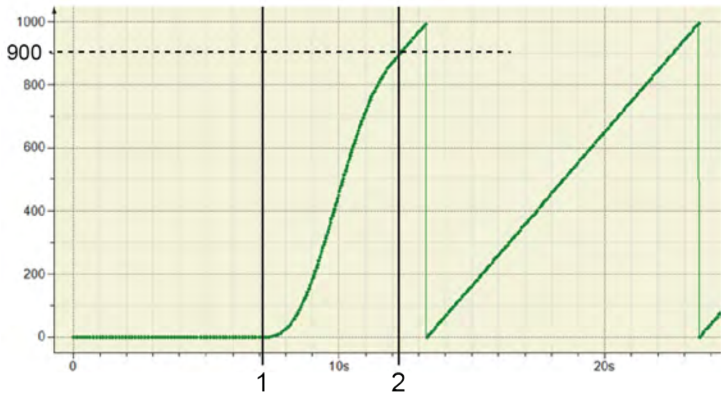
Item	Dis
Master axis type	Modulo (modulo value = 1000)
Slave axis type	Modulo (modulo value = 1000)
Gear ratio	1 : 1
Input MasterSyncPostion	900
Input SlaveSyncPositon	900
Input MasterStartDistance	500

When the master axis position reaches 900 and the salve axis position reaches 900, the master axis starts to synchronize with the slave axis. When the master axis passes the position 400, which is obtained by deducting 500 (MasterStartDistane) from 900 (synchronization start position of the master axis), the slave axis starts traveling to synchronize with the master axis. At this time, velocity, acceleration, and deceleration are automatically determined.

Position of the master axis



Position of the slave axis



■ AvoidReversal

By setting AvoidReversal, the slave axis can be restricted on reverse rotation. If AvoidReversal is set to TRUE, an error occurs under the following conditions.

1. Gear ratio is negative.

   If the gear ratio is negative (for example, RatioNumerator = -1, RatioDenominator = 1), when the axis reaches the position set in GearInPos.StartSync while the slave axis is operating in forward rotation, an error (SMC_GIP_SLAVE_REVERSAL_CANNOT_BE_AVOIDED) occurs

2. The slave axis is rotating in reverse to the rotation of the master axis before the start of synchronization

   When the axis reaches the gear synchronization start position set in StartSync while the slave axis is operating in reverse rotation, an error (SMC_GIP_SLAVE_REVERSAL_CANNOT_BE_AVOIDED) occurs

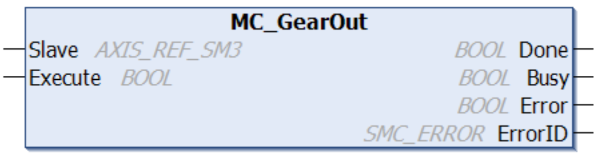
3. Correction of the slave axis is not completed within five cycles.

   Gear synchronization completion (InSync) is not achieved within five cycles after reaching the gear synchronization start (StartSync), an error occurs.

6.2.3 MC_GearOut (Cancel Gear Operation)

This is a function block (FB) that cancels synchronous operation of the gears. Synchronized gear operation between the master axis and slave axis is canceled.

■ Icon



■ Parameter

Scope	Name	Type	Initial	Description
Input / output	Slave	AXIS_REF_SM3	-	Specifies the axis.
Input	Execute	BOOL	FALSE	Starts execution at the rising edge.
Output	Done	BOOL	FALSE	TRUE: Synchronization cancellation is completed.
	Busy	BOOL	FALSE	TRUE: Execution of the FB is not completed.
	Error	BOOL	FALSE	TRUE: An error has occurred within the FB.
	ErrorID	SMC_ERROR	0	An error ID is output.

Info.

- The slave axis operation continues even after the gear operation is canceled. Execute MC_Halt or MC_Stop to stop the slave axis.
- Reference manual

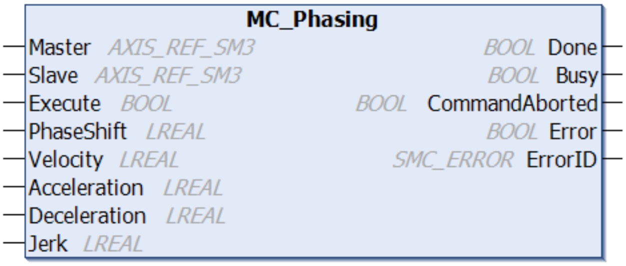
  GM1 Controller RTEX User's Manual (Operation Edition)

6.3 Phase Correction

6.3.1 MC_Phasing (Master Axis Phase Correction)

This is a function block (FB) that performs phase correction between the master axis and slave axis. Phase synchronous operation can be performed by making phase correction for the master axis.

■ Icon



■ Parameter

Scope	Name	Type	Initial	Description
Input / output	Master	AXIS_REF_SM3	-	Specifies the master axis.
	Slave	AXIS_REF_SM3	-	Specifies the slave axis.
Input	Execute	BOOL	FALSE	Starts execution at the rising edge.
	PhaseShift	LREAL	0	Specifies the phase between the master axis and slave axis.
	Velocity	LREAL	0	Specifies the velocity (u/s).
	Acceleration	LREAL	0	Specifies the acceleration (u/s^2).
	Deceleration	LREAL	0	Specifies the deceleration (u/s^2).
	Jerk	LREAL	0	Specifies the jerk (u/s^3).
Output	Done	BOOL	FALSE	TRUE: Phase correction is completed.
	Busy	BOOL	FALSE	TRUE: The FB is in operation.
	CommandAborted	BOOL	FALSE	TRUE: An interruption from other FB has occurred.
	Error	BOOL	FALSE	TRUE: An error has occurred within the FB.
	ErrorID	SMC_ERROR	0	An error ID is output.

Info.

- Reference manual
  GM1 Controller RTEX User's Manual (Operation Edition)

7 Motion Control Function Blocks (Multi-axis Control)

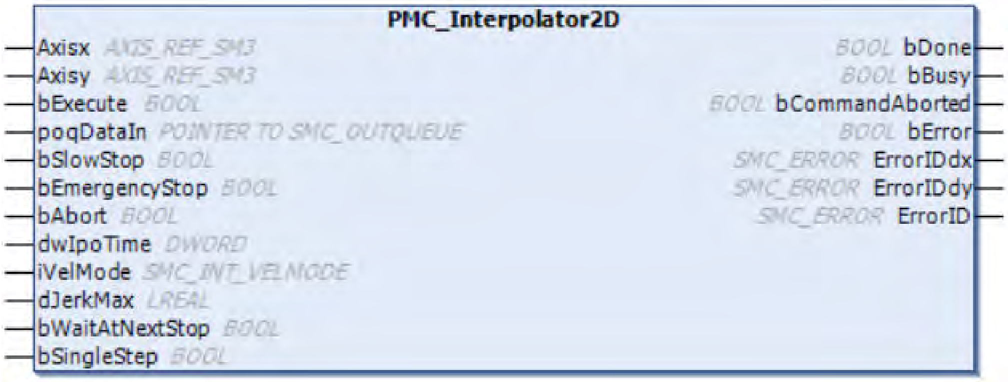
This section describes function blocks used to perform multi-axis control using the CNC program.

7.1 Interpolation Control

7.1.1 PMC_Interpolator2D (Two-dimensional Interpolation Control)

This function block (FB) performs 2-axis interpolation control according to the specified CNC table.

■ **Icon**



■ **Parameter**

Scope	Name	Type	Default value	Description
I/O	Axisx	AXIS_REF_SM3	-	Specifies the x-axis.
	Axisy	AXIS_REF_SM3	-	Specifies the y-axis.
Input	bExecute	BOOL	FALSE	Starts execution at the rising edge.
	poqDataIn	POINTER TO SMC_OUTQUEUE	-	Specifies a pointer to the CNC table.
	bSlowStop	BOOL	FALSE	TRUE: A pause is executed. Deceleration stop is executed according to the velocity profile (iVelMode). FALSE: The pause is canceled.
	bEmergencyStop	BOOL	FALSE	TRUE: An emergency stop is executed. FALSE: The emergency stop is canceled.
	bAbort	BOOL	FALSE	TRUE: Execution of the FB is stopped.
	dwIpoTime	DWORD	0	MotionTask interval (μsec)
	iVelMode	SMC_INT_VELMODE	TRAPEZOID	Specifies a velocity profile.
	dJerkMax	LREAL	LREAL	Specifies the maximum value of jerk.

Scope	Name	Type	Default value	Description
				This parameter must be specified when QUADRATIC is selected for the velocity profile (iVelMode).
	bWaitAtNextStop	BOOL	BOOL	TRUE: A pause is executed in the table where the velocity between paths becomes zero. The conditions that cause the velocity between paths to become zero are set in bSingleStep or dAngleMode. FALSE: The pause is canceled.
	bSingleStep	BOOL	BOOL	TRUE: All connections between paths are established through deceleration stop.
Output	bCommandAborted	BOOL	FALSE	TRUE: An interruption is caused by another FB.
	bBusy	BOOL	-	TRUE: Execution of the FB is not completed.
	bDone	BOOL	FALSE	TRUE: Output is completed.
	bError	BOOL	FALSE	TRUE: An error has occurred within the FB.
	ErrorIDdx	SMC_ERROR	SMC_NO_ERROR	Error ID output during x-axis movement processing
	ErrorIDdy	SMC_ERROR	SMC_NO_ERROR	Error ID output during y-axis movement processing
	ErrorID	SMC_ERROR	SMC_NO_ERROR	Error ID output during interpolation control operation

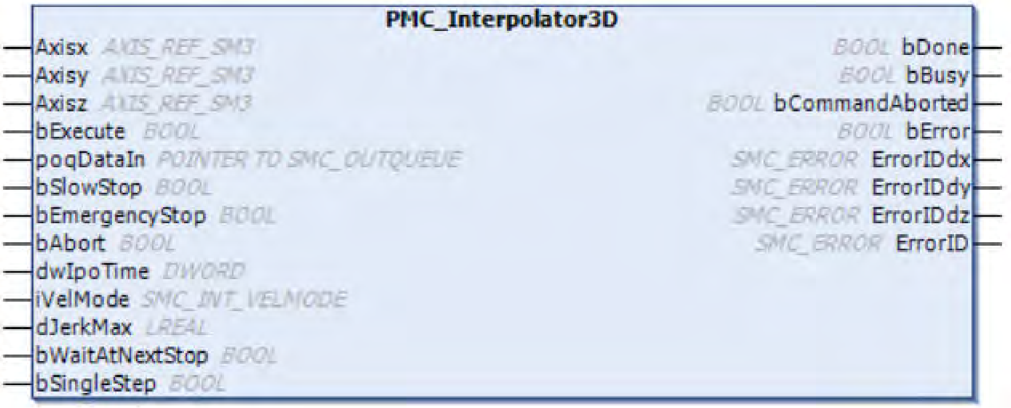
SMC_INT_VELMODE (Enumeration type)

Name	Value	Description
TRAPEZOID	0	Trapezoid
SIGMOID	1	Sin2
SIGMOID_LIMIT	2	Sin2 (limit)
QUADRATIC	3	Quadratic
QUADRATIC_SMOOTH	4	Quadratic (smooth)

7.1.2 PMC_Interpolator3D (Three-dimensional Interpolation Control)

This function block (FB) performs 3-axis interpolation control according to the specified CNC table.

■ Icon



■ Parameter

Scope	Name	Type	Default value	Description
I/O	Axisx	AXIS_REF_SM3	-	Specifies the x-axis.
	Axisy	AXIS_REF_SM3	-	Specifies the y-axis.
	Axisz	AXIS_REF_SM3	-	Specifies the z-axis.
Input	bExecute	BOOL	FALSE	Starts execution at the rising edge.
	poqDataIn	POINTER TO SMC_OUTQUEUE	-	Specifies a pointer to the CNC table.
	bSlowStop	BOOL	FALSE	TRUE: A pause is executed. Deceleration stop is executed according to the velocity profile (iVelMode). FALSE: The pause is canceled.
	bEmergencyStop	BOOL	FALSE	TRUE: An emergency stop is executed. FALSE: The emergency stop is canceled.
	bAbort	BOOL	FALSE	TRUE: Execution of the FB is stopped.
	dwIpoTime	DWORD	0	MotionTask interval (μsec)
	iVelMode	SMC_INT_VELMODE	TRAPEZOID	Specifies a velocity profile.
	dJerkMax	LREAL	LREAL	Specifies the maximum value of jerk. This parameter must be specified when QUADRATIC is selected for the velocity profile (iVelMode).
	bWaitAtNextStop	BOOL	BOOL	TRUE: A pause is executed in the table where the velocity between paths becomes zero. The conditions that cause the velocity between paths to become

Scope	Name	Type	Default value	Description
				zero are set in bSingleStep or dAngleMode. FALSE: The pause is canceled.
	bSingleStep	BOOL	BOOL	TRUE: All connections between paths are established through deceleration stop.
Output	bCommandAborted	BOOL	FALSE	TRUE: An interruption is caused by another FB.
	bBusy	BOOL	-	TRUE: Execution of the FB is not completed.
	bDone	BOOL	FALSE	TRUE: Output is completed.
	bError	BOOL	FALSE	TRUE: An error has occurred within the FB.
	ErrorIDdx	X-axis error ID	SMC_NO_ERROR	Error ID output during x-axis movement processing
	ErrorIDdy	Y-axis error ID	SMC_NO_ERROR	Error ID output during y-axis movement processing
	ErrorIDdz	Z-axis error ID	SMC_NO_ERROR	Error ID output during z-axis movement processing
	ErrorID	SMC_ERROR	SMC_NO_ERROR	Error ID output during interpolation control operation

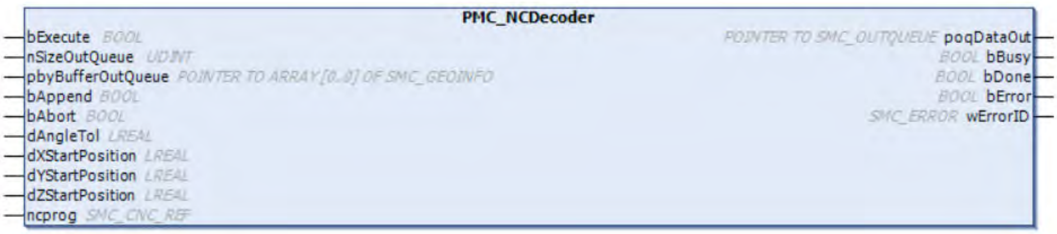
SMC_INT_VELMODE (Enumeration type)

Name	Value	Description
TRAPEZOID	0	Trapezoid
SIGMOID	1	Sin2
SIGMOID_LIMIT	2	Sin2 (limit)
QUADRATIC	3	Quadratic
QUADRATIC_SMOOTH	4	Quadratic (smooth)

7.1.3 PMC_NCDecoder (CNC Table Conversion)

This function block (FB) decodes the specified SMC_CNC_REF value to SMC_OUTQUEUE.

■ **Icon**



■ Parameter

Scope	Name	Type	Default value	Description
I/O	ncprogIn	SMC_CNC_REF	-	Specifies the SMC_CNC_REF value to be decoded.
Input	bExecute	BOOL	FALSE	Starts execution at the rising edge.
	nSizeOutQueue	UDINT	-	Specifies a buffer size. We recommend that a buffer be created and the sizeof operator be specified as shown in the following example. ExampleBuf: ARRAY [0..50] OF SMC_GEOINFO; nSizeOutQueue:=sizeof(ExampleBuf)
	pbyBufferOutQueue	POINTER TO ARRAY [0..0] OF SMC_GEOINFO	-	Specifies the memory space for SMC_OUTUEUE. We recommend that array SMC_GEOINFO be defined and an address be specified as shown in the following example. ExampleBuf: ARRAY [0..50] OF SMC_GEOINFO; (Buffer that can store 50 path elements) pbyBufferOutQueue:=ADR(ExampleBuf)
	dXstartPosition	LREAL	0	Specifies the position of the x-axis at the start of movement(Note 1).
	dYstartPosition	LREAL	0	Specifies the position of the y-axis at the start of movement(Note 1).
	dZstartPosition	LREAL	0	Specifies the position of the z-axis at the start of movement(Note 1).
	bAppend	BOOL	FALSE	TRUE: Decoded data of ncprogIn is appended to the end of poqDataOut without resetting the poqDataOut data within the FB at the rising edge as specified by bExecute.
	bAbort	BOOL	FALSE	TRUE: Execution of the FB is stopped.
Output	poqDataOut	POINTER TO SMC_OUTQUEUE	-	Pointer to SMC_OUTQUEUE which manages decoded SMC_GEOINFO objects
	Busy	BOOL	FALSE	TRUE: Execution of the FB is not completed.
	Done	BOOL	FALSE	TRUE: Output is completed.
	Error	BOOL	FALSE	TRUE: An error has occurred within the FB.
	ErrorID	SMC_ERROR	0	Error ID output

(Note 1) We recommend that fSetPosition be entered. If the entered value and the actual position differ, there is a risk that the axis may move suddenly.

8 Motion Control Function Blocks (Motion Communication Control)

This section describes function blocks used to perform communication control.

8.1 RTEX

8.1.1 Types of Data To Be Handled by AMP Function Blocks

Item	Description	Related function blocks
AMP alarm	This is an AMP alarm that occurs in AMP operation.	RTEX_ClearAmpAlarm RTEX_ReadAmpAlarm RTEX_ReadAmpState
AMP warning	This is an AMP warning that occurs in AMP operation. This occurs before the AMP alarm. If the situation worsens, an AMP alarm occurs.	
Monitor data	This is monitor data (position deviation, load percentage, etc.) of the RTEX communication data.	RTEX_ReadAmpData
AMP parameter	This is configuration data of the AMP device itself.	RTEX_ReadAmpParameter RTEX_WriteAmpParameter
Multi-turn data	There are two types of data in the data read by the absolute encoder (23 bit/r): one type is single-turn data that indicates the position within one motor rotation and the other is multi-turn data that counts one for one turn.	RTEX_ClearAmpMultiTurnData
Deviation counter	This is a processing part in the AMP that receives move commands to the AMP. The motor moves according to the commands accumulated in the deviation counter. The commands used for the motor movement are deleted from the deviation counter. The amount of commands accumulated in the deviation counter is called the position deviation.	RTEX_ClearAmpPositionalDeviation
Limit switch	This data is collected to monitor the POT and NOT states of the AMP.	RTEX_ReadNot, RTEX_ReadPot

8.1.2 RTEX_ClearAmpAlarm (Clear Amplifier Alarm)

This is a function block (FB) that clears the AMP alarm. It deletes the alarm or warning that has occurred in the AMP.

■ Icon

■ Parameter

Scope	Name	Type	Initial	Description
Input / Output	Axis	AXIS_REF_SM3	-	Specifies the axis.
Input	Execute	BOOL	FALSE	TRUE: Starts execution at the rising edge. FALSE: Stops processing.
Output	CommandAborted	BOOL	FALSE	TRUE: An interruption from other FB has occurred.
	Busy	BOOL	FALSE	TRUE: Execution of the FB is not completed.
	Done	BOOL	FALSE	TRUE: Clearing is completed.
	Error	BOOL	FALSE	TRUE: An error has occurred within the FB.
	ErrorID	SMC_ERROR	0	An error ID is output.
	Alarm	ALARM_CODE	-	A deleted alarm code is output.
	Warning	WARNING_CODE	-	A deleted warning code is output.

■ ALARM_CODE (Union)

Member	Type	Description
uiAlarmCode	UINT	Alarm code
tAlarmCodeMember	ALARM_WARNING_CODES	Main alarm code and sub alarm code

■ WARNING_CODE (Union)

Member	Type	Description
uiWarningCode	UINT	Warning code
tWarningCodeMember	ALARM_WARNING_CODES	Main warning code (warning number) and sub warning code (0)

■ ALARM_WARNING_CODES (Structure)

Member	Type	Description
byMainCode	BYTE	Main code
bySubCode	BYTE	Sub code

REFERENCE

11.2.1 RTEX Error ID
11.2.2 Alarm Codes
11.2.3 Warning Codes

8.1.3 RTEX_ReadAmpAlarm (Read Amplifier Alarm)

This is a function block (FB) that reads the AMP alarm. It reads the information of the alarm or warning that has occurred in the AMP.

■ Icon

■ Parameter

Scope	Name	Type	Initial	Description
Input / output	Axis	AXIS_REF_SM3	-	Specifies the axis.
Input	Execute	BOOL	FALSE	TRUE: Starts execution at the rising edge. FALSE: Stops processing.
	AlarmIndex	UINT	-	Specifies the history number (0 to 14). 0 is given for the latest history.
Output	CommandAborted	BOOL	FALSE	TRUE: An interruption from other FB has occurred.
	Busy	BOOL	FALSE	TRUE: Execution of the FB is not completed.
	Done	BOOL	FALSE	TRUE: Reading is completed.
	Error	BOOL	FALSE	TRUE: An error has occurred within the FB.

Scope	Name	Type	Initial	Description
	ErrorID	SMC_ERROR	0	An error ID is output.
	Alarm	ALARM_CODE	-	A read alarm code is output.
	Warning	WARNING_CODE	-	A read warning code is output.

■ ALARM_CODE (Union)

Member	Type	Description
uiAlarmCode	UINT	Alarm code
tAlarmCodeMember	ALARM_WARNING_CODES	Main alarm code and sub alarm code

■ WARNING_CODE (Union)

Member	Type	Description
uiWarningCode	UINT	Warning code
tWarningCodeMember	ALARM_WARNING_CODES	Main warning code and sub warning code

■ ALARM_WARNING_CODES (Structure)

Member	Type	Description
byMainCode	BYTE	Main alarm number code
bySubCode	BYTE	Sub alarm number code

REFERENCE

11.2.1 RTEX Error ID
11.2.2 Alarm Codes
11.2.3 Warning Codes

8.1.4 RTEX_ReadAmpState (Amplifier Alarm Status)

This is a function block (FB) that reads the AMP alarm state. It outputs the information and state of the axis where the AMP alarm or warning has occurred.

■ Icon

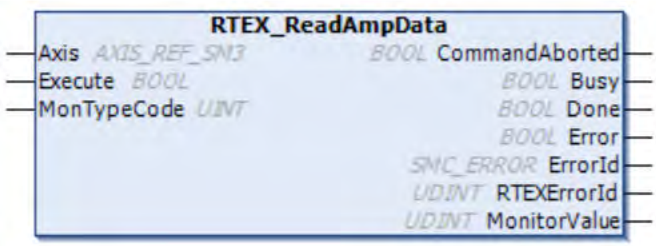
■ **Parameter**

Scope	Name	Type	Initial	Description
Output	RTEXCommState	UDINT	-	Outputs RTEX communication status 0: Initialization in progress 100: Operation in progress 1000 or more: Error
	NumberOfSlaves	UDINT	-	The number of axes connected (1 to 16) is output.
	AlarmState	DWORD	-	The MAC-ID (0 to 15) where the AMP alarm has occurred is output.
	WarningState	DWORD	-	The MAC-ID (0 to 15) where the AMP warning has occurred is output.

8.1.5 RTEX_ReadAmpData (Amplifier Monitor)

This is a function block (FB) that reads the monitor data of the AMP. It reads various monitor data of the AMP.

■ **Icon**



■ **Parameter**

Scope	Name	Type	Initial	Description
Input / output	Axis	AXIS_REF_SM3	-	Specifies the axis.
Input	Execute	BOOL	FALSE	TRUE: Starts execution at the rising edge. FALSE: Stops processing.
	MonTypeCode	UINT		Specifies the type code for the monitor command.
Output	CommandAborted	BOOL	FALSE	TRUE: An interruption from other FB has occurred.
	Busy	BOOL	FALSE	TRUE: Execution of the FB is not completed.
	Done	BOOL	FALSE	TRUE: Monitoring is completed.

Scope	Name	Type	Initial	Description
	Error	BOOL	FALSE	TRUE: An error has occurred within the FB.
	ErrorID	SMC_ERROR	0	An error ID is output.
	MonitorValue	UDINT	-	Read monitor command

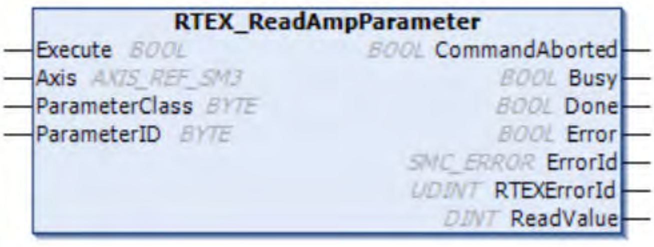
REFERENCE

11.2.1 RTEX Error ID
11.4 Monitor Commands

8.1.6 RTEX_ReadAmpParameter (Read Amplifier Parameter)

This is a function block (FB) that reads the AMP parameter.

■ Icon



■ Parameter

Scope	Name	Type	Initial	Description
Input / output	Axis	AXIS_REF_SM3	-	Specifies the axis.
Input	Execute	BOOL	FALSE	TRUE: Starts execution at the rising edge. FALSE: Stops processing.
	ParameterClass	BYTE	-	Specifies the AMP parameter classification.
	ParameterID	BYTE	-	Specifies the AMP parameter number.
Output	CommandAborted	BOOL	FALSE	TRUE: An interruption from other FB has occurred.
	Busy	BOOL	FALSE	TRUE: Execution of the FB is not completed.
	Done	BOOL	FALSE	TRUE: Reading is completed.
	Error	BOOL	FALSE	TRUE: An error has occurred within the FB.
	ErrorID	SMC_ERROR	0	An error ID is output.

Scope	Name	Type	Initial	Description
	ReadValue	DINT	-	Read AMP parameter value

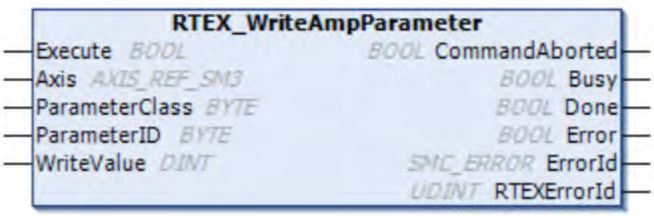
REFERENCE

11.2.1 RTEX Error ID
11.3 List of AMP Parameters

8.1.7 RTEX_WriteAmpParameter (Write Amplifier Parameter)

This is a function block (FB) that writes the AMP parameter.

■ Icon



■ Parameter

Scope	Name	Type	Initial	Description
Input / output	Axis	AXIS_REF_SM3	-	Specifies the axis.
Input	Execute	BOOL	FALSE	TRUE: Starts execution at the rising edge. FALSE: Stops processing.
	ParameterClass	BYTE	-	Specifies the AMP parameter classification.
	ParameterID	BYTE	-	Specifies the AMP parameter number.
	WriteValue	DINT	-	Value to be written in the AMP parameter
Output	CommandAborted	BOOL	FALSE	TRUE: An interruption from other FB has occurred.
	Busy	BOOL	FALSE	TRUE: Execution of the FB is not completed.
	Done	BOOL	FALSE	TRUE: Writing is completed.
	Error	BOOL	FALSE	TRUE: An error has occurred within the FB.
	ErrorID	SMC_ERROR	0	An error ID is output.

REFERENCE

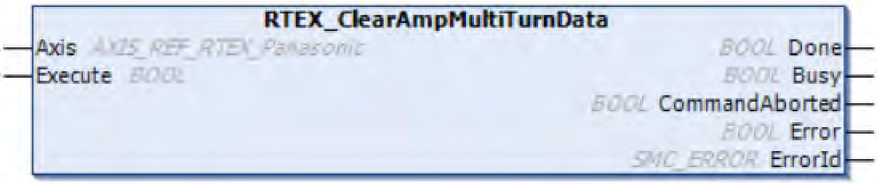
11.2.1 RTEX Error ID

11.3 List of AMP Parameters

8.1.8 RTEX_ClearAmpMultiTurnData (Clear Amplifier Multi-turn Data)

This is a function block (FB) that clears the multi-turn data of the AMP.

■ Icon



■ Parameter

Scope	Name	Type	Initial	Description
Input / output	Axis	AXIS_REF_RTEX_Panasonic	-	Specifies the axis.
Input	Execute	BOOL	FALSE	TRUE: Starts execution at the rising edge. FALSE: Stops processing.
Output	CommandAborted	BOOL	FALSE	TRUE: An interruption from other FB has occurred.
	Busy	BOOL	FALSE	TRUE: Execution of the FB is not completed.
	Done	BOOL	FALSE	TRUE: Clearing is completed.
	Error	BOOL	FALSE	TRUE: An error has occurred within the FB.
	ErrorID	SMC_ERROR	0	An error ID is output.

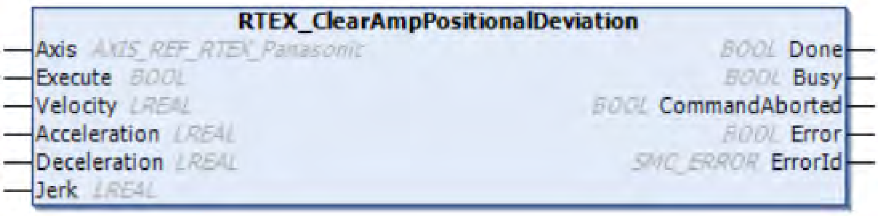
The PMC_ClearAmpMultiTurnData function block outputs the following errors.

Error	Description
SMC_WRONG_CONTROLLER_MODE	Executed in a mode other than the position control mode. Change to SMC_position using SMC_SetControllerMode.
SMC_DI_HOMING_ERROR	The encoder used is an Incremental encoder.
SMC_AXIS_NOT_READY_FOR_MOTION	The axis is in a state where PMC_ClearAmpMultiTurnData cannot be executed. It can be executed only when set to Disabled or Errorstop.
SMC_REGULATOR_OR_START_NOT_SET	The axis is in a servo ON state.
SMC_AXIS_REF_CHANGED_DURING_OPERATION	The Axis was changed during operation.

8.1.9 RTEX_ClearAmpPositionalDeviation (Clear Amplifier Deviation Counter)

This is a function block (FB) that clears the deviation counter of the AMP. It deletes the position deviation data in the deviation counter of the AMP.

■ Icon



■ Parameter

Scope	Name	Type	Initial	Description
Input / output	Axis	AXIS_REF_RTEX_Panasonic	-	Specifies the axis.
Input	Execute	BOOL	FALSE	Starts execution at the rising edge.
	Velocity(Note 1)	LREAL		Information required to execute MC_MoveAbsolute
	Acceleration(Note 1)	LREAL		Information required to execute MC_MoveAbsolute
	Deceleration(Note 1)	LREAL		Information required to execute MC_MoveAbsolute
	Jerk(Note 1)	LREAL		Information required to execute MC_MoveAbsolute
Output	CommandAborted	BOOL	FALSE	TRUE: An interruption from other FB has occurred.
	Busy	BOOL	FALSE	TRUE: Execution of the FB is not completed.
	Done	BOOL	FALSE	TRUE: Clearing is completed.
	Error	BOOL	FALSE	TRUE: An error has occurred within the FB.
	ErrorID	SMC_ERROR	0	An error ID is output.

(Note 1) This function block internally substitutes the command position with an actual position to call MC_MoveAbsolute and, therefore, requires parameters including Velocity, Acceleration, Deceleration, and Jerk.

The RTEX_ClearAmpPositionalDeviation function block outputs the following errors.

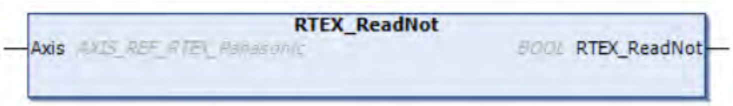
Error	Description
SMC_WRONG_CONTROLLER_MODE	Executed in a mode other than the position control mode. Change to SMC_position using SMC_SetControllerMode.

Error	Description
SMC_AXIS_NOT_READY_FOR_MOTION	The axis is in a state where PMC_ClearAmpMultiTurnData cannot be executed. It can be executed only at the standstill state.
SMC_REGULATOR_OR_START_NOT_SET	The axis is in a servo ON state。
SMC_PP_WRONG_AXIS_TYPE	The axis is a virtual axis.

8.1.10 RTEX_ReadNot (Read NOT of Amplifier)

This is a function that reads the NOT state of the amplifier.

■ Icon



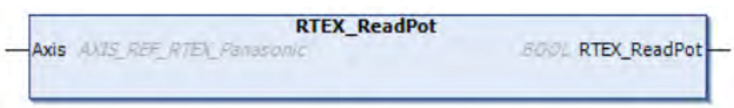
■ Parameter

Type	Parameter name	Type	Description
I/O	Axis	AXIS_REF_RTEX_Panasonic	Specifies the axis.
Output	RTEX_ReadNot	BOOL	TRUE: NOT is ON.

8.1.11 RTEX_ReadPot (Read POT of Amplifier)

This is a function that reads the POT state of the amplifier.

■ Icon



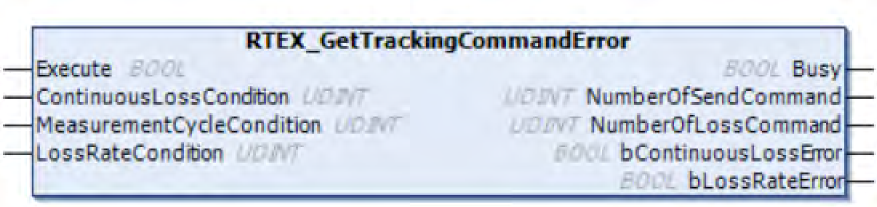
■ Parameter

Type	Parameter name	Type	Description
I/O	Axis	AXIS_REF_RTEX_Panasonic	Specifies the axis.
Output	RTEX_ReadPot	BOOL	TRUE: POT is ON.

8.1.12 RTEX_GetTrackingCommandError

The RTEX periodically sends commands. With the GM1 specifications, when the MotionTask cycle time exceeds the control cycle, the command position for the servo amplifier is not updated for that cycle. (This is called a lost RTEX command.) This function block measures the number of sent RTEX commands and the number of lost RTEX commands. Using this function, you can check if the command position is updated normally for every cycle.

■ Icon



■ Parameter

Scope		Definition	Value	Description
Input	Execute	---		Execute = TRUE: Starts measurement when triggered. Execute = FALSE: Clears output.
	ContinuousLossCondtion	Continuous command loss condition	0: Disabled	If the command loss continuously occurs at ContinuousLossCondtion cycle, bContinuousLossError turns TRUE.
	MesurementCycleCondition	Command loss statistical measurement cycle condition	0: Disabled	If the command loss occurs as many times as specified in LossRateCondition during the MesurementCycleCondition period, bLossRateError turns TRUE.
	LossRateCondition	Under measurement	0 to 100%	
Output	Busy	Under measurement		---
	NumberOfSendCommand	Total number of commands sent		Returns a value when Execute is TRUE. Clears when Execute is FALSE.
	NumberOfLossCommand	Total number of commands lost		Returns a value when Execute is TRUE. Clears when Execute is FALSE.
	bContinuousLossError	Occurrence of a continuous command loss error		Occurrence of a condition error of ContinuousLossCondtion
	bLossRateError	Occurrence of a command loss statistics error		Occurrence of a condition error of MesurementCycleCondition or LossRateCondition

(Note 1) If the number of frames exceeds 32 bits, normal value is not returned.

9 Motion Control Function Blocks (Auxiliary Function)

9.1 Motion Auxiliary Function (Monitoring)

9.1.1 MC_ReadActualPosition (Read Current Position)

This is a function block (FB) that reads the actual position data of the axis.

■ Icon

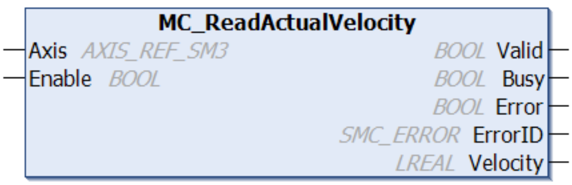
■ Parameter

Scope	Name	Type	Initial	Description
Input / output	Axis	AXIS_REF_SM3	-	Specifies the axis.
Input	Enable	BOOL	FALSE	Reads the actual position while Enable is set to TRUE.
Output	Valid	BOOL	FALSE	TRUE: Valid output
	Busy	BOOL	FALSE	TRUE: Execution of the FB is not completed.
	Error	BOOL	FALSE	TRUE: An error has occurred within the FB.
	ErrorID	SMC_ERROR	0	An error ID is output.
	Position	LREAL	0	Actual position (u) that is read out

9.1.2 MC_ReadActualVelocity (Read Current Velocity)

This is a function block (FB) that reads the actual velocity of the axis.

■ Icon



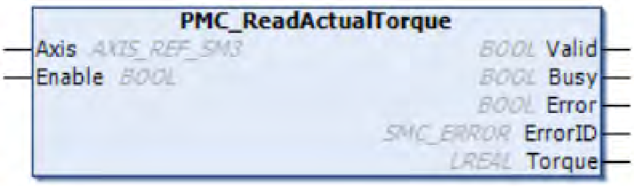
■ **Parameter**

Scope	Name	Type	Initial	Description
Input / output	Axis	AXIS_REF_SM3	-	Specifies the axis.
Input	Enable	BOOL	FALSE	Reads the actual velocity while Enable is set to TRUE.
Output	Valid	BOOL	FALSE	TRUE: Valid output
	Busy	BOOL	FALSE	TRUE: Execution of the FB is not completed.
	Error	BOOL	FALSE	TRUE: An error has occurred within the FB.
	ErrorID	SMC_ERROR	0	An error ID is output.
	Velocity	LREAL	0	Current actual velocity (u/s) that is read out

9.1.3 PMC_ReadActualTorque (Read Current Torque)

This is a function block (FB) that reads the actual torque value of the axis.

■ **Icon**



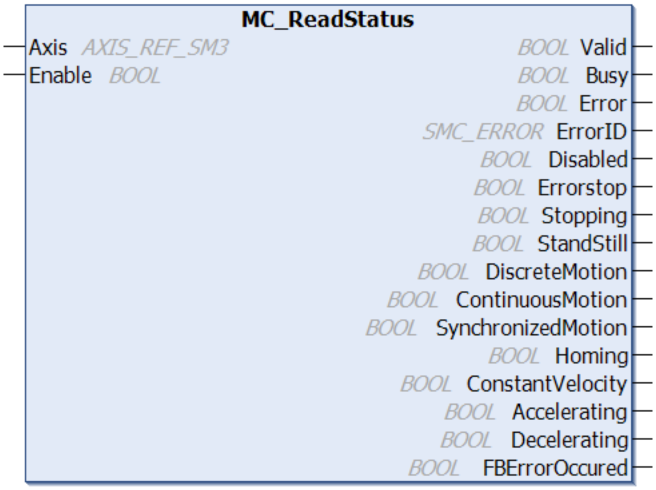
■ **Parameter**

Scope	Name	Type	Initial	Description
Input / output	Axis	AXIS_REF_SM3	-	Specifies the axis.
Input	Enable	BOOL	FALSE	Reads the actual torque value while Enable is set to TRUE.
Output	Valid	BOOL	FALSE	TRUE: Valid output
	Busy	BOOL	FALSE	TRUE: Execution of the FB is not completed.
	Error	BOOL	FALSE	TRUE: An error has occurred within the FB.
	ErrorID	SMC_ERROR	0	An error ID is output.
	Torque	LREAL	0	Current actual torque (%) that is read out

9.1.4 MC_ReadStatus (Read Status)

This is a function block (FB) that reads the status information of the axis. It reads detailed information about the axis state.

■ Icon



■ Parameter

Scope	Name	Type	Initial	Description
Input / output	Axis	AXIS_REF_SM3	-	Specifies the axis.
Input	Enable	BOOL	FALSE	Reads the status information while the input is TRUE.
Output	Valid	BOOL	FALSE	TRUE: Valid output
	Busy	BOOL	FALSE	TRUE: Execution of the FB is not completed.
	Error	BOOL	FALSE	TRUE: An error has occurred within the FB.
	ErrorID	SMC_ERROR	0	An error ID is output.
	Disabled	BOOL	FALSE	TRUE: The axis is in the Disabled state.
	ErrorStop	BOOL	FALSE	TRUE: The axis is in the ErrorStop state.
	Stopping	BOOL	FALSE	TRUE: The axis is in the Stopping state.
	StandStill	BOOL	FALSE	TRUE: The axis is in the StandStill state.
	DiscreteMotion	BOOL	FALSE	TRUE: The axis is in the DiscreteMotion state.
	ContinuousMotion	BOOL	FALSE	TRUE: The axis is in the ContinuousMotion state.

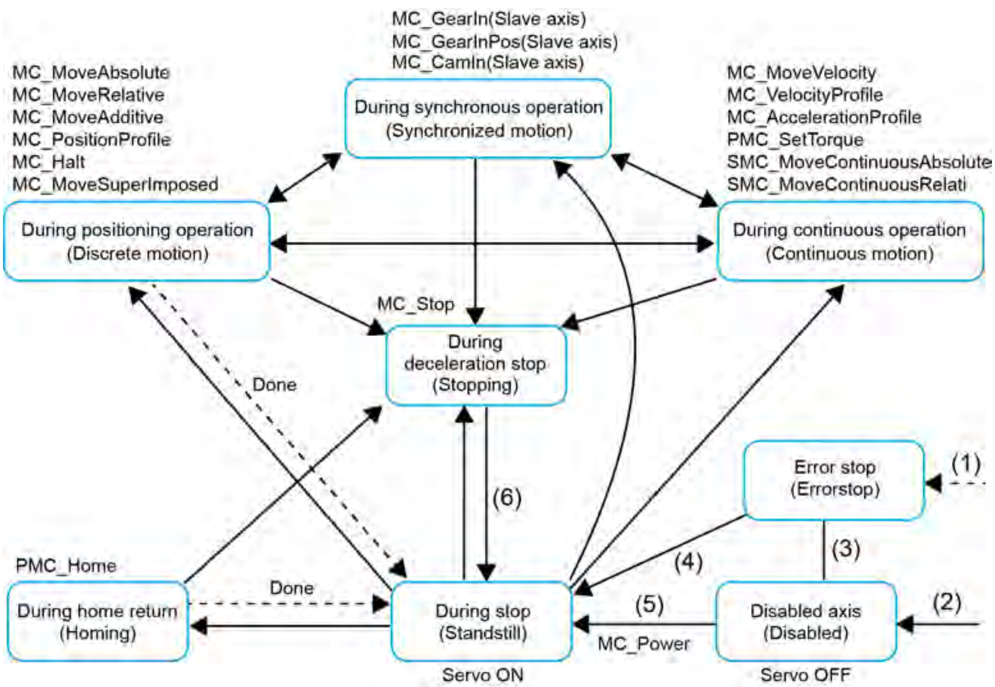
Scope	Name	Type	Initial	Description
	SynchronizedMotion	BOOL	FALSE	TRUE: The axis is in the SynchronizedMotion state.
	Homing	BOOL	FALSE	TRUE: The axis is in the Homing state.
	ConstantVelocity	BOOL	FALSE	TRUE: The axis is moving at a constant velocity.
	Accelerating	BOOL	FALSE	TRUE: The axis is moving in acceleration.
	Decelerating	BOOL	FALSE	TRUE: The axis is moving in deceleration.
	FBErrorOccured	BOOL	FALSE	TRUE: An FB error has occurred.

■ Axis state

The following section describes state transition diagram of the axis when the motion function blocks are executed.

State transition diagram

- The blue frame indicates the state.
- When the function block indicated above the state is executed, the state transitions to the direction indicated by the solid-line arrow.
- When the execution is completed or when an error occurs, the state transitions to the state indicated at the tip of the broken-line arrow.
- The terms in parentheses are Japanese defined in PLCopen.



Number	Transition conditions
(1)	Regardless of the state, when an error occurs in the axis
(2)	Regardless of the state, when Enable of MC_Power is TRUE, bRegulator is FALSE, and there is no error in the axis
(3)	When Status of MC_Reset and Status of MC_Power are FALSE
(4)	When Enable of MC_Reset and Enable of MC_Power are TRUE, bRegulator is TRUE, and Status is TRUE
(5)	When Enable of MC_Power is TRUE, bRegulator is TRUE, and Status is TRUE
(6)	When Done of MC_Stop is TRUE and Execute of MC_Stop is FALSE

9.1.5 SMC_InPosition (In-position Judgment)

This is a function block (FB) that compares the actual position of the AMP with the command value and judges whether the position is within the specified range. The maximum difference between the actual position of the AMP and the command value as well as the dwell time are specified to judge (in-position judgment) whether the specified values are satisfied.

■ Icon

■ Parameter

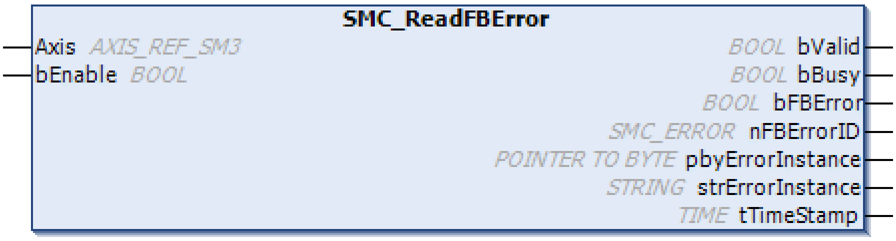
Scope	Name	Type	Initial	Description
Input / output	Axis	AXIS_REF_SM3	-	Specifies the axis.
Input	bEnable	BOOL	FALSE	TRUE: Executes the FB.
	fPosWindow	LREAL	0	The maximum difference between the actual position and the command value to judge whether the target position has been reached.
	fPosTime	LREAL	0	The dwell time (s) to judge whether the axis has reached the position
	fTimeOut	LREAL	0	The time (s) from when the FB is enabled to when judgment is made that timeout has occurred When the value is “0”, the timeout judgment is not made yet.
Output	bInPosition	BOOL	FALSE	TRUE: The target position is reached. While the difference between the actual position and the command

Scope	Name	Type	Initial	Description
				value is within the time specified in fPosTime, it is within the fPosWindow.
	bBusy	BOOL	FALSE	TRUE: The FB is in operation.
	bTimeOut	BOOL	FALSE	TRUE: Timeout has occurred.

9.1.6 SMC_ReadFBError (Read Oldest Error)

This is a function block (FB) that reads the oldest function block error information.

■ Icon



■ Parameter

Scope	Name	Type	Initial	Description
Input / output	Axis	AXIS_REF_SM3	-	Specifies the axis.
Input	bEnable	BOOL	FALSE	TRUE: The FB can be executed.
Output	bValid	BOOL	FALSE	TRUE: Error information is acquired.
	bBusy	BOOL	FALSE	TRUE: The FB is in operation.
	bFBError	BOOL	FALSE	TRUE: An error has occurred within the FB.
	nFBErrorID	SMC_ERROR	0	An error ID is output.
	pbyErrorInstance	POINTER TO BYTE	0	FB instance of the error acquisition source
	strErrorInstance	STRING	''	FB instance name of the error acquisition source
	tTimeStamp	TIME	TIME#0ms	Time stamp of the error information

Note

- The error information is cleared when SMC_ClearFBError is executed. When and error occurs again, SMC_ReadFBError reads the error.

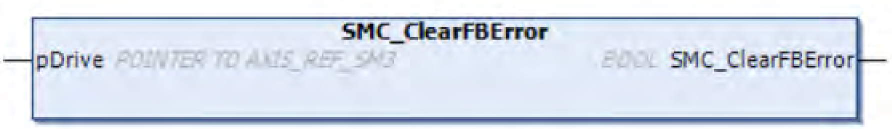
REFERENCE

9.1.7 SMC_ClearFBError (Clear Oldest Error)

9.1.7 SMC_ClearFBError (Clear Oldest Error)

This function clears the oldest FB error information.

■ Icon



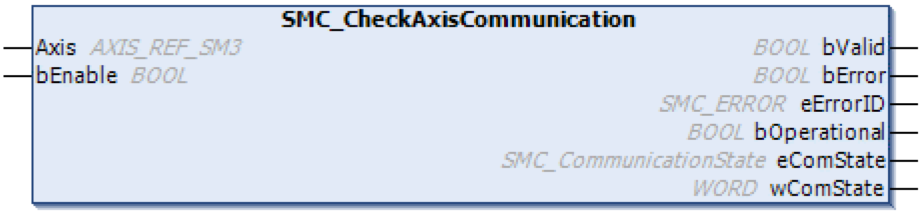
■ Parameter

Type	Parameter name	Type	Default	Description
Input	pDrive	POINTER TO AXIS_REF_SM3	-	Specifies the axis.
Return	SMC_ClearFBError	BOOL		This function always returns FALSE even for normal completion.

9.1.8 SMC_CheckAxisCommunication (Check Axis Communication Status)

This is a function block (FB) that checks the communication state of the axis.

■ Icon



■ Parameter

Scope	Name	Type	Initial	Description
Input / output	Axis	AXIS_REF_SM3	-	Specifies the axis.
Input	bEnable	BOOL	FALSE	TRUE: The FB can be executed.
Output	bValid	BOOL	FALSE	TRUE: The output value is valid.
	bError	BOOL	FALSE	TRUE: An error has occurred within the FB.
	eErrorID	SMC_ERROR	0	An error ID is output.
	bOperational	BOOL	FALSE	TRUE: Communication state is operational. (100)

Scope	Name	Type	Initial	Description
	eComState	SMC_CommunicationState	SMC_COMSTATE_NOT_STARTED	Communication state
	wComState	WORD	0	Internal value of the communication state

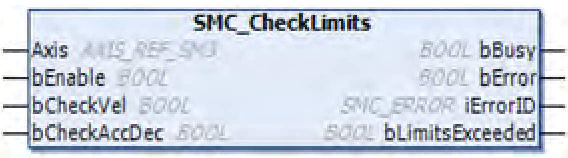
■ **SMC_CommunicationState (Enumeration type)**

Name	Value	Description
SMC_COMSTATE_NOT_STARTED	0	Stop
SMC_COMSTATE_VARIABLE_INITIALIZATION	1	Initialization of variables
SMC_COMSTATE_BASE_COM_INITIALIZATION	2	Initialization of base communication settings
SMC_COMSTATE_DRIVE_INITIALIZATION	3	Initialization of drive settings
SMC_COMSTATE_DRIVE_WAITING_FOR_SYNC	4	Waiting for drive synchronization
SMC_COMSTATE_INITIALIZATION_DONE	5	Initialization completed
SMC_COMSTATE_OPERATIONAL	6	Operational
SMC_COMSTATE_REINITIALIZATION	7	Re-initialization
SMC_COMSTATE_ERROR	8	Error
SMC_COMSTATE_UNKNOWN	9	Unknown

9.1.9 SMC_CheckLimits (Check Exceeding Limits)

This is a function block (FB) that checks whether the velocity, acceleration, or deceleration is in excess of the dynamic limit set value of the device.

■ **Icon**



■ **Parameter**

Scope	Name	Type	Initial	Description
Input / output	Axis	AXIS_REF_SM3	-	Specifies the axis.
Input	bEnable	BOOL	FALSE	TRUE: The FB can be executed.
	bCheckVel	BOOL	TRUE	TRUE: Checks the velocity setting.

Scope	Name	Type	Initial	Description
	bCheckAccDec	BOOL	FALSE	TRUE: Checks the acceleration and deceleration settings.
Output	bBusy	BOOL	FALSE	TRUE: The FB is in operation.
	bError	BOOL	FALSE	TRUE: An error has occurred within the FB.
	iErrorID	SMC_ERROR	0	An error ID is output.
	bLimitsExceeded	BOOL	FALSE	TRUE: Limits are exceeded.

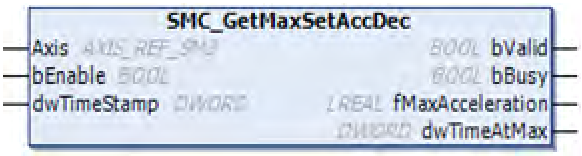
Info.

- Reference manual
  GM1 Controller RTEX User's Manual (Operation Edition)

9.1.10 SMC_GetMaxSetAccDec (Measure Maximum Acceleration / Deceleration)

This is a function block (FB) that measures the maximum value of the axis acceleration/ deceleration command.

■ Icon



■ Parameter

Scope	Name	Type	Initial	Description
Input / output	Axis	AXIS_REF_SM3	-	Specifies the axis.
Input	bEnable	BOOL	FALSE	TRUE: The FB can be executed.
	dwTimeStamp	DWORD	0	Time stamp
Output	bValid	BOOL	FALSE	TRUE: The output value is valid.
	bBusy	BOOL	FALSE	TRUE: The FB is in operation.
	fMaxAcceleration	LREAL	0	Maximum acceleration (u/s^2).
	dwTimeAtMax	DWORD	0	dwTimeStamp value at the maximum acceleration

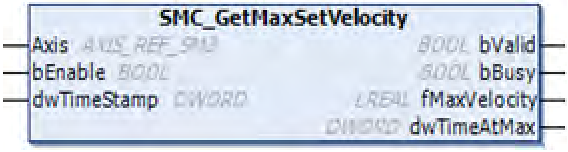
Note

- It is possible to check when the maximum acceleration or deceleration has occurred by entering a call counter value in the input variable "dwTimeStamp".

9.1.11 SMC_GetMaxSetVelocity (Measure Maximum Velocity)

This is a function block (FB) that measures the maximum value of the axis velocity command.

■ Icon



■ Parameter

Scope	Name	Type	Initial	Description
Input / output	Axis	AXIS_REF_SM3	-	Specifies the axis.
Input	bEnable	BOOL	FALSE	TRUE: The FB can be executed.
	dwTimeStamp	DWORD	0	Time stamp
Output	bValid	BOOL	FALSE	TRUE: The output value is valid.
	bBusy	BOOL	FALSE	TRUE: The FB is in operation.
	fMaxVelocity	LREAL	0	Maximum velocity (u/s).
	dwTimeAtMax	DWORD	0	dwTimeStamp value at the maximum acceleration

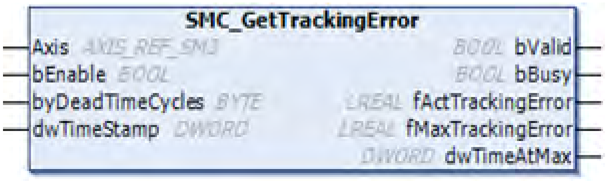
Note

- It is possible to check when the maximum velocity has occurred by entering a call counter value in the input variable "dwTimeStamp".

9.1.12 SMC_GetTrackingError (Measure Tracking Error)

This is a function block (FB) that measures the tracking error of the actual position for the axis command position.

■ Icon



■ Parameter

Scope	Name	Type	Initial	Description
Input / output	Axis	AXIS_REF_SM3	-	Specifies the axis.

Scope	Name	Type	Initial	Description
Input	bEnable	BOOL	FALSE	TRUE: The FB can be executed.
	byDeadTimeCycles	BYTE	2	Number of dead time cycles Compares the command position and actual position between the specified cycles.
	dwTimeStamp	DWORD	0	Time stamp
Output	bValid	BOOL	FALSE	TRUE: The output value is valid.
	bBusy	BOOL	FALSE	TRUE: The FB is in operation.
	fActTrackingError	LREAL	0	Actual tracking error
	fMaxTrackingError	LREAL	0	Maximum tracking error while the function block is being executed
	dwTimeAtMax	DWORD	0	dwTimeStamp value when the maximum tracking error is detected

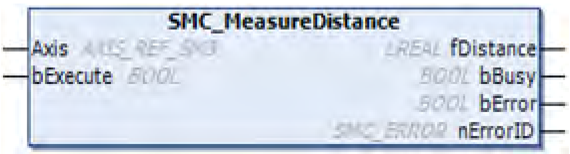
Note

- It is possible to check when the maximum tracking error has occurred by entering a call counter value in the input variable "dwTimeStamp".

9.1.13 SMC_MeasureDistance (Measure Turnaround Travel Distance)

This is a function block (FB) that measures the travel distance. For the modulo axis, the cover distance can be measured considering the laps.

■ Icon



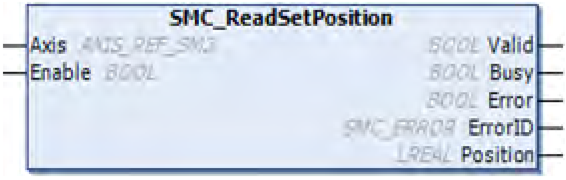
■ Parameter

Scope	Name	Type	Initial	Description
Input / output	Axis	AXIS_REF_SM3	-	Specifies the axis.
Input	bExecute	BOOL	FALSE	TRUE: Starts measurement at the rising edge. FALSE: Ends measurement.
Output	fDistance	LREAL	0	Distance traveled from the start of measurement
	bBusy	BOOL	FALSE	TRUE: The FB is in operation.
	bError	BOOL	0	TRUE: An error has occurred within the FB.
	nErrorID	SMC_ERROR	0	An error ID is output.

9.1.14 SMC_ReadSetPosition (Read Axis Set Position)

This is a function block (FB) that acquires the command position of the axis.

■ Icon



■ Parameter

Scope	Name	Type	Initial	Description
Input / output	Axis	AXIS_REF_SM3	-	Specifies the axis.
Input	Enable	BOOL	FALSE	TRUE: Executes the FB.
Output	Valid	BOOL	FALSE	TRUE: The output value is valid.
	Busy	BOOL	FALSE	TRUE: The FB is in operation.
	Error	BOOL	FALSE	TRUE: An error has occurred within the FB.
	ErrorID	SMC_ERROR	0	An error ID is output.
	Position	LREAL	0	Axis position

9.2 Motion Auxiliary Function (Change / Reset)

9.2.1 MC_Reset (Axis Error Reset)

This is a function block (FB) that resets the state transition error of the axis. It reset the axis error and transitions the state from the ErrorStop state to the StandStill state.

■ Icon

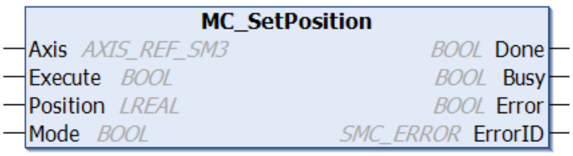
■ Parameter

Scope	Name	Type	Initial	Description
Input / output	Axis	AXIS_REF_SM3	-	Specifies the axis.
Input	Execute	BOOL	FALSE	Starts execution at the rising edge.
Output	Done	BOOL	FALSE	TRUE: Reset done
	Busy	BOOL	FALSE	TRUE: Execution of the FB is not completed.
	Error	BOOL	FALSE	TRUE: An error has occurred within the FB.
	ErrorID	SMC_ERROR	0	An error ID is output.

9.2.2 MC_SetPosition (Change Current Position)

This is a function block (FB) that changes the current command position of the axis.

■ Icon



■ Parameter

Scope	Name	Type	Initial	Description
Input / output	Axis	AXIS_REF_SM3	-	Specifies the axis.
Input	Execute	BOOL	FALSE	Starts execution at the rising edge.

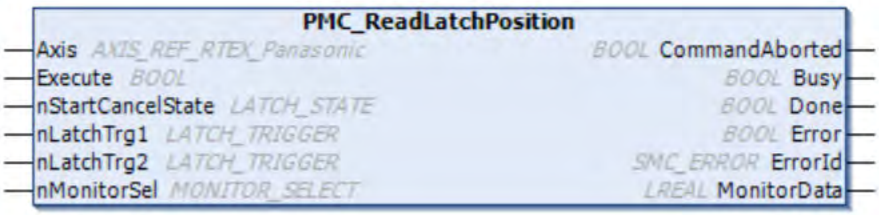
Scope	Name	Type	Initial	Description
	Position	LREAL	0	Specifies the position when the mode is set to ABSOLUTE. Specifies the distance when the mode is set to RELATIVE.
	Mode	BOOL	FALSE	TRUE: RELATIVE (Relative position) FALSE: ABSOLUTE (Absolute position)
Output	Done	BOOL	FALSE	TRUE: Position change is completed.
	Busy	BOOL	FALSE	TRUE: Execution of the FB is not completed.
	Error	BOOL	FALSE	TRUE: An error has occurred.
	ErrorID	SMC_ERROR	0	An error ID is output.

9.3 Motion Auxiliary Function (Other Functions)

9.3.1 PMC_ReadLatchPosition (Amplifier Latch Monitor)

This is a function block (FB) that monitor the AMP latch position. It reads the axis position when a trigger signal occurs.

■ Icon



■ Parameter

Scope	Name	Type	Initial	Description
Input / output	Axis	AXIS_REF_SM3	-	Specifies the axis.
Input	Execute	BOOL	FALSE	TRUE: Starts execution at the rising edge. FALSE: Stops processing.
	nStartCancelState	IoDRVRTEX.LATCH_STATE	MONITOR_LATCH	Specifies the start and cancellation of the latch mode.
	nLatchTrg1	IoDRVRTEX.LATCH_TRIGGER	Z_PHASE	Selects the trigger signal for latch position 1
	nLatchTrg2	IoDRVRTEX.LATCH_TRIGGER	-	Selects the trigger signal for latch position 2
	nMonitorSel	IoDRVRTEX.MONITOR_SELECT		Selects the latch position to be output as the output MonitorData.
Output	CommandAborted	BOOL	FALSE	TRUE: An interruption from other FB has occurred.
	Busy	BOOL	FALSE	TRUE: Execution of the FB is not completed.
	Done	BOOL	FALSE	TRUE: Output is completed.
	Error	BOOL	FALSE	TRUE: An error has occurred within the FB.
	ErrorID	SMC_ERROR	0	An error ID is output.
	MonitorData	LREAL		Axis position is output.

■ IoDRVRTEX.LATCH_STATE (Enumeration type)

Name	Value	Description
MONITOR_LATCH	80	Monitors the position latch state.

Name	Value	Description
		Monitors the state without newly starting or canceling.
START_LATCH1	81	Starts the position latch 1 (CH1).
START_LATCH2	82	Starts the position latch 2 (CH2).
START_LATCH1_AND2	83	Starts the position latch 1 (CH1) and position latch 2 (CH2).
CANCEL_LATCH1	84	Cancels the position latch 1 (CH1).
CANCEL_LATCH2	88	Cancels the position latch 2 (CH2).
CANCEL_LATCH1_AND2	92	Cancels the position latch 1 (CH1) and position latch 2 (CH2).

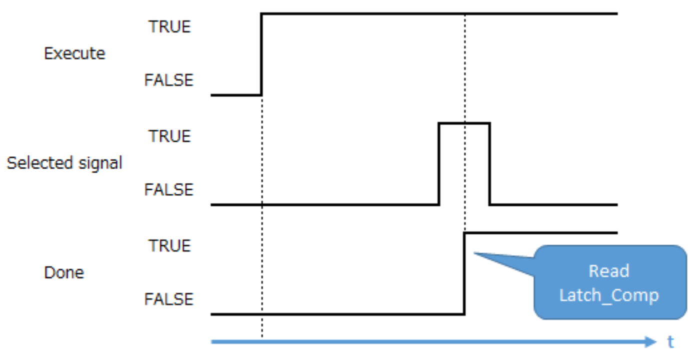
■ IoDRVRTEX.LATCH_TRIGGER (Enumeration type)

Name	Value	Description
Z_PHASE	0	Z phase
EXT1_RISING_EDGE	1	Rising edge of EXT1
EXT2_RISING_EDGE	2	Rising edge of EXT2
EXT3_RISING_EDGE	3	Rising edge of EXT3
PR7_111_RISING_EDGE	7	Not used for this FB.
EXT1_FALLING_EDGE	9	Falling edge of EXT1
EXT2_FALLING_EDGE	10	Falling edge of EXT2
EXT3_FALLING_EDGE	11	Falling edge of EXT3
PR7_111_FALLING_EDGE	15	Not used for this FB.

■ IoDRVRTEX.MONITOR_SELECT (Enumeration type)

Name	Value	Description
LPOS1	9	Latch position 1
LPOS2	10	Latch position 2

■ Operations when the function block is executed



The PMC_ReadLatchPosition function block outputs the following error.

Error	Description
SMC_WRONG_CONTROLLER_MODE	Executed in a mode other than the position control mode. Change to SMC_position using SMC_SetControllerMode.
SMC_RP_DRIVE_PARAMETER_NOT_MAPPED	Specified nLatchTrg1 and nLatchTrg2 to not use.
	Allocation of EXT1, EXT2, and EXT3 to the servo amplifier is faulty. Change the settings for Pr4.04 to Pr4.06.

As the PMC_ReadLatchPosition function block uses the RTEX home return command, it cannot be executed together with PMC_Home.

If PMC_ReadLatchPosition is executed while PMC_Home is being executed, the CommandAborted parameter of PMC_ReadLatchPosition becomes TRUE.

When using EXT1, EXT2, and EXT3 for nLatchTrg1 and nLatchTrg2, set amplifier parameters as shown in the following table.

Parameter	Parameter name	Settings
Pr4.04	SI5 input selection	EXT1
Pr4.05	SI6 input selection	EXT2
Pr4.06	SI7 input selection	EXT3

9.3.2 PMC_StopLatchPosition (Stop Amplifier Latch)

This is a function block (FB) that stops monitoring the AMP latch position. It stops the axis when a trigger event occurs.

■ Icon

■ Parameter

Scope	Name	Type	Initial	Description
Input / output	Axis	AXIS_REF_SM3	-	Specifies the axis.
Input	Execute	BOOL	FALSE	TRUE: Starts execution at the rising edge. FALSE: Stops processing.

Scope	Name	Type	Initial	Description
	nLatchTrg1	IoDRVRTEX.LATCH_TRIGGER	EXT1_RISING_EDGE	Selects the trigger signal for latch position
	Velocity	LREAL	0	Specifies the velocity (u/s).
	Acceleration	LREAL	0	Specifies the acceleration (u/s^2).
	Deceleration	LREAL	0	Specifies the deceleration (u/s^2).
	Jerk	LREAL	0	Specifies the jerk (u/s^3).
	Direction	MC_Direction	negative	Specifies the traveling direction of the axis.
Output	InVelocity	BOOL	FALSE	TRUE: The axis has reached the specified velocity for the first time.
	CommandAborted	BOOL	FALSE	TRUE: An interruption from other FB has occurred.
	Busy	BOOL	FALSE	TRUE: Execution of the FB is not completed.
	Done	BOOL	FALSE	TRUE: Stopping is completed.
	Error	BOOL	FALSE	TRUE: An error has occurred within the FB.
	ErrorID	SMC_ERROR	0	An error ID is output.
	MonitorData	LREAL		Axis position is output.

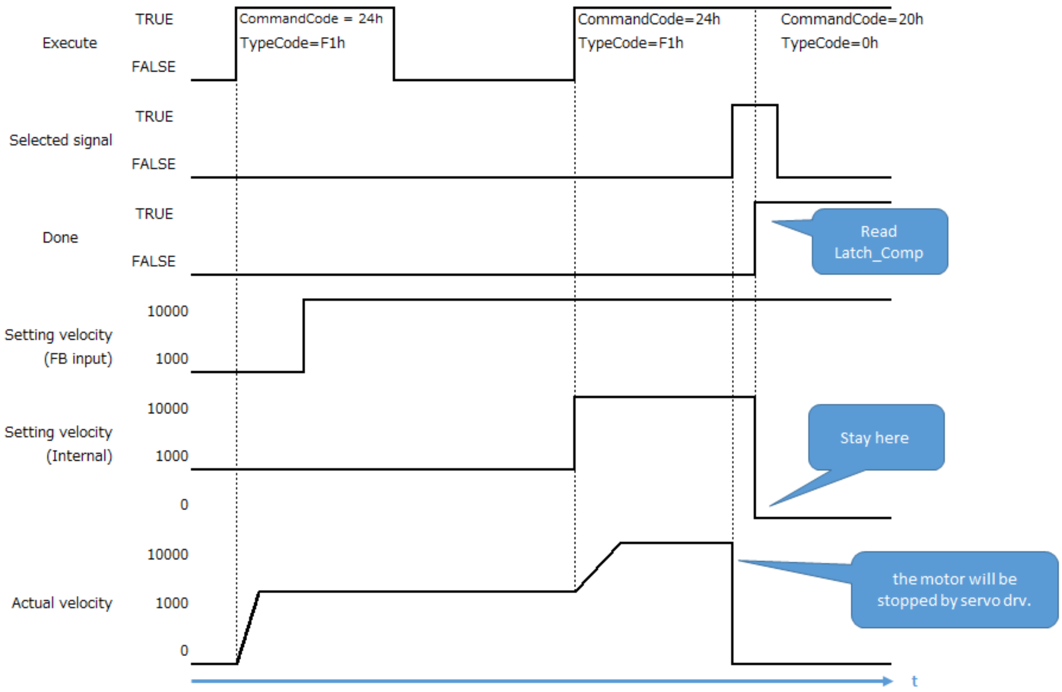
■ IoDRVRTEX.LATCH_TRIGGER (Enumeration type)

Name	Value	Description
Z_PHASE	0	Not used for this FB.
EXT1_RISING_EDGE	1	Rising edge of EXT1
EXT2_RISING_EDGE	2	Rising edge of EXT2
EXT3_RISING_EDGE	3	Rising edge of EXT3
PR7_111_RISING_EDGE	7	Condition set by MINAS amplifier parameter Pr7.111
EXT1_FALLING_EDGE	9	Falling edge of EXT1
EXT2_FALLING_EDGE	10	Falling edge of EXT2
EXT3_FALLING_EDGE	11	Falling edge of EXT3
PR7_111_FALLING_EDGE	15	Condition set by MINAS amplifier parameter Pr7.111

■ MC_Direction (Enumeration type)

Name	Value	Description
positive	1	Travels in the positive direction.
negative	-1	Travels in the negative direction.
shortest	0	Not available. Do not specify this.
fastest	3	Not available. Do not specify this.
current	2	Travels to the current direction. Possible to use only for the modulo axis.

Operations when the function block is executed



- Execute = TRUE: Starts the latch mode. Execute = FALSE: Ends the latch, however, the axis operation continues as long as PMC_StopLatchPosition is called. Stop the axis using either MC_Stop or MC_Halt.
- When a trigger signal is input, the PMC_StopLatchPosition function block ignores the command value from the GM1 and stops at the latch position.

Execution errors

The PMC_StopLatchPosition function block outputs the following error.

Error	Description
SMC_WRONG_CONTROLLER_MODE	Executed in a mode other than the position control mode. Change to SMC_position using SMC_SetControllerMode.
SMC_RP_DRIVE_PARAMETER_NOT_MAPPED	Specified nLatchTrg1 to not use.
	Allocation of EXT1, EXT2, and EXT3 to the servo amplifier is faulty. Change the settings for Pr4.04 to Pr4.06.
SMC_DI_HOMING_ERROR	Servo amplifier version is lower than V1.24.
SMC_AXIS_NOT_READY_FOR_MOTION	The axis is in a state (Stopping, Disabled, or Errorstop) where PMC_StopLatchPosition cannot be executed.
SMC_REGULATOR_OR_START_NOT_SET	The servo OFF or brake is applied.

Error	Description
SMC_3SH_INVALID_VELACC_VALUES	The input (Velocity, Acceleration, or Deceleration) is faulty.
SMC_AXIS_REF_CHANGED_DURING_OPERATION	The Axis was changed during operation.

■ Execution conditions

- As the PMC_StopLatchPosition function block uses the RTEX home return command, it cannot be executed together with PMC_Home.
- To use the PMC_StopLatchPosition function block, the MINAS version must be V1.23 or higher.
- The function block supports only the control cycle of 1.0 ms and communication cycle of 0.5 ms.

■ Amplifier parameter conditions

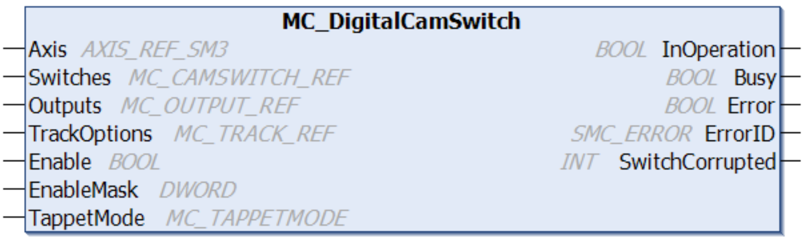
When using EXT1, EXT2, and EXT3 for nLatchTrg1, set amplifier parameters as shown in the following table.

Parameter	Parameter name	Settings
Pr4.04	SI5 input selection	EXT1
Pr4.05	SI6 input selection	EXT2
Pr4.06	SI7 input selection	EXT3

9.3.3 MC_DigitalCamSwitch (Enable Digital Cam Switch)

This is a function block (FB) that performs ON / OFF control on the digital output according to the axis position. It assigns digital cam switches to tracks (maximum of 32). Switching operations can be controlled by specifying the ON / OFF position for each digital camp switch.

■ Icon



■ Parameter

Scope	Name	Type	Initial	Description
Input / output	Axis	AXIS_REF_SM3	-	Specifies the axis where the switch is connected.
	Switches	MC_CAMSWITCH_REF	-	Specifies the switching operation.

Scope	Name	Type	Initial	Description
	Outputs	MC_OUTPUT_REF	-	ON or OFF of the switch is output. ARRAY [1..32] OF BOOL
	TrackOptions	MC_TRACK_REF	-	Specifies the property of the track. ARRAY [1..32] OF MC_TRACK_TR
Input	Enable	BOOL	FALSE	TRUE: The FB can be executed.
	EnableMask	DWORD	16#FFFFFFFF	Specifies the track to be enabled. 1: Enabled, 0: Disabled The least significant bit is the 1st track. The most significant bit is the 32nd track.
	TappetMode	MC_TAPPETMODE	tp_mode_auto	Specifies the tappet mode.
Output	InOperation	BOOL	FALSE	TRUE: The track is enabled.
	Busy	BOOL	FALSE	TRUE: Execution of the FB is not completed.
	Error	BOOL	FALSE	TRUE: An error has occurred within the FB.
	ErrorID	SMC_ERROR	0	An error ID is output.
	SwitchCorrupted	INT	-1	Index output of a faulty switch -1: No problem 0 to 31: A problem has occurred in switches 1 to 32.

■ MC_CAMSWITCH_REF (Structure)

Member	Type	Description
NoOfSwitches	BYTE	Number of switches Specifies the number of switches to be enabled when the FB is executed in the MC_CAMSWITCH_TR type array (1 to 32).
CamSwitchPtr	POINTER TO MC_CAMSWITCH_TR	Pointer to the first element of the MC_CAMSWITCH_TR type array

■ MC_CAMSWITCH_TR (Structure)

Member	Type	Description
TrackNumber	INT	Switch track number (1 to 32)
FirstOnPosition	LREAL	Position where the switch turns ON when the axis is moving in the positive direction
LastOnPosition	LREAL	Position where the switch turns OFF when the axis is moving in the positive direction Not used when CamSwitchMode is set to 1.
AxisDirection	INT	Movement direction where the switch is enabled 0: Both positive and negative directions 1: Only positive direction 2: Only negative direction

Member	Type	Description
CamSwitchMode	INT	Control method that performs switch ON / OFF control 0: ON and OFF are both controlled by the position. 1: ON is controlled by the position and OFF is controlled by the time.
Duration	TIME	Specifies the time during which the switch remains ON for when CamSwitchMode is set to 1.
bOn	BOOL	Used within the FB.
CounterOff	INT	Used within the FB.

■ MC_TRACK_REF (Structure)

Member	Type	Description
OnCompensation	LREAL	Specifies the switch ON delay time in seconds. When a positive value is specified, the switch turns ON later by the time specified. When a negative value is specified, the switch turns ON earlier by the time specified.
OffCompensation	LREAL	Specifies the switch OFF delay time in seconds. When a positive value is specified, the switch turns OFF after a delay of the time specified. When a negative value is specified, the switch turns OFF earlier by the time specified.
Hysteresis	LREAL	Specifies the hysteresis value (position).

■ MC_TAPPETMODE (Enumeration type)

Name	Value	Description
tp_mode_auto	0	Automatically determined according to the state. Servo ON state: Command position (fSetPosition) of the master axis Servo OFF state: Actual position (fActPosition) of the master axis
tp_mode_demandposition	1	Command state (fSetPosition) of the master axis
tp_mode_actualposition	2	Actual state (fActPosition) of the master axis

Regarding the method for entering defaults for variables of the MC_CAMSWITCH_TR type structure, refer to “Default Setting for Variables of the MC_TP_REF Type Structure”.

■ Operations when the function block is executed

The following sections shows switching operations (Outputs) of each track when the function block is executed after setting the parameter as follows. The axis is set to the modulo (modulo value: 1000).

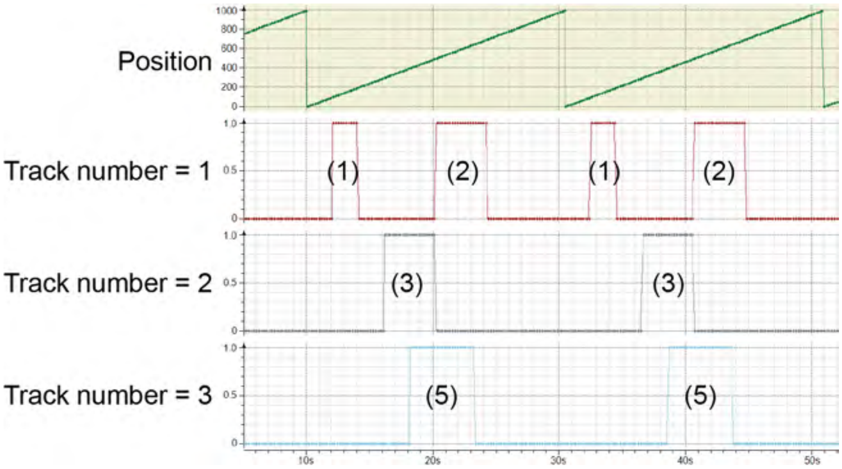
Function block input parameters

Five switches (CamSwitchPtr) are set.

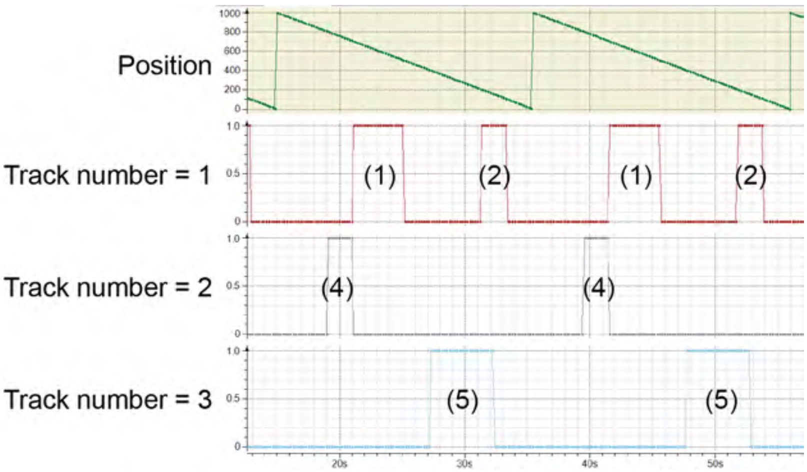
Switch	Index	Track Number	FirstOn Position	LastOn Position	Axis Direction	Cam SwitchMode	Duration
(1)	1	1	100	200	0 (Both)	0 (Position)	T#0ms

Switch	Index	Track Number	FirstOn Position	LastOn Position	Axis Direction	Cam SwitchMode	Duration
(2)	2	1	500	700	0 (Both)	0 (Position)	T#0ms
(3)	3	2	300	500	1 (Positive direction)	0 (Position)	T#0ms
(4)	4	2	700	800	2 (Negative direction)	0 (Position)	T#0ms
(5)	5	3	400	0	0 (Both)	1 (Time)	T#5s

Switching operations when the axis is moved in the positive direction
(1) to (5) are switch numbers.



Switching operations when the axis is moved in the negative direction



■ Detection of faulty switch operation (SwitchCorrupted)

SwitchCorrupted occurs when the switch does not turn ON/OFF as set.

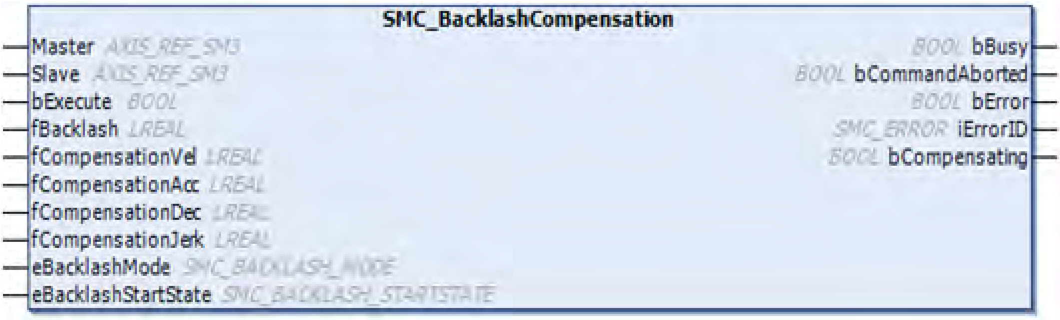
REFERENCE

5.6.6 Default Setting for Variables of the MC_TP_REF Type Structure

9.3.4 SMC_BacklashCompensation (Compensate Backlash)

This is a function block (FB) that compensates the backlash.

■ Icon



■ Parameter

Scope	Name	Type	Initial	Description
Input / output	Master	AXIS_REF_SM3	-	Specifies the master axis.
	Slave	AXIS_REF_SM3	-	Specifies the slave axis.
Input	bExecute	BOOL	FALSE	Starts execution at the rising edge. Remains enabled until the slave axis is interrupted by another operation or until an error occurs.
	fBacklash	LREAL	0	Distance to compensate (backlash)
	fCompensationVel	LREAL	0	Additional velocity used when compensation is performed (A value to be added to the master axis velocity)
	fCompensationAcc	LREAL	0	Additional acceleration used when compensation is performed (A value to be the maximum acceleration when compensation is performed)
	fCompensationDec	LREAL	0	Additional deceleration used when compensation is performed. (A value to be the maximum deceleration when compensation is performed)
	fCompensationJerk	LREAL	0	Additional jerk used when compensation is performed (Even if any value is set, the setting is disabled.)

Scope	Name	Type	Initial	Description
	eBacklashMode	SMC_BACKLASH_MODE	SMC_BL_AUTO	Backlash compensation mode
	eBacklashStartState	SMC_BACKLASH_STARTSTATE	SMC_BL_START_NONE	Specifies the start conditions whether compensation is required or not when starting the backlash compensation.
Output	bBusy	BOOL	FALSE	TRUE: The FB is in operation.
	bCommandAborted	BOOL	FALSE	TRUE: An interruption from other FB has occurred.
	bError	BOOL	FALSE	TRUE: An error has occurred within the FB.
	iErrorID	SMC_ERROR	0	An error ID is output.
	bCompensating	BOOL	FALSE	TRUE: Backlash compensation in operation

■ MC_BACKLASH_MODE (Enumeration type)

Name	Value	Description
SMC_BL_AUTO	2	Compensation in the traveling direction of the master axis
SMC_BL_POSITIVE	1	Compensation in the positive direction
SMC_BL_NEGATIVE	-1	Compensation in the negative direction
SMC_BL_OFF	0	No backlash compensation

■ SMC_BACKLASH_STARTSTATE (Enumeration type)

Name	Value	Description
SMC_BL_START_NEGATIVE	-1	If the slave axis is driven in the negative direction when compensation is started: • To make the axis travel in the positive direction, compensation is required for the backlash distance (fBacklash). • No compensation is required for the travels in the negative direction.
SMC_BL_START_NONE	0	If the slave axis is not driven in either direction when compensation is started: To make the axis travel in the positive or negative direction, compensation is required for half the amount of the backlash distance (fBacklash).
SMC_BL_START_POSITIVE	1	If the slave axis is driven in the positive direction when compensation is started: • No compensation is required for the travels in the positive direction. • To make the axis travel in the negative direction, compensation is required for the backlash distance (fBacklash).

- When starting operation, make sure that both the master axis and slave axis are in the same position. If they are not set at the same position, the slave axis travels to the master axis position at the moment when SMC_BacklashCompensation is executed.
- SMC_BacklashCompensation functions in the same way as the phase synchronous operation (MC_Phasing) and the phase depends on the master axis direction.

(MEMO)

10 Other Function Blocks

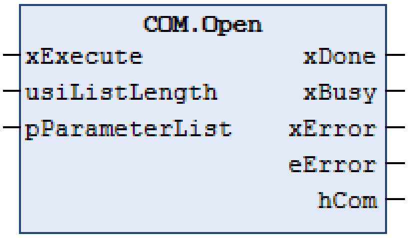
10.1 COM Port (General-purpose Communication)

This section describes function blocks that are used to perform general-purpose communication with the COM port.

10.1.1 COM.Open (Open COM port)

This is a function block that opens a COM port. It reads from and writes to the COM port using the output handle. Close the opened COM port using the COM.Close instruction.

■ Icon



■ Parameter

Scope	Name	Type	Description
Input	xExecute	BOOL	Starts execution at the rising edge.
	usiListLength	USINT	Number of pParameterList entries
	pParameterList	COM.CAA.PVOID	A pointer to the communication setting parameter list for the COM port. Specifies the pointer to the COM.PARAMETER structure array.
Output	xDone	BOOL	TRUE: Execution is completed.
	xBusy	BOOL	TRUE: Execution of the FB is not completed.
	xError	BOOL	TRUE: An error has occurred within the FB.
	eError	COM.ERROR	An error ID is output. Refer to "10.1.5 COM.ERROR (Error ID)".
	hCom	COM.CAA.HANDLE	Handle of the opened COM port.

■ COM.PARAMETER (Structure)

Member	Type	Description
udiParameterId	UDINT	Parameter ID to be set in the COM port. For a list of parameters, refer to "COM.CAA_Parameter_Constants (Constants)".
udiValue	UDINT	Value to be set in the COM port

■ COM.CAA_Parameter_Constants (Constants)

Name	Value	Support	Description
udiPort	16#1	Supported	Port number (Fixed to 1.)
udiStopBits	16#2	Supported	Stop bit Refer to "COM.STOPBIT (Enumeration type)".
udiParity	16#3	Supported	Parity Refer to "COM.PARITY (Enumeration type)".
udiBaudrate	16#4	Supported	Baud rate (Can be selected from 1200, 2400, 4800, 9600, 19200, 38400, 57600, and 115200)
udiTimeout	16#5	Not supported	Timeout
udiBufferSize	16#6	Not supported	Buffer size parameter Specifies a serial buffer size.
udiByteSize	16#7	Supported	Byte size parameter Sets the number of data bits to 4 to 8. (Specify 7 or 8 for the GM1 Controller.)
udiBinary	16#8	Not supported	Binary parameter Enables the binary mode. (With the GM1 Controller, it is fixed to 0 (binary mode).)
udiOutxCtsFlow (Note 1)	16#9	Not supported	CTS handshake for the output parameter
udiOutxDsrFlow (Note 1)	16#A	Not supported	DSR handshake for the output parameter
udiDtrControl (Note 1)	16#B	Not supported	DTR flow control parameter
udiDsrSensitivity (Note 1)	16#C	Not supported	DSR sensitivity parameter
udiRtsControl (Note 1)	16#D	Not supported	Rts flow control parameter
udiTXContinueOnXoff (Note 1)	16#E	Not supported	XOFF continues Tx parameter.
udiOutX (Note 1)	16#F	Not supported	XON / XOFF output flow control parameter
udiInX (Note 1)	16#10	Not supported	XON / XOFF of the flow control parameter
udiXonChar (Note 1)	16#11	Not supported	Tx AND Rx XON character parameter
udiXoffChar (Note 1)	16#12	Not supported	Tx AND Rx XOFF character parameter
udiXonLim (Note 1)	16#13	Not supported	Sends XON threshold parameter
udiXoffLim (Note 1)	16#14	Not supported	Sends XOFF threshold parameter

(Note 1) The GM1 Controller does not support the flow control.

■ COM.STOPBIT (Enumeration type)

Name	Value	Description
ONESTOPBIT	0	1 stop bit
ONE5STOPBITS	1	1.5 stop bit (Not available)
TWOSTOPBITS	2	2 stop bit

■ COM.PARITY (Enumeration type)

Name	Value	Description
EVEN	0	Even
ODD	1	Odd
NONE	2	None

■ Program example

Declaration section of ST program

```
Open : COM.Open;
OpenParam : ARRAY [1..7] OF COM.PARAMETER := [
    (udiParameterId := COM.CAA_Parameter_Constants.udiPort,      udiValue := 2),
    (udiParameterId := COM.CAA_Parameter_Constants.udiBaudrate,  udiValue := 115200),
    (udiParameterId := COM.CAA_Parameter_Constants.udiParity,    udiValue := INT_TO_UDINT(COM.PARITY.ODD)),
    (udiParameterId := COM.CAA_Parameter_Constants.udiStopBits,  udiValue := INT_TO_UDINT(COM.STOPBIT.ONESTOPBIT)),
    (udiParameterId := COM.CAA_Parameter_Constants.udiTimeout,   udiValue := 0),
    (udiParameterId := COM.CAA_Parameter_Constants.udiByteSize,  udiValue := 8),
    (udiParameterId := COM.CAA_Parameter_Constants.udiBinary,    udiValue := 1)
];
```

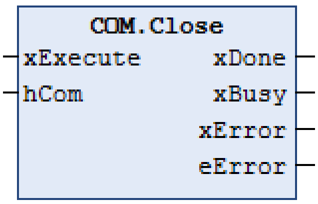
Implementation section of ST program

```
Open( xExecute:=TRUE , pParameterList:=ADR(OpenParam) , usiListLength:=SIZEOF(OpenParam)/SIZEOF(COM.PARAMETER) );
```

10.1.2 COM.Close (Close COM Port)

This is a function block that closes the COM port.

■ Icon



■ Parameter

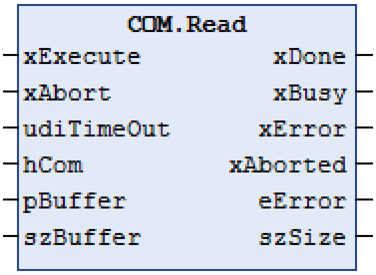
Scope	Name	Type	Description
Input	xExecute	BOOL	Starts execution at the rising edge.

Scope	Name	Type	Description
	hCom	COM.CAA. HANDLE	Handle of the COM port to be closed
Output	xDone	BOOL	TRUE: Execution is completed.
	xBusy	BOOL	TRUE: Execution of the FB is not completed.
	xError	BOOL	TRUE: An error has occurred within the FB.
	eError	COM.ERRO R	An error ID is output. Refer to "10.1.5 COM.ERROR (Error ID)".

10.1.3 COM.Read (Read COM Port)

This is a function block that reads data from the COM port.

■ Icon



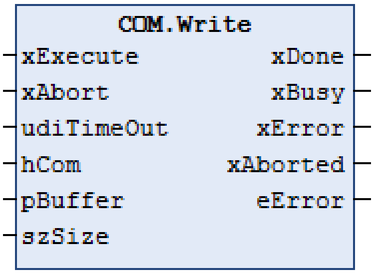
■ Parameter

Scope	Name	Type	Description
Input	xExecute	BOOL	Starts execution at the rising edge.
	xAbort	BOOL	TRUE: Stops execution and resets all outputs.
	udiTimeOut	UDINT	Timeout time until the execution is stopped (μs)
	hCom	COM.CAA. HANDLE	Handle of the COM port
	pBuffer	CAA.PVOID	Pointer to the buffer that acquires data read from the COM port
	szBuffer	CAA.SIZE	Maximum byte of pBuffer
Output	xDone	BOOL	TRUE: Execution is completed.
	xBusy	BOOL	TRUE: Execution of the FB is not completed.
	xError	BOOL	TRUE: An error has occurred within the FB.
	xAborted	BOOL	TRUE: Execution is stopped by the user.
	eError	COM.ERRO R	An error ID is output. Refer to "10.1.5 COM.ERROR (Error ID)".
	szSize	COM.CAA.S IZE	Data size (bytes) acquired by the pBuffer

10.1.4 COM.Write (Write COM Port)

This is a function block that writes data to the COM port.

■ Icon



■ Parameter

Scope	Name	Type	Description
Input	xExecute	BOOL	Starts execution at the rising edge.
	xAbort	BOOL	TRUE: Stops execution and resets all outputs.
	udiTimeOut	UDINT	Timeout time until the execution is stopped (μs)
	hCom	COM.CAA.HANDLE	Handle of the COM port
	pBuffer	CAA.PVOID	Pointer to the buffer of the data written to the COM port
	szSize	COM.CAA.SIZE	Data size (bytes) of the pBuffer to be written to the COM port
Output	xDone	BOOL	TRUE: Execution is completed.
	xBusy	BOOL	TRUE: Execution of the FB is not completed.
	xError	BOOL	TRUE: An error has occurred within the FB.
	xAborted	BOOL	TRUE: Execution is stopped by the user.
	eError	COM.ERROR	An error ID is output. Refer to "10.1.5 COM.ERROR (Error ID)".

10.1.5 COM.ERROR (Error ID)

This is an enumeration type error ID that is output when the COM port (general-purpose communication) function block is executed.

■ COM .ERROR (Enumeration type)

Name	Value	Description
NO_ERROR	0	No error
TIME_OUT	5001	Timeout error
ABORT	5002	xAbort input enabled

Name	Value	Description
HANDLE_INVALID	5003	Invalid handle
ERROR_UNKNOWN	5004	Unknown error
WRONG_PARAMETER	5005	Wrong parameter
WRITE_INCOMPLETE	5006	Incomplete write

10.2 COM port (Modbus COM)

This section describes the instructions that are used to perform ModbusRTU communication with the COM port.

10.2.1 IoDrvModbusComPort

This is a function block that controls the Modbus_Master_COM_Port device.

■ **Icon**

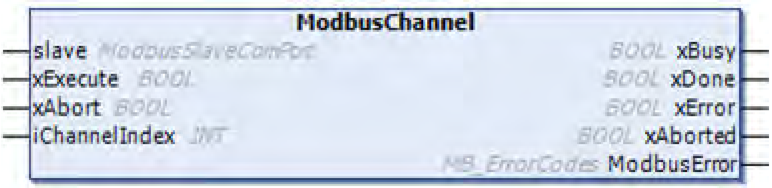
■ **Parameter**

Scope	Name	Type	Description
Input	xStop	BOOL	TRUE: Stops sending a new request to the slave. FALSE: Continues the current request.
	xResetCom Port	BOOL	Closes the COM port at a rising edge.
Output	uiNumberOf Communicat ingSlaves	UINT	Number of remote slaves under communication.
	xAllSlavesO k	BOOL	TRUE: All slaves are communicating normally. FALSE: An error has occurred in one of the slaves.

10.2.2 IoDrvModbus.ModbusChannel(Start Sending Modbus Command)

This is a function block that sends the commands set in the Modbus Slave channel of the ModbusSlaveCOM_Port device.

■ **Icon**



■ **Parameter**

Scope	Name	Type	Description
Input	xExecute	BOOL	Starts sending commands at the rising edge.
	xAbort	BOOL	TRUE: Stops execution and resets all outputs.

Scope	Name	Type	Description
	iChannelIndex	INT	Channel number where commands to be sent are set
I/O	slave	ModbusSlaveComPort	Handle of the ModbusSlaveComPort device
Output	xBusy	BOOL	TRUE: Processing of the FB is not completed.
	xDone	BOOL	TRUE: Processing is completed.
	xError	BOOL	TRUE: An error has occurred within the FB.
	xAborted	BOOL	TRUE: Execution is stopped by the user's xAbort input.
	ModbusError	MB_ErrorCodes	An error code is output. Refer to "10.5.5 IoDrvModbus.MB_ErrorCodes (Error Codes)".

10.2.3 IoDrvModbus.ModbusRequest (Modbus Request)

This is a function block that processes the Modbus command specified by I/O without using the ModbusMasterComPort device.

■ Supported commands

- Command 1 (Read multi-point coil state)
- Command 2 (Read multi-point input state)
- Command 3 (Read multi-point holding register)
- Command 4 (Read multi-point input register)
- Command 5 (Write single-point coil)
- Command 6 (Write single-point holding register)
- Command 15 (Write multi-point coil)
- Command 16 (Write multi-point holding register)
- Command 23 (Read / write multi-point holding register)

■ Icon

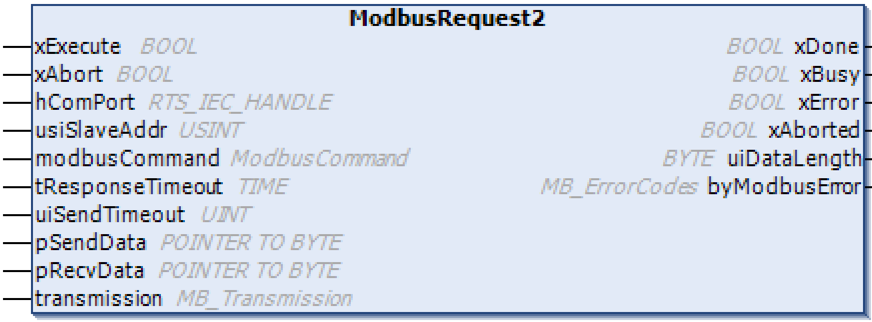
■ **Parameter**

Scope	Name	Type	Description
Input	hComPort	RTS_IEC_HANDLE	COM port handle acquired by COM.Open
	xExecute	BOOL	Starts sending commands at the rising edge.
	xAbort	BOOL	TRUE: Stops execution and resets all outputs.
	usiSlaveAddr	USINT	Slave address 1 to 247
	uiFunctionCode	UINT	Modbus function code
	uiReadOffset	UINT	Read address offset (0 to 65535)
	uiReadLen	UINT	Read length (1 to 125)
	uiWriteOffset	UINT	Write address offset (0 to 65535)
	uiWriteLen	UINT	Write length (1 to 121)
	tTimeout	UINT	Timeout value (in ms units)
	pWriteBuf	POINTER TO BYTE	Pointer to the send buffer.
	pReadBuf	POINTER TO BYTE	Pointer to the receive buffer
	transmission	MB_Transmission	Transmission type (RTU / ASCII) * Supports only RTU.
Output	xBusy	BOOL	TRUE: Processing of the FB is not completed.
	xDone	BOOL	TRUE: Processing is completed.
	xError	BOOL	TRUE: An error has occurred within the FB.
	xAborted	BOOL	TRUE: Execution is stopped by the user's xAbort input.
	byModbusErrorCode	BYTE	An error code is output. Refer to "10.2.6 IoDrvModbus.MB_ErrorCodes (Error Codes)".

10.2.4 IoDrvModbus.ModbusRequest 2 (Modbus Request 2)

This is a function block that processes, like the ModbusRequest, the Modbus command specified by I/O without using the ModbusMasterComPort device. It is different from ModbusRequest in that the structure type is used to specify the Modbus command.

■ Icon



■ Parameter

Scope	Name	Type	Description
Input	xExecute	BOOL	Starts sending commands at the rising edge.
	xAbort	BOOL	TRUE: Stops execution and resets all outputs.
	hComPort	RTS_IEC_HANDLE	COM port handle acquired by COM.Open
	usiSlaveAddr	USINT	Slave address 1 to 247
	modbusCommand	ModbusCommnad	Modbus command
	tResponseTimeout	TIME	Timeout (in ms units) of the response for a request
	uiSendTimeout	UINT	Transmission timeout
	pSendData	UINT	Pointer to the send data
	pRecvData	UINT	Pointer to the receive data
	transmission	MB_Transmission	Transmission type (RTU / ASCII) * Supports only RTU.
Output	xDone	BOOL	TRUE: Processing is completed.
	xBusy	BOOL	TRUE: Processing of the FB is not completed.
	xError	BOOL	TRUE: An error has occurred within the FB.
	xAborted	BOOL	TRUE: Execution is stopped by the user’s xAbort input.
	uiDataLength	BYTE	Received data length (byte)
	byModbusError	MB_ErrorCodes	An error code is output. Refer to "10.2.6 IoDrvModbus.MB_ErrorCodes (Error Codes)".

■ ModbusCommand (Structure)

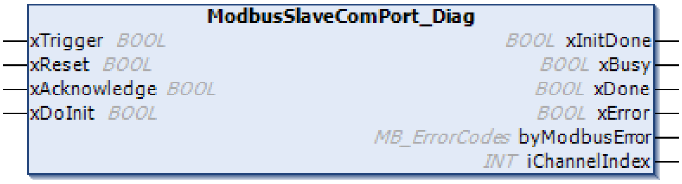
Name	Type	Description
uiFunctionCode	UINT	Modbus command code
uiReadOffset	UINT	Read address 0 to 65535

Name	Type	Description
uiReadLen	UINT	Range in the number of read instances varies depending on commands.
uiWriteOffset	UINT	Write address 0 to 65535
uiWriteLen	UINT	Range in the number of write instances varies depending on commands.

10.2.5 IoDrvModbus.ModbusSlaveComPort

This is a function block that controls the Modbus_Slave_COM_Port device.

■ Icon



■ Parameter

Scope	Name	Type	Description
Input	xTrigger	BOOL	Sends all the commands of the Modbus channel at the rising edge.
	xReset	BOOL	Resets xError and byModbusError and resumes communication.
	xAcknowledge	BOOL	Resumes communication without resetting xError and byModbusError.
	xDoInit	BOOL	TRUE: Sends a slave initialization command when communication is resumed.
Output	xInitDone	BOOL	TRUE: Modbus slave initialization command is fully completed.
	xBusy	BOOL	TRUE: Processing of the FB is not completed.
	xDone	BOOL	TRUE: Processing is completed.
	xError	BOOL	TRUE: An error has occurred within the FB.
	byModbusError	MB_ErrorCodes	An error code is output. Refer to "10.2.6 IoDrvModbus.MB_ErrorCodes (Error Codes)".
	iChannelIndex	INT	Channel index

10.2.6 IoDrvModbus.MB_ErrorCodes (Error Codes)

This is an enumeration type error code that is output when the function block for Modbus communication instruction that uses the COM port is executed.

■ IoDrvModbus.MB_ErrorCodes (Enumeration type)

Name	Value	Description
RESPONSE_SUCCESS	16#0	Succeeded
ILLEGAL_FUNCTION	16#1	Function code not supported by the slave
ILLEGAL_DATA_ADDRESS	16#2	Register offset not supported by the slave
ILLEGAL_DATA_VALUE	16#3	Illegal data writing
SLAVE_DEVICE_FAILURE	16#4	Non-recoverable error
ACKNOWLEDGE	16#5	Start operation
SLAVE_DEVICE_BUSY	16#6	During operation
MEMORY_PARITY_ERROR	16#8	Memory parity error
GATEWAY_PATH_UNAVAILABLE	16#A	Gateway path unavailable
GATEWAY_DEVICE_FAILED_TO_RESPOND	16#B	Gateway device failed to respond
RESPONSE_TIMEOUT	16#A1	Timeout
RESPONSE_CRC_FAIL	16#A2	CRC error
RESPONSE_WRONG_SLAVE	16#A3	Wrong response
RESPONSE_WRONG_FUNCTIONCODE	16#A4	Wrong function code in the response
REQUEST_FAILED_TO_SEND	16#A5	Request not sent
RESPONSE_INVALID_DATA	16#A6	Invalid response data
RESPONSE_INVALID_PROTOCOL	16#A7	Invalid response protocol
RESPONSE_INVALID_HEADER	16#A8	Invalid response header
UNDEFINED	16#FF	Undefined

10.3 LAN port (IoDrvEthernet)

This section describes the library functions that are used for the network interface to perform communication with the LAN port.

10.3.1 IoDrvEthernet

This is a function block that acquires the status of the LANPort device.

■ Icon

■ Parameter

Scope	Name	Type	Description
Output	eState	EthernetState	Ethernet state Refer to "EthernetState (Enumeration type)".

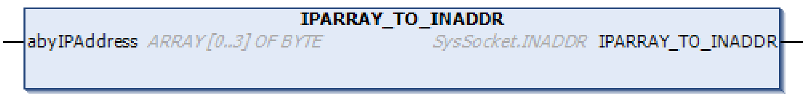
■ EthernetState (Enumeration type)

Name	Value	Description
NOT_CONFIGURED	0	Before configuration
CONFIGURED	1	After configuration
DISCONNECTED	2	Disconnected
RUNNING	3	Being executed
ERROR	4	An error has occurred.
SET_IP_ERROR	5	An IP error has occurred.

10.3.2 IoDrvEthernet.IPARRAY_TO_INADDR (Array Type to Union Type)

This is a function that converts an array type IP address to an INADDR (union type).

■ Icon



■ Parameter

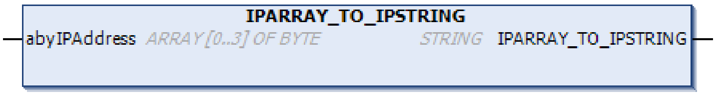
Scope	Name	Type	Description
Input	abyIPAddress	ARRY[0..3] OF BYTE	IP address array

Scope	Name	Type	Description
Output	IPARRAY_TO_INADDR	SysSocket.INADDR	Union type IP address

10.3.3 IoDrvEthernet.IPARRAY_TO_IPSTRING (Array Type to Character String Type)

This is a function that converts an array type IP address to a character string type.

■ Icon



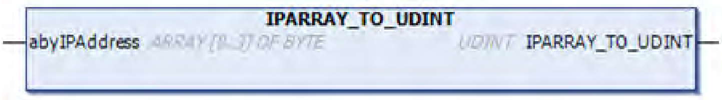
■ Parameter

Scope	Name	Type	Description
Input	abyIPAddress	ARRY[0..3] OF BYTE	IP address array
Output	IPARRAY_TO_STRING	STRING	Character string type IP address

10.3.4 IoDrvEthernet.IPARRAY_TO_UDINT (Array Type to UDINT Type)

This is a function that converts an array type IP address to a UDINT type.

■ Icon



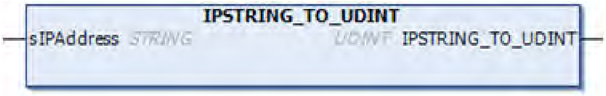
■ Parameter

Scope	Name	Type	Description
Input	abyIPAddress	ARRY[0..3] OF BYTE	IP address array
Output	IPARRAY_TO_UDINT	UDINT	UDINT type IP address

10.3.5 IoDrvEthernet.IPSTRING_TO_UDINT (Character String Type to UDINT Type)

This is a function that converts a character string type IP address to a UDINT type.

■ Icon



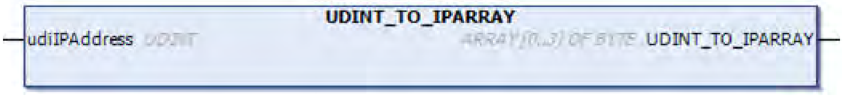
■ Parameter

Scope	Name	Type	Description
Input	abyIPAddres s	STRING	Character string type IP address
Output	IPARRAY_T O_UDINT	UDINT	UDINT type IP address

10.3.6 IoDrvEthernet.UDINT_TO_IPARRAY (UDINT Type to Array Type)

This is a function that converts a UDINT type IP address to an array type.

■ Icon



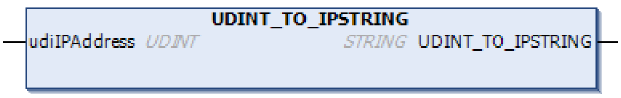
■ Parameter

Scope	Name	Type	Description
Input	abyIPAddres s	UDINT	UDINT type IP address
Output	UDINT_TO_ IPARRAY	ARRY[0..3] OF BYTE	IP address array

10.3.7 IoDrvEthernet.UDINT_TO_IPSTRING (UDINT Type to Character String Type)

This is a function that converts a UDINT type IP address to an array type.

■ Icon



■ Parameter

Scope	Name	Type	Description
Input	abyIPAddress	UDINT	UDINT type IP address
Output	UDINT_TO_STRING	STRING	Character string type IP address

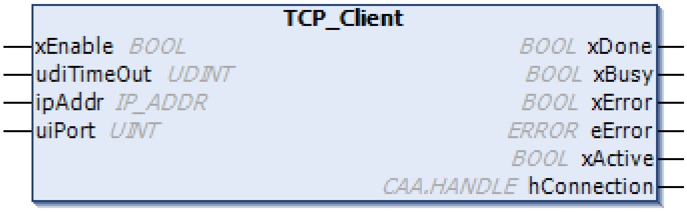
10.4 LAN Port (General-purpose Communication)

This section describes the library functions that are used to perform general-purpose communication with the LAN port using the TCP or UDP protocol.

10.4.1 NBS.TCP_Client (Connect to TCP Client)

This is a function block that connects to the TCP/IP client.

■ Icon



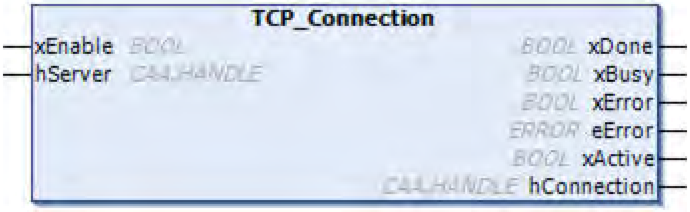
■ Parameter

Scope	Name	Type	Description
Input	xEnable	BOOL	TRUE: Active
	udiTimeOut	UDINT	Connection timeout (us) No timeout when set to 0.
	ipAddr	NBS.IP ADDR	Server IP address (character string type)
	uiPort	UINT	Server port No.
Output	xDone	BOOL	TRUE: Processing is completed.
	xBusy	BOOL	TRUE: Processing of the FB is not completed.
	xError	BOOL	TRUE: An error has occurred within the FB.
	eError	NBS.ERRO R	Connection result Refer to "10.4.8 NBS.ERROR (Error Code)".
	xActive	BOOL	TRUE: Connection is established.
	hConnection	CAA.HAND LE	Connection handle (Valid when xActive = TRUE)

10.4.2 NBS.TCP_Connection (Connect TCP)

This is a function block that establishes the connection of the client connecting to the connection port opened by TCP_Server.

■ Icon



■ Parameter

Scope	Name	Type	Description
Input	xEnable	BOOL	TRUE: Active
	hServer	CAA.HANDLE	Connection port handle acquired by TCP_Server
Output	xDone	BOOL	TRUE: Processing is completed.
	xBusy	BOOL	TRUE: Processing of the FB is not completed.
	xError	BOOL	TRUE: An error has occurred within the FB.
	eError	NBS.ERROR	Connection result Refer to "10.4.8 NBS.ERROR (Error Code)".
	xActive	BOOL	TRUE: Connection is established.(Note 1)
	hConnection	CAA.HANDLE	Connection handle (Valid when xActive = TRUE)

(Note 1) To detect a disconnection from the client after the line is connected, it is necessary to periodically call TCP_Read.

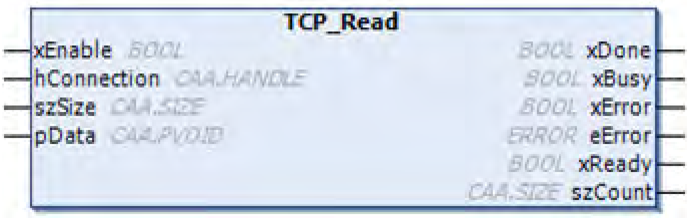
Info.

- When multiple clients are connected simultaneously to the same port, multiple TCP_Connection instances are created.
- The hServer handle acquired by one TCP_Server is set to the multiple TCP_Connection instances.

10.4.3 NBS.TCP_Read (Receive TCP Data)

This is a function block that acquires data received by the connection port that is established by TCP_Connection.

■ Icon



■ Parameter

Scope	Name	Type	Description
Input	xEnable	BOOL	TRUE: Active
	hConnection	CAA.HANDLE	Connection port handle acquired by TCP_Connection
	szSize	CAA.SIZE	Received buffer size (byte)
	pData	CAA.PVOID	Pointer to the receive buffer
Output	xDone	BOOL	TRUE: Processing is completed.
	xBusy	BOOL	TRUE: Processing of the FB is not completed.
	xError	BOOL	TRUE: An error has occurred within the FB.
	eError	NBS.ERROR	Connection result Refer to "10.4.8 NBS.ERROR (Error Code)".
	xReady	BOOL	TRUE: Data is received.
	szCount	CAA.SIZE	Received data size (byte)

10.4.4 NBS.TCP_Server (Connect TCP Server)

This is a function block that opens the specified port as a TCP/IP connection port.

■ Icon

■ Parameter

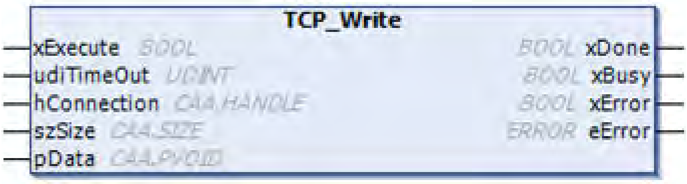
Scope	Name	Type	Description
Input	xEnable	BOOL	TRUE: Active
	ipAddr	NBS.IP_ADDR	Home IP address (character string), LANPort1 or LANPort2 IP address
	uiPort	UINT	Home waiting port number

Scope	Name	Type	Description
Output	xDone	BOOL	TRUE: Processing is completed.
	xBusy	BOOL	TRUE: Processing of the FB is not completed.
	xError	BOOL	TRUE: An error has occurred within the FB.
	eError	NBS.ERROR	Connection result Refer to "10.4.8 NBS.ERROR (Error Code)".
	hServer	CAA.HANDLE	Connection handle used by TCP_Connection

10.4.5 NBS.TCP_Write (Send TCP Data)

This is a function block that sends data to the connection port that is established by TCP_Connection.

■ Icon



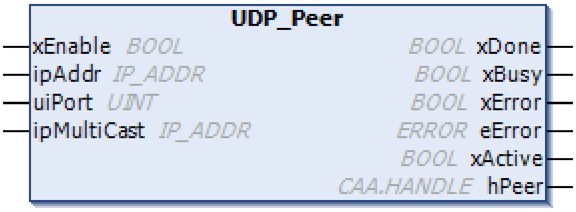
■ Parameter

Scope	Name	Type	Description
Input	xExecute	BOOL	TRUE: Send started (edge) FALSE: Processing ended (edge)
	udiTimeOut	UDINT	Timeout (us)
	hConnection	CAA.HANDLE	Connection port handle acquired by TCP_Connection
	szSize	CAA.SIZE	Send data size (byte)
	pData	CAA.PVOID	Pointer to the send data buffer.
Output	xDone	BOOL	TRUE: Processing is completed.
	xBusy	BOOL	TRUE: Processing of the FB is not completed.
	xError	BOOL	TRUE: An error has occurred within the FB.
	eError	NBS.ERROR	Connection result Refer to "10.4.8 NBS.ERROR (Error Code)".

10.4.6 NBS.UDP_Peer (Open UDP Port)

This is a function block that opens the UDP/IP port.

■ Icon



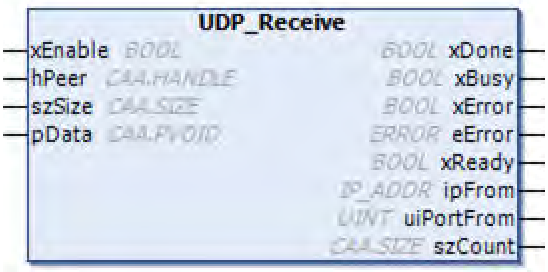
■ Parameter

Scope	Name	Type	Description
Input	xEnable	BOOL	TRUE: Active FALSE: Stop (xDone, xBusy, and xError are reset.)
	ipAddr	NBS.IP_ADDR	Home IP address (character string), LANPort1 or LANPort2 IP address
	uiPort	UINT	Home port number; Not possible to set to 0
	ipMultiCast	NBS.IP_ADDR	Multicast address ("255.255.255.255"=> INADDR_NONE)
Output	xDone	BOOL	TRUE: Processing is completed.
	xBusy	BOOL	TRUE: Processing of the FB is not completed.
	xError	BOOL	TRUE: An error has occurred within the FB.
	eError	NBS.ERROR	Connection result Refer to "10.4.8 NBS.ERROR (Error Code)".
	xActive	BOOL	TRUE: Connection is established.
	hPeer	CAA.HANDLE	Connection handle (Valid when xActive = TRUE)

10.4.7 NBS.UDP_Receive (Receive UDP Data)

This is a function block that receives data to the connection handle acquired by UDP_Peer.

■ Icon



■ Parameter

Scope	Name	Type	Description
Input	xEnable	BOOL	TRUE: Active FALSE: Stop (xDone, xBusy, and xError are reset.)
	hPeer	CAA.HANDLE	Connection handle acquired by UDP_Peer
	szSize	CAA.SIZE	Receive data buffer size (byte)
	pData	CAA.PVOID	Pointer to the receive data buffer
Output	xDone	BOOL	TRUE: Processing is completed.
	xBusy	BOOL	TRUE: Processing of the FB is not completed.
	xError	BOOL	TRUE: An error has occurred within the FB.
	eError	NBS.ERROR	Connection result Refer to "10.4.8 NBS.ERROR (Error Code)".
	xReady	BOOL	TRUE: Data acquired, FALSE: No received data
	ipFrom	NBS.IP_ADDR	Data sending source IP address
	uiPortFrom	UINT	Data sending source port No.
	szCount	CAA.SIZE	Received data size (byte)

(Note 1) If the szSize (receive data buffer size) is smaller than the received data size, only the data equivalent to the size specified by szSize is stored in pData and the data exceeding the size specified by szSize is discarded.

10.4.8 NBS.ERROR (Error Code)

This is an enumeration type error code that is output when the function block for communication instruction that uses the LAN port is executed.

■ NBS.ERROR (Enumeration type)

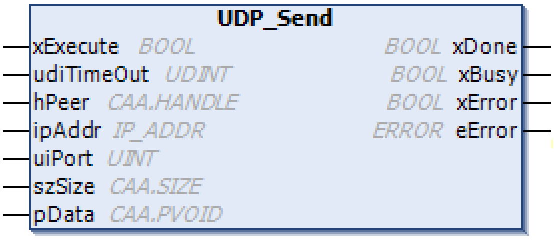
Name	Value	Description
NO_ERROR	0	No error is occurring.
FIRST_ERROR	6000	Reserved
TIME_OUT	6001	Reserved
INVALID_ADDR	6002	IP address is invalid.
INVALID_HANDLE	6003	Handle is invalid.
INVALID_DATAPOINTER	6004	Data pointer is invalid.
INVALID_DATASIZE	6005	Data size is invalid.
UDP_RECEIVE_ERROR	6006	UDP datagram cannot be received.
UDP_SEND_ERROR	6007	UDP datagram cannot be sent.
UDP_SEND_NOT_COMPLETE	6008	Reserved
UDP_OPEN_ERROR	6009	Port cannot be opened.

Name	Value	Description
UDP_CLOSE_ERROR	6010	Port cannot be released.
TCP_SEND_ERROR	6011	TCP message cannot be sent.
TCP_RECEIVE_ERROR	6012	TCP message cannot be received.
TCP_OPEN_ERROR	6013	TCP port cannot be created.
TCP_CONNECT_ERROR	6014	TCP connection cannot be established.
TCP_CLOSE_ERROR	6015	TCP port cannot be released.
TCP_SERVER_ERROR	6016	Reserved
WRONG_PARAMETER	6017	The parameter contains an invalid value.
ERROR_UNKNOWN	6018	Reserved
TCP_NO_CONNECTION	6019	There is no TCP connection.
IOCTL_ERROR	6020	Internal error (IOCTL is not supported.)
FIRST_MF	6050	Reserved
LAST_ERROR	6099	Reserved

10.4.9 NBS.UDP_Send (Send UDP Data)

This is a function block that sends data to the connection handle acquired by UDP_Peer.

■ Icon



■ Parameter

Scope	Name	Type	Description
Input	xExecute	BOOL	TRUE: Send started (edge) FALSE: Processing ended (edge)
	udiTimeOut	UDINT	Timeout (us)
	hPeer	CAA.HANDLE	Connection port handle acquired by UDP_Peer
	ipAddr	NBS.IP_ADDR	Destination IP address
	uiPort	UINT	Destination port No.
	szSize	CAA.SIZE	Send data size (byte)
	pData	CAA.PVOID	Pointer to the send data buffer.

Scope	Name	Type	Description
Output	xDone	BOOL	TRUE: Processing is completed.
	xBusy	BOOL	TRUE: Processing of the FB is not completed.
	xError	BOOL	TRUE: An error has occurred within the FB.
	eError	NBS.ERROR	Connection result Refer to "10.4.8 NBS.ERROR (Error Code)".

10.5 LAN Port (Modbus TCP)

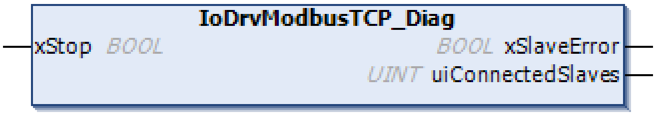
This section describes the library functions that are used to perform ModbusTCP communication with the LAN port.

It is created from Modbus master TCP available in the device tree.

10.5.1 IoDrvModbusTCP

This is a function block that controls the Modbus_TCP_Master device.

■ **Icon**



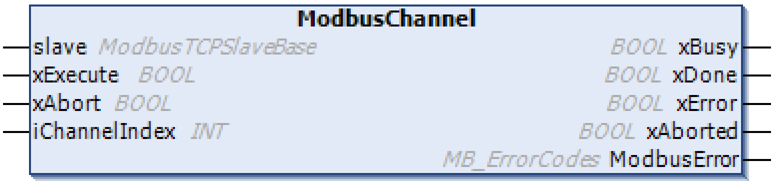
■ **Parameter**

Scope	Name	Type	Description
I/O	xStop	BOOL	TRUE: Stops sending commands to the slave.
Output	xSlaveError	BOOL	There is an error in the slave function
	uiConnectes Slaves	UINT	Number of slaves connected via TCP/IP

10.5.2 IoDrvModbusTCP.ModbusChannel (Start Sending Modbus Command)

This is a function block that sends the commands set in the Modbus Slave channel of the ModbusTCP_Slave device.

■ **Icon**



■ **Parameter**

Scope	Name	Type	Description
Input	xExecute	BOOL	Starts sending commands at the rising edge.
	xAbort	BOOL	TRUE: Stops execution and resets all outputs.
	iChannelInd ex	INT	Channel number where commands to be sent are set
I/O	slave	ModbusTCP SlaveBase	Handle of the Modbus_TCP_Slave device Output

Scope	Name	Type	Description
Output	xBusy	BOOL	TRUE: Processing of the FB is not completed.
	xDone	BOOL	TRUE: Processing is completed.
	xError	BOOL	TRUE: An error has occurred within the FB.
	xAborted	BOOL	TRUE: Execution is stopped by the user’s xAbort input
	ModbusError	MB_ErrorCodes	An error code is output. Refer to "10.5.5 IoDrvModbus.MB_ErrorCodes (Error Codes)".

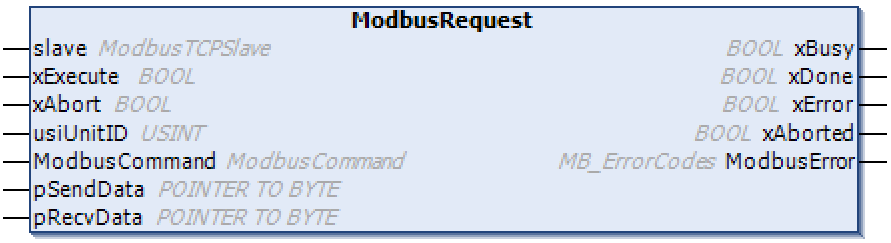
10.5.3 IoDrvModbusTCP.ModbusRequest (Modbus Request)

This is a function block that processes the Modbus command specified by I/O without using the Modbus_TCP_Slave device.

■ Supported commands

- Command 1 (Read multi-point coil state)
- Command 2 (Read multi-point input state)
- Command 3 (Read multi-point holding register)
- Command 4 (Read multi-point input register)
- Command 5 (Write single-point coil)
- Command 6 (Write single-point holding register)
- Command 15 (Write multi-point coil)
- Command 16 (Write multi-point holding register)
- Command 23 (Read / write multi-point holding register)

■ Icon



■ Parameter

Scope	Name	Type	Description
Input	slave	ModbusTCPSlave	Handle of the Modbus_TCP_Slave device
	xExecute	BOOL	Starts sending commands at the rising edge.
	xAbort	BOOL	TRUE: Stops execution and resets all outputs.
	usiUnitID	USINT	Slave address 1 to 247
	ModbusCommand	ModbusCommand	Structure that stores parameters of the commands issued.

Scope	Name	Type	Description
	pSendData	POINTER TO BYTE	Pointer to the send data buffer.
	pRecvData	POINTER TO BYTE	Pointer to the receive data buffer
Output	xBusy	BOOL	TRUE: Processing of the FB is not completed.
	xDone	BOOL	TRUE: Processing is completed.
	xError	BOOL	TRUE: An error has occurred within the FB.
	xAborted	BOOL	TRUE: Execution is stopped by the user's xAbort input
	ModbusError	BYTE	An error code is output. Refer to "10.5.5 IoDrvModbus.MB_ErrorCodes (Error Codes)". Also possible to convert the type and use as enumeration type MB_ErrorCodes.

■ ModbusCommand (Structure)

Name	Type	Description
uiFunctionCode	UINT	Modbus command code
uiReadOffset	UINT	Read address 0 to 65535
uiReadLen	UINT	Range in the number of read instances varies depending on commands.
uiWriteOffset	UINT	Write address 0 to 65535
uiWriteLen	UINT	Range in the number of write instances varies depending on commands.

10.5.4 IoDrvModbusTCPSlave

This is a function block that controls the Modbus_TCP_Slave device.

■ Icon

■ Parameter

Scope	Name	Type	Description
Input	xConfirmError	BOOL	Resets xError and byModbusError and resumes communication.

Scope	Name	Type	Description
	xDoInit	BOOL	TRUE: Sends a slave initialization command when communication is resumed.
Output	xInitDone	UINT	TRUE: Modbus slave initialization command is fully completed.
	xDone	BOOL	TRUE: Processing is completed.
	xBusy	BOOL	TRUE: Processing of the FB is not completed.
	xError	BOOL	TRUE: An error has occurred within the FB.
	byModbusError	MB_ErrorCodes	An error code is output. Refer to "10.5.5 IoDrvModbus.MB_ErrorCodes (Error Codes)".
	ComSettings	ModbusTCPComSettings	IP address and port number registered in the Modbus_TCP_Slave device.
	ComState	ModbusTCPComState	Communication status
	iChannelIndex	INT	Channel number

10.5.5 IoDrvModbus.MB_ErrorCodes (Error Codes)

This is an enumeration type error code that is output when the function block for Modbus communication instruction that uses the COM port is executed.

■ IoDrvModbus.MB_ErrorCodes (Enumeration type)

Name	Value	Description
RESPONSE_SUCCESS	16#0	Succeeded
ILLEGAL_FUNCTION	16#1	Function code not supported by the slave
ILLEGAL_DATA_ADDRESS	16#2	Register offset not supported by the slave
ILLEGAL_DATA_VALUE	16#3	Illegal data writing
SLAVE_DEVICE_FAILURE	16#4	Non-recoverable error
ACKNOWLEDGE	16#5	Start operation
SLAVE_DEVICE_BUSY	16#6	During operation
MEMORY_PARITY_ERROR	16#8	Memory parity error
GATEWAY_PATH_UNAVAILABLE	16#A	Gateway path unavailable
GATEWAY_DEVICE_FAILED_TO_RESPOND	16#B	Gateway device failed to respond
RESPONSE_TIMEOUT	16#A1	Timeout
RESPONSE_CRC_FAIL	16#A2	CRC error
RESPONSE_WRONG_SLAVE	16#A3	Wrong response
RESPONSE_WRONG_FUNCTIONCODE	16#A4	Wrong function code in the response
REQUEST_FAILED_TO_SEND	16#A5	Request not sent

Name	Value	Description
RESPONSE_INVALID_DATA	16#A6	Invalid response data
RESPONSE_INVALID_PROTOCOL	16#A7	Invalid response protocol
RESPONSE_INVALID_HEADER	16#A8	Invalid response header
UNDEFINED	16#FF	Undefined

10.6 LAN Port (EtherNet/IP)

This section describes the instructions that are used to control EtherNet/IP scanner and adapter functions using the SMC.

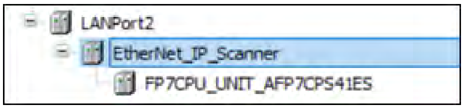
10.6.1 IoDrvEtherNetIP (EtherNet/IP Scanner Device)

This is a function block (FB) that controls the EtherNet/IP scanner device.

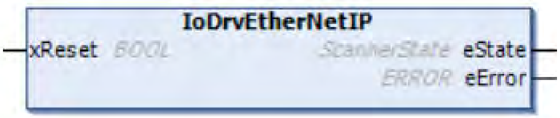
This function block is automatically generated by adding an EtherNet/IP scanner device and the name of the device that is added is used as the instance name.

Example

Adding an EtherNet/IP scanner device named "EtherNet_IP_Scanner" to LANPort2



■ Icon



■ Parameter

Scope	Name	Type	Description
Input	xReset	BOOL	Resets the scanner function at the rising edge
Output	eState	ScannerState	EtherNet/IP scanner device state
	eError	ERROR	Error state code of EtherNet/IP scanner

■ ScannerState (EtherNet/IP scanner device state)

Name	Description
INITIALIZING	The device is setting up a CIP object. It is continuing IP_CONFIG.
IP_CONFIG	The device creates an IP configuration for Ethernet interface and waits until it enters a RUNNING state.
UDP_CONFIG	The device opens the socket for UDP default port 2222.
ENCAPSULATION_CONFIG	The encapsulation server for the scanner is started via the default TCP port (44818).
ADAPTER_CONFIG	The device is in an empty state. It is continuing OPEN_CONNECTIONS.
OPEN_CONNECTIONS	The CIP ID status is set to "configured" and the RUNNING state continues.
RUNNING	The device opens a connection to the adapter and processes explicit messages with I/O communication.

Name	Description
DIAGNOSTIC_AVAILABLE	There are diagnostic messages from the configurator or editor.
BUS_ERROR	The UDP or TCP port failed to open.
RESET	xReset for the CIP ID object was received.
ERROR	When the network interface is in a continued state, the scanner enters the INITIALIZING state.

■ ERROR (Error state code of EtherNet/IP scanner)

Name	Description
NO_ERROR	No error is occurring.
INVALID_COMMAND	The command is invalid.
OUT_OF_MEMORY	A memory shortage occurred.
INVALID_DATA	The data is invalid.
INVALID_SESSION_HANDLE	The session handle is invalid.
INVALID_LENGTH	The data length is invalid.
UNSUPPORTED_PROTOCOL_VERSION	The protocol version is unsupported.
NBS_ERROR	An NBS error occurred.
NBS_RCV_ERROR	Data cannot be received via NBS.
NBS_SND_ERROR	Data cannot be sent via NBS.
ENCAPSULATION_ERROR	An encapsulation error occurred.
TCPIP_CONFIG_ERROR	TCP IP settings are incorrect.
UDP_CONFIG_ERROR	UDP settings are incorrect.
UDP_RECV_ERROR	UDP datagrams cannot be received.
UDP_SEND_ERROR	UDP datagrams cannot be sent.
UDP_CLOSE_ERROR	UDP ports cannot be released.
NULL_POINTER	This is a null pointer.
DEVICE_STATE_ERROR	An error is occurring on the device.
RECONFIGURATION_FAILED	Reconfiguration failed.
PERFORMANCE_MONITOR_DISABLED	The performance monitor is disabled.
INVALID_MEASURING_POINT	Measuring points are invalid.
IP_CONFIG_ERROR	IP settings are faulty.

10.6.2 RemoteAdapter (Remote Adapter Device)

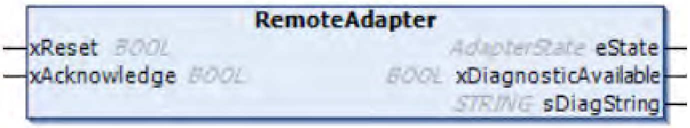
This is a function block (FB) for the remote adapter device linked to the EtherNet/IP scanner device.

This function block is automatically generated by adding an EtherNet/IP remote adapter device and the name of the device that is added is used as the instance name.

Example

Adding a remote adapter device named "FP7CPU_UNIT_AFP7CPS41ES" to EtherNet_IP_Scanner

■ Icon



■ Parameter

Scope	Name	Type	Description
Input	xReset	BOOL	Resets the remote adapter function at the rising edge
	xAcknowledge	BOOL	Acknowledges the diagnostic information at the rising edge
Output	eState	Adapter State	Remote adapter state
	xDiagnostic Available	BOOL	The output remains TRUE when there is diagnostic information
	sDiagString	STRING	Diagnosis string

■ AdapterState (Adapter device state)

Name	Description
DISABLED	The device is disabled in device tree
NOT_CONFIGURED	Parameters are being loaded
IP_CONFIG	The device has configured a TCP object and is waiting for an Ethernet node
ENCAPSULATION_CONFIG	Encapsulation is being configured
LIST_SERVICES	List services are being executed
REGISTER_SESSION	Register session is in progress
PARAMETER_CONFIG	Parameters are being configured
CONFIGURED	The device is in configuration completion state
RUNNING	The device is in running state
IDLE	The device is in idle state
RESET	UDP and TCP connection is closing
RESET_SERVICE	Reset service is being executed
CONNECTIVITY_CHECK	Connectivity check is in progress
BUS_ERROR	Bus error is occurring

Name	Description
ERROR	Error is occurring

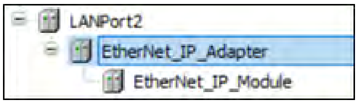
10.6.3 IoDrvEtherNetIPAdapter (EtherNet/IP adapter device)

This is a function block (FB) that controls the EtherNet/IP adapter device.

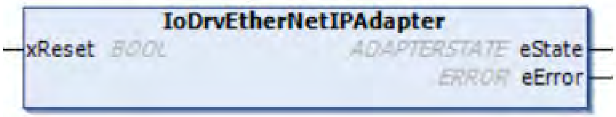
This function block is automatically generated by adding an EtherNet/IP adapter device and the name of the device that is added is used as the instance name.

Example

Adding an EtherNet/IP adapter device named "EtherNet_IP_Adapter" to LANPort2



■ Icon



■ Parameter

Scope	Name	Type	Description
Input	xReset	BOOL	Resets the adapter function at the rising edge
Output	eState	ADAPTERSTATE	EtherNet/IP adapter device state
	eError	ERROR	Error state code of EtherNet/IP adapter

■ ADAPTERSTATE (EtherNet/IP adapter device state)

Name	Description
UPDATE_CONFIGURATION	Startup phase
NOT_CONFIGURED	Parameters are being loaded
DISABLED	The device is disabled in device tree
CONFIGURED	A CIP object has been created
IP_CONFIG	The device has configured a TCP object and is waiting for an Ethernet node
IMPLICITMESSAGING_CONFIG	UDP port has been opened
EXPLICITMESSAGING_CONFIG	TCP port has been opened
NO_CONNECTION	The protocol stack has been started, but the scanner is unconnected.
RUNNING	The protocol stack is running, and the scanner is connected.

Name	Description
STOPPED	The Ethernet node is inactive, and the device is waiting for the Ethernet node to return.
RESET	UDP and TCP connection is closing.
SCANNER_EXTENSION	If the scanner registered this adapter as an I/O extension, the adapter is active in this state.
ERROR	Critical error
BUS_ERROR	Ethernet is not ready yet or is unavailable.

■ ERROR (EtherNet/IP adapter error state)

Name	Description
NO_ERROR	No error
TIME_OUT	Timeout
CONFIGURATION_FAILED	Failed to initialize resources, load connector parameters, or communicate with sub-connectors (modules)
IP_CONFIG_FAILED	The Ethernet node issued an error
IMPLICITMESSAGING_CONFIG_FAILED	Failed to create UDP port "CIP_ENC.ParameterList.gc_uiUDPPort" (default: 2222)
EXPLICITMESSAGING_CONFIG_FAILED	Failed to create TCP / UDP port "IP_ENC.ParameterList.gc_uiTCPPort" (default: 44818)
EXPLICITMESSAGE_RECEIVE_FAILED	Problem related to TCP or UDP port socket | CIP_ENC.ParameterList.gc_uiTCPPort | (default: 44818)
EXPLICITMESSAGE_SEND_FAILED	Problem related to TCP or UDP port socket | CIP_ENC.ParameterList.gc_uiTCPPort | (default: 44818)
LICENSE_MISSING	No license

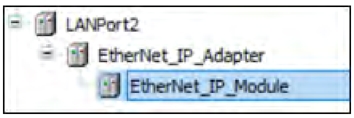
10.6.4 Module (EtherNet/IP Module Device)

This is a function block (FB) that controls the EtherNet/IP module device.

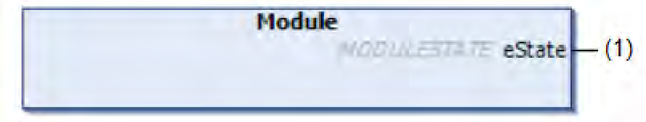
This function block is automatically generated by adding an EtherNet/IP module device and the name of the device that is added is used as the instance name.

Example

Adding an EtherNet/IP module device named "EtherNet_IP_Module" to EtherNet/IP adapter device



■ Icon



■ Parameter

Scope	Name	Type	Description
Output	eState	MODULESTATE	Module device state

■ MODULESTATE (EtherNet/IP module device state)

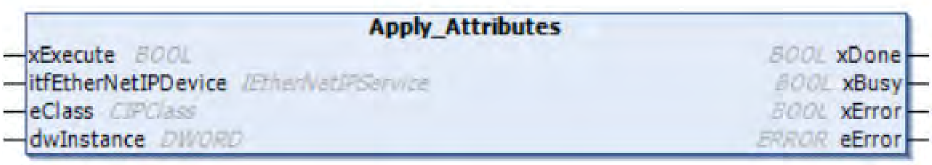
Name	Description
NOT_CONFIGURED	Parameters are being loaded.
CONFIGURED	A CIP object has been created.
NO_CONNECTION	The protocol stack has been started, but the scanner is unconnected.
RUNNING	The protocol stack is running, and the scanner is connected.
STOPPED	The Ethernet node is inactive, and the device is waiting for the Ethernet node to return.
DISABLED	The device is disabled in device tree.
ERROR	Critical error

10.6.5 Apply_Attributes (Apply_Attributes Service)

This is a function block (FB) that calls the "Apply_Attributes" service of the CIP object instance.

The attribute set in "Get_Attribut_Single" or "Get_Attribut_All" is adopted and saved in the adapter.

■ Icon



■ Parameter

Scope	Name	Type	Description
Input	xExecute	BOOL	Execution flag
	itfEtherNetIPDevice	IEtherNetIPService	EtherNet/IP device that implements the EtherNet/IP service

Scope	Name	Type	Description
	eClass	ENIP.CIPClass	Class that executes the service
	dwInstance	DWORD	Instance that executes the service (0: Class level, 1-x: Instance level)
Output	xDone	BOOL	Completion flag
	xBusy	BOOL	Busy flag
	xError	BOOL	Error flag
	eError	ENIP.ERROR	Error (0-255: CIP error, 256-x: Library error)

10.6.6 Generic_Service (Generic Service Execution)

This is a function block (FB) that executes generic services with the EtherNet/IP adapter.

Messages are sent as unconnected explicit message requests.

Note

- The endianness of data to be sent or received must be exchanged by devices.

■ Icon

■ Parameter

Scope	Name	Type	Description
Input	xExecute	BOOL	Execution flag
	itfEtherNetIPDevice	IEtherNetIPService	EtherNet/IP device that implements the EtherNet/IP service
	eClass	ENIP.CIPClass	Class that executes the service
	dwInstance	DWORD	Instance that executes the service (0: Class level, 1-x: Instance level)
	wAttribute	WORD	Attribute corresponding to the service

Scope	Name	Type	Description
	eService	ENIP.CIPCommonService	CIPCommonService member service code or vendor-specific service code
	pWriteData	POINTER TO BYTE	Pointer to data to be written to the EtherNet/IP adapter. The parameter is set to 0 when no data is sent.
	udiWriteDataSize	UDINT	Size of data to be written to the EtherNet/IP adapter. The parameter is set to 0 when no data is sent.
	pReadData	POINTER TO BYTE	Storage pointer to data received from the EtherNet/IP adapter. The parameter is set to 0 when no data is received.
	udiReadDataSize	UDINT	Size of storage buffer for data received from the EtherNet/IP adapter. The parameter is set to 0 when no data is received.
Output	xDone	BOOL	Completion flag
	xBusy	BOOL	Busy flag
	xError	BOOL	Error flag
	eError	ENIP.ERROR	Error (0-255: CIP error, 256-x: Library error)
	udiReceivedDataSize	UDINT	Size of received data

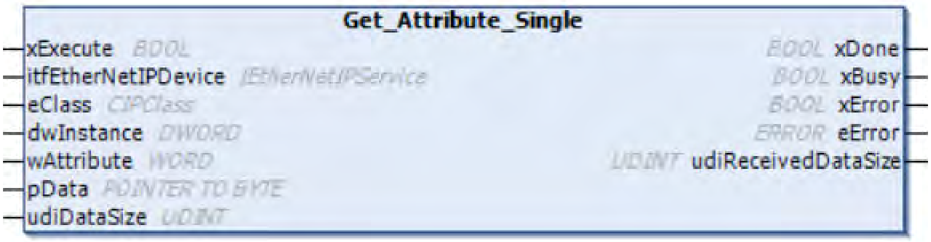
■ **ENIP.CIPCommonService (CIPCommonService member service code)**

Name	Value
None	16#0
GET_ATTRIBUTES_ALL	16#1
SET_ATTRIBUTES_ALL	16#2
RESET	16#5
START	16#6
STOP	16#7
APPLY_ATTRIBUTES	16#D
GET_ATTRIBUTE_SINGLE	16#E
SET_ATTRIBUTE_SINGLE	16#10
NO_OPERATION	16#17

10.6.7 Get_Attribute_Single (Inquire Specific Attributes of a Specific Instance)

This is a function block (FB) that inquires specific attributes of a specific instance of the CIP object.

■ Icon



■ Parameter

Scope	Name	Type	Description
Input	xExecute	BOOL	Execution flag
	itfEtherNetIPDevice	IEtherNetIPService	EtherNet/IP device that implements the EtherNet/IP service
	eClass	ENIP.CIPClass	Class that executes the service
	dwInstance	DWORD	Instance that executes the service (0: Class level, 1-x: Instance level)
	wAttribute	WORD	Attribute corresponding to the service
	pData	POINTER TO BYTE	Storage pointer to data received from the EtherNet/IP adapter
	udiDataSize	UDINT	Size of storage buffer for data received from the EtherNet/IP adapter
Output	xDone	BOOL	Completion flag
	xBusy	BOOL	Busy flag
	xError	BOOL	Error flag
	eError	ENIP.ERROR	Error (0-255: CIP error, 256-x: Library error)
	udiReceivedDataSize	UDINT	Size of received data

10.6.8 Get_Attributes_All (Inquire All Attributes of a Specific Instance)

This is a function block (FB) that inquires all attributes of a specific instance of the CIP object.

■ Icon

■ Parameter

Scope	Name	Type	Description
Input	xExecute	BOOL	Execution flag
	itfEtherNetIPDevice	IEtherNetIPService	EtherNet/IP device that implements the EtherNet/IP service
	eClass	ENIP.CIPClass	Class that executes the service
	dwInstance	DWORD	Instance that executes the service (0: Class level, 1-x: Instance level)
	pData	POINTER TO BYTE	Storage pointer to data received from the EtherNet/IP adapter
	udiDataSize	UDINT	Size of storage buffer for data received from the EtherNet/IP adapter
Output	xDone	BOOL	Completion flag
	xBusy	BOOL	Busy flag
	xError	BOOL	Error flag
	eError	ENIP.ERROR	Error (0-255: CIP error, 256-x: Library error)
	udiReceivedDataSize	UDINT	Size of received data

10.6.9 Set_Attribute_Single (Set Specific Attributes of a Specific Instance)

This is a function block (FB) that sets specific attributes of a specific instance of the CIP object

■ Icon

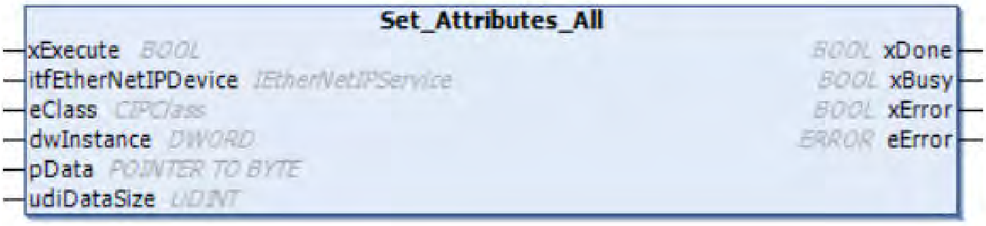
■ Parameter

Scope	Name	Type	Description
Input	xExecute	BOOL	Execution flag
	itfEtherNetIPDevice	IEtherNetIP Service	EtherNet/IP device that implements the EtherNet/IP service
	eClass	ENIP.CIPClass	Class that executes the service
	dwInstance	DWORD	Instance that executes the service (0: Class level, 1-x: Instance level)
	wAttribute	WORD	Attribute corresponding to the service
	pData	POINTER TO BYTE	Pointer to data to be written
	udiDataSize	UDINT	Size of data to be written
Output	xDone	BOOL	Completion flag
	xBusy	BOOL	Busy flag
	xError	BOOL	Error flag
	eError	ENIP.ERROR	Error (0-255: CIP error, 256-x: Library error)

10.6.10 Set_Attributes_All (Set All Attributes of a Specific Instance)

This is a function block (FB) that sets all attributes of a specific instance of the CIP object.

■ Icon



■ Parameter

Scope	Name	Type	Description
Input	xExecute	BOOL	Execution flag
	itfEtherNetIPDevice	IEtherNetIP Service	EtherNet/IP device that implements the EtherNet/IP service
	eClass	ENIP.CIPClass	Class that executes the service
	dwInstance	DWORD	Instance that executes the service (0: Class level, 1-x: Instance level)

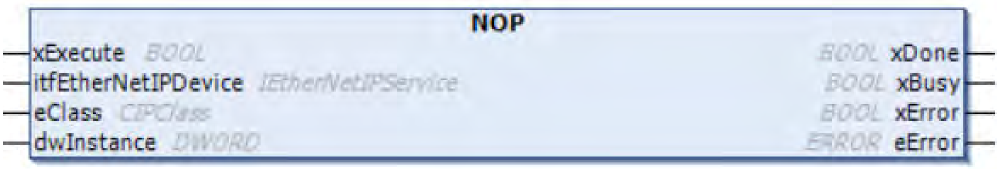
Scope	Name	Type	Description
	pData	POINTER TO BYTE	Pointer to data to be written
	udiDataSize	UDINT	Size of data to be written
Output	xDone	BOOL	Completion flag
	xBusy	BOOL	Busy flag
	xError	BOOL	Error flag
	eError	ENIP.ERROR	Error (0-255: CIP error, 256-x: Library error)

10.6.11 NOP (NOP Service)

This is a function block (FB) that executes the NOP service of a specific instance of the CIP object.

Normally, this service is used to check whether the adapter can still be used in the network.

■ Icon



■ Parameter

Scope	Name	Type	Description
Input	xExecute	BOOL	Execution flag
	itfEtherNetIPDevice	IEtherNetIPService	EtherNet/IP device that implements the EtherNet/IP service
	eClass	ENIP.CIPClass	Class that executes the service
	dwInstance	DWORD	Instance that executes the service (0: Class level, 1-x: Instance level)
Output	xDone	BOOL	Completion flag
	xBusy	BOOL	Busy flag
	xError	BOOL	Error flag
	eError	ENIP.ERROR	Error (0-255: CIP error, 256-x: Library error)

10.6.12 Reset (Reset Service)

This is a function block (FB) that executes the Reset service of a specific instance of the CIP object.

The effects of this service differ according to the CIP object.

■ Icon

■ Parameter

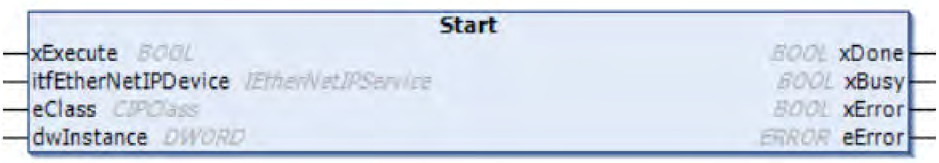
Scope	Name	Type	Description
Input	xExecute	BOOL	Execution flag
	itfEtherNetIPDevice	IEtherNetIPService	EtherNet/IP device that implements the EtherNet/IP service
	eClass	ENIP.CIPClass	Class that executes the service
	dwInstance	DWORD	Instance that executes the service (0: Class level, 1-x: Instance level)
Output	xDone	BOOL	Completion flag
	xBusy	BOOL	Busy flag
	xError	BOOL	Error flag
	eError	ENIP.ERROR	Error (0-255: CIP error, 256-x: Library error)

10.6.13 Start (Start Service)

This is a function block (FB) that executes the Start service of a specific instance of the CIP object.

The effects of this service differ according to the CIP object.

■ Icon



■ Parameter

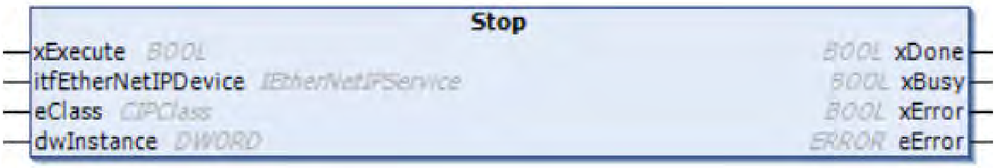
Scope	Name	Type	Description
Input	xExecute	BOOL	Execution flag
	itfEtherNetIPDevice	IEtherNetIP Service	EtherNet/IP device that implements the EtherNet/IP service
	eClass	ENIP.CIPClass	Class that executes the service
	dwInstance	DWORD	Instance that executes the service (0: Class level, 1-x: Instance level)
Output	xDone	BOOL	Completion flag
	xBusy	BOOL	Busy flag
	xError	BOOL	Error flag
	eError	ENIP.ERROR	Error (0-255: CIP error, 256-x: Library error)

10.6.14 Stop (Stop Service)

This is a function block (FB) that executes the Stop service of a specific instance of the CIP object.

The effects of this service differ according to the CIP object.

■ Icon



■ Parameter

Scope	Name	Type	Description
Input	xExecute	BOOL	Execution flag
	itfEtherNetIPDevice	IEtherNetIP Service	EtherNet/IP device that implements the EtherNet/IP service
	eClass	ENIP.CIPClass	Class that executes the service
	dwInstance	DWORD	Instance that executes the service (0: Class level, 1-x: Instance level)
Output	xDone	BOOL	Completion flag
	xBusy	BOOL	Busy flag
	xError	BOOL	Error flag

Scope	Name	Type	Description
	eError	ENIP.ERROR	Error (0-255: CIP error, 256-x: Library error)

10.6.15 ENIP.ERROR (Message Service Instruction Error Code)

Name	Value	Description
NO_ERROR	0	The service was executed normally by the specified object.
CONNECTION_FAILURE	16#1	The connection-related service failed due to the connection path.
RESOURCE_UNAVAILABLE	16#2	The object was unable to use the resources that it required to execute the requested service.
INVALID_PARAM_VALUE	16#3	Refer to status code 0x20 that is an appropriate value to be used in this situation.
PATH_SEGMENT_ERROR	16#4	The path segment identifier or segment syntax was not recognized by the processing node. Path processing stops when a path segment error occurs.
PATH_DESTINATION_UNKNOWN	16#5	The path refers to an object class, instance, or structure element that is unknown or not included in the processing node. If an unknown path destination error occurs, path processing will stop.
PARTIAL_TRANSFER	16#6	Only part of the expected data was transferred.
CONNECTION_LOST	16#7	The messaging connection was lost.
SERVICE_NOT_SUPPORTED	16#8	The requested service is not implemented or defined for this object class or instance.
INVALID_ATTRIBUTE_VALUE	16#9	Invalid attribute data was detected.
ATTRIBUTE_LIST_ERROR	16#A	The status of the attribute of Get_Attribute_List or Set_Attribute_List response is other than zero.
ALREADY_IN_REQUEST_STATE	16#B	The object is already in the mode or state requested by the service.
OBJECT_STATE_ERROR	16#C	The object cannot execute the requested service in the current mode or state.
OBJECT_ALREADY_EXISTS	16#D	An instance requested for the object to be created already exists.
ATTRIBUTE_NOT_SETTABLE	16#E	A request to change a read-only attribute was received.
PRIVILEGE_VIOLATION	16#F	An authority / privilege check failed.
DEVICE_STATE_ERROR	16#10	The current mode or state of the device prohibits the requested service from being executed.
REPLY_DATA_TOO_LARGE	16#11	The size of data to be sent via a response buffer is larger than the capacity of the allocated response buffer.
FRAGMENTATION_OF_VALUE	16#12	The service specifies an operation that fragmentates half of primitive data values which are a REAL data type.
NOT_ENOUGH_DATA	16#13	The service did not provide enough data to execute the specified operation.

Name	Value	Description
ATTRIBUTE_NOT_SUPPORTED	16#14	The attribute specified in the request is not supported.
TOO_MUCH_DATA	16#15	The service provided more data than expected.
OBJECT_DOES_NOT_EXIST	16#16	The specified object does not exist in the device.
SERVICE_FRAGMENTATION_ SEQUENCE_NOT_IN_PROGRESS	16#17	The fragmentation sequence of this service is currently not active for this data.
NO_STORED_ATTRIBUTE_DATA	16#18	The attribute data of this object has not been saved before the requested service is executed.
STORE_OPERATION_FAILURE	16#19	The attribute data of this object has not been saved because an error occurred during the attempt to save the data.
ROUTING_FAILURE_REQUEST_ PACKET_TOO_LARGE	16#1A	The service request packet was too large to send through the network existing in the path to the destination. The routing device forcibly canceled the service.
ROUTING_FAILURE_RESPONSE_ PACKET_TOO_LARGE	16#1B	The service response packet was too large to send through the network existing in the path from the destination. The routing device forcibly canceled the service.
MISSING_ATTRIBUTE_LIST_ ENTRY_DATA	16#1C	The service did not provide attributes in the list of attributes that it requires to execute the requested operation.
INVALID_ATTRIBUTE_VALUE_LIST	16#1D	The service returned a list of provided attributes together with status information of invalid attributes.
EMBEDDED_SERVICE_ERROR	16#1E	An error occurred in the embedded service.
VENDOR_SPECIFIC_ERROR	16#1F	A vendor-specific error occurred. The additional code field for error response is used to define a specific error that occurred. Use this field only if the error in question does not apply to any of the error codes shown in these tables or those shown in the object class definition.
INVALID_PARAMETER	16#20	The parameter associated with the request is invalid. This code is used when the parameter does not meet the requirements of this specification or the requirements defined in the application object specification.
WRITE_ONEC_VALUE_OR_ MEDIUM_ALREADY_WRITTEN	16#21	An attempt was made to write to a write-once medium (such as WORM drive or PROM) to which data has already been written or to change a value that cannot be changed once set.
INVALID_REPLY_RECEIVED	16#22	An invalid response was received (for example, the response service code does not match the request service code or the response message is shorter than the expected minimum response size). This status code is useful to investigate other causes of invalid responses.
BUFFER_OVERFLOW	16#23	The size of the received message exceeds the maximum size of messages that can be handled by the receiver buffer. The entire message was discarded.
MESSAGE_FORMAT_ERROR	16#24	The format of the received message is not supported by the server.
KEY_FAILURE_IN_PATH	16#25	The key segment included as the first segment of the path does not match the destination module. The object-specific status indicates which part of the key check has failed.

Name	Value	Description
PATH_SIZE_INVALID	16#26	The size of the path sent with the service request is not large enough to route the request to the object or routing data included in the path is too much.
UNEXPECTED_ATTRIBUTE_IN_LIST	16#27	An attempt was made to set an attribute that cannot currently be set.
INVALID_MEMBER_ID	16#28	The member ID specified in the request does not exist in the specified class, instance, or attribute.
MEMBER_NOT_SETTABLE	16#29	A request to change an unchangeable member was received.
GROUP_2_ONLY_SERVER_ GENERAL_FAILURE	16#2A	This error code is issued only by DeviceNet Group 2 Only servers with 4K or less code space and is supported only instead of the server. Attributes are not supported and cannot be set.
UNKNOWN_MODBUS_ERROR	16#2B	The program for conversion from CIP to Modbus received an unknown Modbus exception code.
ATTRIBUTE_NOT_GETTABLE	16#2C	A request to read an unreadable attribute was received.
INSTANCE_NOT_DELETABLE	16#2D	The requested object instance cannot be deleted.
SERVICE_NOT_SUPPORTED_ FOR_SPECIFIED_PATH	16#2E	The object supports the service but does not support the specified application path (such as attributes). Note: Do not use this code for the set service. (Instead, use general status code 0x0E or 0x29.)
TIME_OUT	16#100	The request has timed out.
INTERFACE_MISSING		IEtherNetIPService is not implemented.
REMOTE_CALL_FAILED		There is no physical connection.
NULL_POINTER		A null value was entered by mistake.
INVALID_DATA_SIZE		The data size is invalid.
WRONG_INTERFACE_VERSION		The versions do not match. The device is not equipped with the same version of interface as the called method.
NO_MEMORY		There is not enough memory.
UNKNOWN_ERROR		An unknown error occurred.
ABORTED		The service was aborted.

10.6.16 ENIP.CIPClass (Service Class Code)

Name	Value
IdentityObject	16#1
MessageRouterObject	16#2
DeviceNetObject	16#3
AssemblyObject	16#4
ConnectionObject	16#5
ConnectionManagerObject	16#6
RegisterObject	16#7

Name	Value
DiscreteInputPointObject	16#8
DiscreteOutputPointObject	16#9
AnalogInputPointObject	16#A
AnalogOutputPointObject	16#B
PresenceSensingObject	16#E
ParameterObject	16#F
ParameterGroupObject	16#10
GroupObject	16#12
DiscreteInputGroupObject	16#1D
DiscreteOutputGroupObject	16#1E
DiscreteGroupObject	16#1F
AnalogInputGroupObject	16#20
AnalogOutputGroupObject	16#21
AnalogGroupObject	16#22
PositionSensorObject	16#23
PositionControllerSupervisorObject	16#24
PositionControllerObject	16#25
BlockSequencerObject	16#26
CommandBlockObject	16#27
MotorDataObject	16#28
ControlSupervisorObject	16#29
ACDCDriveObject	16#2A
AcknowledgeHandlerObject	16#2B
OverloadObject	16#2C
SoftstartObject	16#2D
SelectionObject	16#2E
S_DeviceSupervisorObject	16#30
S_AnalogSensorObject	16#31
S_AnalogActuatorObject	16#32
S_SingleStageControllerObject	16#33
S_GasCalibrationObject	16#34
TripPointObject	16#35
FileObject	16#37
S_PartialPressureObject	16#38
SafetySupervisorObject	16#39
SafetyValidatorObject	16#3A
SafetyDiscreteOutputPointObject	16#3B

Name	Value
SafetyDiscreteOutputGroupObject	16#3C
SafetyDiscreteInputPointObject	16#3D
SafetyDiscreteInputGroupObject	16#3E
SafetyDualChannelOutputObject	16#3F
S_SensorCalibrationObject	16#40
EventLogObject	16#41
MotionDeviceAxisObject	16#42
TimeSyncObject	16#43
ModbusObject	16#44
OriginatorConnectionListObject	16#45
ModbusSerialLinkObject	16#46
DeviceLevelRingObject	16#47
QoSObject	16#48
SafetyAnalogInputPointObject	16#49
SafetyAnalogInputGroupObject	16#4A
SafetyDualChannelAnalogInputObject	16#4B
SERCOSIIILinkObject	16#4C
TargetConnectionListObject	16#4D
EnergyObject	16#4E
ElectricalEnergyObject	16#4F
Non_ElectricalEnergyObject	16#50
BaseSwitchObject	16#51
SNMPObject	16#52
PowerManagementObject	16#53
ControlNetObject	16#F0
ControlNetKeeperObject	16#F1
ControlNetSchedulingObject	16#F2
ConnectionConfigurationObject	16#F3
PortObject	16#F4
TCPIPInterfaceObject	16#F5
EthernetLinkObject	16#F6
CompoNetLink	16#F7
CompoNetRepeater	16#F8

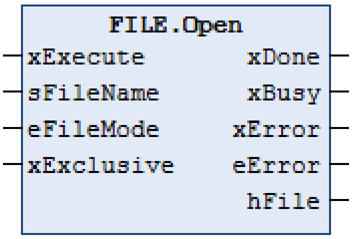
10.7 SD Card Operation (File Operation)

Files in the SD card inserted in the SD memory card slot can be operated.
In file operation using the GM1 Controller, WSTRING (kanji) cannot be used in the file name and directory name.

10.7.1 FILE.Open (Open File)

This is a function block that opens a file or creates a new file.

■ Icon



■ Parameter

Scope	Name	Type	Description
Input	xExecute	BOOL	Starts execution at the rising edge.
	sFileName	FILE.CAA.FILENAME	Specifies the file name with an absolute path or relative path.
	eFileMode	FILE.MODE	File mode
	xExclusive	BOOL	TRUE: Exclusive access mode FALSE: Multiple access mode xExclusive is not supported.
Output	xDone	BOOL	TRUE: Execution is completed.
	xBusy	BOOL	TRUE: Execution of the FB is not completed.
	xError	BOOL	TRUE: An error has occurred within the FB.
	eError	FILE.ERROR	An error ID is output. Refer to "10.7.15 FILE.ERROR (Error ID)".
	hFile	FILE.CAA.HANDLE	Handle of a file

■ FILE.MODE (Enumeration type)

Name	Value	Description
MWRITE	0	Overwrite mode (When the specified file does not exist, a new file is created.)
MREAD	1	Read mode
MRDWR	2	Read / write mode (When the specified file does not exist, a new file is created.)

Name	Value	Description
MAPPD	3	Append write mode

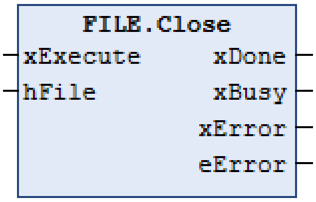
Info.

- You cannot use full size characters and the following symbols in a file name: [\], [/], [:], [*], [?], ["], [<], [>], [|].

10.7.2 FILE.Close (Close File)

This is a function block that closes a file.

■ Icon



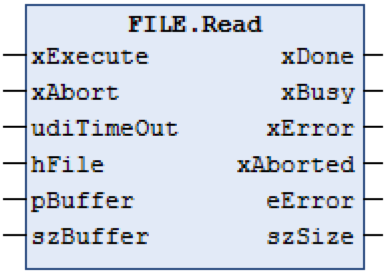
■ Parameter

Scope	Name	Type	Description
Input	xExecute	BOOL	Starts execution at the rising edge.
	hFile	FILE.CAA.HANDLE	Handle of a file to be closed Specifies the handle output by FILE.Open.
Output	xDone	BOOL	TRUE: Execution is completed.
	xBusy	BOOL	TRUE: Execution of the FB is not completed.
	xError	BOOL	TRUE: An error has occurred within the FB.
	eError	FILE.ERROR	An error ID is output. Refer to "10.7.15 FILE.ERROR (Error ID)".

10.7.3 FILE.Read (Read File)

This is a function block that reads data from the file opened.

■ Icon



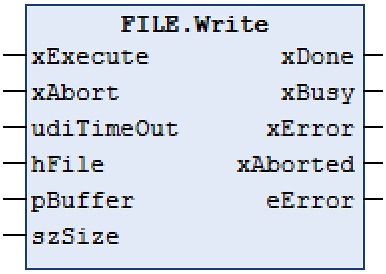
■ Parameter

Scope	Name	Type	Description
Input	xExecute	BOOL	Starts execution at the rising edge.
	xAbort	BOOL	TRUE: Stops execution and resets all outputs.
	udiTimeOut	UDINT	Timeout time until the execution is stopped (μs)
	hFile	FILE.CAA.HANDLE	Handle of a file Specifies the handle output by FILE.Open.
	pBuffer	FILE.CAA.PVOID	Pointer to the data buffer to be read Gets a pointer by the ADR operator.
	szBuffer	FILE.CAA.SIZE	Size of the data buffer to be read Gets a pointer by the SIZEOF operator.
Output	xDone	BOOL	TRUE: Execution is completed.
	xBusy	BOOL	TRUE: Execution of the FB is not completed.
	xError	BOOL	TRUE: An error has occurred within the FB.
	xAborted	BOOL	TRUE: Execution is stopped by the user.
	eError	FILE.ERROR	An error ID is output. Refer to "10.7.15 FILE.ERROR (Error ID)".
	szSize	FILE.CAA.SIZE	Size of the read data buffer

10.7.4 FILE.Write (Write File)

This is a function block that writes data to the file opened.

■ Icon



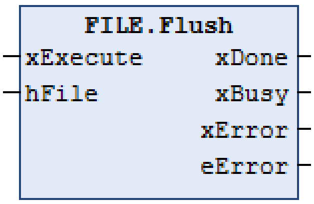
■ Parameter

Scope	Name	Type	Description
Input	xExecute	BOOL	Starts execution at the rising edge.
	xAbort	BOOL	TRUE: Stops execution and resets all outputs.
	udiTimeOut	UDINT	Timeout time until the execution is stopped (μs)
	hFile	FILE.CAA.HANDLE	Handle of a file Specifies the handle output by FILE.Open.
	pBuffer	FILE.CAA.PVOID	Pointer to the data buffer to be written Gets a pointer by the ADR operator.
	szSize	FILE.CAA.SIZE	Size of the data buffer to be written Gets a pointer by the SIZEOF operator.
Output	xDone	BOOL	TRUE: Execution is completed.
	xBusy	BOOL	TRUE: Execution of the FB is not completed.
	xError	BOOL	TRUE: An error has occurred within the FB.
	xAborted	BOOL	TRUE: Execution is stopped by the user.
	eError	FILE.ERROR	An error ID is output. Refer to "10.7.15 FILE.ERROR (Error ID)".

10.7.5 FILE.Flush (Flush File)

This is a function block that flushes buffer contents to a file.

■ Icon



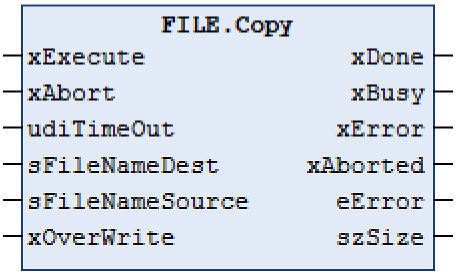
■ Parameter

Scope	Name	Type	Description
Input	xExecute	BOOL	Starts execution at the rising edge.
	hFile	FILE.CAA.HANDLE	Handle of a file Specifies the handle output by FILE.Open.
Output	xDone	BOOL	TRUE: Execution is completed.
	xBusy	BOOL	TRUE: Execution of the FB is not completed.
	xError	BOOL	TRUE: An error has occurred within the FB.
	eError	FILE.ERROR	An error ID is output. Refer to "10.7.15 FILE.ERROR (Error ID)".

10.7.6 FILE.Copy (Copy File)

This is a function block that copies a file.

■ Icon



■ Parameter

Scope	Name	Type	Description
Input	xExecute	BOOL	Starts execution at the rising edge.
	xAbort	BOOL	TRUE: Stops execution and resets all outputs.
	udiTimeOut	UDINT	Timeout time until the execution is stopped (μs)
	sFileNameDest	FILE.CAA.FILENAME	Copy destination file name
	sFileNameSource	FILE.CAA.FILENAME	Copy source file name
	xOverWrite	BOOL	TRUE: Copies to overwrite an existing file. FALSE: Outputs an error without copying to overwrite. If FALSE is specified in a case where there is an existing file, copy is not executed. No error is output.
Output	xDone	BOOL	TRUE: Execution is completed.
	xBusy	BOOL	TRUE: Execution of the FB is not completed.
	xError	BOOL	TRUE: An error has occurred within the FB.

Scope	Name	Type	Description
	xAborted	BOOL	TRUE: Execution is stopped by the user.
	eError	FILE.ERROR	An error ID is output. Refer to "10.7.15 FILE.ERROR (Error ID)".
	szSize	FILE.CAA.SIZE	Size of the copied file

Info.

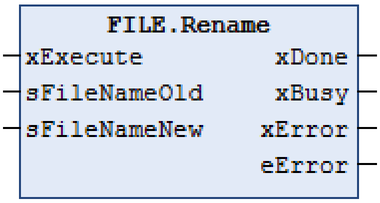
- You cannot use full size characters and the following symbols in a file name: [\], [/], [:], [*], [?], ["], [<], [>], [|].

10.7.7 FILE.Rename (Rename File)

This is a function block that changes a file name.

It is not possible to change the directory name of a directory that is currently open. Close it using the DirClose function block.

■ Icon



■ Parameter

Scope	Name	Type	Description
Input	xExecute	BOOL	Starts execution at the rising edge.
	sFileNameOld	FILE.CAA.FILENAME	File name before change
	sFileNameOld	FILE.CAA.FILENAME	File name after change
Output	xDone	BOOL	TRUE: Execution is completed.
	xBusy	BOOL	TRUE: Execution of the FB is not completed.
	xError	BOOL	TRUE: An error has occurred within the FB.
	eError	FILE.ERROR	An error ID is output. Refer to "10.7.15 FILE.ERROR (Error ID)".

Info.

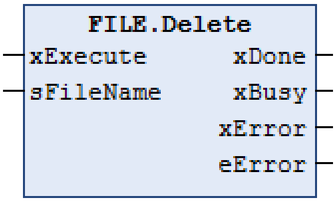
- You cannot use full size characters and the following symbols in a file name: [\], [/], [:], [*], [?], ["], [<], [>], [|].

10.7.8 FILE.Delete (Delete File)

This is a function block that deletes a file.

It is not possible to delete a file that is currently open. Close it using the Close function block.

■ Icon



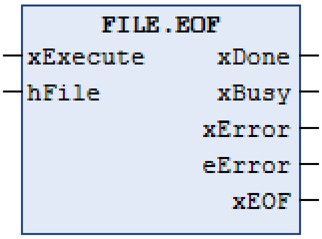
■ Parameter

Scope	Name	Type	Description
Input	xExecute	BOOL	Starts execution at the rising edge.
	sFileName	FILE.CAA.FILENAME	File to be deleted
Output	xDone	BOOL	TRUE: Execution is completed.
	xBusy	BOOL	TRUE: Execution of the FB is not completed.
	xError	BOOL	TRUE: An error has occurred within the FB.
	eError	FILE.ERROR	An error ID is output. Refer to "10.7.15 FILE.ERROR (Error ID)".

10.7.9 FILE.EOF (End of File)

This is a function block that determines whether the current offset of a file is EOF (End Of File) or not. It can be used only when the OPEN mode is set to MREAD/MREADPLUS.

■ Icon



■ Parameter

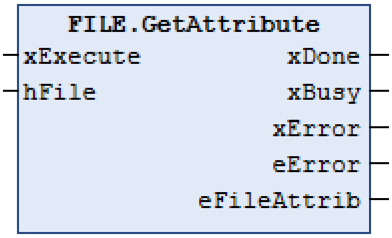
Scope	Name	Type	Description
Input	xExecute	BOOL	Starts execution at the rising edge.

Scope	Name	Type	Description
	hFile	FILE.CAA.FILENAME	Handle of a file Specifies the handle output by FILE.Open.
Output	xDone	BOOL	TRUE: Execution is completed.
	xBusy	BOOL	TRUE: Execution of the FB is not completed.
	xError	BOOL	TRUE: An error has occurred within the FB.
	eError	FILE.ERROR	An error ID is output. Refer to "10.7.15 FILE.ERROR (Error ID)".
	xEOF	BOOL	File: The current offset is EOF.

10.7.10 FILE.GetAttribute (Get File Attribute)

This is a function block that gets file attributes.

■ Icon



■ Parameter

Scope	Name	Type	Description
Input	xExecute	BOOL	Starts execution at the rising edge.
	hFile	FILE.CAA.FILENAME	Handle of a file Specifies the handle output by FILE.Open.
Output	xDone	BOOL	TRUE: Execution is completed.
	xBusy	BOOL	TRUE: Execution of the FB is not completed.
	xError	BOOL	TRUE: An error has occurred within the FB.
	eError	FILE.ERROR	An error ID is output. Refer to "10.7.15 FILE.ERROR (Error ID)".
	eFileAttrib	FILE.ATTRIB	TRUE: The current offset is EOF. FALSE: The current offset is not EOF.

■ FILE.ATTRIB (Enumeration type)

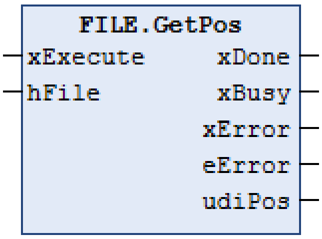
Name	Value	Description
ARCHIVE	0	Archive file
HIDDEN	1	Hidden file

Name	Value	Description
NORMAL	2	File without any other attributes
READONLY	3	Read only

10.7.11 FILE.GetPos (Get File Offset)

This is a function block that gets the current offset of a file.

■ Icon



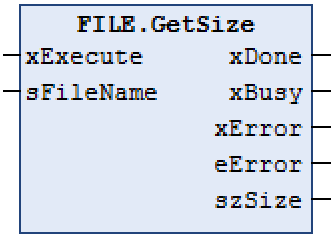
■ Parameter

Scope	Name	Type	Description
Input	xExecute	BOOL	Starts execution at the rising edge.
	hFile	FILE.CAA.FILENAME	Handle of a file Specifies the handle output by FILE.Open.
Output	xDone	BOOL	TRUE: Execution is completed.
	xBusy	BOOL	TRUE: Execution of the FB is not completed.
	xError	BOOL	TRUE: An error has occurred within the FB.
	eError	FILE.ERROR	An error ID is output. Refer to "10.7.15 FILE.ERROR (Error ID)".
	udiPos	__UXINT	The current offset (byte) is output.

10.7.12 FILE.GetSize (Get File Size)

This is a function block that gets the file size.

■ **Icon**



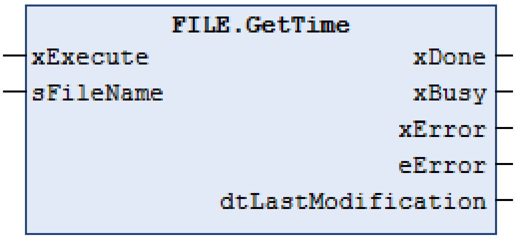
■ **Parameter**

Scope	Name	Type	Description
Input	xExecute	BOOL	Starts execution at the rising edge.
	sFileName	FILE.CAA.FILENAME	File from which to get the file size
Output	xDone	BOOL	TRUE: Execution is completed.
	xBusy	BOOL	TRUE: Execution of the FB is not completed.
	xError	BOOL	TRUE: An error has occurred within the FB.
	eError	FILE.ERROR	An error ID is output. Refer to "10.7.15 FILE.ERROR (Error ID)".
	szSize	FILE.CAA.SIZE	The file size (byte) is output.

10.7.13 FILE.GetTime (Get File Update Time)

This is a function block that gets the update time of a file.

■ **Icon**



■ **Parameter**

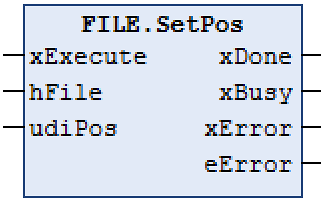
Scope	Name	Type	Description
Input	xExecute	BOOL	Starts execution at the rising edge.
	sFileName	FILE.CAA.FILENAME	File from which to get the file update time
Output	xDone	BOOL	TRUE: Execution is completed.

Scope	Name	Type	Description
	xBusy	BOOL	TRUE: Execution of the FB is not completed.
	xError	BOOL	TRUE: An error has occurred within the FB.
	eError	FILE.ERROR	An error ID is output. Refer to "10.7.15 FILE.ERROR (Error ID)".
	dtLastModification	DATE_AND_TIME	The last update date and time is output. Example: DATE_AND_TIME#2020-01-11-15:12:30

10.7.14 FILE.SetPos (Set File Offset)

This is a function block that sets the offset of a file.

■ Icon



■ Parameter

Scope	Name	Type	Description
Input	xExecute	BOOL	Starts execution at the rising edge.
	hFile	FILE.CAA.HANDLE	Handle of a file Specifies the handle output by FILE.Open.
	udiPos	__UXINT	Offset to be set (byte)
Output	xDone	BOOL	TRUE: Execution is completed.
	xBusy	BOOL	TRUE: Execution of the FB is not completed.
	xError	BOOL	TRUE: An error has occurred within the FB.
	eError	FILE.ERROR	An error ID is output. Refer to "10.7.15 FILE.ERROR (Error ID)".

10.7.15 FILE.ERROR (Error ID)

This is an enumeration type error ID that is output when a function block for file operation is executed. It is used to output an error in a file operation or directory operation of the SD card.

■ FILE.ERROR (Enumeration type)

Name	Value	Description
NO_ERROR	0	Normal end

Name	Value	Description
FIRST_ERROR	5100	First library specific error
TIME_OUT	5101	Timeout
ABORT	5102	Aborts processing by xAbort.
HANDLE_INVALID	5103	Invalid handle
NOT_EXIST	5104	No file or directory exists.
EXIST	5105	A file or directory already exists
NO_MORE_ENTRIES	5106	There are no other entries.
NOT_EMPTY	5107	The file or directory is not empty.
READ_ONLY_CAA	5108	The file or directory is write protected.
WRONG_PARAMETER	5109	Wrong parameter
ERROR_UNKNOWN	5110	Unknown error
WRITE_INCOMPLETE	5111	Not all the data is written.
FILE_NOT_IMPLEMENTED	5112	The function is not implemented.
ASM_CREATEJOB_FAILED	5113	Failed to create an AsyncManager job.
FILE_OPERATION_DENIED	5114	No access due to ForceFilePath / ForceIecFilePath
FIRST_MF	5150	First error unique to the manufacturer
LAST_ERROR	5199	insert manuf. specific errors here Last library specific error

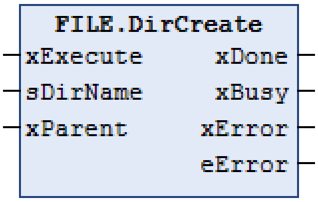
10.8 SD Card Operation (Directory Operation)

Directories in the SD card inserted in the SD memory card slot can be operated.

10.8.1 FILE.DirCreate (Create Directory)

This is a function block that creates a directory. An error occurs if there already exists a sub-directory.

■ Icon



■ Parameter

Scope	Name	Type	Description
Input	xExecute	BOOL	Starts execution at the rising edge.
	sDirName	FILE.CAA.FILENAME	Specifies a new directory name with an absolute path or relative path.
	xParent	BOOL	TRUE: Automatically creates a non-existing sub-directory. FALSE: An error occurs if there already exists a sub-directory.
Output	xDone	BOOL	TRUE: Execution is completed.
	xBusy	BOOL	TRUE: Execution of the FB is not completed.
	xError	BOOL	TRUE: An error has occurred within the FB.
	eError	FILE.ERROR	An error ID is output. Refer to "10.7.15 FILE.ERROR (Error ID)".

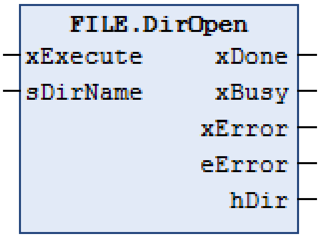
Info.

- You cannot use full size characters and the following symbols in a directory name: [\], [/], [:], [*], [?], ["], [<], [>], [|].

10.8.2 FILE.DirOpen (Open Directory)

This is a function block that opens a directory.

■ **Icon**



■ **Parameter**

Scope	Name	Type	Description
Input	xExecute	BOOL	Starts execution at the rising edge.
	sDirName	FILE.CAA.FILENAME	Specifies a directory name with an absolute path or relative path.
Output	xDone	BOOL	TRUE: Execution is completed.
	xBusy	BOOL	TRUE: Execution of the FB is not completed.
	xError	BOOL	TRUE: An error has occurred within the FB.
	eError	FILE.ERROR	An error ID is output. Refer to "10.7.15 FILE.ERROR (Error ID)".
	hDir	FILE.CAA.HANDLE	Handle of the FILE.CAA.HANDLE directory

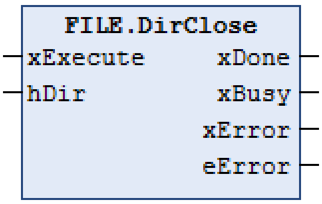
Info.

- You cannot use full size characters and the following symbols in a directory name: [\], [/], [:], [*], [?], ["], [<], [>], [|].

10.8.3 FILE.DirClose (Close Directory)

This is a function block that closes a directory.

■ **Icon**



■ **Parameter**

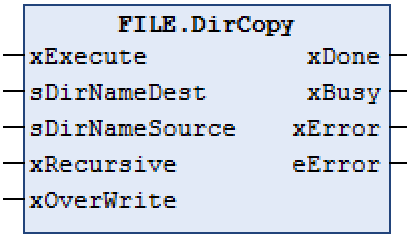
Scope	Name	Type	Description
Input	xExecute	BOOL	Starts execution at the rising edge.

Scope	Name	Type	Description
	hFile	FILE.CAA.FILENAME	Handle of the directory to be closed Specifies the handle output by FILE.Open.
Output	xDone	BOOL	TRUE: Execution is completed.
	xBusy	BOOL	TRUE: Execution of the FB is not completed.
	xError	BOOL	TRUE: An error has occurred within the FB.
	eError	FILE.ERROR	An error ID is output. Refer to "10.7.15 FILE.ERROR (Error ID)".

10.8.4 FILE.DirCopy (Copy Directory)

This is a function block that copies a directory.

■ Icon



■ Parameter

Scope	Name	Type	Description
Input	xExecute	BOOL	Starts execution at the rising edge.
	sDirNameDest	FILE.CAA.FILENAME	Directory name of the copy destination
	sDirNameSource	FILE.CAA.FILENAME	Directory of the copy source
	xRecursive	BOOL	TRUE: Copies the sub-directory and files.
	xOverWrite	BOOL	TRUE: Copies to overwrite an existing file.
Output	xDone	BOOL	TRUE: Execution is completed.
	xBusy	BOOL	TRUE: Execution of the FB is not completed.
	xError	BOOL	TRUE: An error has occurred within the FB.
	eError	FILE.ERROR	An error ID is output. Refer to "10.7.15 FILE.ERROR (Error ID)".

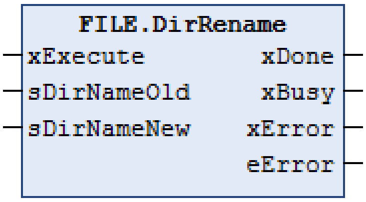
Info.

- You cannot use full size characters and the following symbols in a directory name: [\], [/], [:], [*], [?], [”], [<], [>], [|].

10.8.5 FILE.DirRename (Rename Directory)

This is a function block that renames a directory name. It is not possible to change the directory name of a directory that is currently open. Close it using the DirClose function block.

■ Icon



■ Parameter

Scope	Name	Type	Description
Input	xExecute	BOOL	Starts execution at the rising edge.
	sDirNameOld	FILE.CAA.FILENAME	Directory name before change
	sDirNameNew	FILE.CAA.FILENAME	Directory name after change
Output	xDone	BOOL	TRUE: Execution is completed.
	xBusy	BOOL	TRUE: Execution of the FB is not completed.
	xError	BOOL	TRUE: An error has occurred within the FB.
	eError	FILE.ERROR	An error ID is output. Refer to "10.7.15 FILE.ERROR (Error ID)".

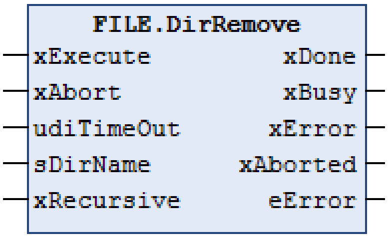
Info.

- You cannot use full size characters and the following symbols in a directory name: [\], [/], [:], [*], [?], [”], [<], [>], [|].

10.8.6 FILE.DirRemove (Delete Directory)

This is a function block that deletes a directory. It is not possible to delete a directory that is currently open. Close it using the DirClose function block.

■ **Icon**



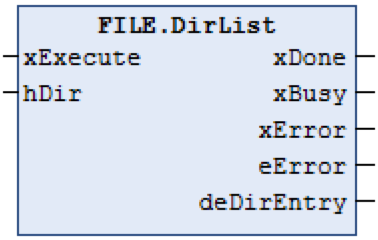
■ **Parameter**

Scope	Name	Type	Description
Input	xExecute	BOOL	Starts execution at the rising edge.
	xAbort	BOOL	TRUE: Stops execution and resets all outputs.
	udiTimeOut	UDINT	Timeout time until the execution is stopped (μs)
	sDirName	FILE.CAA.FILENAME	Specifies a directory name with an absolute path or relative path.
	xRecursive	BOOL	TRUE: Deletes the sub-directory and all files. FALSE: Deletes only when the directory is empty. An error occurs if the directory is not empty.
Output	xDone	BOOL	TRUE: Execution is completed.
	xBusy	BOOL	TRUE: Execution of the FB is not completed.
	xError	BOOL	TRUE: An error has occurred within the FB.
	xAborted	BOOL	TRUE: Execution is stopped by the user.
	eError	FILE.ERROR	An error ID is output. Refer to "10.7.15 FILE.ERROR (Error ID)".

10.8.7 FILE.DirList (Directory List)

This is a function block that outputs a list of directories and files inside the directory.

■ **Icon**



■ Parameter

Scope	Name	Type	Description
Input	xExecute	BOOL	Starts execution at the rising edge.
	hDir	FILE.CAA.HANDLE	Directory from which to output a list Specifies the handle output by FILE.Open.
Output	xDone	BOOL	TRUE: Execution is completed.
	xBusy	BOOL	TRUE: Execution of the FB is not completed.
	xError	BOOL	TRUE: An error has occurred within the FB.
	eError	FILE.ERROR	An error ID is output. Refer to "10.7.15 FILE.ERROR (Error ID)".
	deDirEntry	FILE.FILE_DIR_ENTRY	Files and directories are output.

■ FILE_DIR_ENTRY (Structure)

Member	Type	Description
sEntry	FILE.CAA.FILENAME	Directory or file name
szSize	FILE.CAA.SIZE	File size
xDirectory	BOOL	TRUE: Directory FALSE: File
xExclusive	BOOL	TRUE: Exclusive access mode FALSE: Multiple access mode
dtLastModification	DATE_AND_TIME	Last update date and time.

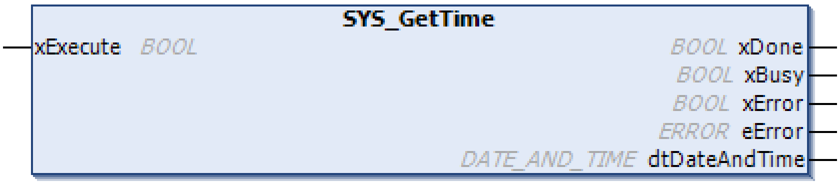
10.9 Clock Setting

This section describes function blocks that are used to set the clock of the GM1 Controller.
Enter a function block name by using the RTCLK (namespace).

10.9.1 SYS_GetTime (Get Time)

This is a function block (FB) that gets the current local time

■ Icon



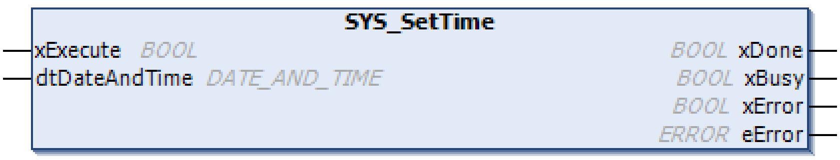
■ Parameter

Scope	Name	Type	Description
Input	xExecute	BOOL	TRUE: Active FALSE: Stop
Output	xDone	BOOL	TRUE: The function block is normally ended.
	xBusy	BOOL	TRUE: The function block is active.
	xError	BOOL	TRUE: An error has occurred.
	eError	ERROR	Details of error contents
	dtDateAndTime	DT	Current local time

10.9.2 SYS_SetTime (Set Time)

This is a function block (FB) that sets the current local time.

■ Icon



■ Parameter

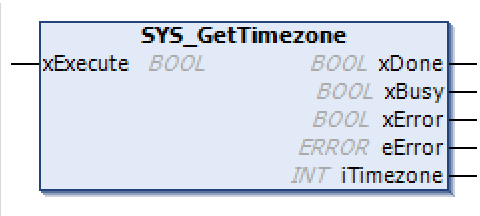
Scope	Name	Type	Description
Input	xExecute	BOOL	TRUE: Active

Scope	Name	Type	Description
			FALSE: Stop
	dtDateAndTime	DT	Current time to be set
Output	xDone	BOOL	TRUE: The function block is normally ended.
	xBusy	BOOL	TRUE: The function block is active.
	xError	BOOL	TRUE: An error has occurred.
	eError	ERROR	Details of error contents

10.9.3 SYS_GetTimezone (Get Time Zone Information)

This is a function block (FB) that gets the time zone information.

■ Icon



■ Parameter

Scope	Name	Type	Description
Input	xExecute	BOOL	TRUE: Active FALSE: Stop
Output	xDone	BOOL	TRUE: The function block is normally ended.
	xBusy	BOOL	TRUE: The function block is active.
	xError	BOOL	TRUE: An error has occurred.
	eError	ERROR	Details of error contents
	iTimezone	INT	Time zone information (Offset from UTC)

10.9.4 SYS_SetTimezone (Set Time Zone Information)

This is a function block (FB) that sets the time zone information.

■ **Icon**

■ **Parameter**

Scope	Name	Type	Description
Input	xExecute	BOOL	TRUE: Active FALSE: Stop
	iTimezone	INT	Time zone information (Offset from UTC)
Output	xDone	BOOL	TRUE: The function block is normally ended.
	xBusy	BOOL	TRUE: The function block is active.
	xError	BOOL	TRUE: An error has occurred.
	eError	ERROR	Details of error contents

10.9.5 SYS_DateConcat (Convert from UINT Type to DATE Type)

This is a function (FUN) that converts a UINT type date to a DATE type.

■ **Icon**

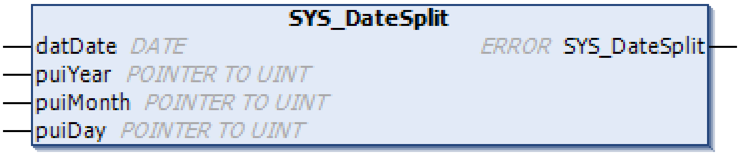
■ **Parameter**

Scope	Name	Type	Description
Input	uiYear	UINT	Year: 1970 to 2099
	uiMonth	UINT	Month: 1 to 12
	uiDay	UINT	Day: 1 to 31
	peError	POINTER TO ERROR	Pointer to the error information storage location
Output	SYS_DateConcat	DATE	Return value: Returns DT#1970-01-01 if the input value is invalid.

10.9.6 SYS_DateSplit (Convert from DATE Type to UINT Type)

This is a function (FUN) that converts a DATE type date to a UINT type.

■ Icon



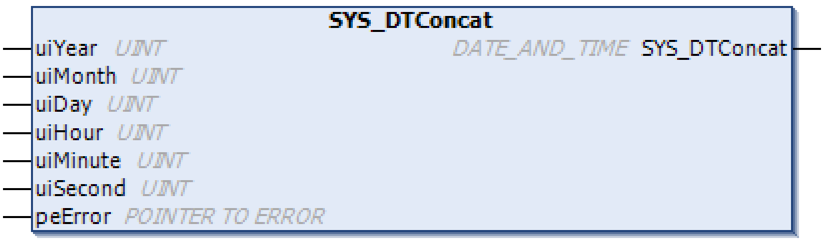
■ Parameter

Scope	Name	Type	Description
Input	datDate	DATE	Date data
	puiYear	POINTER TO UINT	Pointer to the year data storage location: 1970 to 2099
	puiMonth	POINTER TO UINT	Pointer to the month data storage location: 1 to 12
	puiDay	POINTER TO UINT	Pointer to the day data storage location: 1 to 31
Output	SYS_DateSplit	ERROR	Return value: Error information

10.9.7 SYS_DTConcat (Convert from UINT Type to DT Type)

This is a function (FUN) that converts a UINT type date and time to a DT type.

■ Icon



■ Parameter

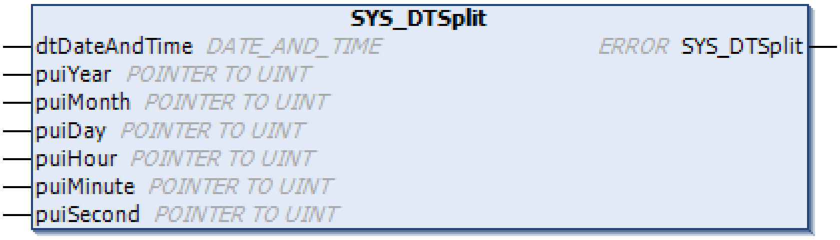
Type	Name	Type	Description
Input	uiYear	UINT	Year: 1970 to 2099
	uiMonth	UINT	Month: 1 to 12
	uiDay	UINT	Day: 1 to 31

Type	Name	Type	Description
	uiHour	UINT	Hour: 0 to 23
	uiMinute	UINT	Minute: 0 to 59
	uiSecond	UINT	Second: 0 to 59
	peError	POINTER TO ERROR	Pointer to the error information
Output	SYS_DTConcat	DT	Return value: Returns DT#1970-01-01-00:00:00 if the input value is invalid.

10.9.8 SYS_DTSplit (Convert from UINT Type to DT Type)

This is a function (FUN) that converts a UINT type date and time to a DT type.

■ Icon



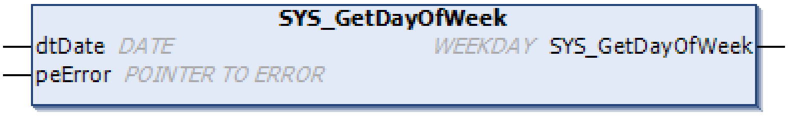
■ Parameter

Scope	Name	Type	Description
Input	dtDateAndTime	DT	Date and time data
	uiYear	POINTER TO UINT	Pointer to the year data storage location: 1970 to 2099
	uiMonth	POINTER TO UINT	Pointer to the month data storage location: 1 to 12
	uiDay	POINTER TO UINT	Pointer to the day data storage location: 1 to 31
	uiHour	POINTER TO UINT	Pointer to the hour data storage location: 0 to 23
	uiMinute	POINTER TO UINT	Pointer to the minute data storage location: 0 to 59
	uiSecond	POINTER TO UINT	Pointer to the second data storage location: 0 to 59
Output	SYS_DTSplit	ERROR	Return value: Error information

10.9.9 SYS_GetDayOfWeek (Get Day of the Week)

This is a function (FUN) that gets the day of the week from the DATE type date.

■ Icon



■ Parameter

Scope	Name	Type	Description
Input	dtDate	DATE	Date data
	peError	POINTER TO ERROR	Pointer to the error information
Output	SYS_GetDayOfWeek	RTCLK.WEEKDAY	Return value: Day of the week

RTCLK.WEEKDAY (Day of the week)

Name	Value	Description
SUNDAY	16#00	Sunday
MONDAY	16#01	Monday
TUESDAY	16#02	Tuesday
WEDNESDAY	16#03	Wednesday
THURSDAY	16#04	Thursday
FRIDAY	16#05	Friday
SATURDAY	16#06	Saturday

10.9.10 SYS_TODConcat (Convert from UINT Type to TOD Type)

This is a function (FUN) that converts a UINT type time with milliseconds to a TOD type.

■ Icon

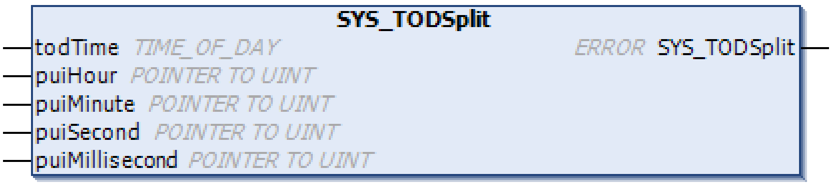
■ Parameter

Scope	Name	Type	Description
Input	uiHour	UINT	Hour: 0 to 23
	uiMinute	UINT	Minute: 0 to 59
	uiSecond	UINT	Second: 0 to 59
	uiMillisecond	UINT	Millisecond: 0 to 999
	peError	POINTER TO ERROR	Pointer to the error information
Output	SYS_TODConcat	TOD	Return value Returns TOD#00:00:00 if the input value is invalid.

10.9.11 SYS_TODSplit (Convert from TOD Type to UINT Type)

This is a function (FUN) that converts a TOD type time with milliseconds to a UINT type.

■ Icon



■ Parameter

Scope	Name	Type	Description
Input	todTime	TOD	Time (hour, minute, second) with millisecond data
	puiHour	POINTER TO UINT	Pointer to the hour data storage location: 0 to 23
	puiMinute	POINTER TO UINT	Pointer to the minute data storage location: 0 to 59
	puiSecond	POINTER TO UINT	Pointer to the second data storage location: 0 to 59
	puiMillisecond	POINTER TO UINT	Pointer to the millisecond data storage location: 0 to 999
Output	SYS_TODSplit	ERROR	Return value: Error information

10.9.12 ERROR (Clock Instruction Error Code)

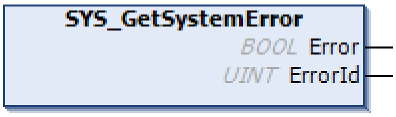
Name	Value	Description
NO_ERROR	0	No error
FIRST_ERROR	5700	First error unique to the library
TIME_OUT	5751	The time limit is exceeded.
NOT_AVAILABLE	5752	Not available.
INPUT_VALID	5753	Invalid input value
DTU_ERROR_UNKNOWN	5754	Unknown error
DTU_WRONG_PARAMETER	5755	Wrong parameter
DTU_TZI_NOT_SET	5756	The time zone information has not been initialized.
FIRST_MF	5770	First error unique to the manufacturer
LAST_ERROR	5799	Last error unique to the library

10.10 System Data

10.10.1 SYS_GetSystemError (Get System Error)

This is a function block that gets the information of a system error that has occurred in the GM1 Controller.

■ Icon



■ Parameter

Scope	Name	Type	Description
Input	None		
Output	Error	BOOL	TRUE: An error has occurred.
	ErrorId	UDINT	Error ID of the error that has occurred

Info.

- For the error ID, refer to the *GM1 Controller RTEX User’s Manual (Operation Edition)*.

11 Reference Information

11.1 Motion Errors (SMC_ERROR Type)

This section describes errors that are output in motion control instructions and their contents.
Motion control errors are defined in SMC_ERROR.

11.1.1 Error Check Method

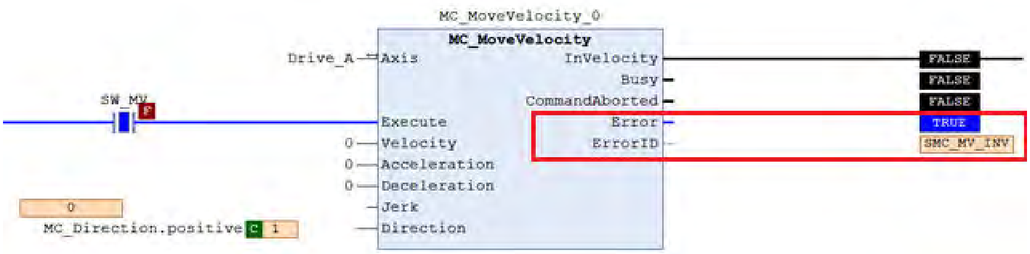
This section describes errors that are output in motion control instructions and their contents.
Motion control errors are defined in SMC_ERROR.

■ Error check method

With a function block that has an output Error and output ErrorID, it is possible to check that an error has occurred.

The following shows an example of an error that has occurred when the MC_MoveVelocity function block is executed.

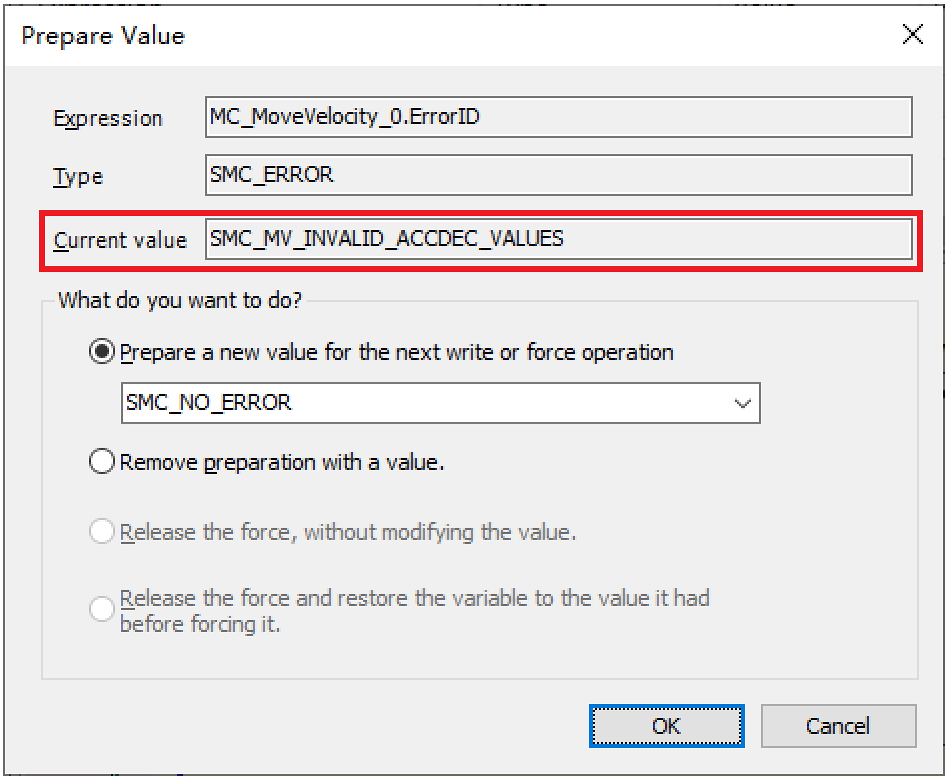
“TRUE” is output to the output Error and an error is output to the output ErrorID.



The error name can be checked by double-clicking the output ErrorID.

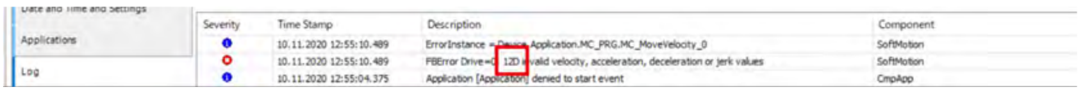
An error name defined in the enumeration type SMC_ERROR is displayed in the "Current value" field in the "Presetting values" dialog box.

Double-clicking SMC_MV_INV in the above execution example displays the following dialog box where error name "SMC_MV_INVALID_ACCDEC_VALUES" can be checked.



When an error occurs, the value of the error that has occurred (SMC_ERROR) is also recorded in hexadecimal number on the "Log" screen of the device editor.

The following example shows a record when an error ("SMC_MV_INVALID_ACCDEC_VALUES") with an error value of 12D (301 in decimal) has occurred.



11.1.2 SMC_ERROR

Error name	Value	Description
SMC_NO_ERROR	0	No error
SMC_DI_GENERAL_COMMUNICATION_ERROR	1	Communication error Communication disconnection or another communication problem occurred.
SMC_DI_AXIS_ERROR	2	Axis error Amplifier alarm or another axis problem occurred.
SMC_DI_FIELDBUS_LOST_SYNCRONICITY	3	The fieldbus lost synchronicity.
SMC_DI_SWLIMITS_EXCEEDED	10	The software limit has been exceeded.
SMC_DI_HWLIMITS_EXCEEDED	11	The hardware end switch is active.

Error name	Valu e	Description
SMC_DI_LINEAR_AXIS_OUTOFRANGE	12	An overflow occurred in the linear axis.
SMC_DI_HALT_OR_QUICKSTOP_NOT_SUPPO RTED	13	The drive state is set to Halt or the Quickstop is unsupported.
SMC_DI_VOLTAGE_DISABLED	14	No power is supplied to the drive.
SMC_DI_IRREGULAR_ACTPOSITION	15	This error is not used.
SMC_DI_POSITIONLAGERROR	16	Position lag error The difference between the commanded position and actual position has exceeded the specified limit when position lag monitoring is active.
SMC_DI_HOMING_ERROR	17	Home return error
SMC_REGULATOR_OR_START_NOT_SET	20	Either the controller is disabled or the brake has been applied. Servo OFF or another similar problem occurred during axis movement.
SMC_WRONG_CONTROLLER_MODE	21	The executed function block is set to unsupported controller mode (SMC_SetControllerMode).
SMC_INVALID_ACTION_FOR_LOGICAL	25	Invalid operation was performed on the logical axis.
SMC_FB_WASNT_CALLED_DURING_MOTION	30	The function block was not called on the POU while the motion instruction was being executed ("Busy"). The operation was stopped while the motion instruction was being executed ("Busy").
SMC_AXIS_IS_NO_AXIS_REF	31	The type of the AXIS_REF type variable is different.
SMC_AXIS_REF_CHANGED_DURING_OPERATI ON	32	The AXIS_REF variable was changed during operation.
SMC_FB_ACTIVE_AXIS_DISABLED	33	The axis became disabled (MC_Power.bRegulatorOn) during movement.
SMC_AXIS_NOT_READY_FOR_MOTION	34	The axis cannot execute a motion instruction (an attempt was made to execute a motion instruction with MC_Stop enabled, for example).
SMC_AXIS_ERROR_DURING_MOTION	35	An error (such as amplifier alarm) occurred during motion operation.
SMC_VD_MAX_VELOCITY_EXCEEDED	40	The maximum velocity (fMaxVelocity) was exceeded.
SMC_VD_MAX_ACCELERATION_EXCEEDED	41	The maximum acceleration (fMaxAccleration) was exceeded.
SMC_VD_MAX_DECELERATION_EXCEEDED	42	The maximum deceleration (fMaxDeceleration) was exceeded.
SMC_3SH_INVALID_VELACC_VALUES	50	Either an invalid velocity or acceleration was specified.
SMC_3SH_MODE_NEEDS_HWLIMIT	51	For safety reasons, a request is made to invoke the mode in which the end switch is used.
SMC_FRC_NO_FREE_HANDLE	60	There is no file open handle.

Error name	Value	Description
SMC_MAC_INITIALIZATION_FAILED	65	SMC_MultiAcyclicCommunicator initialization failed.
SMC_MAC_INVALID_TASK_HANDLE	66	There is an invalid handle for the axis.
SMC_MAC_TOO_MANY_TASKS	67	There are too many tasks that use an axis generating SDO.
SMC_MAC_ATOMIC_ADD_FAILED	68	An attempt to add Atomic failed.
SMC_SDO_INVALID_DATALENGTH	69	An invalid data length (> 4) occurred due to SDO reading.
SMC_SCM_NOT_SUPPORTED	70	An invalid controller mode was set for SMC_SetControllerMode.
SMC_SCM_AXIS_IN_WRONG_STATE	71	The controller mode cannot be changed in the current axis state (an attempt was made to execute SMC_SetControllerMode with MC_Stop enabled, for example).
SMC_SCM_INTERRUPTED	72	SMC_SetControllerMode was interrupted by MC_Stop or ErrorStop.
SMC_ST_WRONG_CONTROLLER_MODE	75	The motion instruction was executed in an incorrect controller mode.
SMC_RAG_ERROR_DURING_STARTUP	80	An error occurred when the axis group was started up.
SMC_RAG_ERROR_AXIS_NOT_INITIALIZED	81	The axis is not in the specified state.
SMC_PP_WRONG_AXIS_TYPE	85	The function block does not support virtual axes or logical axes.
SMC_PP_NUMBER_OF_ABSOLUTE_BITS_INVALID	86	The number of bits is invalid (between 8 and 32 bits).
SMC_CGR_ZERO_VALUES	90	An invalid value was specified.
SMC_CGR_DRIVE_POWERED	91	A gear parameter was changed while the drive was in operation.
SMC_CGR_INVALID_POSPERIOD	92	An invalid position (0 or less, or half or more than the bus bandwidth) was specified.
SMC_CGR_POSPERIOD_NOT_INTEGRAL	93	The modulo period is not an integer.
SMC_P_FTASKCYCLE_EMPTY	110	There is no information in one cycle time. (fTaskCycle = 0)
SMC_R_NO_ERROR_TO_RESET	120	There is no error to be reset (MC_Reset was executed when there was no function block error, for example).
SMC_R_DRIVE_DOESNT_ANSWER	121	There is no response to an error reset.
SMC_R_ERROR_NOT_RESETTABLE	122	An error reset cannot be executed.
SMC_R_DRIVE_DOESNT_ANSWER_IN_TIME	123	Communication with the axis is not working.
SMC_R_CANNOT_RESET_COMMUNICATION_ERROR	124	A reset cannot be executed due to a communication error.
SMC_RP_PARAM_UNKNOWN	130	The parameter number is undefined.
SMC_RP_REQUESTING_ERROR	131	An error occurred in communication with the drive.
SMC_RP_DRIVE_PARAMETER_NOT_MAPPED	132	Parameters are not assigned to the drive.

Error name	Value	Description
SMC_RP_PARAM_CONVERSION_ERROR	133	Conversion of drive parameter values failed. Soft motion parameters are undefined.
SMC_WP_PARAM_INVALID	140	The parameter number is undefined or write operations are inhibited.
SMC_WP_SENDING_ERROR	141	Refer to the error number for WriteDriveParameter.
SMC_WP_DRIVE_PARAMETER_NOT_MAPPED	142	Parameters are undefined for the drive.
SMC_WP_PARAM_CONVERSION_ERROR	143	Conversion of drive parameter values failed. Soft motion parameters are undefined.
SMC_H_AXIS_WASNT_STANDSTILL	170	The axis is not in a standstill state.
SMC_H_AXIS_DIDNT_START_HOMING	171	An error occurred when home return was started.
SMC_H_AXIS_DIDNT_ANSWER	172	An error occurred when home return was started.
SMC_H_ERROR_WHEN_STOPPING	173	An error occurred after the axis stopped in home return mode. It is possible that deceleration was not set.
SMC_H_AXIS_IN_ERRORSTOP	174	The drive is in the ErrorStop state. Home return cannot be executed.
SMC_MS_UNKNOWN_STOPPING_ERROR	180	Undefined error
SMC_MS_INVALID_ACCDEC_VALUES	181	Either an invalid velocity or acceleration was specified.
SMC_MS_DIRECTION_NOT_APPLICABLE	182	"shortest" cannot be applied to the direction.
SMC_MS_AXIS_IN_ERRORSTOP	183	Because the drive is in the ErrorStop state, stop operation cannot be executed with MC_Stop.
SMC_BLOCKING_MC_STOP_WASNT_CALLED	184	MC_Stop (Execute=TRUE) blocks the axis. MC_Stop (Execute=FALSE) must be executed.
SMC_UNKNOWN_TASK_INTERVAL	200	The task interval of the bus task is undefined.
SMC_MA_INVALID_VELACC_VALUES	201	Either an invalid velocity or acceleration was specified.
SMC_MA_INVALID_DIRECTION	202	There is an error in the specified direction ("Direction").
SMC_MR_INVALID_VELACC_VALUES	226	Either an invalid velocity or acceleration was specified.
SMC_MR_INVALID_DIRECTION	227	There is an error in the specified direction ("Direction").
SMC_MAD_INVALID_VELACC_VALUES	251	Either an invalid velocity or acceleration was specified.
SMC_MAD_INVALID_DIRECTION	252	There is an error in the specified direction ("Direction").
SMC_MSI_INVALID_VELACC_VALUES	276	Either an invalid velocity or acceleration was specified.
SMC_MSI_INVALID_DIRECTION	277	There is an error in the specified direction ("Direction").
SMC_MSI_INVALID_EXECUTION_ORDER	278	The same instance of MC_MoveSuperImposed was called more than once in a single cycle.

Error name	Valu e	Description
SMC_LOGICAL_NO_REAL_AXIS	300	Unused error
SMC_MV_INVALID_ACCDEC_VALUES	301	Either an invalid velocity or acceleration was specified.
SMC_MV_DIRECTION_NOT_APPLICABLE	302	"shortest" or "fastest" cannot be applied to the direction ("Direction").
SMC_PP_ARRAYSIZE	325	There is an error in the specified array size.
SMC_PP_STEP0MS	326	The step time is 0s.
SMC_VP_ARRAYSIZE	350	There is an error in the specified array size.
SMC_VP_STEP0MS	351	The step time is 0s.
SMC_AP_ARRAYSIZE	375	There is an error in the specified array size.
SMC_AP_STEP0MS	376	The step time is 0s.
SMC_TP_TRIGGEROCCUPIED	400	The trigger is already enabled.
SMC_TP_COULDNT_SET_WINDOW	401	The drive does not support the window function.
SMC_TP_COMM_ERROR	402	Communication error
SMC_AT_TRIGGERNOTOCCUPIED	410	The trigger is already disabled.
SMC_MCR_INVALID_VELACC_VALUES	426	Either an invalid velocity or acceleration was specified.
SMC_MCR_INVALID_DIRECTION	427	An invalid direction was specified.
SMC_MCA_INVALID_VELACC_VALUES	451	Either an invalid velocity or acceleration was specified.
SMC_MCA_INVALID_DIRECTION	452	An invalid direction was specified.
SMC_MCA_DIRECTION_NOT_APPLICABLE	453	"fastest" cannot be applied to the direction ("Direction").
SMC_SDL_INVALID_AXIS_STATE	475	Function block "SMC_ChangeDynamicLimits" was executed in a state other than "Standstill" or "Power_off".
SMC_SDL_INVALID_VELACC_VALUES	476	An invalid velocity, acceleration, deceleration, or jerk was specified.
SMC_CR_NO_TAPPETS_IN_CAM	600	The cam is not equipped with a tappet.
SMC_CR_TOO_MANY_TAPPETS	601	The tappet group ID exceeds MAX_NUM_TAPPETS.
SMC_CR_MORE_THAN_32_ACCESSES	602	There are 32 or more accesses to one cam.
SMC_CI_NO_CAM_SELECTED	625	No cam is selected. It is possible that the correct cam table is not set in the CamTableID parameter of MC_CamIn.
SMC_CI_MASTER_OUT_OF_SCALE	626	The current commanded position on the master axis is outside the range of the cam table.
SMC_CI_RAMPIN_NEEDS_VELACC_VALUES	627	A velocity and acceleration must be specified when StartMode is set to ramp_in.
SMC_CI_SCALING_INCORRECT	628	The scaling variables (fEditor, TableMasterMin, and Max) are incorrect.
SMC_CI_TOO_MANY_TAPPETS_PER_CYCLE	629	The number of tappet outputs is too many to be enabled in one cycle.

Error name	Value	Description
SMC_CB_NOT_IMPLEMENTED	640	The function block for cam format is not implemented.
SMC_GI_RATIO_DENOM	675	RatioDenominator (denominator of the gear ratio) is set to 0.
SMC_GI_INVALID_ACC	676	The value specified in "Acceleration" is invalid.
SMC_GI_INVALID_DEC	677	The value specified in "Deceleration" is invalid.
SMC_GI_MASTER_REGULATOR_CHANGED	678	The Enable / Disable state of the master axis was changed without permission.
SMC_GI_INVALID_JERK	679	The value specified in "Jerk" is invalid.
SMC_PH_INVALID_VELACCDEC	725	The specified velocity, acceleration, or deceleration were invalid.
SMC_PH_ROTARYAXIS_PERIOD0	726	fPositionPeriod for the rotation axis is set to 0.
SMC_NO_CAM_REF_TYPE	750	The cam type is not an MC_CAM_REF structure.
SMC_CAM_TABLE_DOES_NOT_COVER_MASTER_SCALE	751	The master axis area (xStart and xEnd) of the cam table is outside the curve data range.
SMC_CAM_TABLE_EMPTY_MASTER_RANGE	752	The cam data table is empty.
SMC_CAM_TABLE_INVALID_MASTER_MINMAX	753	The maximum value and minimum value of the master axis in the cam data are invalid.
SMC_CAM_TABLE_INVALID_SLAVE_MINMAX	754	The maximum value and minimum value of the slave axis in the cam data are invalid.
SMC_GIP_MASTER_DIRECTION_CHANGE	775	The rotation direction of the master axis was changed while the salve axis was connected.
SMC_GIP_SLAVE_REVERSAL_CANNOT_BE_AVOIDED	776	The AvoidReversal input is set, but reverse rotation of the slave axis cannot be avoided.
SMC_GIP_AVOID_REVERSAL_FOR_FINITE_AXIS	777	The AvoidReversal input cannot be set for the finite slave axis.
SMC_BC_BL_TOO_BIG	800	The fBacklash gear backlash is too large (larger than position period/2).
SMC_QPROF_DIVERGES	825	Internal error: Failed in calculating the secondary path
SMC_QPROF_INVALID_PARAMETER	826	Internal error: Failed in calculating the secondary path
SMC_QPROF_NO_RESULT	827	Internal error: Failed in calculating the secondary path
SMC_QPROF_INVALID_NEW_LBD	828	Internal error: Failed in calculating the secondary path
SMC_QPROF_BAD_NEGOTIATION	829	Internal error: Failed in calculating the secondary path
SMC_QPROF_INVALID_INTERVAL	830	Internal error: Failed in calculating the secondary path
SMC_QPROF_NOT_ENOUGH_PHASES	831	Internal error: Failed in calculating the secondary path
SMC_TG_INTERNAL_ERROR	832	Internal error: Failed in calculating the secondary path

Error name	Value	Description
SMC_SRT_NOT_STANDSTILL_OR_POWEROFF	850	Execution is possible only in the standstill or power_off state.
SMC_SRT_INVALID_RAMPTYPE	851	The value specified in RampType is invalid.
SMC_SMT_NOT_STANDSTILL_OR_POWEROFF	852	Execution is possible only in the standstill or power_off state.
SMC_SMT_INVALID_MOVEMENTTYPE_OR_POSITIONPERIOD	853	The value specified in MovementType or PositionPeriod is invalid.
SMC_SMT_AXIS_NOT_VIRTUAL	854	The function block is valid only for the virtual axis.
SMC_NO_LICENSE	1000	License error
SMC_INT_VEL_ZERO	1001	Because Velocity is set to 0, path processing cannot be performed.
SMC_INT_NO_STOP_AT_END	1002	The final velocity of the path is other than 0.
SMC_INT_DATA_UNDERRUN	1003	GEOINFO-List was processed by DataIn, but the end of the list has not been reached.
SMC_INT_VEL_NONZERO_AT_STOP	1004	The velocity at the time of stoppage is greater than 0.
SMC_INT_TOO_MANY_RECURSIONS	1005	There are too many recursions of SMC_Interpolator.
SMC_INT_NO_CHECKVELOCITIES	1006	SMC_CheckVelocities is not called by SMC_OUTQUEUE.
SMC_INT_PATH_EXCEEDED	1007	Internal error or calculation error
SMC_INT_VEL_ACC_DEC_ZERO	1008	The specified velocity and acceleration / deceleration are 0 or less.
SMC_INT_DWIPOTIME_ZERO	1009	The motion task was called when dwIpoTime = 0.
SMC_INT_JERK_NONPOSITIVE	1010	A negative value was set for "Jerk".
SMC_INT_QPROF_DIVERGES	1011	Internal error The calculation algorithm is incorrect.
SMC_INT_INVLALID_VELOCITY_MODE	1012	The specified velocity mode is invalid.
SMC_INT_TOO_MANY_AXES_INTERPOLATED	1013	The maximum allowable number of axes for interpolation has been exceeded.
SMC_INT_DEGENERATE_SEGMENT	1014	
SMC_INT2DIR_BUFFER_TOO_SMALL	1050	
SMC_INT2DIR_PATH_FITS_NOT_IN_QUEUE	1051	
SMC_XINT_INVALID_DIRECTION	1070	
SMC_XINT_NOINTERSECTION	1071	
SMC_WAR_INT_OUTQUEUE_TOO_SMALL	1080	
SMC_WAR_END_VELOCITIES_INCORRECT	1081	The specified final velocity is incorrect.
SMC_CV_ACC_DEC_VEL_NONPOSITIVE	1100	Negative values are specified for the velocity and acceleration/deceleration.
SMC_CA_INVALID_ACCDEC_VALUES	1120	Negative values are specified for fGapVelocity, fGapAcceleration, and fGapDeceleration.
SMC_DEC_ACC_TOO_LITTLE	1200	The specified acceleration is unacceptable.

Error name	Valu e	Description
SMC_DEC_RET_TOO_LITTLE	1201	The specified deceleration is unacceptable.
SMC_DEC_OUTQUEUE_RAN_EMPTY	1202	Data underrun The queue was read, but it was empty.
SMC_DEC_JUMP_TO_UNKNOWN_LINE	1203	Because the line number is unknown, the cursor cannot jump to the line.
SMC_DEC_INVALID_SYNTAX	1204	The syntax is invalid.
SMC_DEC_3DMODE_OBJECT_NOT_SUPPORTED	1205	The object is not supported in 3D mode.
SMC_DEC_NEGATIVE_PERIOD	1206	A negative value is specified for the period during which an additional axis is disabled.
SMC_DEC_DIMENSIONS_EXCLUSIVE_AU	1207	Both axis A and axis U are not always interpolated. PA and PU are mutually exclusive.
SMC_DEC_DIMENSIONS_EXCLUSIVE_BV	1208	Both axis B and axis V are not always interpolated. PB and PV are mutually exclusive.
SMC_DEC_DIMENSIONS_EXCLUSIVE_CW	1209	Both axis C and axis W are not always interpolated. PC and PW are mutually exclusive.
SMC_IPR_TOO_SMALL__BUFFER	1259	The buffer size specified for OutOueue is too small.
SMC_GCV_BUFFER_TOO_SMALL	1300	
SMC_GCV_BUFFER_WRONG_TYPE	1301	
SMC_GCV_UNKNOWN_IPO_LINE	1302	
SMC_NO_CNC_REF_TYPE	1500	
SMC_NO_OUTQUEUE_TYPE	1501	The specified pointer is not SMC_OUTQUEUE.
SMC_GEOINFO_BUFFER_MISALIGNED	1502	The buffer segments aligned by four-byte boundaries are not used by pbyBuffer.
SMC_3D_MODE_NOT_SUPPORTED	1600	The FB functions only with 2D paths.

11.2 RTEX communication error

11.2.1 RTEX Error ID

■ WARNING_CODE (Union type)

Member	Type	Description
uiWarningCode	UINT	Warning code
tWarningCodeMember	ALARM_WARNING_CODES	Main code (warning number) and sub-code (0) of the warning code

■ List of RTEX Error IDs

Category	Error_Code / Sub_Error_Code	Cause
Command header related	0011h	● Mismatched node address (MAC-ID)
	0012h	● C/R bit is 1 despite of command. ● Sub_Chk is 0 in 32-byte mode.
Command code, control mode related	0021h	● Cyclic command is not defined.
	0022h	● Non-cyclic command is not defined (when cyclic command is normal). ● Combination error of control mode and non-cyclic command. ● Subcommand is undefined in 32-byte mode.
	002Eh	● Combination of communication cycle, semi-closed/full-closed, 16 / 32 byte mode, and control mode is not correct. ● Control mode has been changed in less than 2 ms. ● Control mode has been changed during profile position latch positioning / profile home return (Type_Code = 12h, 13h, 31h, 32h, 33h, 34h, 36h) operation. ● Control mode has been changed during execution of non-cyclic command (Busy = 1). ● Run the home return command (□4h) Type_Code = 1□h / 2□h during the velocity control (CV) / torque control (CT). ● Control mode has been changed to the velocity control during the 2 degrees of freedom control (synchronous) mode. ● Control mode has been changed to the torque control during the 2 degrees of freedom control (standard / synchronous) mode. ● Control mode has been changed during the retracting operation.
Argument related	0031h	● Type_Code / Sub_Type_Code is not defined.
	0032h	● Non-cyclic data / sub-command data other than Type_Code / Sub_Type_Code is out of setup range.
	0033h	● Cyclic data (Command_Data1) is out of setup range.
	0034h	● Feed forward data (Command_Data3, Sub_Command_Data2 / 3) is out of setup range.
Not executable 1 (general)	0041h	● Write access is attempted to read only media.
	0042h	● Alarm clear command is executed while an alarm that cannot be cleared has occurred and no warning was issued.

Category	Error_Code / Sub_Error_Code	Cause
	0043h	● External scale error clear command is executed when not in full-closed control mode or when no external scale error is detected.
	0045h	● In servo on state, reset command is executed in attribute C parameter validation mode.
	0046h	● After deceleration and stop according to the drive inhibit input (POT / NOT), direction command POT / NOT is applied. ● During deceleration according to the drive inhibit input (POT / NOT), a profile operation (except Type_Code = 31h, 32h, 33h, 34h, and 36h) is started.
Not executable 2 (related to home return)	0051h	● Multi-turn clearing of the home return command was executed while the encoder was in the incremental mode. ● Multi-turn clearing of the home return command was executed even when the single-turn absolute function was effective.
	0052h	● During cyclic position control (CP) (* including full-closed control) in absolute mode, Type_Code = 1□h of the home return command (□4h) has been executed. ● During profile position control (PP) (* including full-closed control) in absolute mode, profile home return has been executed.
	0053h	● During cyclic position control (CP) (* including full-closed control) in absolute mode, actual position set / command position set (Type_Code = 21h, 22h) of the home return command (□4h) have been executed.
	0055h	● Multi-turn clearing of the home return command is executed while in the full-closed control mode.
	0056h	● Multi-turn clearing of the home return command is executed while in the servo-on condition.
	0057h	● Type_Code = 1□h of the home return command is executed while in the servo-off state.
	0058h	● While the external input is not assigned to the latch correction terminal, Type_Code is executed by using the external input as a trigger. ● Started the latch mode with a stop function operated by the amplifier output signal as the trigger signal when Pr7.111 “Trigger signal assignment setting for the latch mode with a stop function” was set to 0 “Disabled”.
	0059h	● Executed the home return command (□4h) while the profile position latch positioning / profile home return (Type_Code = 12h, 13g, 31h, 32h, 33h, 34h, 36h) was operated. ● During profile positioning / profile continuous revolution (Type_Code = 10h, 11h, 20h), initialization mode (Type_Code = 1□h, 31h) of home return command (□4h) has been executed.
	005Ah	● Z phase is set to latch trigger signal despite absolute external scale.
	005Bh	● Received the following commands in the virtual full-closed control mode. • Home return command (□4h) • Profile position latch absolute positioning (12h) of the profile command (17h) • Profile position latch absolute positioning (13h) of the profile command (17h) • Profile home return (31h to 34h, 36h) of the profile command (17h) • Config command

Category	Error_Code / Sub_Error_Code	Cause
		● Received a command to change to the virtual full-closed control mode under the following conditions. • While initialization mode of home return command (□4h) was operated, latch mode was operated, or latch mode with stop function was operated • Changed to a command other than command code (□4h) after starting home return command (Type_Code: 51h to 53h) • During a period from starting the latch to detecting the latch after starting home return command (Type_Code: 51h to 53h) • While profile position latch absolute positioning (12h) of the profile command (17h) was operated • While profile position latch absolute positioning (13h) of the profile command (17h) was operated • While profile home return (31h to 34h, 36h) of the profile command (17h) was operated • After starting profile command (12h, 13h, 31h to 34h, 36h), during the period from when a change was made to a command other than command code (17h) until the latch or home was detected • While Config command was executed
	005Fh	● Latch mode with stop function (Type_Code = F1h) was used in a setting other than the cyclic position control (CP). ● Latch mode with stop function (Type_Code = F1h) was used in a setting other than the communication cycle of 0.5 ms/command update cycle of 1.0 ms ● Latch mode with stop function (Type_Code = F1h) was used in a setting other than the electronic gear ratio of less than 1.
Not executable 3 (Related to hardware factor)	0061h	● EEPROM writing is not permitted because of under voltage of the control power.
Not executable 4 (in process)	0101h	● Not permitted to be accepted because the previous command is in process.
	0102h	● Command is not permitted to be accepted because the servo driver is accessing to the encoder now.
	0103h	● Command is not permitted to be accepted because the servo driver is accessing to the external scale now.
	0104h	● Type_Code has been changed while operating under profile position control (PP).
	0105h	● During execution of the PANATERM command (test run operation, FFT, Z phase search, pin assignment setting, or fit gain), received the RTEX command (reset command, home return command, or parameter command).
Not executable 5 (access inhibited)	0201h	● Command is not permitted to be accepted because parameter writing or writing to EEPROM is inhibited now. ● Write parameter command or write EEPROM command is issued while bit 0 of Pr7.23 RTEX function expansion setup 2 is set at 1.

11.2.2 Alarm Codes

■ ALARM_CODE (Union type)

Member	Type	Description
uiAlarmCode	UINT	Alarm code
tAlarmCodeMember	ALARM_WARNING_CODES	Main code and sub-code of the alarm code

■ List of alarm codes

Error No.		Alarm name	Attribute		
Main	Sub		History	Can be cleared	Emergency stop(Note 6)
11	0	Control power supply undervoltage protection		○	
12	0	Over-voltage protection	○	○	
13	0	Main power supply undervoltage protection (Insufficient voltage between P and N)		○	○
	1	Main power supply undervoltage protection (AC interception detection)		○	○
14	0	Over-current protection	○		
	1	IPM error protection	○		
15	0	Overheat protection	○		○
	1	Encoder overheat error protection	○		○
16	0	Overload protection	○	○(Note 1)	
	1	Torque saturation error protection	○	○	
18	0	Regenerative overload protection	○		○
	1	Regenerative transistor error protection	○		
21	0	Encoder communication line breakage fault protection	○		
	1	Encoder communication error protection	○		
23	0	Encoder communication data error protection	○		
24	0	Position deviation excess protection	○	○	○
	1	Speed deviation excess protection	○	○	○
25	0	Hybrid deviation excess protection	○		○
26	0	Overspeed protection	○	○	○
	1	2nd overspeed protection	○	○	
27	1	Absolute clearing protection	○		
	4	Command error protection	○		○
	5	Command generation error protection	○		○
	6	Operation command contention protection	○	○	
	7	Position information initialization error protection	○		

Error No.		Alarm name	Attribute		
Main	Sub		History	Can be cleared	Emergency stop(Note 6)
28	0	Pulse regeneration limit protection	○	○	○
29	1	Counter overflow protection 1	○		
	2	Counter overflow protection 2	○		
31	0	Safety function error protection 1	○		
	2	Safety function error protection 2	○		
33	0	Input duplicated allocation error-1 protection	○		
	1	Input duplicated allocation error-2 protection	○		
	2	Input function number error-1 protection	○		
	3	Input function number error-2 protection	○		
	4	Output function number error-1 protection	○		
	5	Output function number error-2 protection	○		
	8	Latch input allocation error protection	○		
34	0	Motor operable range setting error protection	○	○	
	1	One revolution absolute operable range error protection	○	○	
36	0 to 1	EEPROM parameter error protection			
37	0 to 2	EEPROM check code error protection			
38	0	Over-travel inhibit input protection 1		○	
	1	Over-travel inhibit input protection 2		○	
	2	Over-travel inhibit input protection 3	○		
40	0	Absolute system failure protection	○	○(Note 2)	
41	0	Absolute counter limit excess protection	○		
42	0	Absolute overspeed protection	○	○(Note 2)	
44	0	Single-turn counter error protection	○		
45	0	Multi-turn counter error protection	○		
47	0	Absolute status error protection	○		
50	0	External scale wiring error protection	○		
	1	External scale communication error protection	○		
	2	External scale communication data error protection	○		
51	0	External scale ST error protection 0	○		
	1	External scale ST error protection 1	○		
	2	External scale ST error protection 2	○		
	3	External scale ST error protection 3	○		
	4	External scale ST error protection 4	○		
	5	External scale ST error protection 5	○		
55	0	Phase-A wiring error protection	○		

Error No.		Alarm name	Attribute		
Main	Sub		History	Can be cleared	Emergency stop(Note 6)
	1	Phase-B wiring error protection	○		
	2	Phase-Z wiring error protection	○		
70	0	Phase U current detector error protection	○		
	1	Phase W current detector error protection	○		
72	0	Thermal relay error protection	○		
80	3	PLL incomplete error protection	○	○	
82	0	RTEX node addressing error protection	○		
83	0	RTEX continuous communication error protection 1	○	○	○
	1	RTEX continuous communication error protection 2	○	○	○
84	0	RTEX Communication timeout error protection	○	○	○
	3	RTEX communication synchronization error protection	○		
	5	RTEX communication cycle error protection	○	○	○
85	0	Retracting operation completion (I/O)(Note 7)	○	(Note 8)	○
	2	Retracting operation error(Note 7)	○	(Note 8)	○
86	0	RTEX cyclic data error protection 1	○	○	○
	1	RTEX cyclic data error protection 2	○	○	○
	2	RTEX update counter error protection	○		○
87	0	Forced alarm input protection		○	○
	1	Retracting operation completion (I/O)(Note 7)	○	(Note 8)	○
	3	Retracting operation error(Note 7)	○	(Note 8)	○
90	2	RTEX multi-axis synchronization establishment error protection	○		
91	1	RTEX command error protection	○	○	
	3	RTEX command error protection 2	○	○	
92	0	Encoder data restoration error protection	○		
	1	External scale data restoration error protection	○		
	3	Multi-turn data upper-limit value mismatch error protection	○		
93	0	Parameter setting error protection 1	○		
	2	Parameter setting error protection 2	○		
	3	External scale connection error protection	○		
	5	Parameter setting error protection 4	○		
	8	Parameter setting error protection 6	○		
94	2	Home return error protection	○	○	
	3	Home return error protection 2	○	○	

Error No.		Alarm name	Attribute		
Main	Sub		History	Can be cleared	Emergency stop(Note 6)
95	0 to 4	Motor automatic recognition error protection			
96	2	Control unit error protection 1	○		
	3	Control unit error protection 2	○		
	4	Control unit error protection 3	○		
	5	Control unit error protection 4	○		
	6	Control unit error protection 5	○		
	7	Control unit error protection 6	○		
98	1	RTEX hardware error protection 1	○		
	2	RTEX hardware error protection 2	○		
	3	RTEX hardware error protection 3	○		
Other numbers		Other error protections	-	-	-

(Note 1) When Err 16.0 “Over-load protection” occurs, it can be cleared approx. 10 seconds after it occurs. The alarm clear command is received as is and clearing process takes place after it is ready to be cleared.

(Note 2) When Err 40.0 “Absolute system failure protection” or Err 42.0 ”Absolute overspeed protection” occurs, the error cannot be cleared until absolute clear is performed.

(Note 3) When an alarm that cannot be cleared occurs, cycle the control power supply after removing the cause of the error or use RTEX software reset command to clear the alarm.

(Note 4) When an alarm that can be cleared occurs, use RTEX communication or USB communication (setup support software) to clear the alarm. Always clear the alarm while all axes are stopped and after securing safety.

(Note 5) If the internal control circuit of the servo amplifier malfunctions due to excessive noise etc., the display will be as shown below.

In such a case, immediately turn OFF the power.

(Note 6) Emergency stop refers to an alarm that is triggered if Pr 5.10 “Sequence at alarm” is set to 4 to 7 and that causes an immediate stop. For details, refer to the instruction manual and other technical references for the servo amplifier.

(Note 7) The alarm generated during retracting operation is switched by Pr 6.86 “Retreat operation alarm setup” bit 15.

Example: When bit 15 = 0, Err 85.0 and Err 85.2 will occur (A5N compatible specification).

When bit 15 = 1, Err 87.1 and Err 87.3 will occur (A6B compatible specification).

(Note 8) Whether alarm can be cleared or not is determined by the setting (bit 0 or 2) of Pr 6.86.

Bit 0: Err 85.0 / Err 87.1 (Retracting operation completion (I/O)) alarm clear attribute

Bit 2: Err 85.2 / Err 87.3 (Retracting operation error) alarm clear attribute; For either case, 0: Alarm clear invalid, 1: Alarm clear valid

11.2.3 Warning Codes

■ ALARM_WARNING_CODES (Structure)

Member	Type	Description
byMainCode	BYTE	Main code
bySubCode	BYTE	Sub-code

■ General warnings

Warning No. (hexadecimal)	Warning name	Description	Warning latch	Output setting	Warning mask
			Pr6.27 (Note 1)	Pr4.40 / Pr4.41 (Note 2)	Pr6.38 / Pr6.39 Corresponding bit (Note 3)
A0	Overload warning Warning	Load factor is 85% or more of the protection level.	○	1	Pr6.38 bit7
A1	Over-regeneration warning	Regenerative load factor has exceeded 85% of the protection level.	○	2	Pr6.38 bit5
A2	Battery warning (Note 4)	Battery voltage is 3.2 V or less.	Latch fixed	3	Pr6.38 bit0
A3	Fan warning	Fan has stopped for 1 second.	○	4	Pr6.38 bit6
A4	Encoder communication warning	The number of successive encoder communication errors has exceeded the specified value.	○	5	Pr6.38 bit4
A5	Encoder overheat warning	The encoder temperature exceeds the specified value.	○	6	Pr6.38 bit3
A6	Oscillation detection warning	Oscillation state was detected.	○	7	Pr6.38 bit13
A7	Lifetime detection warning	The remaining life expectancy of a capacitor or a fan dropped below the specified value.	Latch fixed	8	Pr6.38 bit2
A8	External scale error warning	The external scale detected a warning.	○	9	Pr6.38 bit8
A9	External scale communication warning	The number of successive external scale communication errors has exceeded the specified value.	○	10	Pr6.38 bit14
AC	Deterioration diagnosis warning (Note 6)	Load characteristic estimated value or torque command value at a constant velocity has exceeded the set range.	○	22	Pr6.39 bit7

■ Extended warning

Warning No. (hexadecimal)	Warning name	Description	Warning latch Pr6.27 (Note 1)	Output setting Pr4.40 / Pr4.41 (Note 2)	Warning mask Pr6.38 / Pr6.39 Corresponding bit (Note 3)
C0	RTEX continuous communication error warning	The number of successive errors (CRC error) detected when reading the received data sent to the local node The number of successive errors (CRC error) has exceeded the value set by Pr 7.26 “RTEX continuous communication error warning setup”.	○	11	Pr6.38 bit9
C1	RTEX accumulated communication error warning	The number of successive errors (CRC error) detected when reading the received data sent to the local node has exceeded the value set by Pr 7.27 “RTEX accumulated communication error warning setup”.	Latch fixed	12	Pr6.38 bit10
C2	RTEX_ Update_Counter error warning	The Update_Counter was not updated properly because the data accumulated exceeded the count value set by Pr 7.28 “RTEX_Update_Counter error warning setup”.	Latch fixed	13	Pr6.38 bit11
C3	Main power OFF warning	When Pr 7.14 “Main power OFF warning detection time” was set to 10 to 1999, instantaneous power failure that occurred between L1 and L3 exceeded the time set by Pr 7.14.	○	14	Pr6.38 bit12
D2	PANATERM command execution warning	While bit 0 of Pr 7.99 “RTEX function enhancement setting 6” was set to 1 and RTEX communication was established, an operation command (test run, FFT, etc.) was executed by the setup support software “PANATERM”.	○	30	Pr6.39 bit8

(Note 1) The symbol “○” marked in the “Warning latch” column indicates that it is possible to switch the mode between non-latch mode (latch for 1 second) and latch mode by using Pr 6.27 ”Warning latch state setup”. Only latch mode is available for the battery warning and the lifetime detection warning.

(Note 2) Select the warning output signal 1 (WARN 1) or warning output signal 2 (WARN 2) through Pr 4.40 “Warning output select 1” or Pr 4.41 “Warning output select 2”. When the set value is 0, all warnings are Ored before being output. Do not use any settings other than the settings shown in the above table.

(Note 3) Each warning detection can be disabled by Pr 6.38 “Warning mask setting” or Pr 6.39 “Warning mask setting 2”. The corresponding bits are shown in the table. Set the bit to 1 to disable the warning detection. For extended warning, warning detection can be disabled by parameter settings.

Also note that bit arrangements of these masks are different from those of general-purpose type MINAS-A6 series.

(Note 4) When the single-turn absolute function is enabled, a battery alarm is not detected.

(Note 5) Warning can be cleared by alarm clear. If warning cause is not resolved yet, the warning is cleared once, but a warning is issued again.

(Note 6) If bit 1 of Pr 6.97 “Function enhancement setup 3” is set to 0, it is disabled.

11.3 List of AMP Parameters

11.3.1 Class 0: Basic Setting

Class	No.	Parameter name	Unit	Setting range
0	00	Rotational direction setup	-	0 to 1
	01	Control mode setup	-	0 to 6
	02	Real-time auto-gain tuning setup	-	0 to 6
	03	Selection of machine stiffness at real-time auto-gain tuning	-	0 to 31
	04	Inertia ratio	%	0 to 10000
	08	Command pulse counts per one motor revolution	pulse	0 to 2^{23}
	09	Numerator of electronic gear	-	0 to 2^{30}
	10	Denominator of electronic gear	-	1 to 2^{30}
	11	Number of output pulses per motor rotation	pulse/r	1 to 2097152
	12	Reversal of pulse output logic / output source selection	-	0 to 3
	13	1st torque limit	%	0 to 500
	14	Position deviation excess setup	Command unit	0 to 2^{30}
	15	Absolute encoder setup	-	0 to 4
	16	External regenerative resistor setup	-	0 to 3
	17	Load factor of external regenerative resistor selection	-	0 to 4

11.3.2 Class 1: Gain Adjustment

Class	No.	Parameter name	Unit	Setting range
1	00	1st gain of position loop	0.1/s	0 to 30000
	01	1st gain of velocity loop	0.1 Hz	1 to 32767
	02	1st time constant of velocity loop integration	0.1 ms	1 to 10000
	03	1st filter of speed detection	-	0 to 5
	04	1st time constant of torque filter	0.01 ms	0 to 2500
	05	2nd gain of position loop	0.1/s	0 to 30000
	06	2nd gain of velocity loop	0.1 Hz	1 to 32767
	07	2nd time constant of velocity loop integration	0.1 ms	1 to 10000
	08	2nd filter of speed detection	-	0 to 5

Class	No.	Parameter name	Unit	Setting range
	09	2nd time constant of torque filter	0.01 ms	0 to 2500
	10	Velocity feed forward gain	0.1%	0 to 4000
	11	Velocity feed forward gain	0.1%	0 to 4000
	12	Velocity feed forward gain	0.01 ms	0 to 6400
	13	Torque feed forward filter	0.01 ms	0 to 6400
	14	2nd gain setup	-	0 to 1
	15	Mode of position control switching	-	0 to 10
	16	Delay time of position control switching	0.1 ms	0 to 10000
	17	Level of position control switching	-	0 to 20000
	18	Hysteresis at position control switching	-	0 to 20000
	19	Position gain switching time	0.1 ms	0 to 10000
	20	Mode of velocity control switching	-	0 to 5
	21	Delay time of velocity control switching	0.1 ms	0 to 10000
	22	Level of velocity control switching	-	0 to 20000
	23	Hysteresis at velocity control switching	-	0 to 20000
	24	Mode of torque control switching	-	0 to 3
	25	Delay time of torque control switching	0.1 ms	0 to 10000
	26	Level of torque control switching	-	0 to 20000
	27	Hysteresis at torque control switching	-	0 to 20000

11.3.3 Class 2: Vibration Suppression Function

Class	No.	Parameter name	Unit	Setting range
2	00	Adaptive filter mode setup	-	0 to 6
	01	1st notch frequency	Hz	50 to 5000
	02	1st notch width selection	-	0 to 20
	03	1st notch depth selection	-	0 to 99
	04	2nd notch frequency	Hz	50 to 5000
	05	2nd notch width selection	-	0 to 20
	06	2nd notch depth selection	-	0 to 99
	07	3rd notch frequency	Hz	50 to 5000
	08	3rd notch width selection	-	0 to 20
	09	3rd notch depth selection	-	0 to 99

Class	No.	Parameter name	Unit	Setting range
	10	4th notch frequency	Hz	50 to 5000
	11	4th notch width selection	-	0 to 20
	12	4th notch depth selection	-	0 to 99
	13	Selection of damping filter switching	-	0 to 6
	14	1st damping frequency	0.1 Hz	0 to 3000
	15	1st damping filter setup	0.1 Hz	0 to 1500
	16	2nd damping frequency	0.1 Hz	0 to 3000
	17	2nd damping filter setup	0.1 Hz	0 to 1500
	18	3rd damping frequency	0.1 Hz	0 to 3000
	19	3rd damping filter setup	0.1 Hz	0 to 1500
	20	4th damping frequency	0.1 Hz	0 to 3000
	21	4th damping filter setup	0.1 Hz	0 to 1500
	22	Command smoothing filter	0.1 ms	0 to 10000
	23	Command FIR filter	0.1 ms	0 to 10000
	24	5th notch frequency	Hz	50 to 5000
	25	5th notch width selection	-	0 to 20
	26	5th notch depth selection	-	0 to 99
	27	1st vibration control width setting	-	0 to 1000
	28	2nd vibration control width setting	-	0 to 1000
	29	3rd vibration control width setting	-	0 to 1000
	30	4th vibration control width setting	-	0 to 1000

11.3.4 Class 3: Speed, Torque Control, Full-closed Control

Class	No.	Parameter name	Unit	Setting range
3	12	Acceleration time setting	ms/(1000 r/min)	0 to 10000
	13	Deceleration time setting	ms/(1000 r/min)	0 to 10000
	14	Sigmoid acceleration/ deceleration time setup	ms	0 to 10000
	17	Speed limit selection	-	0 to 1
	21	Speed limit value 1	r/min	0 to 20000
	22	Speed limit value 2	r/min	0 to 20000
	23	External scale selection	-	0 to 6
	24	External scale numerator of division	-	0 to 2^{23}
	25	External scale denominator of division	-	1 to 2^{23}

Class	No.	Parameter name	Unit	Setting range
	26	External scale reversal of direction	-	0 to 3
	27	External scale Z phase disconnection detection disable	-	0 to 1
	28	Hybrid deviation excess protection	Command unit	1 to 2^{27}
	29	Hybrid deviation clear setting	Rotation	0 to 100
	32	External scale movement judgment threshold at virtual full-closed control mode	External scale unit	0 to 65534

11.3.5 Class 4: I/O Monitor Setting

Class	No.	Parameter name	Unit	Setting range
4	00	SI1 input selection	-	0 to 00FFFFFFh
	01	SI2 input selection	-	0 to 00FFFFFFh
	02	SI3 input selection	-	0 to 00FFFFFFh
	03	SI4 input selection	-	0 to 00FFFFFFh
	04	SI5 input selection	-	0 to 00FFFFFFh
	05	SI6 input selection	-	0 to 00FFFFFFh
	06	SI7 input selection	-	0 to 00FFFFFFh
	07	SI8 input selection	-	0 to 00FFFFFFh
	10	SO1 output selection	-	0 to 00FFFFFFh
	11	SO2 output selection	-	0 to 00FFFFFFh
	12	SO3 output selection	-	0 to 00FFFFFFh
	16	Type of analog monitor 1	-	0 to 28
	17	Analog monitor 1 output gain	-	0 to 214748364
	18	Type of analog monitor 2	-	0 to 28
	19	Analog monitor 2 output gain	-	0 to 214748364
	21	Analog monitor output setup	-	0 to 2
	31	Positioning complete range	Command unit	0 to 2097152
	32	Positioning complete (In-position) output setup	-	0 to 10
	33	INP hold time	ms	0 to 30000
	34	Zero-speed	r/min	10 to 20000
	35	Speed coincidence range	r/min	10 to 20000
	36	At-speed (Speed arrival)	r/min	10 to 20000

Class	No.	Parameter name	Unit	Setting range
	37	Mechanical brake action at stalling setup	ms	0 to 10000
	38	Mechanical brake action at running setup	ms	0 to 32000
	39	Brake release speed setup	r/min	30 to 3000
	40	Selection of alarm output 1	-	0 to 40
	41	Selection of alarm output 2	-	0 to 40
	42	2nd Positioning complete (In-position) range	Command unit	0 to 2097152
	44	Position compare output pulse width setting	0.1 ms	0 to 32767
	45	Position compare output polarity select	-	0 to 7
	47	Pulse output select	-	0 to 1
	48	Position compare value 1	Command unit	-2147483648 to 2147483647
	49	Position compare value 2	Command unit	-2147483648 to 2147483647
	50	Position compare value 3	Command unit	-2147483648 to 2147483647
	51	Position compare value 4	Command unit	-2147483648 to 2147483647
	52	Position compare value 5	Command unit	-2147483648 to 2147483647
	53	Position compare value 6	Command unit	-2147483648 to 2147483647
	54	Position compare value 7	Command unit	-2147483648 to 2147483647
	55	Position compare value 8	Command unit	-2147483648 to 2147483647
	56	Position compare output delay compensation amount	0.1 us	-32768 to 32767
	57	Position compare output assignment setting	-	-2147483648 to 2147483647

11.3.6 Class 5: Enhancing Setting

Class	No.	Parameter name	Unit	Setting range
5	03	Denominator of pulse output division	-	0 to 8388608
	04	Over-travel inhibit input setup	-	0 to 2
	05	Sequence at over-travel inhibit	-	0 to 2

Class	No.	Parameter name	Unit	Setting range
	06	Sequence at Servo-Off	-	0 to 9
	07	Sequence at main power OFF	-	0 to 9
	08	LV trip selection at main power OFF	-	0 to 3
	09	Detection time of main power off	ms	20 to 2000"11.3.9 Class 8: Special Setting 3"
	10	Sequence at alarm	-	0 to 7
	11	Torque setup for emergency stop	%	0 to 500
	12	Over-load level setup	%	0 to 500
	13	Over-speed level setup	r/min	0 to 20000
	14	Motor working range setup	0.1 revolution	0 to 1000
	15	Control input signal read setting	-	0 to 3
	20	Position setup unit select	-	0 to 1
	21	Selection of torque limit	-	0 to 4
	22	2nd torque limit	%	0 to 500
	23	Torque limit switching setup 1	ms/100 %	0 to 4000
	24	Torque limit switching setup 2	ms/100 %	0 to 4000
	25	Positive direction torque limit	%	0 to 500
	26	Negative direction torque limit	%	0 to 500
	31	USB axis address	-	0 to 127
	33	Pulse regenerative output limit setup	-	0 to 1
	45	Quadrant projection positive direction compensation value	0.1%	-1000 to 1000
	46	Quadrant projection negative direction compensation value	0.1%	-1000 to 1000
	47	Quadrant projection compensation delay time	ms	0 to 1000
	48	Quadrant projection compensation filter setting L	0.01 ms	0 to 6400
	49	Quadrant projection compensation filter setting H	0.1 ms	0 to 10000
	56	Slow stop deceleration time setting	ms/(1000 r/min)	0 to 10000

Class	No.	Parameter name	Unit	Setting range
	57	Slow stop S-shape acceleration and deceleration setting	ms	0 to 1000
	66	Deterioration diagnosis convergence judgment time	0.1 s	0 to 10000
	67	Deterioration diagnosis inertia ratio upper limit	%	0 to 10000
	68	Deterioration diagnosis inertia ratio lower limit	%	0 to 10000
	69	Deterioration diagnosis unbalanced load upper limit	0.1%	-1000 to 1000
	70	Deterioration diagnosis unbalanced load lower limit	0.1%	-1000 to 1000
	71	Deterioration diagnosis dynamic friction upper limit	0.1%	-1000 to 1000
	72	Deterioration diagnosis dynamic friction lower limit	0.1%	-1000 to 1000
	73	Deterioration diagnosis viscous friction upper limit	0.1%/ (10000 r/min)	0 to 10000
	74	Deterioration diagnosis viscous friction lower limit	0.1%/ (10000 r/min)	0 to 10000
	75	Deterioration diagnosis velocity setting	r/min	-20000 to 20000
	76	Deterioration diagnosis torque average time	ms	0 to 10000
	77	Deterioration diagnosis torque upper limit	0.1%	-1000 to 1000
	78	Deterioration diagnosis torque lower limit	0.1%	-1000 to 1000

(Note 1) When using this setup value at a value smaller than the default value, confirm that it matches the user’s power supply environment.

11.3.7 Class 6: Special Setting 1

Class	No.	Parameter name	Unit	Setting range
6	02	Speed deviation excess setup	r/min	0 to 20000
	05	Position control 3rd gain effective time	0.1 ms	0 to 10000
	06	Position control 3rd gain scale factor	%	50 to 1000
	07	Additional value to torque command	%	-100 to 100

Class	No.	Parameter name	Unit	Setting range
	08	Torque compensation value in positive direction	%	-100 to 100
	09	Torque compensation value in negative direction	%	-100 to 100
	10	Function expansion setup	-	-32768 to 32767
	11	Current response setup	%	10 to 100
	14	Immediate stop time at the time of alarming	ms	1000
	15	2nd over-speed level setup	r/min	0 to 20000
	18	Power turn-on wait time	0.1 s	0 to 100
	22	A, B phase external scale pulse output method selection	-	0 to 1
	23	Load fluctuation correction gain	%	-100 to 100
	24	Load fluctuation correction filter	0.01 ms	10 to 2500
	27	Alarm latch time selection	-	0 to 3
	31	Real time auto tuning estimation speed	-	0 to 3
	32	Real time auto tuning custom setup	-	-32768 to 32767
	34	Hybrid vibration suppression gain	-	0 to 30000
	35	Hybrid vibration suppression filter	0.1/s	0 to 32000
	36	Dynamic brake operation input setup	0.01 ms	0 to 1
	37	Oscillation detecting level	-	0 to 1000
	38	Alarm mask setup	0.1%	-32768 to 32767
	39	Alarm mask setup 2	-	-32768 to 32767
	41	1st damping depth	-	0 to 1000
	42	Two-stage torque filter time constant	-	0 to 2500
	43	Two-stage torque filter damping term	0.01 ms	0 to 1000
	47	Function expansion settings 2	-32768 to 32767	-32768 to 32767
	48	Adjustment filter	0 to 2000	0 to 2000
	49	Command response filter / adjustment filter damping term setting	0 to 99	0 to 99
	50	Viscous friction compensation gain	0.1%/ (10000 r/min)	0 to 10000
	51	Immediate stop completion wait time	ms	0 to 10000
	57	Torque saturation error protection detection time	ms	0 to 5000
	60	2nd damping depth	-	0 to 1000
	61	1st resonance frequency	0.1 Hz	0 to 3000

Class	No.	Parameter name	Unit	Setting range
	62	1st resonance damping ratio	-	0 to 1000
	63	1st anti-resonance frequency	0.1 Hz	0 to 3000
	64	1st anti-resonance damping ratio	-	0 to 1000
	65	1st response frequency	0.1 Hz	0 to 3000
	66	2nd resonance frequency	0.1 Hz	0 to 3000
	67	2nd resonance damping ratio	-	0 to 1000
	68	2nd anti-resonance frequency	0.1 Hz	0 to 3000
	69	2nd anti-resonance damping ratio	-	0 to 1000
	70	2nd response frequency	0.1 Hz	0 to 3000
	71	3rd damping depth	-	0 to 1000
	72	4th damping depth	-	0 to 1000
	73	Load estimation filter	0.01 ms	0 to 2500
	74	Torque compensation frequency 1	0.1 Hz	0 to 5000
	75	Torque compensation frequency 2	0.1 Hz	0 to 5000
	76	Load estimation count	-	0 to 8
	85	Retracting operation condition setting	-	-32768 to 32767
	86	Retracting operation alarm setting	-	-32768 to 32767
	88	Absolute multi-rotation data upper limit	-	0 to 65534
	97	Function expansion setting 3	-	-2147483648 to 2147483647
	98	Function expansion setting 4	-	-2147483648 to 2147483647

11.3.8 Class 7: Special Setting 2

Class	No.	Parameter name	Unit	Setting range
7	00	Display on LED	-	0 to 32767
	01	Display time setup upon power-up	100 ms	-1 to 1000
	03	Output setup during torque limit	-	0 to 1
	09	Correction time of latch delay 1	25 ns	-2000 to 2000
	10	Soft limit function	-	0 to 3
	11	Positive side software limit value	Command unit	-1073741823 to 1073741823
	12	Negative side software limit value	Command unit	-1073741823 to 1073741823
	13	Absolute home position offset	Command unit	-1073741823 to 1073741823

Class	No.	Parameter name	Unit	Setting range
	14	Main power OFF warning detection time	ms	0 to 2000
	15	Positioning adjacent range	Command unit	0 to 1073741823
	16	Torque saturation error protection frequency	No. of times	0 to 30000
	20	RTEX communication cycle setup	-	-1 to 12
	21	RTEX command updating cycle ratio setting	-	1 to 2
	22	RTEX function extended setup 1	-	-32768 to 32767
	23	RTEX function extended setup 2	-	-32768 to 32767
	24	RTEX function extended setup 3	-	-32768 to 32767
	25	RTEX speed unit setup	-	0 to 1
	26	RTEX continuous error warning setup	No. of times	0 to 32767
	27	RTEX accumulated error warning setup	No. of times	0 to 32767
	28	RTEX_Update_Counter error warning setup	No. of times	0 to 32767
	29	RTEX monitor select 1	-	0 to 32767
	30	RTEX monitor select 2	-	0 to 32767
	31	RTEX monitor select 3	-	0 to 32767
	32	RTEX monitor select 4	-	0 to 32767
	33	RTEX monitor select 5	-	0 to 32767
	34	RTEX monitor select 6	-	0 to 32767
	35	RTEX command setting 1	-	0 to 2
	36	RTEX command setting 2	-	0 to 2
	37	RTEX command setting 3	-	0 to 2
	38	RTEX_Update_Counter error protection setup	No. of times	0 to 32767
	41	RTEX function extended setup 5	-	-32768 to 32767
	78	Signal reading setting for latch trigger with stop function	-	0 to 3
	91	RTEX communication cycle extended setup	ns	0 to 2000000
	92	Correction time of latch delay 2	25 ns	-2000 to 2000
	93	Home return limit speed	r/min	0 to 20000
	95	Number of RTEX continuous communication error protection 1 detections	No. of times	0 to 17
	96	Number of RTEX continuous communication error protection 2 detections	No. of times	0 to 17

Class	No.	Parameter name	Unit	Setting range
	97	Number of RTEX communication timeout error protection detections	No. of times	0 to 17
	98	Number of RTEX cyclic data error protection 1 / 2 detections	No. of times	0 to 17
	99	RTEX function extended setup 6	-	-32768 to 32767
	108	RTEX communication synchronization setup	-	0 to 7
	110	RTEX function extended setup 7	-	-2147483648 to 2147483647
	111	Trigger signal allocation setting of latch mode with stop function	-	0 to 64
	112	Selection of RTEX communication status flag	-	0 to 1

11.3.9 Class 8: Special Setting 3

Class	No.	Parameter name	Unit	Setting range
8	01	Profile linear acceleration constant	10000 Command unit/s^2	1 to 429496
	04	Profile linear deceleration constant	10000 Command unit/s^2	1 to 429496
	10	Amount of travel after profile position latch detection	Command unit	-1073741823 to 1073741823
	12	Profile home return position mode setup	-	0 to 1
	13	Profile home return velocity 1	Command unit/s or r/min	0 to 2147483647
	14	Profile home return velocity 2	Command unit/s or r/min	0 to 2147483647
	17	Relative movement of retracting operation	Command unit	-2147483648 to 2147483647
	18	Retracting operation speed	Command unit/s or r/min	0 to 2147483647

11.4 Monitor Commands

These commands are specified with RTEX_ReadAmpData (amplifier monitor).

<table>
<tr><th colspan="2">Type_Code
(Note 1) (Note 3)</th><th colspan="2" rowspan="2">Name</th><th rowspan="2">Index
(Note 2)</th><th rowspan="2">Unit</th><th rowspan="2">Description</th></tr>
<tr><th>A4N compatible</th><th>Standard</th></tr>
<tr><td>101h</td><td>01h</td><td>Position deviation</td><td>PERR</td><td>0
(1,2)</td><td>Command unit</td><td><In position control mode>
Position deviation
<In full-closed control mode>
External scale deviation
* The computation method (reference) of position deviation and external scale deviation is set in bit 14 of Pr 7.23 “Command position deviation output switching”.
<table>
<tr><th>r7.23 bit14</th><th>Computation method of positional deviation</th></tr>
<tr><td>0</td><td>Deviation from the command after filtering</td></tr>
<tr><td>1</td><td>Deviation from the command before filtering</td></tr>
</table>
<In speed / torque control mode>
Undefined
Note: Although the same data is returned whether Index is 1 or 2, use Index = 0.</td></tr>
<tr><td>102h</td><td>02h</td><td>Encoder resolution</td><td>-</td><td>0</td><td>pulse/r</td><td>Encoder resolution of the motor connected</td></tr>
<tr><td>104h</td><td>04h</td><td>Command position
(after filtering)</td><td>MPOS</td><td>0</td><td>Command unit</td><td>Command position (after filtering)</td></tr>
<tr><td>105h</td><td>05h</td><td>Actual speed</td><td>ASPD</td><td>0</td><td>Set the unit through Pr 7.25.</td><td>Motor actual speed
* Set the unit through Pr 7.25 “RTEX speed unit setup”.
<table>
<tr><th>Pr7.25</th><th>Unit</th></tr>
<tr><td>0</td><td>[r/min]</td></tr>
<tr><td>1</td><td>[Command unit/s]</td></tr>
</table></td></tr>
<tr><td>106h</td><td>06h</td><td>Internal command torque</td><td>TRQ</td><td>0</td><td>0.1%</td><td>Command torque to motor</td></tr>
<tr><td>-</td><td>07h</td><td>Actual position</td><td>APOS</td><td>0</td><td>Command unit</td><td>Motor actual position
* Position of the external scale in full-closed mode</td></tr>
<tr><td>-</td><td>08h</td><td>Internal command position (before filtering)</td><td>IPOS</td><td>0</td><td>Command unit</td><td>Internal command position before filtering</td></tr>
</table>

<table>
<tr><th colspan="2">Type_Code
(Note 1) (Note 3)</th><th colspan="2" rowspan="2">Name</th><th rowspan="2">Index
(Note 2)</th><th rowspan="2">Unit</th><th rowspan="2">Description</th></tr>
<tr><th>A4N compatible</th><th>Standard</th></tr>
<tr><td>-</td><td>09h</td><td>Latch position 1</td><td>LPOS1</td><td>0</td><td>Command unit</td><td>Motor actual position latched in CH1</td></tr>
<tr><td>-</td><td>0Ah</td><td>Latch position 2</td><td>LPOS2</td><td>0</td><td>Command unit</td><td>Motor actual position latched in CH2</td></tr>
<tr><td>-</td><td>0Ch</td><td>Command velocity
(after filtering)</td><td>MSPD</td><td>0</td><td>Set the unit through Pr 7.25.</td><td>Command velocity after filtering
* Set the unit through Pr 7.25 “RTEX speed unit setup”.<table><tr><th>Pr7.25</th><th>Unit</th></tr><tr><td>0</td><td>[r/min]</td></tr><tr><td>1</td><td>[Command unit/s]</td></tr></table>* The value is undefined in torque control mode.</td></tr>
<tr><td>-</td><td>0Dh</td><td>External scale position(Note 4)</td><td>EXPOS</td><td>0</td><td>Pulse
(External scale)</td><td>External scale position</td></tr>
</table>

(Note 1) When a Type_Code error occurs, command error (0031h) will be returned.
Manufacturer will use a Type_Code not listed above.
When a Type_Code used by the manufacturer is set, undefined value will be returned in place of command error (0031h).

(Note 2) When an Index error occurs, command error (0032h) will be returned.

(Note 3) A4N compatible: Type_Code compatible with A4N series can be used, but only with main commands.
Standard: Type_Code newly created for A5N and A5N series and can be used with both main commands and subcommands. When using with main commands, set leftmost 4 bits to 0.
* Although the product supports the A4N compatible Typer_Code to maintain compatibility, basically use the standard Type_Code.

(Note 4) The version before the function extended version 1 is not supported.

Type_Code		Name		Index	Unit	Description
A4N compatible	Standard					
111h	11h	Regenerative load ratio	-	0	% (Note 2)	Ratio of the regenerative overload protection to the alarm occurrence level
112h	12h	Overload ratio	-	0	0.1%	Ratio of the actual load to the rated motor load
-	21h	Logical input signal	-	0	-	Logic level of input signal
-	22h	Logical output signal	-	0	-	Logic level of output signal
-	23h	Logical input signal	-	0	-	Logic level of input signal (expansion portion)

Type_Code		Name		Index	Unit	Description
A4N compatible	Standard					
		(expansion portion)				
-	24h	Logical output signal (expansion portion)	-	0	-	Logic level of output signal (expansion portion)
-	25h	Physical input signal	-	0	-	Physical level of input signal
-	26h	Physical output signal	-	0	-	Physical level of output signal
131h	31h	Inertia ratio	-	0	%	The ratio of load inertia to the motor's rotor inertia (equivalent of value in Pr 0.04) Inertia ratio = (load inertia / rotor inertia) × 100
132h	32h	Automatic motor recognition	-	0	-	0: Invalid 1: Valid
133h	33h	Cause of no revolution	-	0	-	The number which shows the cause that the motor is not running.
134h	34h	Warning flags	-	0	-	The flag that shows the state of the waring currently occurring. * The corresponding bit is set to 1 to activate the flag (showing warning status).
-	37h	Multiple alarm occurrences /Warning information(Note 1)	-	Refer to Section 6-9-6.	-	Information of all the alarms or warnings currently occurring
201h	41h	Mechanical angle (Single turn data)	-	0	pulse	The mechanical angle (one revolution data of an absolute encoder) of the motor * The polarity is fixed and data increases at CCW rotation. One revolution data = 0 to (Encoder resolution - 1) .
202h	42h	Electrical angle	-	0	0.7031°	Motor electrical angle * The polarity is fixed and data increases at CCW rotation. Electrical angle = 0 to 1FF [Hex]
-	43h	Multi-turn data	-	0	Turn	Multi-turn data of the absolute encoder * In the incremental mode (Pr 0.15 = 1), multi-turn data becomes an indefinite value.
-	44h	Encoder status(Note 1)	-	0	-	The status of the encoder
-	47h	Encoder pulse	-	0	pulse	The sum of encoder feedback pulses

Type_Code		Name		Index	Unit	Description
A4N compatible	Standard					
		sum(Note 1)				
-	48h	External scale pulse sum(Note 1)	-	0	Pulse (External scale)	The sum of external scale feedback pulses
-	49h	External scale absolute position(Note 1)	-	0	Pulse (External scale)	The absolute position of the external scale
-	61h	Power on cumulative time	-	-	30 min	Cumulative on-time of control power to the servo amplifier * Because the power ON time is recored in unit of 30 minutes, a turn-on period shorter than 30 minutes is not recorded in the cumulative on-time. not recorded in the cumulative on-time.

(Note 1) The version before the function extended version 1 is not supported.

(Note 2) Be careful that the unit is different from that used for A4N and A5N. (A4N, A5N: [0.1%], A6N: [%])

* With the function extended version 3 or higher, the unit can be changed through bit 7 of Pr 7.99.
  Pr7.99 bit7 0: [%], 1: [0.1%]

Type_Code		Name		Index	Unit	Description
A4N compatible	Standard					
-	62h	Servo amplifier temperature	-	-	°C	Temperature inside the servo amplifier
-	63h	Encoder temperature	-	-	°C	Temperature inside the encoder * Applicable only to 23-bit encoder. 0 for unsupported encoder.
-	64h	Number of inrush resistance relay operations	-	-	Cycle	Operating cycles of inrush current suppression resistor relay * Saturation will occur at maximum value of 40000000h. * Because the power ON time is recored in unit of 30 minutes, a turn-on period shorter than 30 minutes is not recorded in the cumulative cycles.
-	65h	No. of dynamic brake operations	-	-	Cycle	Number of operations of dynamic brake relay * Saturation will occur at maximum value of 40000000h. * Because the power ON time is recored in unit of 30 minutes, a turn-on period shorter than 30 minutes is not recorded in the cumulative time.
-	66h	Fan operating time	-	-	30 min	Operating time of cooling fan * Because the power ON time is recored in unit of 30 minutes, a turn-on period

Type_Code		Name		Index	Unit	Description
A4N compatible	Standard					
						shorter than 30 minutes is not recorded in the cumulative time. * 0 when no fan is installed.
-	67h	Fan life expectancy	-	-	0.1%	Percent of fan life expectancy * Because the power ON time is recored in unit of 30 minutes, a turn-on period shorter than 30 minutes is not recorded in the cumulative time. * 0 when no fan is installed.
-	68h	Capacitor life expectancy	-	-	0.1%	Percent of life expectancy of main power source capacitor * Because the power ON time is recored in unit of 30 minutes, a turn-on period shorter than 30 minutes is not recorded in the cumulative time.
-	69h	Voltage across PN	-	-	V	Main power source PN voltage
-	6Ch	Consumed power of motor (Note 1)	-	-	W	Momentary power consumption of the motor
-	6Dh	Motor power consumption (Note 1)	-	-	Wh	Power consumption of the motor
-	6Eh	Cumulative motor power consumption (Note 1)	-	-	Wh	Cumulative value of motor power consumption
401h	71h	RTEX Cumulative communication errors	-	0	Cycle	Cumulative number of RTEX communication errors * Saturation will occur at maximum value of FFFFh. The count will be cleared upon restarting of servo amplifier or resetting of control power source.
-	77h	RTEX UpdateCounter cumulative error count(Note 1)	-	0	Cycle	Cumulative number of communication errors of RTEX UpdateCounter * Saturation will occur at maximum value of 7FFFh. The count will be cleared upon restarting of servo amplifier or resetting of control power source.
-	78h	RTEX communication Cumulative RTEX communication timeout errors(Note 1)	-	0	Cycle	Cumulative number of RTEX communication data reception interruption errors * Saturation will occur at maximum value of FFFFh. The count will be cleared upon restarting of servo amplifier or resetting of control power source.

Type_Code		Name		Index	Unit	Description
A4N compatible	Standard					
411h	81h	Encoder cumulative communication errors	-	0	Cycle	Cumulative number of communication errors between encoders * Saturation will occur at maximum value of FFFFh. The count will be cleared upon restarting of servo amplifier or resetting of control power source.

(Note 1) The version before the function extended version 1 is not supported.

Type_Code		Name		Index	Unit	Description
A4N compatible	Standard					
413h	83h	External scale cumulative communication errors(Note 1)	-	0	Cycle	Cumulative number of communication errors between external scales * Saturation will occur at maximum value of FFFFh. The count will be cleared upon restarting of servo amplifier or resetting of control power source.
-	84h	External scale abnormal communication data errors (Note 1)	-	0	Cycle	Cumulative number of communication data errors in communication between external scales * Saturation will occur at maximum value of FFFFh. The count will be cleared upon restarting of servo amplifier or resetting of control power source.
-	85h	For manufacturer's use	-	-	-	-
-	86h	Hybrid position deviation (Note 1)	-	-	Command unit	Tolerance between encoder position and external scale position
-	87h	External scale data(Note 1) (Leftmost 24 bits)	-	0	Pulse (External scale)	Rightmost 24 bits of external scale data
-	88h	External scale data(Note 1) (Rightmost 24 bits)	-	0	Pulse (External scale)	<Virtual full-close control mode function disabled> Leftmost 24 bits of external scale data is output. <Virtual full-close control mode function enabled> ● When an AB-phase output type scale is connected, position data (16 bits) is output that is set to 0 when the power is turned ON. Note that it is not affected by Pr 3.26 Reversal of direction.

Type_Code		Name		Index	Unit	Description
A4N compatible	Standard					
						● When a serial incremental scale is connected, position data (24 bits) of the serial incremental scale is output. Note that the data output is position data affected by Pr 3.26 Reversal of direction.
-	89h	External scale status (Note 1)	-	0	-	Status of external scale
-	A1h	Velocity control command (Note 1)	-	0	r/min	Velocity control command
-	A5h	Internal position command speed (Note 1)	-	0	r/min	Internal position command speed
-	A6h	Speed deviation (Note 3)	-	0	r/min	Speed deviation
-	A8h	Positive direction torque limit value (Note 1)	-	0	0.05%	Positive direction torque limit value
-	A9h	Negative direction torque limit (Note 1)	-	0	0.05%	Negative direction torque limit value
-	AAh	Speed limit value (Note 1)	-	0	r/min	Speed limit value
-	ABh	Gain switching flag (Note 1)	-	0	-	Gain switching flag
-	B1h	Deterioration diagnosis state (Note 1)	-	0	-	Deterioration diagnosis state
-	B2h	Deterioration diagnosis torque average time (Note 1)	-	0	0.1% (Note 2)	Deterioration diagnosis torque command average time
-	B3h	Deterioration diagnosis torque command standard value (Note 3)	-	0	0.1%	Deterioration diagnosis torque command standard value
-	B4h	Deterioration diagnosis inertia ratio estimate (Note 1)	-	0	%	Deterioration diagnosis inertia ratio estimate

(Note 1) The version before the function extended version 1 is not supported.

(Note 2) Be careful that the unit is different from the one of the data displayed on the setup support software (PANATERM).

(Note 3) The version before the function extended version 2 is not supported.

<table>
<tr><th colspan="2">Type_Code</th><th colspan="2" rowspan="2">Name</th><th rowspan="2">Index</th><th rowspan="2">Unit</th><th rowspan="2">Description</th></tr>
<tr><th>A4N compatible</th><th>Standard</th></tr>
<tr><td>-</td><td>B5h</td><td>Deterioration diagnosis unbalanced load estimate (Note 1)</td><td>-</td><td>0</td><td>0.1% *2)</td><td>Deterioration diagnosis unbalanced load estimate</td></tr>
<tr><td>-</td><td>B6h</td><td>Deterioration diagnosis unbalanced load estimate (Note 1)</td><td>-</td><td>0</td><td>0.1% *2)</td><td>Deterioration diagnosis unbalanced load estimate</td></tr>
<tr><td>-</td><td>B7h</td><td>Deterioration diagnosis unbalanced load estimate (Note 1)</td><td>-</td><td>0</td><td>0.1%/ (10000 r/ min) *2)</td><td>Deterioration diagnosis unbalanced load estimate</td></tr>
<tr><td>-</td><td>FAh</td><td>Monitor flag (Note 1)</td><td>-</td><td>0</td><td>-</td><td>Various flag information of the servo amplifier
The contents of Monitor_Data, the response data, are as follows.
<table>
<tr><th>Byte</th><th>bit</th><th>Description</th></tr>
<tr><td>12, 20</td><td>7 to 0</td><td>For manufacturer’s use</td></tr>
<tr><td>13, 21</td><td>7 to 0</td><td>For manufacturer’s use</td></tr>
<tr><td rowspan="4">14, 22</td><td>7 to 6</td><td>For manufacturer’s use</td></tr>
<tr><td>5</td><td>Semi-closed / full-closed selection state
0: Semi-closed
1: Full-closed</td></tr>
<tr><td>4</td><td>Incremental / absolute selection state
0: Incremental mode
1: Absolute mode</td></tr>
<tr><td>3 to 0</td><td>For manufacturer’s use</td></tr>
<tr><td>15, 23</td><td>7 to 0</td><td>For manufacturer’s use</td></tr>
</table>
</td></tr>
</table>

(Note 1) The version before the function extended version 2 is not supported.

(MEMO)

Revision History

The manual code is shown at the bottom of the cover page.

Date of issue	Manual No.	Revision details
February 2021	WUME-GM1PGR-01	First edition

(MEMO)

(MEMO)

Please contact ....

Industrial Device Business Division,
Panasonic Corporation
7-1-1 Morofuku, Daito City, Osaka, 574-0044, Japan
industrial.panasonic.com/ac/e/

February,2021

WUME-GM1PGR-01